FANUC Robot series

R-30*i*B Plus/R-30*i*B Mate Plus/R-30*i*B Compact Plus/
R-30*i*B Mini Plus 控制装置

*i*RPickTool

操作说明书

B-83924CM/03

© FANUC CORPORATION, 2017

非常感谢您购买 FANUC 机器人。
在使用机器人之前，务须仔细阅读“FANUC Robot series 安全手册(B-80687CM)”，并在理解该内容的基础上使用机器人。

- 本说明书的任何内容不得以任何方式复制。
- 本机的外观及规格如需改良而变更，恕不另行通知。

本说明书中所载的商品，受到日本国《外汇和外国贸易法》的限制。从日本出口该商品时，可能需要日本国政府的出口许可。另外，将该商品再出口到其他国家时，应获得再出口该商品的国家的政府许可。此外，某些商品可能还受到美国政府的再出口法的限制。若要出口或再出口该商品时，请向我公司洽询。

我们试图在本说明书中描述尽可能多的情况。然而，要在本说明书中注明所有禁止或不能做的事宜，需要占用说明书的大量篇幅，所以本说明书中没有一一列举。因此，对于那些在说明书中没有特别指明可以做的事，都应解释为“不可”。

安全使用须知

本章对安全使用机器人的注意事项进行说明，在使用机器人之前，务必熟读并理解本章中所载的内容。

有关操作机器人时的详细功能，请用户通过说明书充分理解其规格。

在使用机器人和外围设备及其组合的机器人系统时，必须充分考虑作业人员和系统的安全措施。有关安全使用 FANUC 机器人的注意事项，归纳在“FANUC Robot series 安全手册 (**B-80687CM**)”中，可同时参阅该手册。

1 使用者

使用者的定义如下所示。

- **操作者**
  进行机器人的电源 ON/OFF 操作。
  从操作面板启动机器人程序。
- **程序员/示教作业者**
  进行机器人的操作。
  在安全保护范围内进行机器人的示教等。
- **维护技术人员**
  进行机器人的操作。
  在安全保护范围内进行机器人的示教等。
  进行机器人的维修（修理、调整、更换）作业。

“操作者”不能在安全保护范围内进行作业。
“程序员/示教作业者”、“维护技术人员”可以在安全保护范围内进行作业。
安全保护范围内的作业，包括搬运、设置、示教、调整、维修等。
要在安全保护乏味内进行作业，必须接受过机器人的专业培训。

表 1 (a)表示安全保护范围外的作业。各个机器人作业者可以执行在此表中有「○」标示的作业项目。

表 1 (a)安全保护范围外的作业

	操作者	程序员 /示教作业者	维护技术人员
控制装置电源的 ON/OFF	○	○	○
运行模式的选择 (AUTO, T1, T2)		○	○
遥控/本地模式的选择		○	○
以示教器选择程序		○	○
以外部设备选择程序		○	○
以操作盘开始程序	○	○	○
以示教器开始程序		○	○
以操作盘复位报警		○	○
以示教器复位报警		○	○
以示教器的数据设定		○	○
以示教器的示教		○	○
以操作盘的紧急停止	○	○	○
以示教器的紧急停止	○	○	○
操作盘的维修			○
示教器的维修			○

在进行机器人的操作、编程、维修时，操作者、程序员、维护技术人员必须注意安全，至少应穿戴下列物品进行作业。

- 适合于作业内容的工作服
- 安全鞋
- 安全帽

2 有关安全的记载的定义

本说明书包括保证使用者人身安全以及防止机床损坏的有关安全的注意事项，并根据它们在安全方面的重要程度，在正文中以“警告”和“注意”来叙述。
此外，有关的补充说明以“注释”来叙述。
用户在使用之前，必须熟读“警告”、“注意”和“注释”中所叙述的事项。

标识	定义
⚠ 警告	用于在错误操作时，有可能会出现使用者死亡或者受重伤等危险的情况。
⚠ 注意	用于在错误操作时，有可能会出现人员轻伤或中度受伤、物品受损等危险的情况。
注释	用于记述补充说明属警告或者注意以外的事项。

目录

安全使用须知 s-1

1 前言 1
1.1 关于本说明书 1
1.2 本说明书的使用方法 1
1.3 关于其它说明书 2
1.4 关于 Microsoft®产品 3

2 概要 4
2.1 软件选项的构成 5
2.2 机器构成 6
2.2.1 带队列管理的追踪 6
2.2.2 视觉追踪 7
2.3 用语 9
2.4 系统构成例 14
2.5 适用探讨 21
2.6 制约事项 26
2.7 推荐事项 26

3 *i*RPickTool Basic（基本功能）启动调试步骤 27
3.1 设定应用程序前的准备工作 28
3.1.1 脉冲编码器的连接与设定 28
3.1.1.1 脉冲编码器的连接 28
3.1.1.2 脉冲编码器的设定 29
3.1.1.3 脉冲编码器的设定确认 33
3.1.2 相机的连接确认 34
3.1.3 触发传感器的输入确认 39
3.1.3.1 使用［DI］/［RI］ 39
3.1.3.2 使用［HDI］ 40
3.1.4 设定碰触销的工具坐标系 45
3.1.5 机器人程序的创建 50
3.1.6 负载设定（动作性能） 68
3.2 应用程序的设定 71
3.2.1 工作单元设置 71
3.2.2 托盘设定 79
3.2.3 传送带设定 84
3.2.3.1 供给传送带的设定 84
3.2.3.2 排出传送带的设定 85
3.2.4 追踪坐标系设定 86
3.2.5 传感器设定 94
3.2.5.1 供给传送带传感器的设定 94
3.2.5.2 排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时） 95
3.2.5.3 排出传送带传感器的设定（［HDI］＋［放入工件］时） 96
3.2.6 传感器位置设定 97
3.2.7 视觉设定 103
3.2.7.1 相机校准 103
3.2.7.2 视觉程序 115
3.2.8 传送带工作站设定 124
3.2.9 机械手工具坐标系的设定 129
3.2.10 基准位置的设定、机器人的位置示教 133
3.2.10.1 供给基准位置的设定、机器人的位置示教 133

3.2.10.2 排出基准位置的设定、机器人的位置示教 144
3.2.11 追踪动作的微调整 150

4 *i*RPickTool（基本功能+Plug & Play）的启动调试步骤 151
4.1 设定应用程序前的准备工作 152
4.1.1 脉冲编码器的连接与设定 152
4.1.1.1 脉冲编码器的连接 152
4.1.1.2 脉冲编码器的设定 153
4.1.1.3 脉冲编码器的设定确认 157
4.1.2 相机的连接确认 157
4.1.3 触发传感器的输入确认 162
4.1.3.1 使用 ［DI］/［RI］ 162
4.1.3.2 使用［HDI］ 163
4.1.4 设定碰触销的工具坐标系 167
4.1.5 负载设定（动作性能） 172
4.2 应用程序的设定 175
4.2.1 工作单元设置 175
4.2.2 抓手的设定 184
4.2.3 托盘设定 188
4.2.4 传送带设定 193
4.2.4.1 供给传送带的设定 193
4.2.4.2 排出传送带的设定 194
4.2.5 追踪坐标系设定 195
4.2.6 传感器设定 204
4.2.6.1 供给传送带传感器的设定 204
4.2.6.2 排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时） 205
4.2.6.3 排出传送带传感器的设定（［HDI］＋［放入工件］时） 206
4.2.7 传感器位置设定 207
4.2.8 视觉设定 213
4.2.8.1 相机校准 213
4.2.8.2 视觉程序 225
4.2.9 传送带工作站设定 233
4.2.10 操作的设定 238
4.2.11 机械手工具坐标系的设定 239
4.2.12 基准位置的设定、机器人的位置示教 244
4.2.12.1 供给基准位置的设定、机器人的位置示教 244
4.2.12.2 排出基准位置的设定、机器人的位置示教 251
4.2.12.3 机器人的原位置示教 258
4.2.13 追踪动作的微调整 258

5 设定前的其他准备工作 259
5.1 网络的构筑 259
5.1.1 网络电缆的连接 259
5.1.2 构造网络时的注意事项 260
5.1.3 IP 地址的设定 260
5.1.4 机器人间的通信设定 265
5.2 脉冲编码器的设置和设定 274
5.2.1 设置脉冲编码器 274
5.2.2 连接脉冲编码器 275
5.2.3 设定连接到主板的脉冲编码器 275
5.2.4 在多台机器人上设定脉冲编码器 276
5.3 相机的设置和连接 277
5.4 工具坐标系的设定 278
5.5 点阵板夹具 278

6 基本功能参考 279
6.1 工作单元设置 288

6.1.1 对象数量设定 289
6.1.2 方案 291
6.2 机器人设定 291
6.3 托盘设定 292
6.3.1 单元 295
6.3.2 层操作 296
6.3.3 基准单元的设定 299
6.4 传送带设定 300
6.4.1 追踪坐标系设定 304
6.4.1.1 圆弧传送带的情况 305
6.5 传感器设定 312
6.5.1 传感器详细设定 312
6.5.1.1 ［距离］ ＋ ［放入工件］的情况 313
6.5.1.2 ［距离］＋［通过视觉检出］的情况 314
6.5.1.3 ［距离］+［程序的执行］ 315
6.5.1.4 ［DI］＋［放入工件］的情况 316
6.5.1.5 ［DI］＋［通过视觉检出］的情况 317
6.5.1.6 ［DI］+［程序的执行］ 318
6.5.1.7 ［HDI］＋［放入工件］的情况 319
6.5.1.8 ［标记］＋［放入工件］的情况 320
6.5.1.9 ［RI］＋［放入工件］的情况 321
6.5.1.10 ［RI］＋［通过视觉检出］的情况 322
6.5.1.11 ［RI］+［程序的执行］时 323
6.5.2 每次触发条件和动作组合的必要设定 324
6.5.3 传感器位置设定 324
6.5.3.1 不使用托盘的情况 325
6.5.4 视觉设定 327
6.5.5 托盘位置设定 328
6.5.5.1 ［距离］＋［通过视觉检出］的情况。 328
6.5.5.2 ［DI］＋［通过视觉检出］的情况 331
6.5.6 目标位置设定 334
6.5.6.1 不使用托盘的情况 334
6.5.6.2 使用托盘的情况 337
6.5.7 可同时执行的传感器的限制 340
6.6 传送带工作站设定 341
6.6.1 追踪坐标系的追加 346
6.6.1.1 线传送带的情况 347
6.6.1.2 圆弧传送带的情况 350
6.7 固定工作站设定 353
6.8 KAREL 程序 354
6.8.1 有关工作单元的程序 354
6.8.1.1 PKWCSTART.PC 354
6.8.1.2 PKWCEND.PC 354
6.8.1.3 PKWCCHECK.PC 354
6.8.1.4 PKWCRSTPRDST.PC 354
6.8.1.5 PKGETVAR.PC 355
6.8.1.6 PKSETVAR.PC 356
6.8.1.7 PKPRGRGETEN.PC 357
6.8.1.8 PKRSTWC.PC 357
6.8.1.9 PKABO.PC 357
6.8.1.10 PKLABO.PC 357
6.8.1.11 PKPOSTERR.PC 358
6.8.1.12 PKCOPYPR2POS.PC 358
6.8.2 有关机器人的程序 358
6.8.2.1 PKRBGETID.PC 358
6.8.2.2 PKRBGETSFLG.PC 359
6.8.3 有关机器人的程序 359
6.8.3.1 PKCVSETLBD.PC 359

6.8.3.2 PKCVSETTRNAM.PC 360
6.8.3.3 PKCVENBLB.PC 361
6.8.3.4 PKCVENBSC.PC 361
6.8.3.5 PKCVGETNAME.PC 362
6.8.4 有关传感器的程序 362
6.8.4.1 PKSNENBSENS.PC 362
6.8.4.2 PKSNSETVPNAM.PC 363
6.8.4.3 PKSNSETVTOFS.PC 364
6.8.4.4 PKSNSETMTOFS.PC 365
6.8.4.5 PKSNSTART.PC 366
6.8.4.6 PKSNSTOP.PC 366
6.8.5 有关传送带工作站的程序 366
6.8.5.1 PKCSGETID.PC 366
6.8.5.2 PKCSCLRQUE.PC 367
6.8.5.3 PKCSGETQUE.PC 367
6.8.5.4 PKCSGETQCELL.PC 368
6.8.5.5 PKCSACKQUE.PC 369
6.8.5.6 PKCSPUTQUE.PC 369
6.8.5.7 PKCSGETTIME.PC 370
6.8.5.8 PKCSGETDIST.PC 371
6.8.5.9 PKCSGETPFRT.PC 372
6.8.5.10 PKCSGETNPRT.PC 373
6.8.5.11 PKCSSETDSLIN.PC 373
6.8.5.12 PKCSGETENC.PC 374
6.8.5.13 PKCSGETTRID.PC 374
6.8.5.14 PKCSCLRACK.PC 375
6.8.6 有关固定工作站的程序 375
6.8.6.1 PKFSGETID.PC 375
6.8.6.2 PKFSPUTQUE.PC 375
6.8.6.3 PKFSCLRQUE.PC 376
6.8.6.4 PKFSGETQUE.PC 376
6.8.6.5 PKFSACKQUE.PC 377
6.8.6.6 PKFSPUTBUF.PC 377
6.8.6.7 PKFSCLRBUF.PC 377
6.8.6.8 PKFSGETBUF.PC 378
6.8.6.9 PKFSACKBUF.PC 379
6.8.6.10 PKFSGETTRID.PC 379
6.8.6.11 PKFSGETUF.PC 380
6.8.6.12 PKFSCLRACK.PC 380
6.8.7 有关托盘的程序 380
6.8.7.1 PKTRSETCLPOS.PC 380
6.8.7.2 PKTRGETNUMCL.PC 381
6.9 机器人的程序 382
6.9.1 有关追踪程序 382
6.9.2 传送带—传送带的构成的样本程序 383
6.9.2.1 主程序 384
6.9.2.2 进行拾取动作的程序 385
6.9.2.3 进行放置动作的程序 387
6.9.3 传送带—固定工作站的构成的样本程序 387
6.9.3.1 主程序 388
6.9.3.2 进行拾取动作的程序 389
6.9.3.3 进行放置动作的程序 389
6.9.4 传送带—缓冲区传送带的构成的样本程序 390
6.9.4.1 主程序 392
6.9.4.2 进行拾取动作的程序 393
6.9.4.3 进行放置动作的程序 394
6.9.4.4 进行拾取动作的程序(缓冲区) 395
6.9.4.5 进行放置动作的程序(缓冲区) 395

6.10 基准位置设定 396
6.10.1 ［DI］/［RI］＋［通过视觉检出］ 397
6.11 机器人的位置示教 400
6.11.1 传送带工作站 400
6.11.2 固定工作站 400
6.12 追踪动作的微调整 401
6.12.1 最初的微调整 401
6.12.2 追踪坐标系的设定误差的调整 401
6.12.3 机器人的动态误差的调整 403
6.12.4 示教点的平顺指定 403
6.12.5 直线距离指定 405
6.12.6 法兰盘直线指令 405
6.12.7 直线最高速度功能 406
6.12.8 先执行指令 406
6.13 启动多台机器人时的注意事项 407
6.14 预整列设定 408
6.14.1 用语 410
6.14.2 工作单元、机器人设定 410
6.14.3 托盘设定 410
6.14.4 传送带设定 412
6.14.5 传感器设定 418
6.14.6 传送带工作站设定 419
6.14.7 固定工作站设定 425
6.14.8 KAREL 程序 425
6.14.8.1 PKCSGENVRTRY.PC 425
6.14.8.2 PKCSGETQPRGR.PC 426
6.14.8.3 PKCSTRGPRGR.PC 426
6.14.8.4 PKCSACKQPRGR.PC 427
6.14.9 机器人的程序 428
6.14.9.1 传送带一传送带(相同，最上游)的构成的样本程序 428
6.14.9.2 传送带一传送带(相同，下游)的构成的样本程序 431
6.14.9.3 传送带一传送带的构成的样本程序 432
6.14.10 基准位置设定 433
6.14.11 机器人的位置示教 433
6.14.12 追踪动作的微调整 433
6.14.12.1 追踪坐标系的设定误差的调整 433

7 Plug & Play 参考 435
7.1 工作单元设定 436
7.2 抓手和区域的设定 436
7.2.1 抓手和区域的设定 437
7.2.1.1 抓手的详细设定 439
7.2.1.2 区域的详细设定 440
7.3 操作的设定 441
7.3.1 传送带工作站的操作 442
7.3.1.1 传送带工作站的操作的详细设定 443
7.3.2 固定工作站的操作 445
7.3.2.1 固定工作站的操作的详细设定 446
7.4 固定工作站设定 447
7.5 机器人的程序 448
7.5.1 标准 TP 程序下的机器人的动作 449
7.5.2 数值寄存器 450
7.5.3 位置寄存器 451
7.5.4 字符串寄存器 452
7.5.5 工件数据 VR 452
7.5.6 负载设定 452
7.5.7 用户报警 452

7.5.8 用户坐标系 452
7.5.9 工具坐标系 452
7.6 基准位置设定 453
7.7 机器人的位置示教 453
7.8 追踪动作的微调整 453
7.8.1 追踪坐标系的设定误差的调整 453
7.8.2 机器人的动态误差的调整 454
7.9 KAREL 程序 455
7.9.1 有关工作单元的程序 455
7.9.2 有关抓手的程序 455
7.9.2.1 PKGRCHKPP.PC 456
7.9.2.2 PKGRCLOSE.PC 456
7.9.2.3 PKGRGETDPDLY.PC 457
7.9.2.4 PKGRGETID.PC 457
7.9.2.5 PKGRGETPKDLY.PC 457
7.9.2.6 PKGRGETZNCNT.PC 457
7.9.2.7 PKGRGETZNUT.PC 458
7.9.2.8 PKGRINIT.PC 458
7.9.2.9 PKGROPEN.PC 458
7.9.3 有关机器人的程序 459
7.9.3.1 PKRBGETCVCNT.PC 459
7.9.3.2 PKRBGETFSCNT.PC 459
7.9.3.3 PKRBGETPRGRT.PC 460
7.9.4 有关机器人的程序 460
7.9.5 有关传感器的程序 460
7.9.6 有关传送带工作站的程序 460
7.9.6.1 PKCSCALWPOS.PC 461
7.9.6.2 PKCSGETAPPR.PC 461
7.9.6.3 PKCSGETOPPR.PC 462
7.9.6.4 PKCSGETPMUL.PC 462
7.9.6.5 PKCSGETPPCHK.PC 462
7.9.6.6 PKCSGETVR.PC 463
7.9.6.7 PKCSGETWPOS.PC 463
7.9.7 有关固定工作站的程序 463
7.9.7.1 PKFSGETAPPR.PC 463
7.9.7.2 PKFSGETIXDLY.PC 464
7.9.7.3 PKFSGETIXODO.PC 464
7.9.7.4 PKFSGETIXSIM.PC 464
7.9.7.5 PKFSGETOPPR.PC 465
7.9.7.6 PKFSGETPMUL.PC 465
7.9.7.7 PKFSGETPPCHK.PC 465
7.9.7.8 PKFSGETRDY.PC 465
7.9.7.9 PKFSGETRDYDI.PC 466
7.9.7.10 PKFSGETVR.PC 466
7.9.7.11 PKFSMONIDX.PC 466
7.9.7.12 PKFSSETRDY.PC 467
7.9.8 有关托盘的程序 467
7.10 标准 TP 程序 467
7.10.1 有关标准 TP 程序的名称 468
7.10.2 标准 TP 程序的详细 469
7.10.2.1 标准 TP 程序一览 469
7.10.2.2 PK_MAIN1.TP: 组 1 的主程序 469
7.10.2.3 PK_INIT1.TP 470
7.10.2.4 PK_PERCH1.TP 472
7.10.2.5 PK_PICK1.TP 473
7.10.2.6 PK_DROP1.TP 473
7.10.2.7 PK_GR_GET_UTOOLS1.TP 474
7.10.2.8 PK_CV_GET_PICK_DATA11.TP 475

7.10.2.9 PK_CV_GET_DROP_DATA11.TP 476
7.10.2.10 PK_FS_GET_PICK_DATA11.TP 477
7.10.2.11 PK_FS_GET_DROP_DATA11.TP 478
7.10.2.12 PK_CV_PICK11.TP 479
7.10.2.13 PK_CV_DROP11.TP 482
7.10.2.14 PK_CV_WAITPOS1.TP 485
7.10.2.15 PK_FS_PICK11.TP 486
7.10.2.16 PK_FS_DROP11.TP 488
7.10.2.17 PK_FS_INDEX.TP 490
7.10.2.18 PK_FS_READY.TP 491
7.10.2.19 PK_GR_GET_ENDZONE1.TP 492
7.10.2.20 PK_CYCSTOP1.TP 493
7.11 预整列设定 493
7.11.1 抓手和区域的设定 493
7.11.2 操作的设定 493
7.11.3 追踪动作的微调整 493
7.11.3.1 追踪坐标系的设定误差的调整（使用 ADJ_OFS 时） 493
7.11.4 标准 TP 程序 494
7.11.4.1 PK_PRGR_MAIN1.TP 495
7.11.4.2 PK_PRGR_INIT1.TP 496
7.11.4.3 PK_PRGR_CV_PICKDROP1.TP 496
7.11.4.4 PK_PRGR_CV_PICKVTRAY1.TP 498
7.12 Plug & Play 的缓冲区功能 501
7.12.1 与缓冲区中的拾取/放置相关的机器人的动作 501
7.12.2 缓冲区的设定 502
7.12.3 缓冲区功能中使用的 KAREL 程序 503
7.12.3.1 PKRBGETBUFID.PC 503
7.12.3.2 PKFSGETBUFPT.PC 504
7.12.3.3 PKFSGETBUFDT.PC 504
7.12.4 缓冲区功能中使用的 TP 程序 504
7.12.4.1 PK_BF_GET_DATA1.TP 504
7.12.4.2 PK_BF_PICK1.TP 505
7.12.4.3 PK_BF_DROP1.TP 506
7.12.4.4 PK_BF_OK2PICK1.TP 508
7.12.4.5 PK_BF_OK2DROP1.TP 509
7.12.4.6 PK_CV_OK2PICK1.TP 510
7.12.4.7 PK_CV_OK2DROP1.TP 511
7.12.4.8 PK_BUF_PICK1.TP 512
7.12.4.9 PK_BUF_DROP1.TP 512
7.13 使用 *i*RPickTool 的跳过动作实施范围时的安全逃避动作 513
7.14 多层托盘中的固定接近高度 514
7.14.1 PKCSFIXAPRZ.PC 516
7.14.2 PKFSFIXAPRZ.PC 517
7.15 指定防止负载错误的 *i*RPickTool 机器人最高速度 518

8 不同用途的设定例子 519
8.1 根据传送带上的位置，由不同的机器人拾取 519
8.2 根据工件的部分特征，由不同的机器人拾取 523
8.3 将工件多层堆起放入箱中 528
8.4 从箱子一端依次放置工件 531
8.5 在箱子中以各种方向放置 532
8.6 根据工件种类不同，切换放置位置 533
8.7 根据种类，放置于其他的输出传送带 536
8.8 从 2 个以上的供给传送带上拾取 538
8.9 拾取失败时发出通知 540
8.10 仅在单元都已满的情况下，取出托盘 541
8.11 系统重启时处理停止前已检出的工件 543

9 其他有用功能 544
9.1 使用方案变更工作单元 544
9.1.1 方案设定 545
9.1.2 方案的详细设定 546
9.1.3 利用 PLC 的方案操作 555
9.1.3.1 PLC 方案操作所需的设定 555
9.1.4 KAREL 程序 560
9.1.4.1 PKRCCREATE.PC 560
9.1.4.2 PKRCDELETE.PC 560
9.1.4.3 PKRCSAVE.PC 561
9.1.4.4 PKRCRESTORE.PC 561
9.1.4.5 PKRCEXPORT.PC 561
9.1.4.6 PKRCIMPORT.PC 562
9.1.4.7 PKRCACTIVE.PC 562
9.2 确认生产状况 562
9.2.1 确认传送带的状态 563
9.2.2 确认传感器的状态 565
9.2.3 确认传送带工作站的状态 566
9.2.4 确认固定工作站的状态 569
9.3 限制最终轴的轴可动范围 570
9.4 超出领域时，跳过追踪动作 571
9.5 确认机器人的扭矩 573
9.6 *i*RPickTool 的 4D 图形显示 574
9.6.1 4D 工作单元的显示 575
9.6.2 4D 工作单元的生成 575
9.6.3 图标的使用方法 584
9.6.4 特定信息的选择显示 585
9.6.4.1 4D 拾取边界 585
9.6.4.2 4D 拾取坐标系 586
9.6.5 4D 工作单元的删除 586
9.6.6 使用方案的工作单元的变更 586
9.6.7 用户定义对象形状的使用 587
9.6.8 制约事项 587
9.7 伺服传送带追踪拾取功能 587
9.7.1 概要 587
9.7.2 设定 588
9.7.2.1 附加轴设定 588
9.7.2.2 伺服传送带设定 594
9.7.2.3 伺服传送带用 TP 程序的生成 597
9.7.2.4 追踪参数设定表的设定 601
9.7.2.5 主程序的创建例 601
9.7.3 伺服传送带用功能 601
9.7.3.1 伺服传送带用的动态误差的微调整参数 601
9.7.3.2 等待伺服传送带停止的功能 602
9.7.3.3 伺服传送带专用 KAREL 程序 602
9.7.4 伺服传送带追踪制约事项 609
9.8 多视图 610
9.8.1 概要 610
9.8.2 制约事项 610
9.8.3 相机的准备 610
9.8.4 安装 611
9.8.4.1 视觉设定 611
9.8.4.2 传感器设定 614
9.8.4.3 基准位置的设定、机器人的位置示教 616
9.8.5 生产状况 617
9.9 3D 视觉传感器 618
9.9.1 概要 618

9.9.2 制约事项 618
9.9.3 设置安装位置 618
9.9.4 视觉程序（三维） 628
9.9.5 视觉程序（二维） 635
9.10 间隙检查功能 638
9.10.1 概要 638
9.10.2 制约事项 638
9.10.3 注意事项 638
9.10.4 设定 640
9.10.4.1 工作单元、机器人设定 640
9.10.4.2 托盘设定 640
9.10.4.3 传送带设定 640
9.10.4.4 传感器设定 640
9.10.4.5 间隙向导 643
9.10.4.6 传送带工作站设定 649
9.10.4.7 固定工作站设定 650

10 常见问题处理方法 651
10.1 机器人完全不去拾取工件 651
10.2 工件的跳过频发 651
10.3 2 次去取相同的工件 652
10.4 如果工件旋转，则补偿偏移 653
10.5 传送带一端的工件补偿偏移 653
10.6 机器人动作不顺畅 653
10.7 停止/启动传送带不工作 653
10.8 图像读入的时机延迟 653
10.9 超限次数不断增加 654
10.10 生产中无法变更设定 654

附录

A 系统变量文件 657

B 示教用电脑的设定 658
B.1 电脑与浏览器 658
B.2 通信电缆 658
B.3 连接通信电缆 658
B.4 在电脑上设定 IP 地址 658
B.5 变更 Internet Explorer 的设定 661
B.6 变更 Windows 防火墙的设定 667
B.7 打开 *i*RPickTool 的画面 668
B.8 打开 *i*RVision 的画面 669
B.9 常见问题处理方法 672

C HDI 设定 675
C.1 接线图 676
C.1.1 R-30*i*B Plus 时 676
C.1.2 R-30*i*B Mate Plus 时 678
C.1.3 R-30*i*B Compact Plus 时 679
C.1.4 R-30*i*B Mini Plus 时 680
C.2 有关输入信号 681
C.3 设定方法 682
C.3.1 概要 682
C.3.2 追踪拾取 HDI 功能的有效 682

D αA1000S 脉冲编码器 683

D.1 必要的结构 683
D.2 概略图 688
D.3 连接方法 690
D.3.1 R-30*i*B Plus 时 691
D.3.2 R-30*i*B Mate Plus 时 694
D.3.3 R-30*i*B Compact Plus 时 696
D.3.4 R-30*i*B Mini Plus 时 697

E 量增脉冲编码器 699
E.1 必要的结构 699
E.2 概略图 704
E.3 连接方法 707
E.3.1 与机器人控制装置直接连接 707
E.3.2 与脉冲分配器连接 709

F 另置型接口装置（SDU） 712

G 旧软件与 *i*RPickTool 的 KAREL 对照表 719

H 传感器任务的自定义 720
H.1 概要 720
H.2 自定义传感器任务的步骤 720
H.3 自定义的要点 726
H.4 KAREL 函数 729
H.4.1 put_part_cst 729
H.4.2 run_vis_cst 729
H.4.3 get_ecnt_idx 729
H.4.4 update_nfnd 730
H.4.5 start_srv 730
H.4.6 incre_wid 730
H.4.7 put_queue 731
H.5 内嵌函数 731
H.5.1 LINE_COUNT 731
H.5.2 V_RUN_FIND 732
H.5.3 VT_GET_FOUND 732
H.5.4 P_CSGETID 733
H.5.5 P_CVGETID 733

I 3rd Party 接口加载项 734
I.1 概要 734
I.2 用语 734
I.3 制约 735
I.3.1 机器人系统的制约 735
I.3.2 3rd Party 视觉的制约 735
I.4 关于支持范围 736
I.5 启动步骤 737
I.5.1 3rd Party 视觉的连接和设定 737
I.5.1.1 3rd Party 视觉的连接 737
I.5.1.2 3rd Party 视觉的设定 738
I.5.2 通信设定 738
I.5.2.1 IP 地址的设定 738
I.5.2.2 客户标签的设定 738
I.5.3 *i*RPickTool 的设定 738
I.5.4 机器人程序的创建 739
I.5.4.1 主程序 742
I.5.4.2 触发程序 743
I.5.4.3 初始化程序 745

I.5.4.4 检出程序 746
I.5.4.5 锁存程序 747
I.5.4.6 坐标转换参数计算程序 748
I.5.4.7 基准位置设定程序 749
I.5.5 供给传送带传感器的设定 750
I.5.6 位置寄存器操作的设定 751
I.5.7 坐标转换参数计算 751
I.5.8 供给传送带的基准位置设定、机器人的位置示教 751
I.6 KAREL 程序 752
I.6.1 通信相关程序 752
I.6.1.1 PKMVCONNECT.PC 752
I.6.1.2 PKMVDISCO.PC 752
I.6.2 初始化相关程序 752
I.6.2.1 PKMVINIACT.PC 752
I.6.3 输入信息发送相关程序 753
I.6.3.1 PKMVSETSNDPM.PC 753
I.6.3.2 PKMVDELSNDPM.PC 754
I.6.3.3 PKMVSENDMSG1~4.PC 754
I.6.4 输出信息接收相关程序 754
I.6.4.1 PKMVSETRCVPM.PC 754
I.6.4.2 PKMVRCVMSG1~4.PC 754
I.6.5 输出信息解析相关程序 755
I.6.5.1 PKMVSETPRSPM.PC 755
I.6.5.2 PKMVDELPRSPM.PC 755
I.6.5.3 PKMVPRSMSG1~4.PC 756
I.6.6 坐标转换参数计算相关程序 756
I.6.6.1 PKMVSCLPUTP1.PC 756
I.6.6.2 PKMVSCLPUTP2.PC 756
I.6.6.3 PKMVSCLPUTP3.PC 757
I.6.6.4 PKMVSCLSETP1.PC 757
I.6.6.5 PKMVSCLSETP2PC 757
I.6.6.6 PKMVSCLCALPM.PC 758
I.6.7 检出位置转换相关程序 758
I.6.7.1 PKMVCNVPX2MM.PC 758
I.6.7.2 PKMVCALOFS2D.PC 758
I.6.7.3 PKMVSETVR.PC 759
I.6.8 其他程序 759
I.6.8.1 PKMVWAITTASK.PC 759
I.7 对应用途的设定例子 760
I.7.1 一次拍摄检测出多个工件 760
I.8 常见问题处理方法 762
I.8.1 不能与 3rd Party 视觉进行连接 762
I.8.2 在 3rd Party 视觉中未执行拍摄和检出 762
I.8.3 PKMVRCVMSG 中发生超时 762
I.8.4 PKMVPRSMSG 失败 762
I.8.5 锁存失败 763
I.8.6 抓住位置不正确 764

1 前言

在使用 *i*RPickTool 功能之前，对有关该说明书的概要进行说明。

1.1 关于本说明书

本说明书是 R-30*i*B Plus 控制装置及 R-30*i*B Mate Plus 控制装置、R-30*i*B Compact Plus 控制装置、R-30*i*B Mini Plus 控制装置中所包含的 *i*RPickTool 功能相关操作方法的说明手册。
本说明书仅对于客户所使用的机器人安装、组合后的 *i*RPickTool 功能的设定、编程方法进行总结。启动使用 *i*RPickTool 功能的机器人系统时，请参照“1.3 关于其它说明书”中介绍的说明书。

> **⚠注意**
> **本说明书依照软件版次 7DF5 系列 14 版的 R-30*i*B Plus 控制装置进行编写。根据使用的控制装置的软件版次不同，可能发生不存在本说明书中所记载的功能、设定项目，或者有该说明书中未记载的功能、设定项目，或即使有记载，也有部分差异等情况。有关此问题，敬请谅解。**

1.2 本说明书的使用方法

本说明书对构建带队列管理的追踪系统或是视觉追踪系统的步骤等进行说明。
“*i*RPickTool 基本”的启动调试请阅读第 3 章。
“*i*RPickTool”的启动调试请阅读第 4 章。
有关“*i*RPickTool 基本”和“*i*RPickTool”的不同，请参阅“2.1 软件选项的构成”。
第 5、6、7 章针对在第 3、4 章启动调试步骤中未说明的步骤及具体功能进行说明。
此外，第 8 章针对不同用途的具体设定例子进行说明，第 9 章对 *i*RPickTool 的其他有用功能进行说明。请结合您的使用情况进行参阅。

本说明书的章节标题与内容如下。

章	标题	内容
第 1 章	前言	针对本说明书及相关说明书进行说明。 特别是注意事项，请务必阅读。
第 2 章	概要	对 *i*RPickTool 的概要及使用 *i*RPickTool 的构成例子进行说明。
第 3 章	*i*RPickTool Basic（基本功能）启动调试步骤	对启动调试 *i*RPickTool 基本的准备到设定的过程进行说明。如果您是初次使用 *i*RPickTool 基本，请参阅本章内容。
第 4 章	*i*RPickTool（基本功能+Plug & Play）的启动调试步骤	对启动调试 *i*RPickTool 的准备到设定的过程进行说明。如果您是初次使用含 Plug & Play 的 *i*RPickTool，请参阅本章内容。
第 5 章	设定前的其他准备工作	针对第 3、4 章说明的准备工作之外的其他准备工作进行说明。
第 6 章	基本功能参考	针对基本功能的各个功能及画面项目详情进行说明。
第 7 章	Plug & Play 参考	针对 Plug & Play 的各个功能及画面项目详情进行说明。
第 8 章	不同用途的设定例子	针对可利用 *i*RPickTool 设定系统的各种应用例子，举例说明。
第 9 章	其他有用功能	除了前面 8 章介绍的功能之外，针对 *i*RPickTool 的有用功能进行说明。
第 10 章	常见问题处理方法	对在 *i*RPickTool 追踪系统中经常出现的故障及其对策进行说明。
附录 A～I	附录中对以下项目进行说明。 系统变量文件/示教用电脑的设定/HDI 的设定/脉冲编码器及其连接/旧软件与 *i*RPickTool 中使用的 KAREL 的对照表/传感器任务的自定义/3rd Party 接口加载项	

本说明书的表述

本说明书中使用了下列记号。请搜索信息时进行有效利用。

记号	说明
便签	记载了画面操作相关的提示信息及功能说明和设定内容参考信息。

1.3 关于其它说明书

这里介绍使用 *i*RPickTool 功能时可供参考的本说明书以外的手册。

说明书的种类	规格图编号	说明书的内容
操作说明书（基本操作篇）	B-83284CM	这是控制装置的主要手册。主要就如下事项进行说明。 • 机器人系统的设定 • 机器人的操作 • 程序的创建和修正 • 程序的执行 • 机器人的状态显示 • 程序的保存和加载 在进行机器人的应用设计、机器人的引入、示教、现场调试等时使用。
维修说明书	B-83195CM	该说明书对 R-30*i*B/R-30*i*B Plus 控制装置的维修方法、机器人控制装置与外围设备之间的连接方法进行说明。
	B-83525CM	该说明书对 R-30*i*B/R-30*i*B Mate Plus 控制装置的维修方法、机器人控制装置与外围设备之间的连接方法进行说明。
	B-83555CM	该说明书对 R-30*i*B Mate/R-30*i*B Mate Plus 控制装置（外气导入型）的维修方法、机器人控制装置与外围设备之间的连接方法进行说明。
	B-84035 CM	该说明书对 R-30*i*B Compact Plus 控制装置的维修方法、机器人控制装置与外围设备之间的连接方法进行说明。
	B-84175CM	该说明书描述 R-30*i*B Mini Plus 控制装置的维修方法、机器人控制装置与外围设备之间的连接方法进行说明。
操作说明书 （报警代码一览表）	B-83284CM-1	这是报警代码的一览表。该手册就报警的发生原因和应对方法进行说明。
选项功能 操作说明书	B-83284CM-2	该手册就软件选项功能进行说明。
传感器机构部/控制部 操作说明书	B-83984CM	该说明书对在 *i*RVision 中使用的相机、立体传感器等各种传感器和控制装置的连接方法、各种传感器的维修方法进行说明。
*i*RVision 操作说明书 （参考篇）	B-83914CM	有关 *i*RVision 功能的参考手册。对 *i*RVision 提供的各种功能进行说明。对 *i*RVision 功能的各设定项目的正确的意思、各命令的变量的正确的意思进行说明。
*i*RVision 2 维相机 操作说明书	B-83914CM-2	该手册是在启动使用 *i*RVision 而进行 2 维补正的机器人系统和进行 2.5 维补正的机器人系统时最初可供参照的手册。该说明书对使用 *i*RVision 进行 2 维补正、2.5 维补正时的系统启动步骤、程序的编制方法、应该注意的要点、技巧、应对各种情况的方法等进行说明。
*i*RVision 良否检查 操作说明书	B-83914CM-5	该手册是在启动使用 *i*RVision 而进行工件的良否检查的机器人系统时最初可供参照的手册。该手册对使用 *i*RVision 进行工件的良否检查时的系统启动步骤、程序的编制方法、应该注意的要点、技巧、应对各种情况的方法等进行说明。
Ethernet Function OPERATOR’S MANUAL	B-82974EN	该手册就 FTP、RIPE、*i*RConnect、PC 共享等机器人控制装置的网络选项功能进行说明。
Line Tracking OPERATOR’S MANUAL	B-83474EN	该说明书对追踪拾取功能进行说明。该功能是使用 iRPickTool 的追踪系统的基础。
*i*RPickTool（自动视觉跟踪坐标系设定功能） 快速设置指南	B-83924CM-1	该说明书对使用追踪坐标系自动设定功能，简单启动使用 *i*RPickTool 的追踪系统的步骤进行说明。

1.4 关于 Microsoft®产品

本说明书使用以下简称

Microsoft® Windows® 7 Professional ： Windows 7
Windows® operating system ： Windows

关于商标

Windows、Windows XP、Windows7 Professional、Internet Explorer 是美国 Microsoft Corporation 在美国及其他国家的注册商标或商标。

2 概要

*i*RPickTool 是一款机器人根据传送带的流动，针对拾取/放置工件动作应用程序的专业化机器人软件。
机器人判断流动传送带上的工件后，可以做出拾取/放置动作。
这些动作通过组合与流水线传送带或圆弧传送带流动同步动作的功能，利用相机或光电管传感器等传感器检测工件的功能、实现多个机器人分担取出的功能。
此外，支持类似以下的多个传送带。

- 各机器人从一条传送带取出工件，而将工件放置在另一个传送带的“传送带-传送带”应用程序。
- 供给或排出为固定区域（固定工作站）的“传送带-固定工作站”或“固定工作站传送带”的应用程序，而不是传送带。

*i*RPickTool 的构成，主要有以下 4 种类型。

1. *i*RPickTool 基本
2. *i*RPickTool
3. *i*RPickTool 基本+ *i*RVision 选项（视觉追踪）
4. *i*RPickTool + *i*RVision 选项（视觉追踪）

并且，对于各个组合，可以添加“4D 图形”的扩充功能包选项。

所谓视觉追踪，是指上述的应用程序中，特别是使用相机检知工件的应用程序。
不使用相机，而通过光电管传感器等的传感器检知传送带上的工件的应用程序，在本说明书中称为带队列管理的追踪。
标准备有启动该应用程序所需的功能“*i*RPickTool 基本”和“*i*RPickTool”。有关“*i*RPickTool 基本”和“*i*RPickTool”的不同，在后面的“2.1 软件选项的构成”中说明。

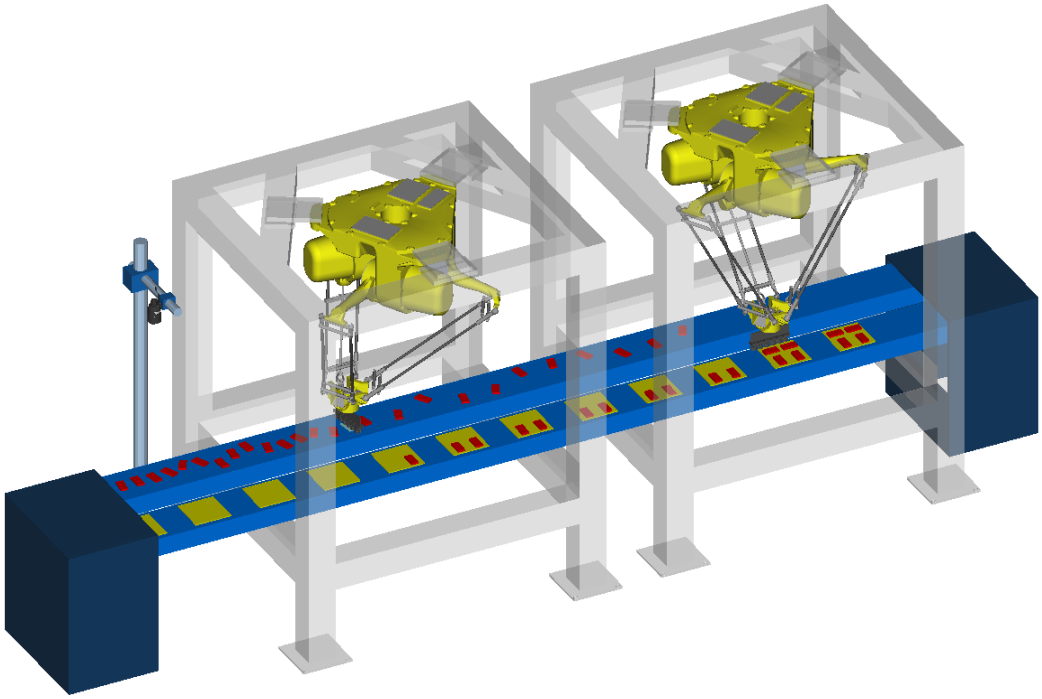


图 2(a) 视觉追踪的使用例子
(使用图左面的相机)

2.1 软件选项的构成

*i*RPickTool 是如下图所示的软件构成。

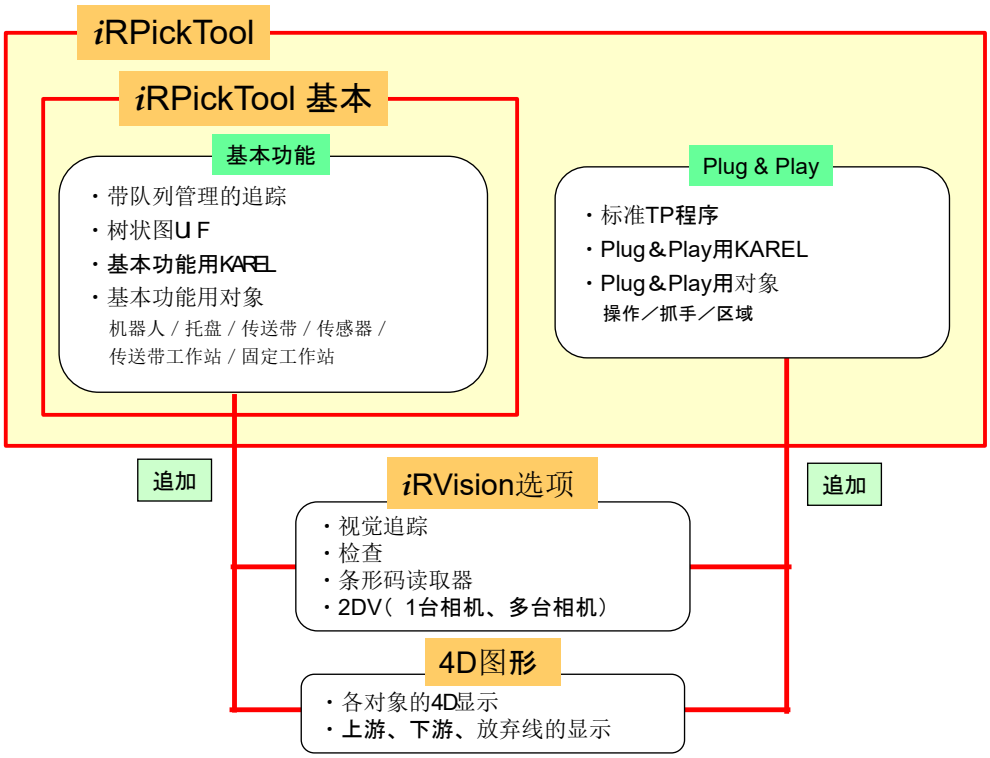


图 2.1(a) *i*RPickTool 软件构成图

*i*RPickTool 基本，具有用于构筑带队列管理的追踪系统的基本功能。由此，通过添加 *i*RVision 选项，可进行使用相机的视觉追踪。

*i*RPickTool 包括有 *i*RPickTool 基本所具有的功能，通过还具有标准 TP 程序等，有助于进一步方便设定。将这个便于进行启动调试的功能称为 Plug & Play。
*i*RPickTool 基本与 *i*RPickTool 的不同在于 Plug & Play 的有无。

> ⚠ **注意**
> **如果安装了 *i*RPickTool(J945)，则不能作为 *i*RPickTool 基本(J944)来使用。例如，不能将 *i*RPickTool 基本(J944)的备份读入到安装有 *i*RPickTool(J945)的机器人控制装置中使用。**

2.2 机器构成

以下对带队列管理的追踪系统，以及使用 *i*RVision 的视觉追踪系统中的机器构成进行说明。

2.2.1 带队列管理的追踪

带队列管理的追踪系统，由如下机器构成。

系统稼动所需的机器

- 机器人控制装置
- 机器人
- 传送带
- 光电管传感器
- 脉冲编码器
- 追踪拾取用接口板（设置于机器人控制装置内）
- 脉冲编码器电缆
- 脉冲分配器（在多台机器人不使用以太网编码器时需要。不使用脉冲编码器电缆，但同时需要脉冲分配器－脉冲编码器用电缆和脉冲分配器用电缆。）
- 以太网通信电缆
- 以太网集线器

仅在系统启动时需要的机器

- 对应触摸屏的 iPendant（或示教用电脑和以太网通信电缆）
- 触针

机器构成例

使用以太网编码器的情况

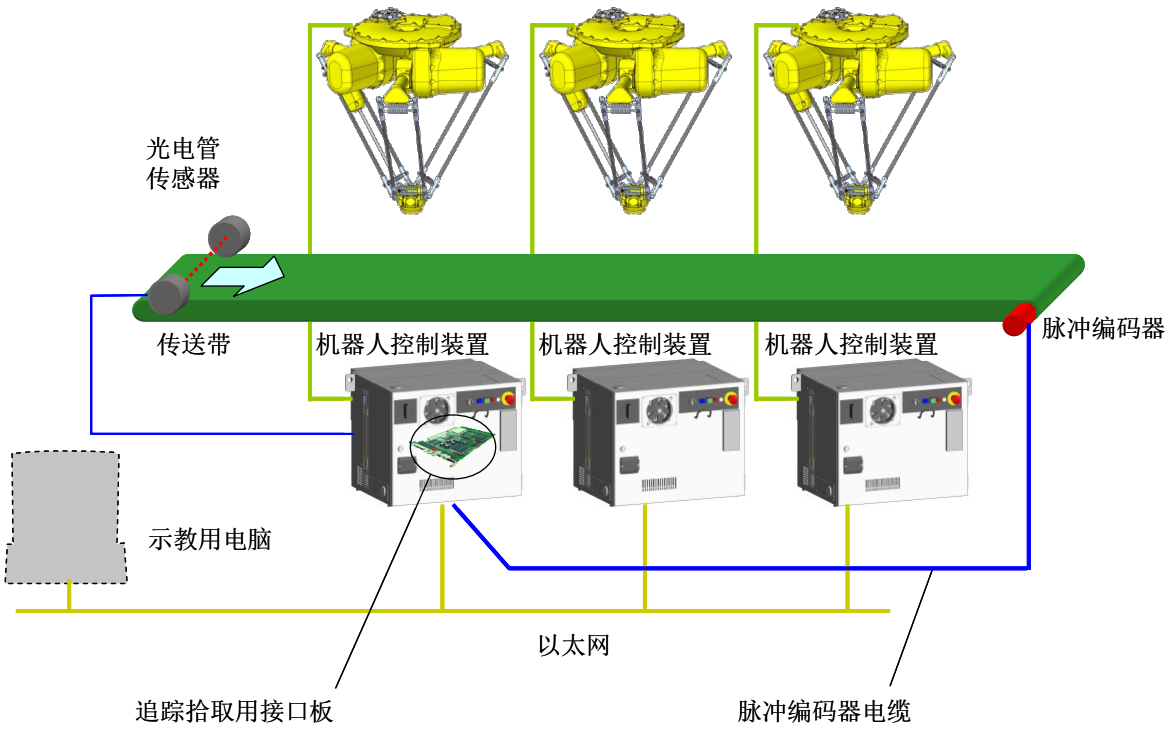


图 2.2.1 (a)

使用脉冲分配器的情况

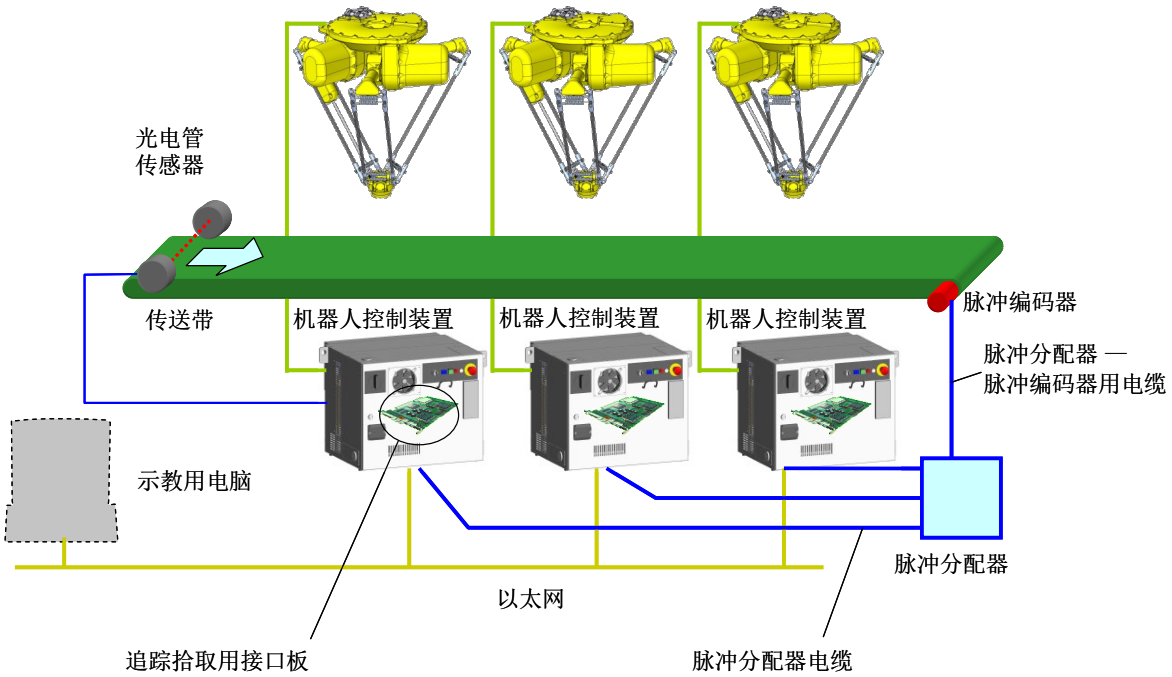


图 2.2.1 (b)

2.2.2 视觉追踪

视觉追踪系统，由如下机器构成。

系统稼动所需的机器

- 机器人控制装置
- 机器人
- 传送带
- 光电管传感器（在相机拍摄的触发器中使用时需要）
- 相机
- 镜头
- 电缆
- 照明
- 脉冲编码器
- 追踪拾取用接口板（设置于机器人控制装置内）
- 脉冲编码器电缆
- 脉冲分配器（在多台机器人不使用以太网编码器时需要。不使用脉冲编码器电缆，但同时需要脉冲分配器－脉冲编码器用电缆和脉冲分配器用电缆。）
- 以太网通信电缆
- 以太网集线器

仅在系统启动时需要的机器

- 对应触摸屏的 *i*Pendant（或示教用电脑和以太网通信电缆）
- 点阵板夹具（相机校准板）
- 触针

机器构成例

使用以太网编码器的情况

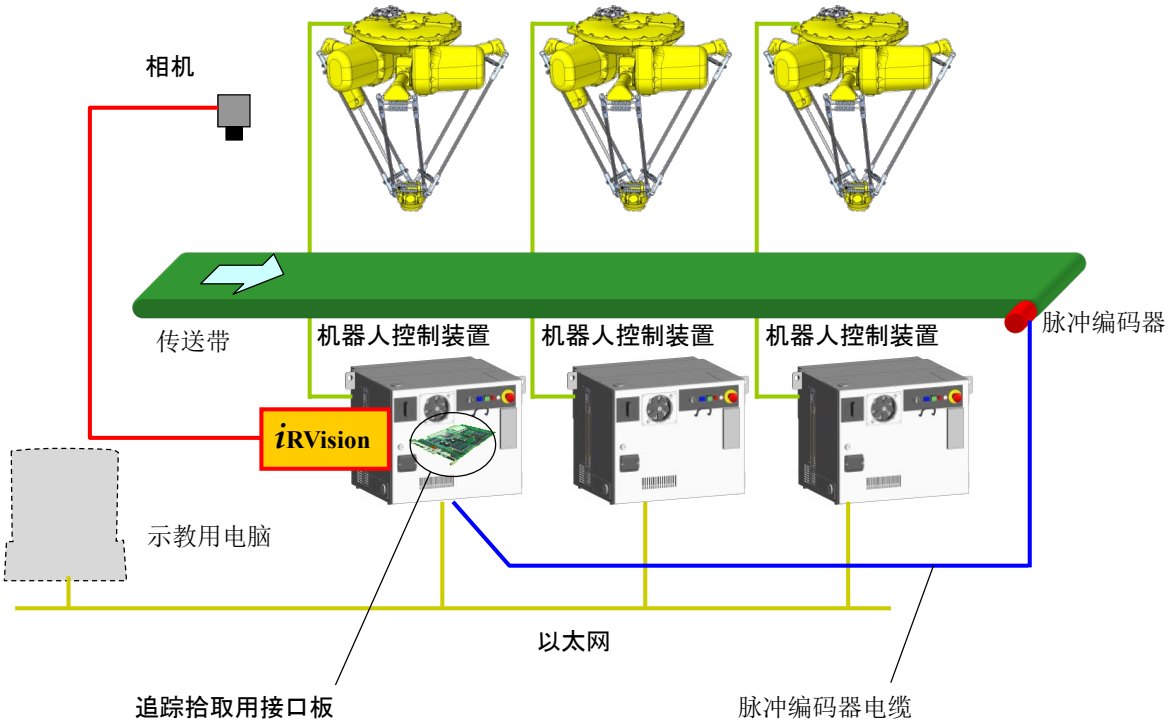


图 2.2.2 (a)

使用脉冲分配器的情况

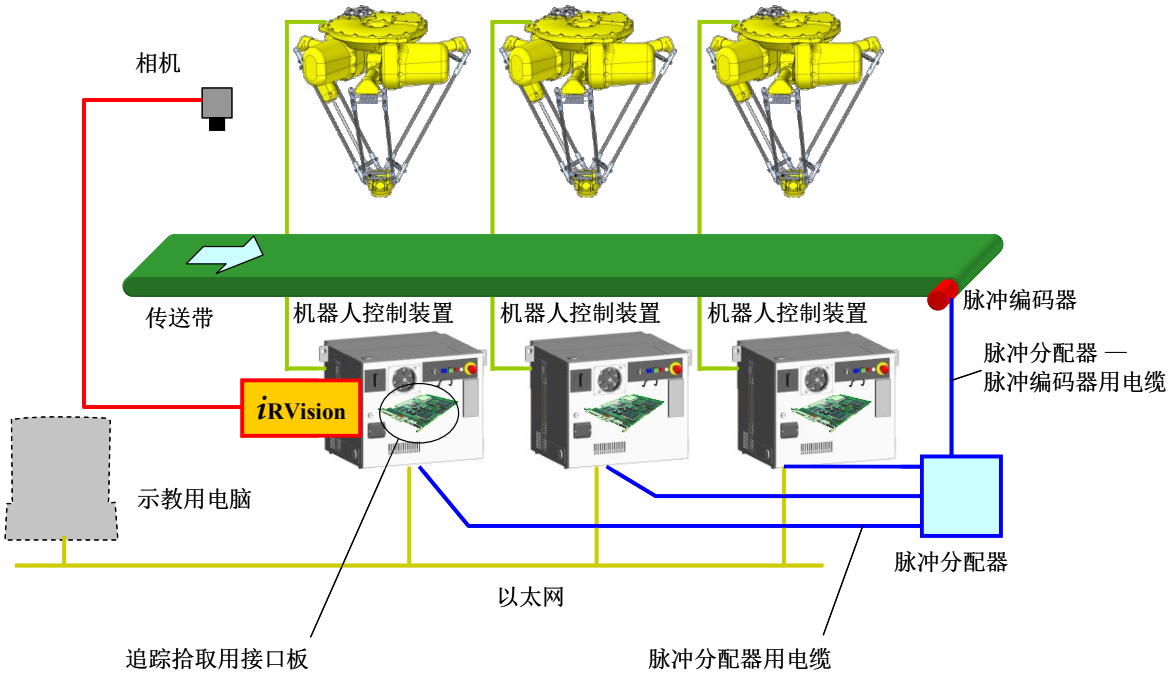


图 2.2.2 (b)

2.3 用语

本节对本册使用的用语进行解说。
用 *i*RPickTool 构造追踪系统时，使用以下概念。

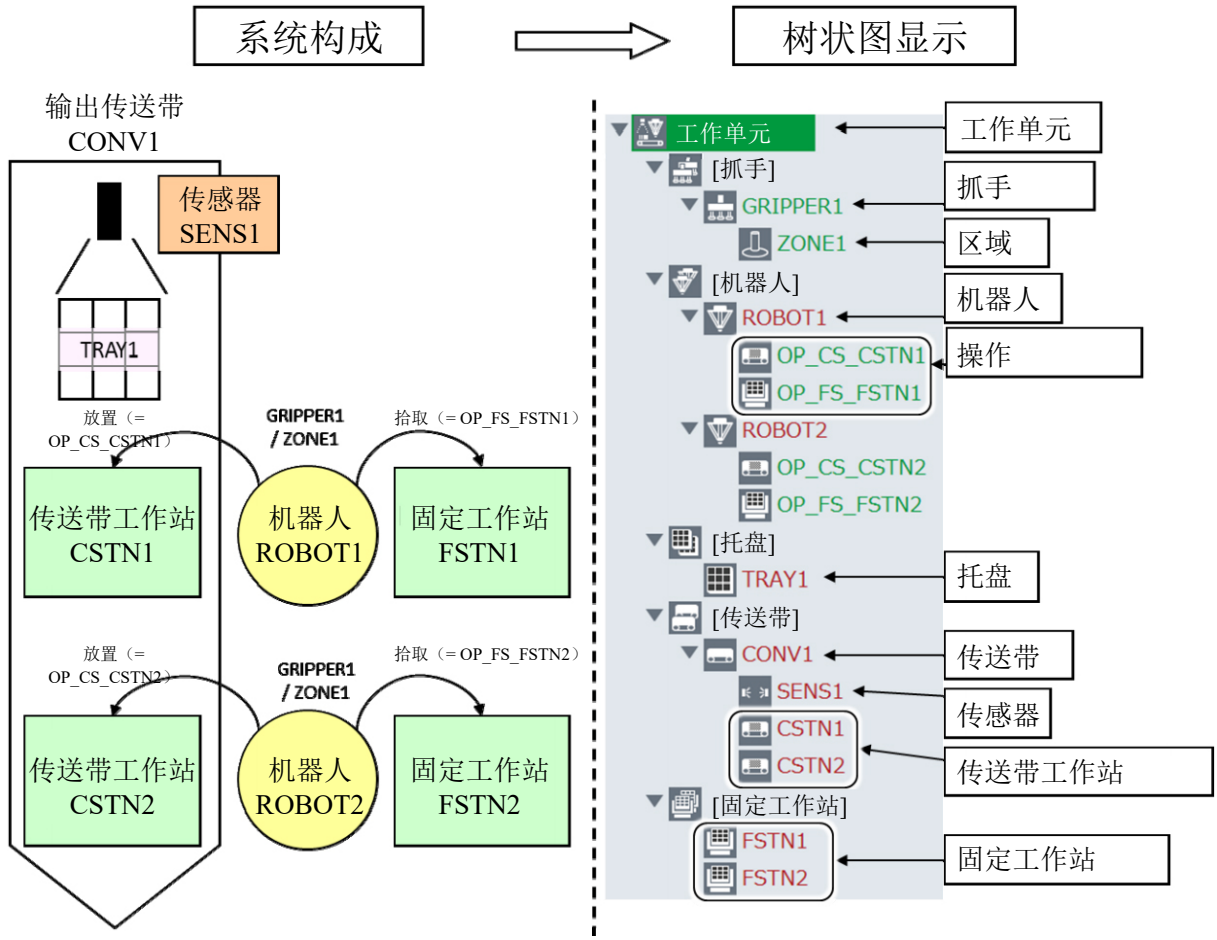


图 2.3(a)系统构成与 *i*RPickTool 树状图标识

*i*RPickTool 中，使用图 2.3(a)中的树状图进行设定。该树状图中使用的 1 个个构成要素称为“对象”。根据树状图，从上方依次设定对象。

[工作单元]

表示追踪系统整体构成的对象。各机器人控制装置中，分别只能具有 1 个。通过改变步骤等变更系统构成时，使用后面叙述的方案，更改工作单元整体。

[抓手]

表示安装在机器人的标牌上的机械手的对象。仅包括于 Plug & Play 中。
从后面叙述的机器人进行参照。即使是相同工作单元的机器人，如果各种具有不同的机械手，也需要进行不同抓手的设定。反之，如果具有相同机械手，在机器人间，可共有相同抓手的设定。

[区域]

在属于抓手的对象中，表示机械手抓住部的对象。仅包括于 Plug & Play 中。
例如，在 1 个机械手中，具有 2 个系统的抓住部的情况（=2 次去取工件的情况），设定 2 个区域。下图中的机械手，有 6 个吸盘，在 3 个空气的系统，用 1 个机械手取 3 次工件。这种情况，需要 3 个区域数。

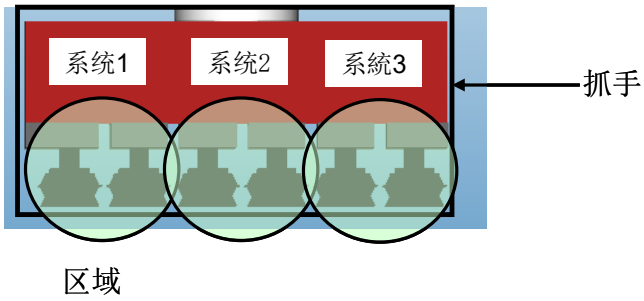


图 2.3 (b)

[机器人]

表示进行作业的机器人的对象。根据机器人的数量，需要生成相同数量的对象。

[操作]

属于机器人的对象只包括在 Plug & Play 中。表示在后述传送工作站与固定工作站进行拾取/放置等机器人动作的对象。传送带工作站和固定工作站 1 对 1 对应，如果生成各个工作站，则也自动生成操作。

[托盘]

根据系统，可能会有将从供给传送带上拾取的工件，装入输出传送带传送过来的“箱”中的作业。通常，“箱”中可放多个工件。托盘是表示该“箱”的对象。

[单元]

托盘中放置工件(或从托盘拾取)的地方的每个位置称为单元。例如，其中可排放有 4 个工件的箱子，是具有 4 个单元的托盘类型。

[托盘坐标系]

记述托盘中单元配置时所使用的坐标系。将各单元的位置和方向，设定为从该托盘坐标系看到的坐标值。可定义于喜欢的托盘位置。

对托盘类型进行定义后，只要对机器人示教相对其中的 1 个单元的放置（或拾取）位置等，就能同样对其他单元放置（或拾取）工件。
此外，托盘的哪个单元放置有工件（或从哪个单元拾取）的信息受到管理，因此可由多台机器人分担处理托盘。

为了方便更具体的理解，以具有 4 个单元的托盘为例进行说明。

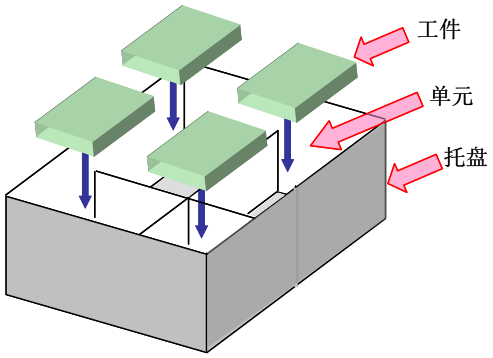


图 2.3(c)例：具有 4 个单元的托盘

图 2.3(c)例子中为将 4 个工件放在箱中（或从箱中取出）。
箱子本身称为“托盘”，箱子中的放置（或拾取）工件的位置称为“单元”。
也就是说，该托盘具有 4 个单元，图示即如下图。
图 2.3(d)为图 2.3(c)托盘的俯视图。本例中，托盘坐标系定义于箱子左下角。

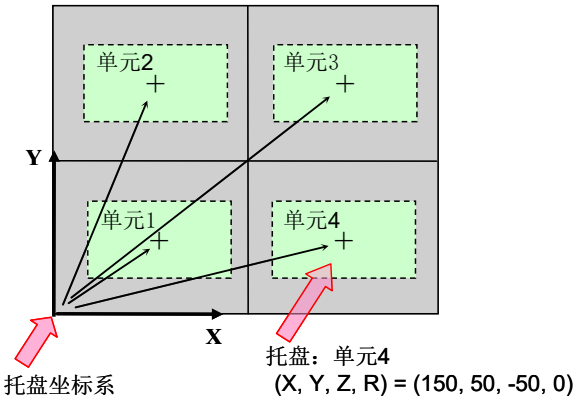


图 2.3(d)将坐标叠放在具有 4 个单元的托盘处的例子

+标记表示各单元的中心位置。单元的位置，设定为从托盘坐标系看到的相对位置。

托盘坐标系的位置，可以在托盘上自由决定，但是在向机器人指导托盘坐标系的位置时，必须用机器人的 TCP 碰触托盘坐标系的原点和 X 轴方向点，因此请定位于容易碰触的位置。

[传送带]

由后面叙述的传感器和传送带工作站构成的对象。表示在传送带上传送的工件，由哪个传感器检知，由哪个传送带工作站向哪个传送带工作站，以什么顺序传送。可以认为以假想方式表示实际的传送带状况。

[传感器]

检出传送带上传送的工件，将检出结果输入在传送带上定义的最上游的传送带工作站的对象。对应使用视觉检出工件，以及不使用视觉，而使用光电管传感器等检出工件的两种情况。

> **注释**
> **实际系统中，自动运行根据“传感器”的设定进行处理的 KAREL 程序。该 KAREL 程序，可根据需要进行自定义，但是一般情况下，不提供 KAREL 程序的源代码。需要自定义时，请参照“附录 H　传感器任务的自定义”或向本公司营业人员咨询。**

[传送带工作站]

传送带工作站，是表示在传送带上，机器人进行作业的区域的对象。机器人从传送带工作站拾取工件，将工件放置于其他的传送带工作站。

为了更具体地理解传送带、传感器、传送带工作站的含意，以下例进行说明。
图 2.3(e)图表示利用 2 台机器人、2 根传送带的视觉追踪应用程序。
传送带 1 为供给传送带，传送带 2 为输出传送带。
机器人 1、2 都从传送带 1 取工件，将拾取的工件放置于传送带 2 上。传送带 1 的上游端设置有 1 台相机，由相机检出传送来的工件。传送带 2 的上游端设置有 1 台光电管传感器，由光电管传感器检出传送来的箱子。将从传送带 1 拾取的工件，装入传送带 2 上传送的箱子内的系统。

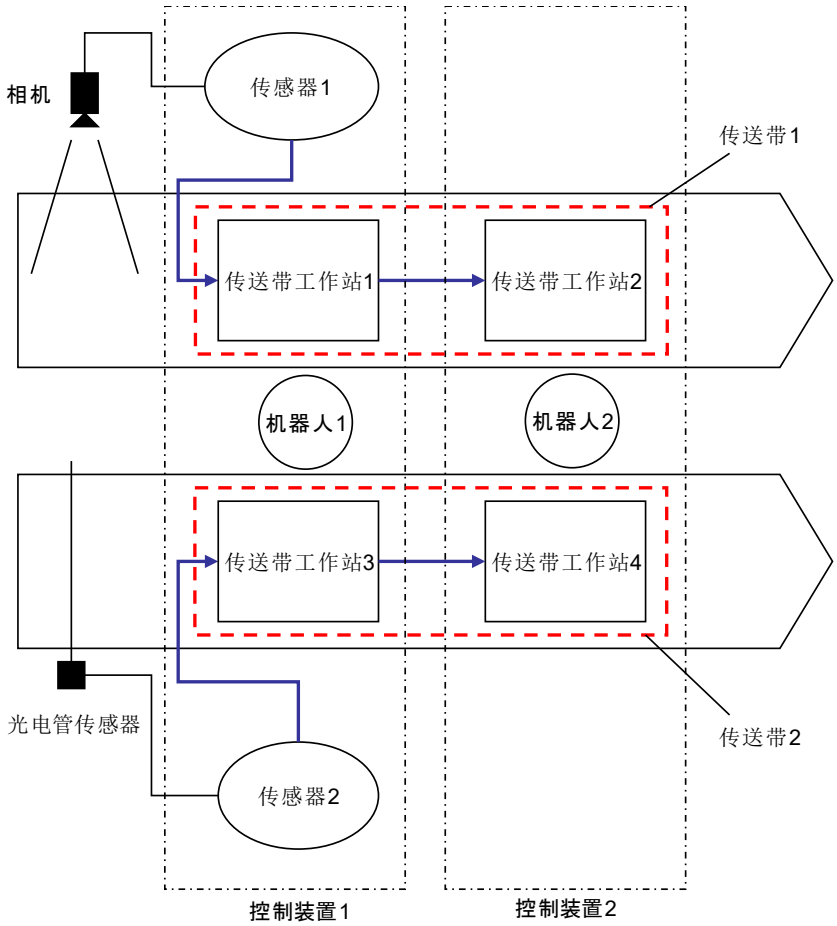


图 2.3(e)传送带工作站的系统示例 1

该系统有 2 根传送带、4 个传送带工作站、2 个传感器和 2 台机器人。

传送带工作站 1 和 2 都定义在传送带 1 上，传送带工作站 3 和 4 都定义在传送带 2 上。
机器人 1 从传送带 1 上的传送带工作站 1 拾取，放置于传动带 2 上的传送带工作站 3 上。机器人 2 从传送带 1 上的传送带工作站 2 拾取，放置于传送带 2 上的传送带工作站 4 上。
传感器 1 利用相机检测传送带 1 上的工件。检出到的工件信息输入到传送带工作站 1。传感器 2 利用光电管传感器检测传送带 2 上的工件。检出到的工件信息输入到传送带工作站 3。
传感器 1 输入传送带工作站 1 的工件信息，时刻配合实际工件的传送，从传送带工作站 1 转移至传送带工作站 2。
同样，传感器 2 输入传送带工作站 3 的工件信息，从传送带工作站 3 转移至传送带工作站 4。

各机器人从传送带工作站取得传送来的工件的相应信息，进行拾取工件，或装入箱子等的放置作业。
如果工件不被传送至传送带工作站的位置，则该工件的信息不会被传达至机器人。
此外，如果工件已通过该传送带工作站，则该工件的信息转移至下一个传送带工作站。
根据各传送带工作站中设定的分配比率、现在位置的信息，工件的信息被传达至各机器人。

一般情况下，对于 1 根传送带定义 1 个传送带，但如下图所示，在相同传送带上串联配置 2 个传感器时，可能对 1 根传送带根据不同区域定义成 2 根传送带。

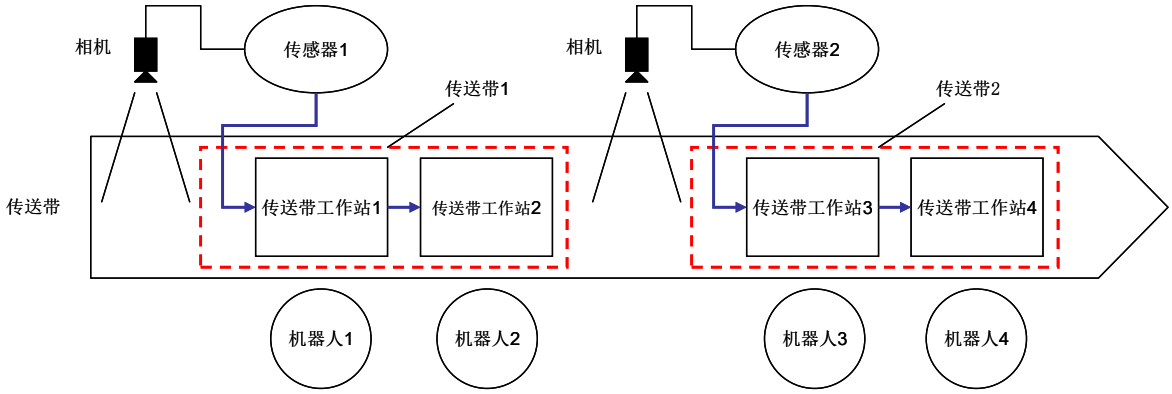


图 2.3(f)传送带工作站的系统示例 2

追踪参数设置

这是机器人为了在传送带工作站上进行作业的设定（追踪坐标系等）。

追踪参数设置可以设定 8 个。若在 TP 程序的属性中指定追踪参数设置编号，机器人就会根据所设定的数据执行追踪动作。

［固定工作站］

相对于传送带工作站定义于“动作”传送带上的机器人作业区域，固定工作站是表示定义于规定为“不动作”位置的作业区域的装置。与传送带工作站相同，机器人可从固定工作站拾取或放置工件。由于是“不动作”领域，与传送带工作站不同，工作站之间没有联系，每个固定工作站各自独立。

下图是从图 2.3(e)的构成，将输出传送带换成固定工作站的情况。机器人 1、2 都从传送带 1 取工件，将拾取的工件放置于固定工作站 1,2。包括间隔进给的传送带，每次将放置拾取的工件的箱子设置于相同位置的系统。

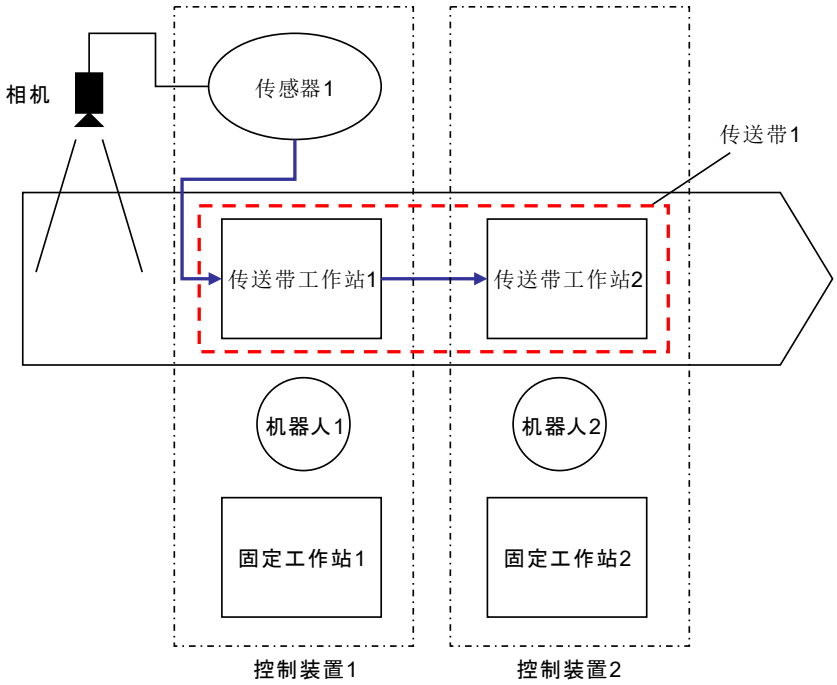


图 2.3 (g)固定工作站的系统示例

2.4 系统构成例

为了更具体地理解可通过 *i*RPickTool 构筑怎样的系统，本节介绍几个系统构成例。

构成例 1

最基本的构成。各机器人拾取供给传送带上传送来的工件，放置于与流向供给传送带相同方向的输出传送带。

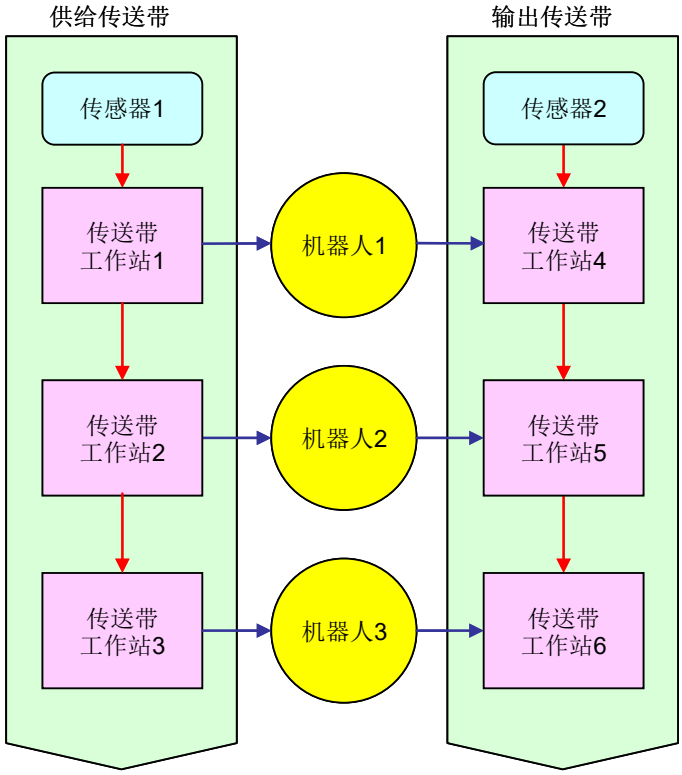


图 2.4 (a)

构成例 2

构成例 1 的输出传送带成为固定工作站的例子。与该构成相反，也可以有供给侧为固定工作站，排出侧为传送带的构成。

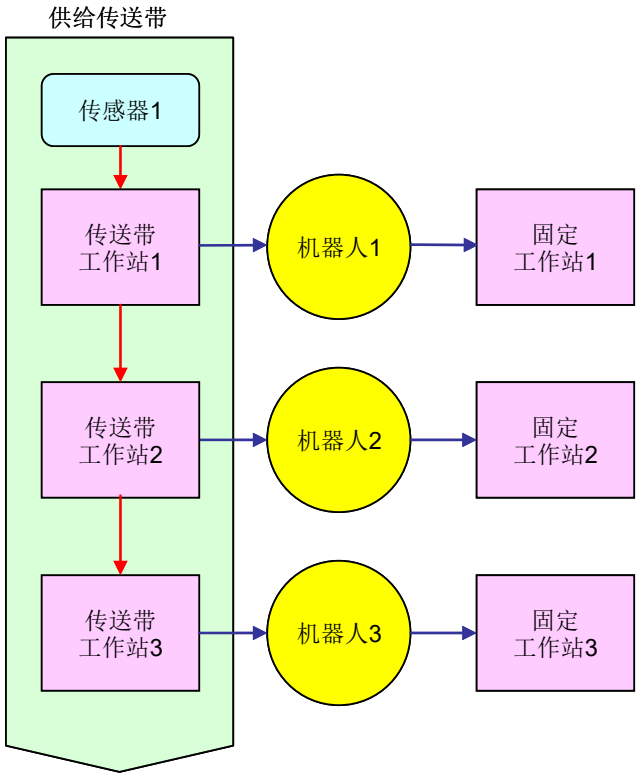


图 2.4 (b)

构成例 3

在构成例 1 中，是缓冲添加固定工作站的构成例子。
如果从供给传送带供给的工件为不规则，在供给过剩时，可先将工件放置在固定工作站积存。
工件供给过少时，从固定工件站取出工件，放置在输出传送带的传送带工作站。

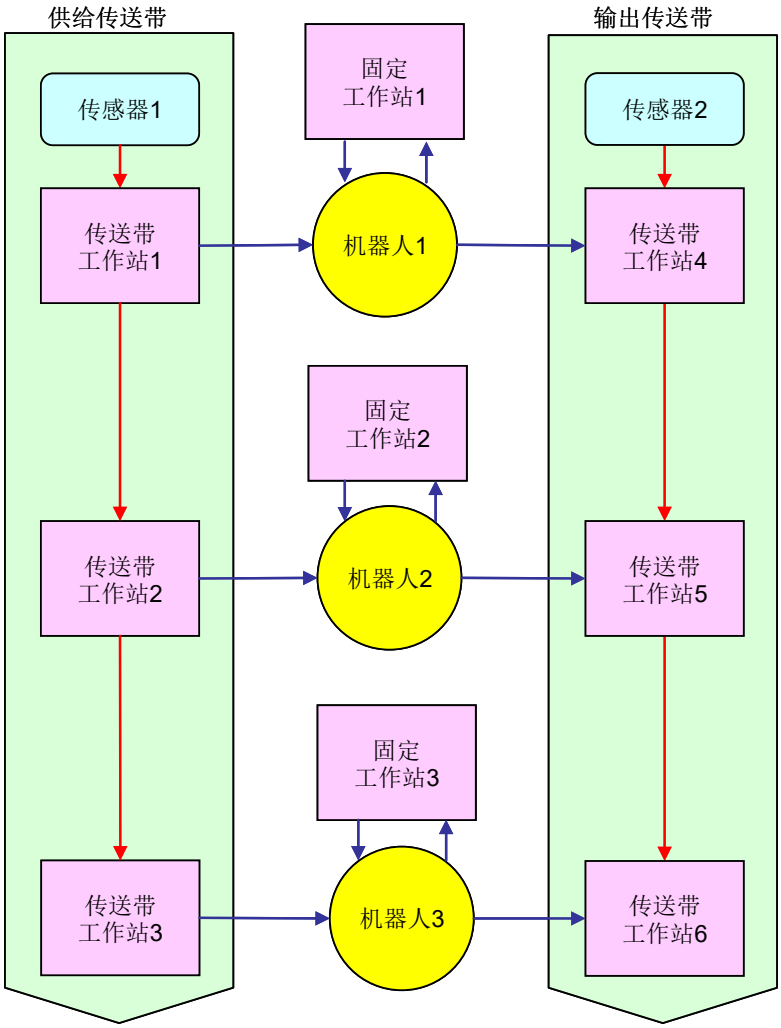


图 2.4 (c)

构成例 4

基本与构成例 1 相同。供给传送带与输出传送带的流向相反。

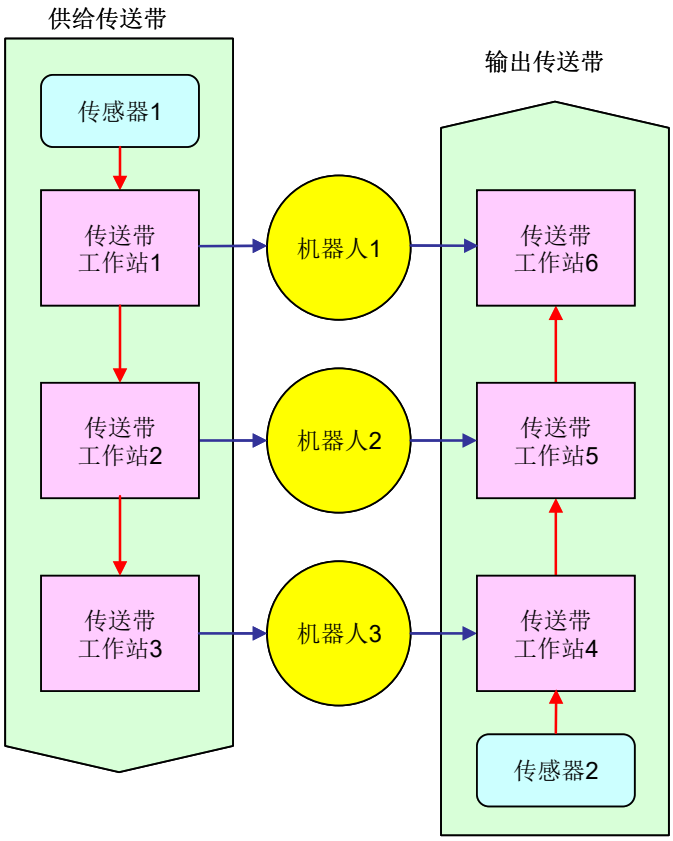


图 2.4 (d)

构成例 5

本例基本也与构成例 1 相同，但是传送带 1 的横向宽度宽，1 台相机无法覆盖传送带整体。这种情况的构成是，在供给传送带上设置 2 个相机，分别覆盖右半部分和左半部分，2 个传感器向相同的传送带工作站输入工件的信息。

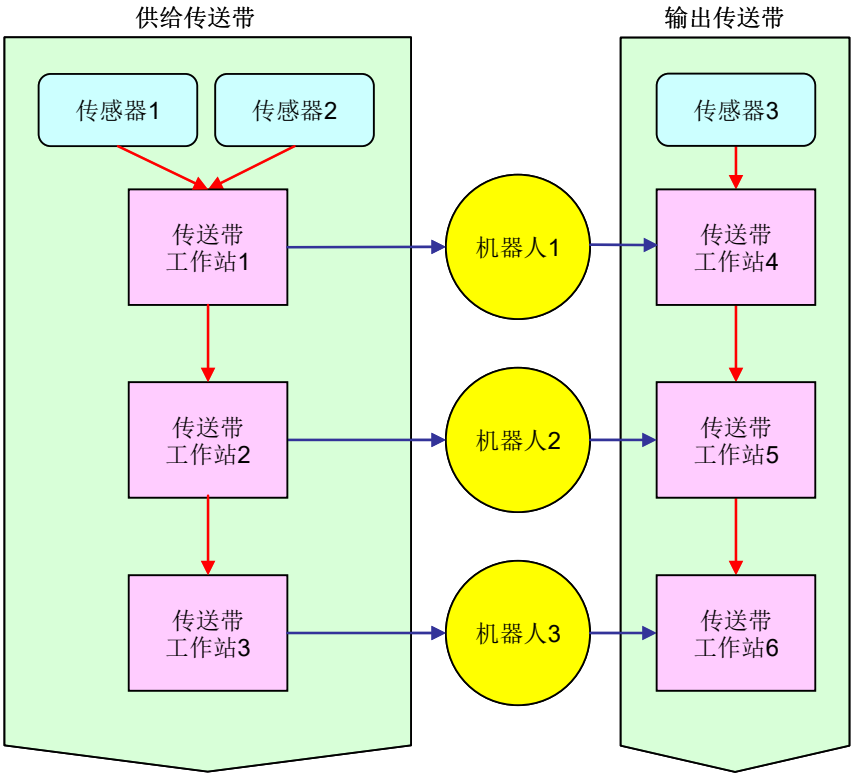


图 2.4 (e)

构成例 6

分别有 2 根供给传送带、2 根输出传送带，供给传送带和输出传送带交叉配置的稍复杂的系统。

虽然看似复杂，但是如果注意各个机器人的动作，会发现例如机器人 1 从传送带工作站 1 拾取工件，放置于传送带工作站 5，机器人的动作本身与构成例 1 几乎没有差异。

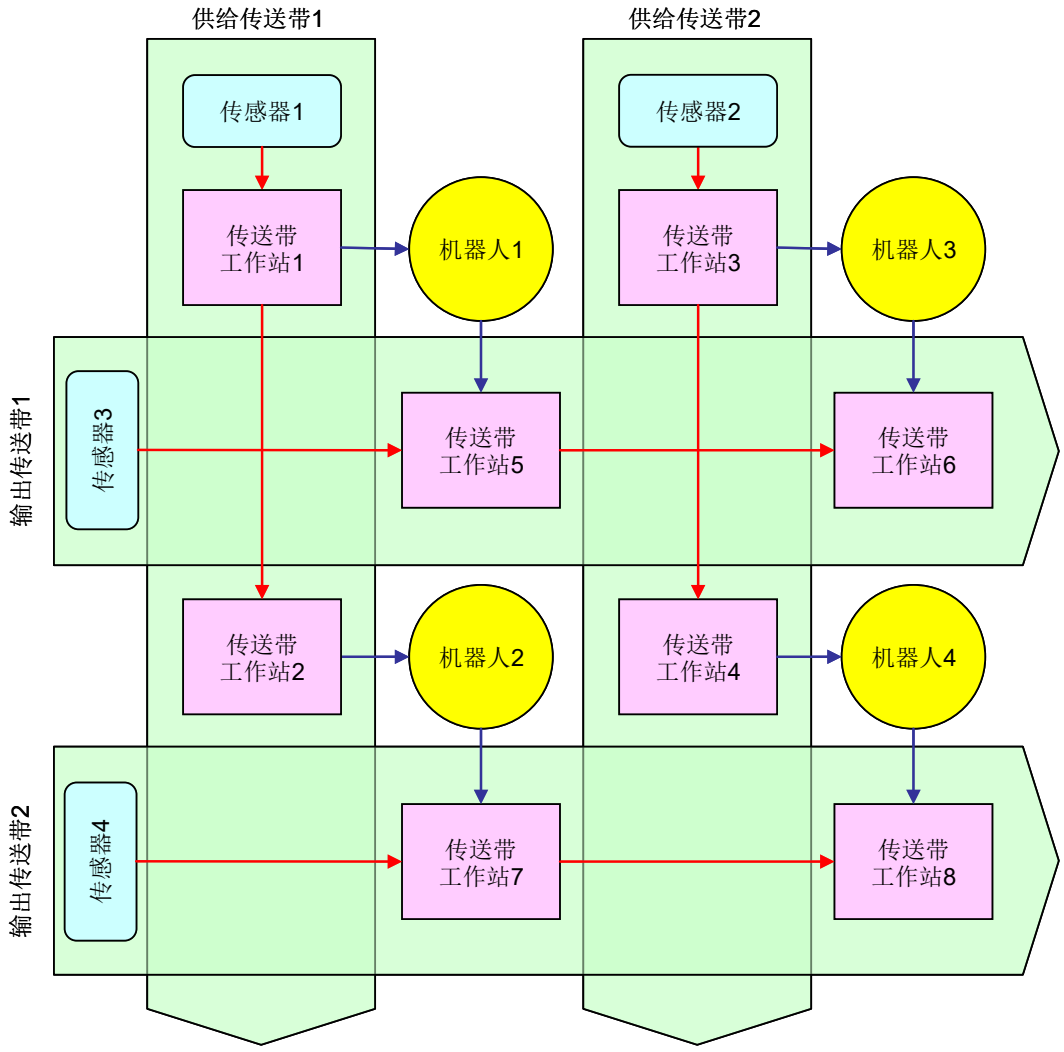


图 2.4 (f)

构成例 7

供给传送带和输出传送带，物理上为相同传送带的情况。
传送带上凌乱传送的工件，在相同传送带上被重新排列的情况，属于该情况。
传感器 1 通过相机检出传送带上的工件，传感器 2 通过接近传感器等，在传送带上等间隔地检出记录的标记，这样可在传送带上等间隔地调整凌乱传送来的工件。
使用 1 根传送带，但需要创建供给侧和输出侧 2 个传送带对象。

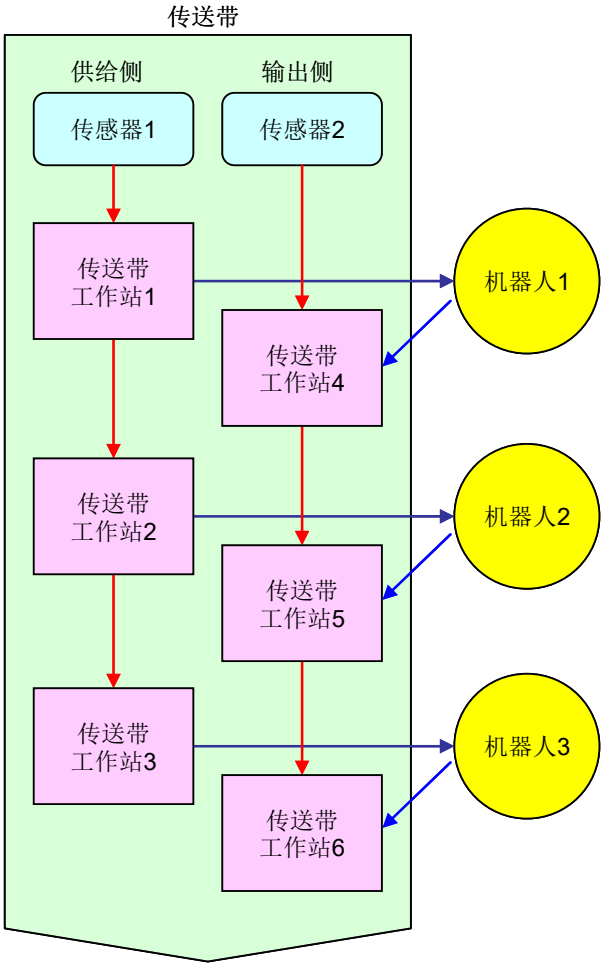


图 2.4 (g)

构成例 8

供给侧的传送带有 2 根，各机器人从其中的供给传送带拾取工件，放置于输出传送带。
有 2 种工件，供给传送带 1 上传送种类 A 工件，供给传送带 2 上传送种类 B 的工件，输出传送带上传送的箱子里混合放置 A 和 B 两种工件的情况属于这种情况。

实施这种构成时，机器人的程序稍复杂。
实际成型的例子，在“8 不同用途的设定例子”中介绍。

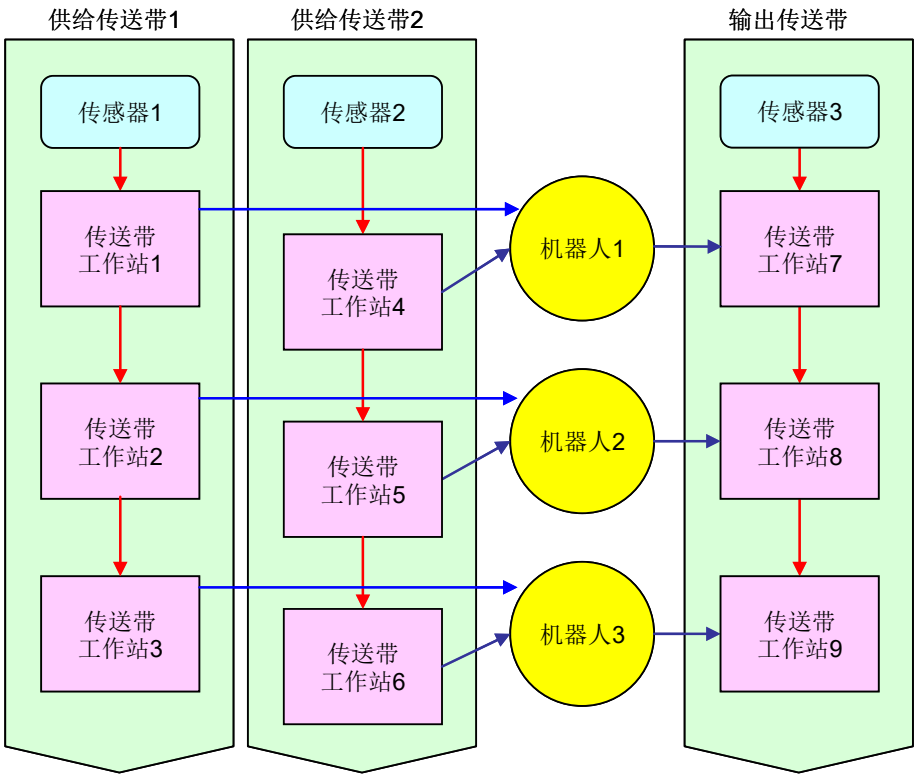


图 2.4 (h)

2.5 适用探讨

本节针对使用 *i*RPickTool 的系统布局、机器人与相机的选定、脉冲计数的分配方法进行说明。

系统布局

将从传送带上拾取的工件放置于其他传送带，或货盘等的移动距离越短，1 次往返所需的时间也越短。对机器人、传送带、货盘等机器进行布局时，应使机器人的动作距离尽量短。

机器人

考虑要求搬送数量、工件的重量等，决定机器人的种类、机器人的台数、机械手等。

根据机器人的种类不同，其最高速度也不同。必须考虑机器人的移动距离、机械手的开闭时间。越是高速的机器人，其 1 分钟内能往返所定距离的次数也越多，因此从搬送数量角度看有利，但是其可搬运量一般较小。

如果使用可取多个的机械手，由于其放置工件的往返次数减少，因此 1 台机器人能处理的搬送数量增加。另一方面，由于机械手变重，请注意机械手+工件的重量不得超过机器人的可搬运质量。此外，一般情况下，由于机械手会变大，可能发生与周围设备/机器人等的干涉问题。

> **⚠ 注意**
> **使用 4 轴～5 轴机器人的情况，设置机器人和传送带，使世界坐标系的 XY 平面和传送带面平行。如果根据本说明书说明的视觉追踪的标准设定方法，设定 4 轴～5 轴机器人，追踪面被自动设定为与世界坐标系的 XY 平面平行。如果机器人的世界坐标系的 XY 平面与传送带面不平行设置，追踪的精度将会变低。**

工件的检出方法

有关各传送带，需要探讨用怎样的传感器检出传送带上传送的工件。大致可分为 2 种方法。

用光电管传感器检出

在传送带的上游侧设置光电管传感器，通过确认工件是否横切光电管来检测工件。光电管传感器只能认识传送带传送方向的位置，传送来的工件的方向、横断传送带的方向的位置必须一定。

用相机检出

在传送带的上游侧设置相机，用相机拍摄传送带上的图像，用 *i*RVision 检出图像中拍摄的工件。无论工件在传送带上的放置位置、方向如何，都可以通过视觉认识工件的位置、方向，进行正确拾取。

此外，检出方法有 2 种，用相机检出工件时，每次传送带进给一定距离，都用相机拍摄图像，无间隙地持续监视传送带上情况的方法；以及同时设置光电管传感器，仅在工件横断光电管传感器时，用相机拍摄图像进行检出的方法。

> **⚠ 注意**
> **以多台机器人使用多台相机的情况，其构成为：1 台机器人控制装置上，不连接 2 台以上的相机。用于将图像处理的负担分散至各机器人控制装置。**

脉冲计数的分配方法

探讨在多台机器人的情况下，如果将脉冲编码器的脉冲计数分配至各机器人。有以下 2 种方法。

通过脉冲分配器进行分配

以脉冲分配器为中继，向各机器人控制装置直接输入脉冲计数。各机器人控制装置认识的脉冲计数全部相等，因此可进行高精度的追踪。

但是，系统嫁动所需的机器增加。全部的控制装置，需要追踪拾取用接口板。

此外，还需要连接各控制装置和脉冲分配器的电缆，以及连接脉冲分配器和脉冲编码器的电缆。

通过以太网编码器进行分配

利用机器人之间的通信，将输入到连接有脉冲编码器的机器人控制装置的脉冲计数分配给其他各机器人控制装置。与使用脉冲分配器时比较，不需要脉冲分配器、连接脉冲分配器和各机器人控制装置的电缆、从控的各机器人控制装置的追踪拾取用接口板，因此可削减硬件成本。

但是，由于通信的影响，各机器人控制装置认识的脉冲计数，会发生稍许差异。对于使用脉冲分配器的情况，换算成时间，可能会有–2.0ms 至+3.0ms 左右的差异。例如，传送带速度为 400mm/s 的情况，搬运位置的差异–0.8mm 至+1.2mm 左右。

⚠ 注意

1 使用以太网编码器时，为减轻通信负担，在机器人间通信的主控制装置上，连接脉冲编码器。

2 使用以太网编码器时，为了取得检出工件瞬间的正确脉冲计数，在连接脉冲编码器的机器人控制装置上，连接相机、传感器。与 1 的注意事项合并，在机器人间通信的主控制装置上，连接脉冲编码器和相机、传感器。

3 使用 2 台以上相机的系统中，建议使用脉冲分配器。如之前注意中所述，如果在各机器人控制装置上连接 1 台相机，在使用以太网编码器时，从控的控制装置上也应连接相机。从控的控制装置无法取得检出工件瞬间的正确脉冲计数，因此搬运位置的差异可能比上述更大。

传送带速度

搬送量由传送带速度和传送带上的工件密度决定。传送带速度为 200 mm/s，工件以平均 100 mm 间隔传送的情况，搬送量为 120 个/分，而传送带速度为 100 mm/s，以 50 mm 间隔传送工件，其搬送量还是 120 个/分，搬送量没有不同。这种情况，传送带速度较慢，则机器人的追踪精度、视觉的检出精度变高，因此较慢的传送带的系统的整体性能较佳。

相机的视野范围

探讨使用相机检出工件时，相机视野的大小应为多大。视野的大小，根据镜头和相机的种类，以及设置相机的高度决定。计算式如下所示。

Y =y × （（L – f ）÷ f ）

Y: 视野范围
y: CCD 尺寸
L: 相机－工件间的距离
f: 焦点距离

如图所示，其位置关系如下。

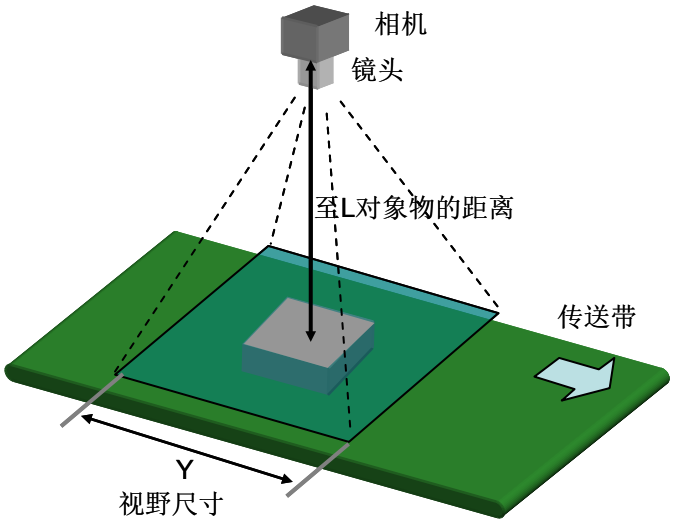


图 2.5(a)决定相机视野范围的位置关系

CCD 尺寸，是相机的 CCD(拍摄元件)的大小。用下面计算式求得。

CCD 尺寸 =单元尺寸 × 图像尺寸

下表是可以在 FANUC 标准配置的相机的 CCD 尺寸。

表 2.5(a)相机的种类与像素

相机	单元尺寸	最大图像尺寸	CCD 尺寸
黑白相机 SC130EF2	**5.300 μm**	**1280 ×1024**	**6.784 mm ×5.427 mm**
彩色相机 SC130EF2C	**10.600 μm**	**640 ×512**	**6.784 mm ×5.427 mm**

例如，焦距 8 mm、相机和工件的距离约 1000 mm 时，视野范围如下。

水平方向：6.784（mm）×（（1000（mm）– 8（mm））÷ 8（mm））= 841.2（mm）
垂直方向：5.427（mm）×（（1000（mm）– 8（mm））÷ 8（mm））= 672.9（mm）

注释
视野范围是可以检出工件的范围，请尽量设定得小。如果视野范围小，每个像素拍摄的实际空间的范围变小，视觉的位置检出的分辨率也提高。

曝光时间

探讨使用相机检出工件时，相机的曝光时间应为多少。
曝光中如果工件移动，则移动部分为模糊图像，检出精度降低。决定曝光时间时，尽量缩小该偏移量。
工件的移动量依存于传送带的速度，因此曝光时间依存于传送带的速度。传送带速度越快，则应使曝光时间越短。相反，传送带速度越慢，则可以使曝光时间变长。
但是，曝光时间的延长，会导致视觉检出整体时间的延长，因此，速度较慢的传送带的情况，也应在一定程度上缩短曝光时间。

根据传送带速度的适当曝光时间，使用下面式子进行计算。

$E \leqq \Delta \div V$

E ：曝光时间（ms）
Δ ：位置的偏移（mm）
V ：传送带速度（mm/s）

例如，传送带速度 100 mm/s 时，为了将利用视觉检出时的位置偏移控制在 0.1 mm 以下，0.1(mm)÷100(mm/s)= 1(ms)
将小于 1 ms 的值设定为曝光时间。

曝光时间越短，越需要强力的照明。请准备照明，在上述概算的曝光时间下，能拍摄足够明亮的图像。

视觉的启动间隔

使用相机检出传送带上的工件时，需要注意以多少间隔运行视觉的程序。为了防止视觉遗漏检出传送带上的工件，传送带每前进一定距离，就需要进行图像处理。因此，启动间隔严重受传送带速度的影响。也就是说，传送带速度越快，就越频繁地进行图像处理。相反，传送带速度越慢，需要的图像处理的频率就下降。

视觉的启动间隔，可通过下面式子概算。

启动间隔（s）=0.5 × 相机视野尺寸（mm）÷传送带速度（mm/s）

如果增大相机视野，则视觉的启动时间增长，另一方面，检出精度随相机视野增大而降低。

搬运精度

视觉追踪的搬运精度，除受视觉的检出精度导致的误差的影响外，还受追踪坐标系的设定误差、TCP 的设定误差、机器人的位置的示教误差、机器人的追踪延迟等的影响。此外，可能发生传送带的传送中有微震的情况，或大物品的冲压品等由于工件本身在传送带上的晃动，而导致搬运精度下降的情况。

可通过实施以下对策，改善搬运精度。

- 正确设定碰触用 TCP。
- 一次检出多个工件等，放慢传送带速度。
- 准备充分的光量照明，缩短曝光时间。
- 调整追踪相关的设定以及动作。

负载平衡

1 根传送带由多台机器人分担进行工件处理时，可指定各机器人拾取工件分量、工件的种类。这种设定“由哪台机器人以何种比例处理哪个工件”的设定，称为负载平衡。

如果使用负载平衡功能，可指定各机器人处理工件的量的比率。通过这类指定，改变交付工件信息的机器人的操作，称为“分配”。典型的分配方法，是各机器人处理相同的量的“等分配”。进行等分配时，负载平衡对例如 2 台机器人指定为 1:1。如果机器人的处理能力不同等，也可以指定为比如 2:1。

如果负载平衡功能无效，则各机器人会尽量多地处理工件。结果，发挥该系统可达成的最大搬送能力，但是如果传送的工件个数较少，则会发生上游侧的机器人处理更多的工件，下游侧机器人处只会有少量的工件的情况。

使用相机检出工件时，除处理量的指定外，还可根据工件的种类，指定由哪个机器人进行处理。例如，圆形工件，由机器人 1 处理，三角形工件，由机器人 2 处理的情况。

根据种类进行分配时，使用 *i*RVision 的模型 ID。模型 ID，典型的是用于工件种类的判别，而通过与 *i*RVision 的条件判断工具的组合使用，也能用于工件种类的判别以外的各种用途。

例如，

- 根据工件的检出位置进行分配(对传送带中央靠右侧的工件，由机器人 1 处理，左侧的工件，由机器人 2 处理等)
- 根据工件的方向进行分配（方向为 0～＋180 度的工件，由机器人 1 处理，方向为－180～0 度的工件，由机器人 2 处理等）

这些分配方法，也可组合使用。具体实施方法，请参照“8 不同用途的设定例子”。

执行“传感器”的机器人控制装置

“2.4 系统构成例”中介绍的几个系统，使用多个传感器。即使是最一般的系统，一般分别在供给侧的传送带和输出侧的传送带上使用，共 2 个传感器。

作为传感器检出传送带上的工件的方法，介绍了 2 大方法：使用视觉检出工件的方法，以及使用光电管传感器检出工件的方法。
其中，使用光电管传感器的“传感器”，其内部处理的负担小，因此无需特别考虑，而使用视觉的“传感器”进行图像处理时，对机器人控制装置施加相当量的内部处理的负担。
为此，如果有 2 个以上的使用视觉的“传感器”时，构成的系统应使多个“传感器”均匀分散于相同机器人控制装置上。更具体而言，其构成为：1 台机器人控制装置上，不连接 2 台以上的相机。

HDI 传感器

需要通过光电管传感器在 1 毫秒内进行检测时，请使用 HDI 选项。

2.6 制约事项

现在的软件版本 7DF5 系列 14 版、带队列管理的追踪、以及视觉追踪中，有以下制约。

- 不对应导轨追踪。
- 不对应热启动。请使停电处理无效。
- 不支持模拟相机。
- 不对应固定于机器人的相机。
- 不能使用先行执行距离指令。（可使用先执行指令。）
- 无法与干涉防止区域功能并用。
- 可同时执行的传感器是有限制的。详细内容请参照“6.5.7 可同时执行的传感器的限制”。
- 使用方案时，需要结合通过 ROS 以太网接口包（RIPE）连接的所有控制装置的软件版本。
- 无法同时使用如下功能。这是由于追踪拾取功能的制约。
  - 码垛堆积功能
  - 协调控制
  - 干涉防止领域功能
  - 远程 TCP
  - 奇异点回避功能（无法与追踪动作同时使用。可与通常动作同时使用。）
  - 精加工机器人操作性能强化软件包
  - 伺服焊枪更换功能
  - 机器人链接功能
  - 基本・扩展干涉检查功能
  - 电弧传感器（TAST）
  - TIG 电弧长度控制功能
  - 多层焊接功能（RPM）
  - 碰触感应器功能
  - 再启动位置检查功能

2.7 推荐事项

在 *i*RPickTool 中推荐进行以下设定。

- 再启动位置检查功能无效。（默认无效。）
- 制动控制无效。（默认无效。）
- 使自动备份无效。

3 *i*RPickTool Basic（基本功能）启动调试步骤

本章针对 *i*RPickTool Basic（基本功能）的启动调试步骤进行说明。
按本章步骤执行，完成使用下图 4 轴拳头机器人的系统启动。

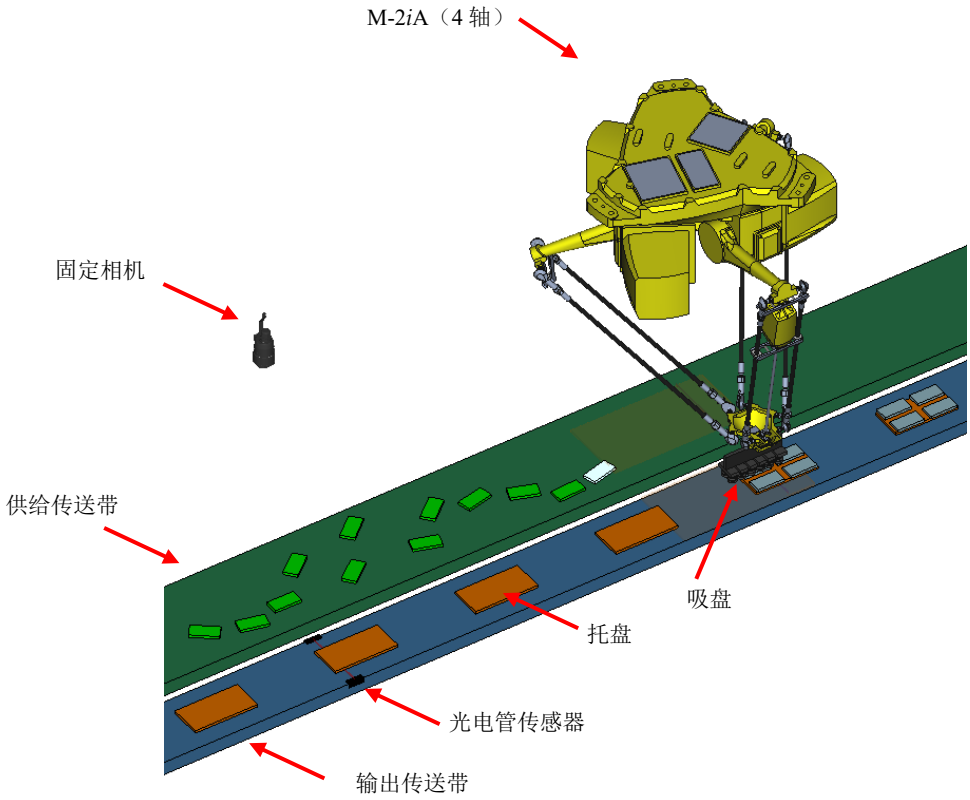


图 3(a) 系统构成

传送带的启动调试步骤内容会因检测的“触发条件”与满足触发条件时执行的“触发”组合以及有无使用托盘而异。
请根据用途选择组合。

本章将以系统构成为例对以下组合进行说明。

供给：［距离］＋［通过视觉检出］
排出：［DI］/［RI］/［HDI］＋［放入工件］（使用托盘）

关于使用其他组合的传送带及固定工作站时的启动调试步骤，各树形图项目详细内容请参阅“6 基本功能参考”。

3.1 设定应用程序前的准备工作

本项将针对软件连接及程序编写等使用 *i*RPickTool 的追踪系统启动调试准备工作进行说明。

3.1.1 脉冲编码器的连接与设定

安装脉冲编码器，与机器人控制装置连接。
连接后，在机器人控制装置的示教器上设定参数。

注释
本说明书中将检测传送带行程的装置称为“脉冲编码器”。
在示教器及 *i*RPickTool 设置画面当中，“编码器”这一术语作为“脉冲编码器”的同义词使用。

可以使用 α A1000S 脉冲编码器与量增脉冲编码器。

注释
使用 1 个或 3 个脉冲编码器时，可以将 α A1000S 脉冲编码器连接到主板使用。连接请参阅“付録 D α A1000S 脉冲编码器”，设定请参阅“5.2.3 设定连接到主板的脉冲编码器”。

3.1.1.1 脉冲编码器的连接

将脉冲编码器连接到机器人控制装置。

连接至生产线追踪用接口板时

参考下图进行连接，使安装在供给传送带的脉冲编码器的编码器编号为 1，安装在排出传送带上的编码器编号为 2。

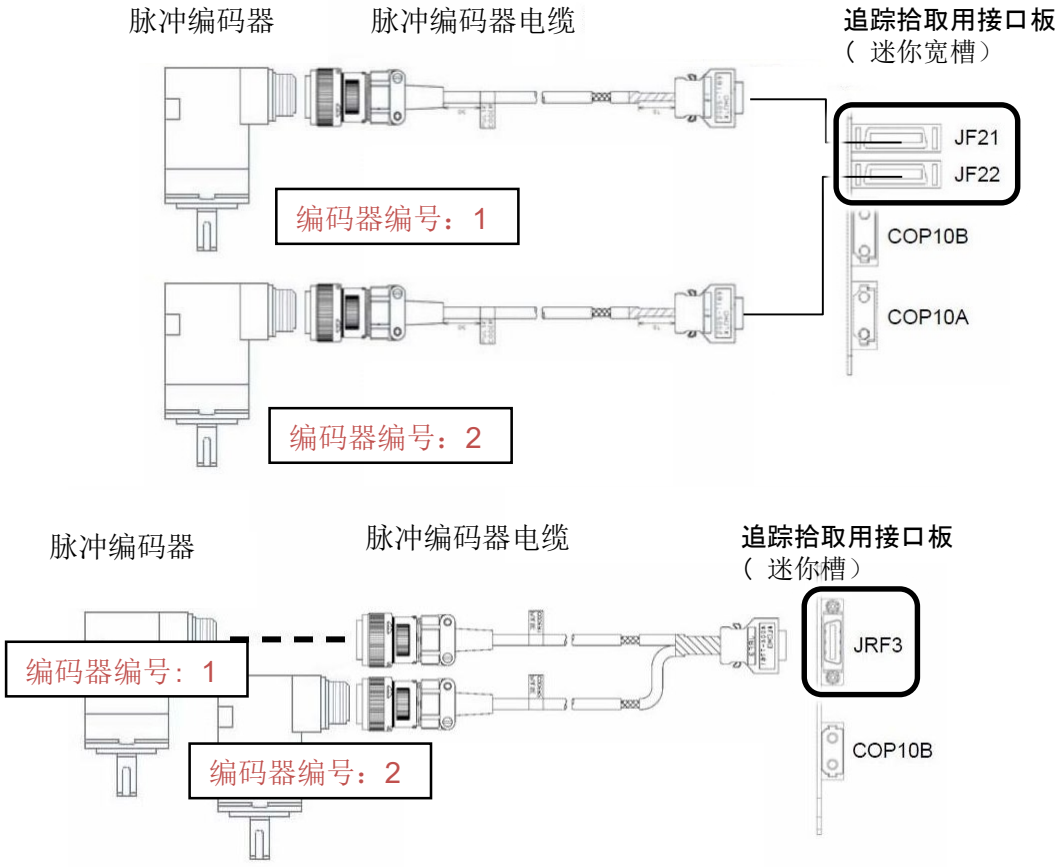


图 3.1.1.1(a) 生产线追踪用接口板与脉冲编码器的连接

3.1.1.2 脉冲编码器的设定

设定脉冲编码器参数。设定以下项目。

- ［编码器轴］
- ［编码器类型］
- ［平均（更新次数）］

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［编码器］。

显示以下画面。
设定项目编号［9］以后为可选功能的菜单。此处不设定。

3 首先设定安装在供给传送带上的脉冲编码器参数。确认［编码器编号］为 1。如果数值不是 1，按下 F3［编码器］，输入 1。

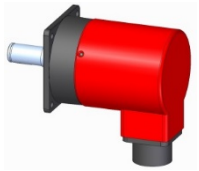
4 把光标对准［编码器轴］，输入 1。但是，在连接到主板的情况下，R-30*i*B Compact Plus / R-30*i*B Mini Plus 时保持初始值 0。

5　将光标对准［编码器类型］，按下 F4［选择］。
　显示菜单。
6　选择与连接脉冲编码器对应的编码器类型。

编码器类型

编码器有以下类型。

- α A1000S 脉冲编码器（红色脉冲编码器）：串行增量式
- 量增脉冲编码器（黑色脉冲编码器）：增加值



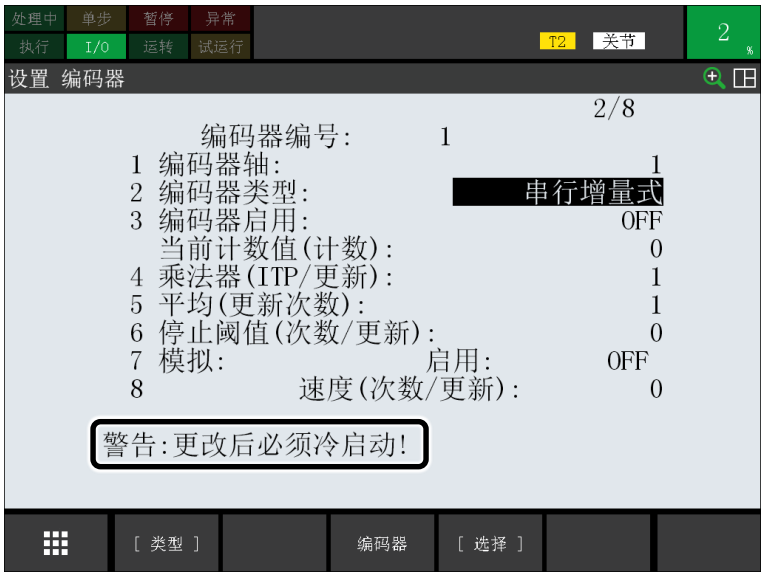
α A1000S（A860-0372-T001）
编码器类型：串行增量式

※在连接到主板的情况下，R-30iB Compact Plus / R-30iB Mini Plus 时为主串行增量式

A860-0301-T001～T004
编码器类型：增加值

※不可连接到主板

如果更改编码器轴或编码器类型，就会显示类似以下要求冷启动的警告。



7　将光标对准［平均（更新次数）］，输入瞬时速度的累积个数（移动平均宽度）。
通过设定瞬时速度的累积个数，即使传送带在机器人追踪中发生突然停止，机器人也不会紧急停止，而是平稳停下。
通常输入［10］。

8　然后设定安装在排出传送带上的脉冲编码器参数。按下 F3［编码器］，输入 2。
9　把光标对准［编码器轴］，输入 2。
10　把光标对准［编码器类型］，按下 F4［选择］，选择与连接脉冲编码器对应的编码器类型。
11　将光标对准［平均（更新次数）］，输入数值。与步骤 7 相同。通常输入［10］。

※　如果在选择中使用 HDI 信号，会显示［9 HDI 端口］，因此输入与使用 HDI 信号相对应的端口编号。

12　最后冷启动机器人控制装置。

3.1.1.3 脉冲编码器的设定确认

1 打开［编码器］画面，将光标对准［编码器启用］，按下 F4［ON］。
2 运转传送带，确认在［当前计数值（计数）］处显示脉冲计数。

3 对于连接至排出传送带的脉冲编码器，也选择编码器编号：2，进行同样确认。

3.1.2 相机的连接确认

确认安装在供给传送带上的相机连接。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［*i*RVision］→［示教和试验］。

显示［*i*RVision 示教和试验］画面。
3 按下 F2［新建］。

4　选择［二维相机］，按下 F4［下一个>］。

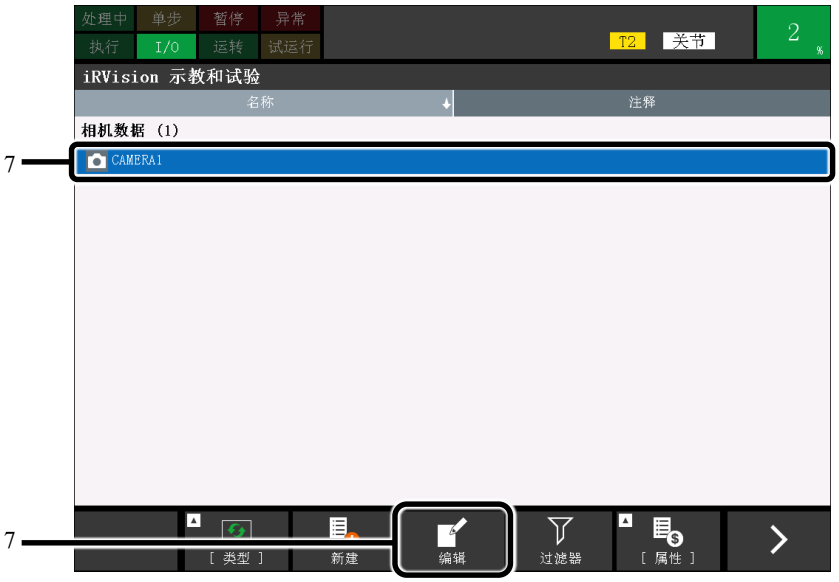
显示以下画面。

5　在［名称］中输入相机名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例：CAMERA1

6　按下 F4［确定］。

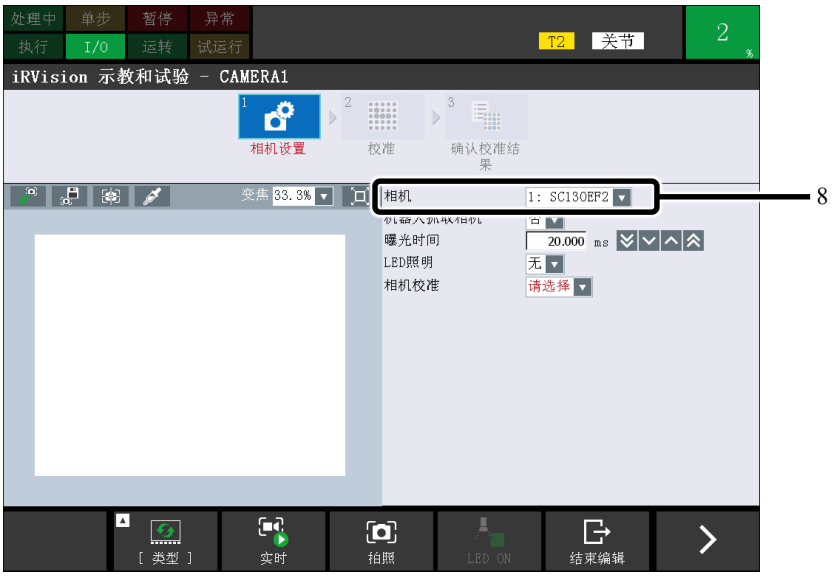
在列表中显示创建的相机数据。

7 选择创建的相机数据，按下 F3［编辑］。



显示以下画面。

8 从菜单选择［相机］。



9 为进行镜头的焦点调整，将写有文字的物品放在相机视野内，按下 F2［实时］。
确认在画面左侧的框内可以拍摄出放置的物品。
※ 图像较暗时，调整以下。
- 打开镜头光圈。
- 更改照明。

10 确认相机视野。
视野过大或过小时，调整相机位置。

11 相机位置确定后，调整镜头焦点。

图 3.1.2(a) 调焦

12 按下 F2［停止］。
实景停止。
13 按下 F5［结束编辑］。
编辑画面关闭。

3.1.3 触发传感器的输入确认

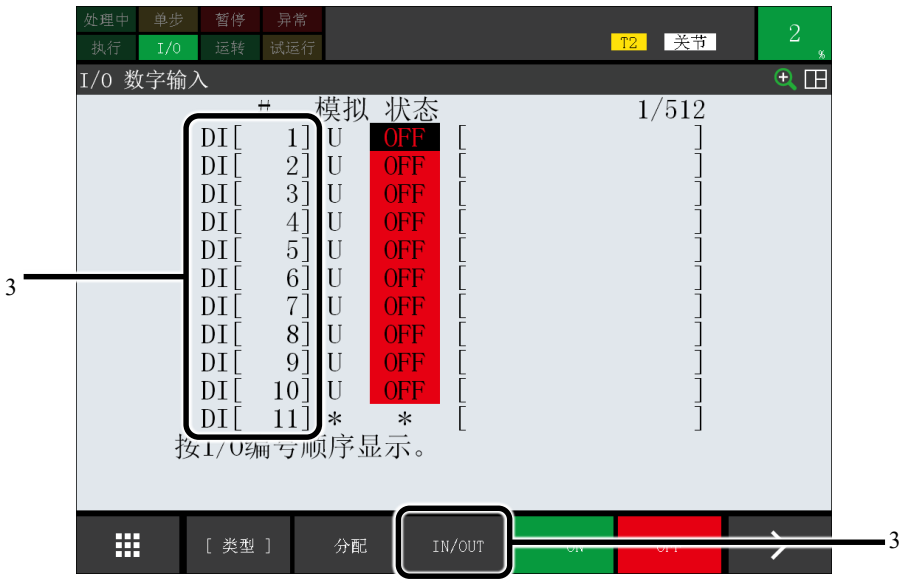
确认触发传感器的输入

3.1.3.1 使用［DI］/［RI］

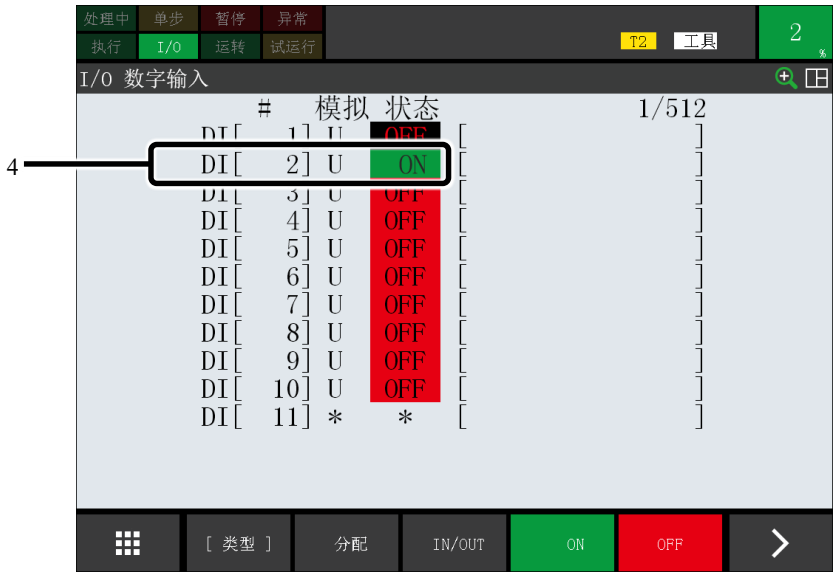
确认使用 DI / RI 时的触发传感器输入

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 使用 DI 时，从菜单选择［I/O］→［数字］。使用 RI 时，从菜单选择［I/O］→［机器人］。
画面为使用 DI 时的样子。

3 按下 F3［IN/OUT］，确认显示为 DI（RI）。



4　通过从传感器的输入确认 DI（RI）是否变成 ON。

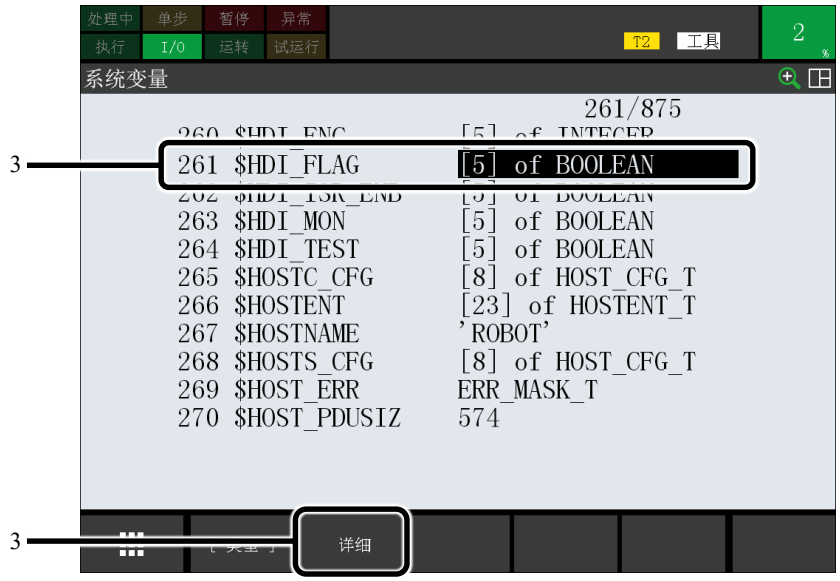


3.1.3.2　使用［HDI］

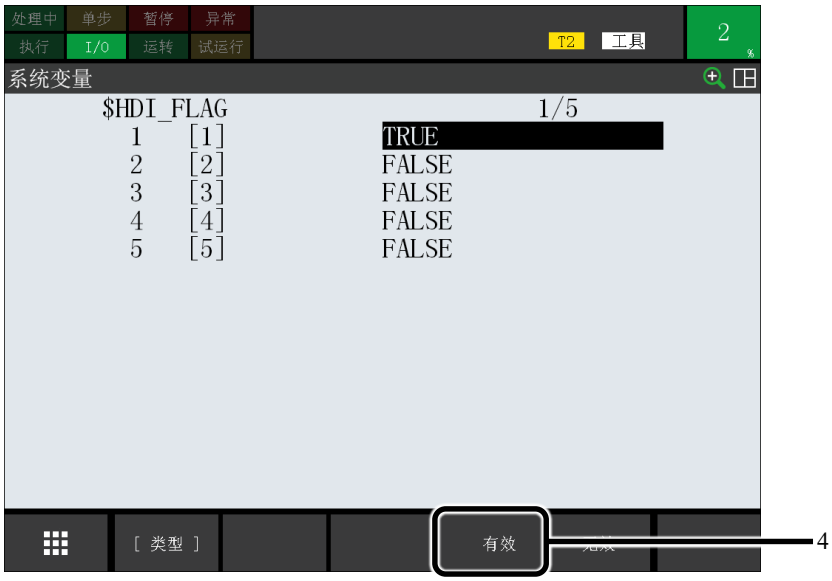
确认使用 HDI 时的触发传感器输入。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。
　显示菜单。
2　从菜单选择［下页］→［系统］→［变量］。

3　将光标对准“$HDI_FLAG”，按下 F2［详细］。

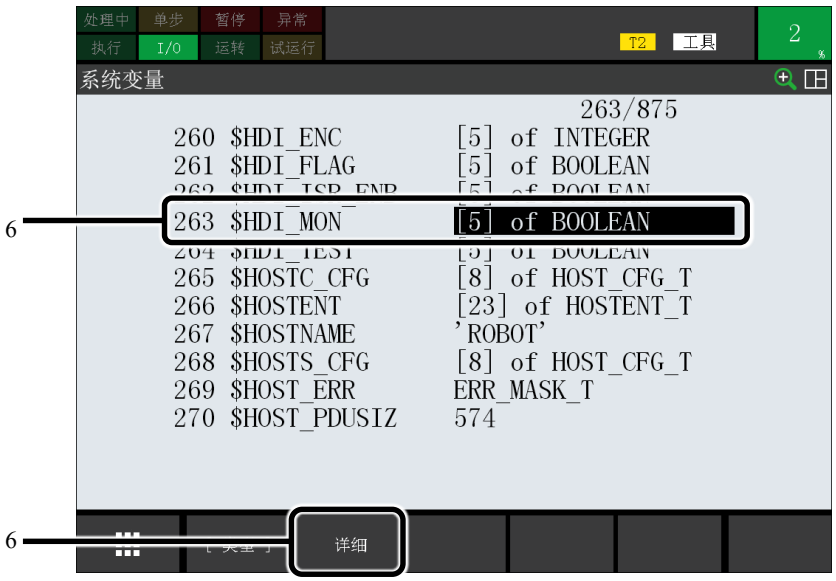


4　将光标对准确认的 HDI 端口号，按下 F4［有效］。

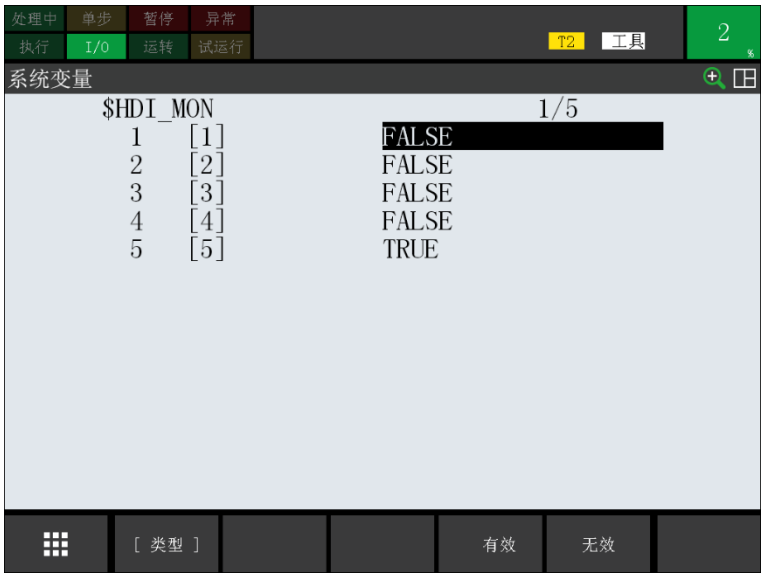


5　在机器人控制装置的示教器上按下［PREV］键。
显示［系统变量］画面。

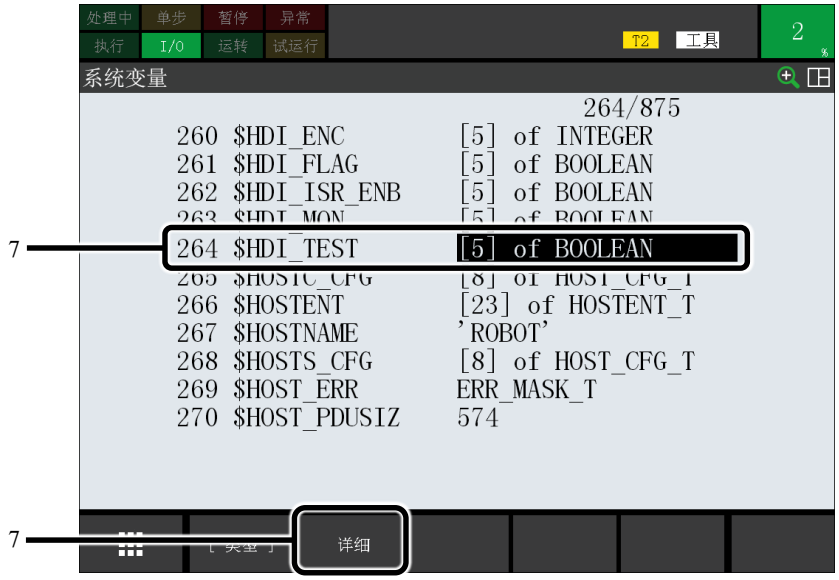
6　将光标对准“$HDI_MON”，按下 F2［详细］。



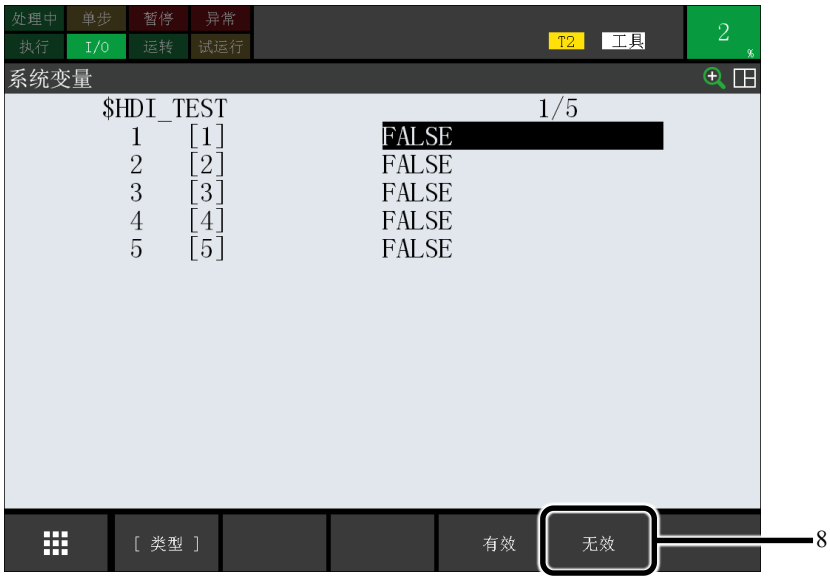
确认与连接到传感器端口号相对应的变量。
输入 ON　　：TRUE
输入 OFF　　：FALSE



输入时间短，无法按步骤 7 的方法确认时，按以下方法确认。
7　将光标对准“$HDI_TEST”，按下 F2［详细］。



8 将光标对准确认的 HDI 端口号，按下 F5［无效］。
输入一设成 ON 就会变成 TRUE，即使输入变成 OFF，也保持 TURE 状态。



3.1.4 设定碰触销的工具坐标系

设定碰触销的工具坐标系。

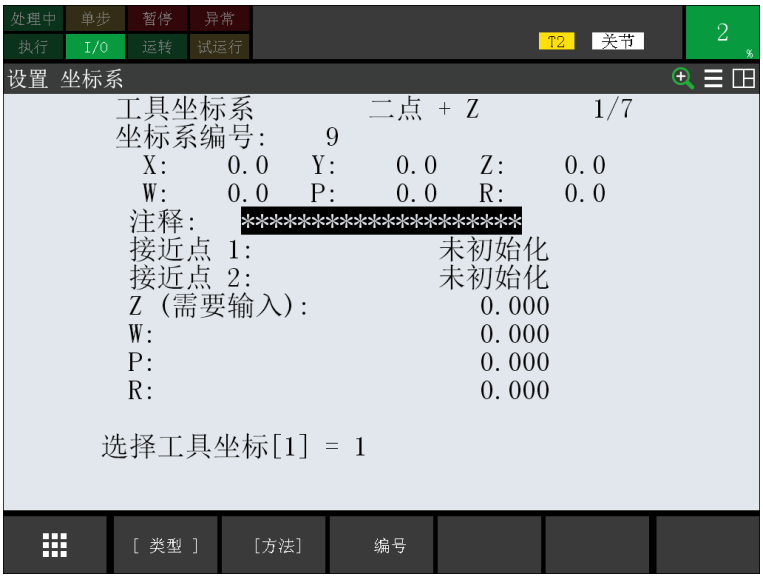
1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 显示工具坐标系之外时，按下 F3［坐标］，从菜单选择［工具］。
显示菜单。

4 将光标对准设定的工具坐标系编号行，按下 F2［详细］。
显示选中坐标系编号的工具坐标系设定画面。
画面为利用工具 9 设定时的例子。

5 按下 F2［方法］。
显示菜单。
6 从菜单选择［二点＋Z］。

显示以下画面。

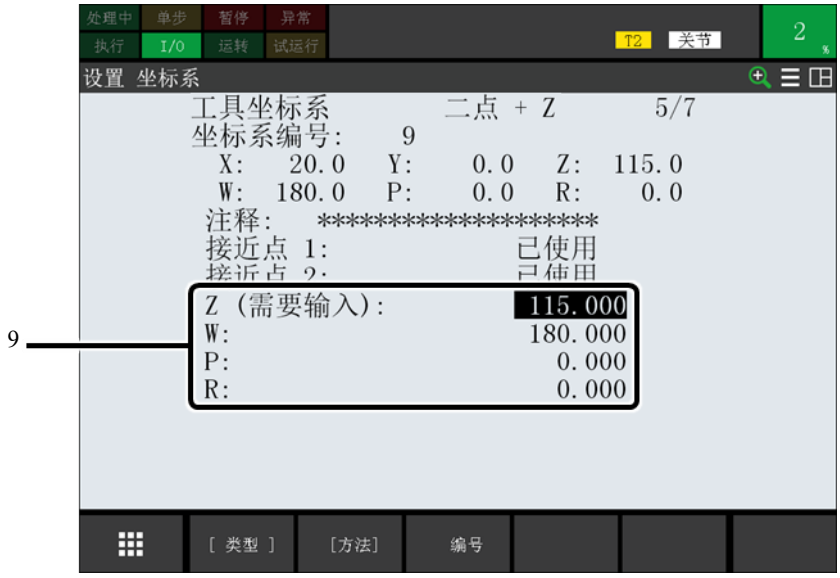


7 将光标对准［接近点 1］，输入接近点 1。
通过点动移动到应记录机器人的点。
在按下示教器［SHIFT］键的同时，按下 F5［记录］，将现值的数据作为接近点输入。
显示“已记录”。

8 将光标对准［接近点 2］，输入接近点 2。
通过点动移动到应记录机器人的点。
在按下示教器［SHIFT］键的同时，按下 F5［记录］，将现值的数据作为接近点输入。
输入所有的接近点后，显示“已使用”，设定工具坐标系的 X、Y。

⚠ 注意
1 输入接近点时，请将法兰盘面朝下。
2 请在不同的位置输入接近点 1 和 2。
3 如果没有满足以上条件，将显示“输入点设置无效”的消息。

9 将光标对准［Z］及［W］、［P］、［R］，输入各自的值。
［Z］用输入规尺等测量的数值。
［W］、［P］、［R］直接输入。
※推荐（W、P、R）=（180、0、0）。



10 在机器人控制装置的示教器上按下［PREV］键。
显示工具坐标系一览画面。
可以确认所有工具坐标系的设定值。

11 按下 F5［切换］。
12 输入设定的工具坐标系的坐标系编号。

启用工具坐标系。
※ 如不进行步骤 11、12，设定的坐标系就不会有效。
13 将机器人的点动进给设定成［工具］。
在机器人控制装置的示教器上，在将点动进给切换成“工具”之前按下［手动进给］键。

14 在机器人控制装置的示教器中按下 -Z (J6) 键或 +Z (J6) 键。
Z 轴旋转进行点动动作。
确认工具坐标系（TCP）设定正确。

3.1.5 机器人程序的创建

创建机器人程序
机器人程序要创建以下程序。

- 主程序
- 追踪程序。
- 吸附 ON / OFF 用程序

主程序创建

创建主程序。

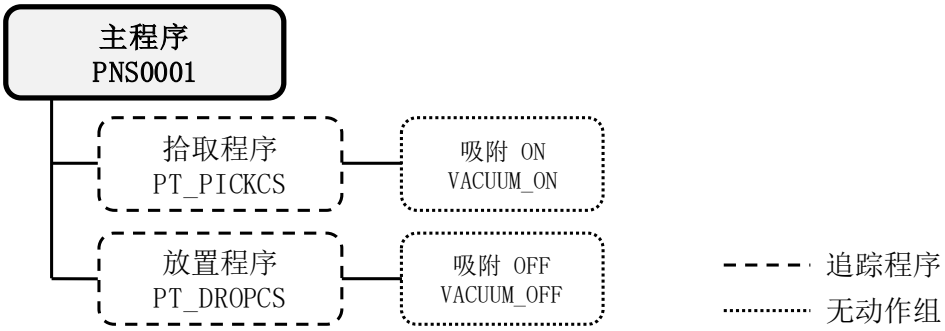


图 3.1.5(a)主程序的位置关系

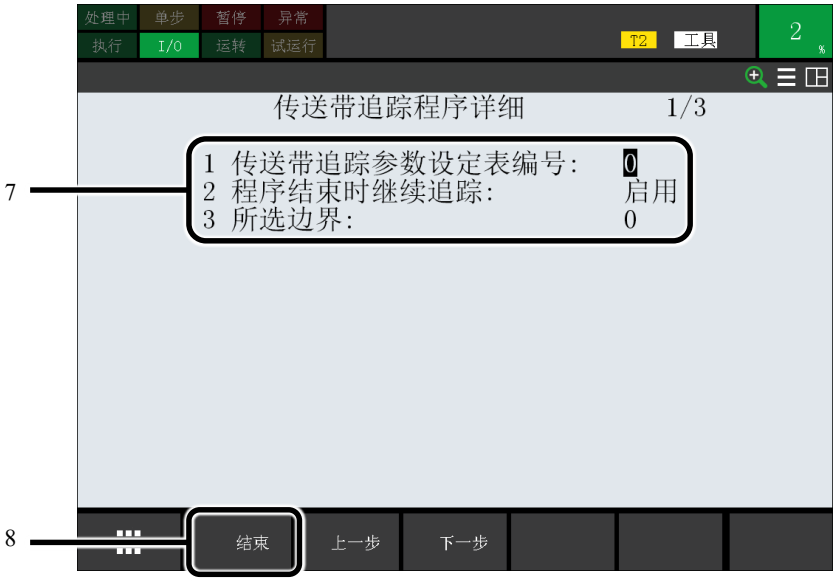
1 在机器人控制装置的示教器上按下［SELECT］键。
显示程序目录画面。
2 按下 F2［创建］。
显示程序创建画面。
如果未显示［创建］、按下［>（下一页）］切换画面。

3 将光标对准［程序名］，输入程序名称。
程序名称的输入方法可以从弹出显示的［单词］、［大写］、［小写］、[**其他**／键盘］选择。
例：此处在 PNS 启动的前提下，将名称设成 PNS0001。
4 在机器人控制装置的示教器上按下［ENTER］键。

5 按下 F2［详细］。
显示程序详细画面。

6　确认在［下一步］的第 1 组中输入了“1”，再按下 F3［组掩码］。
显示追踪程序的详细画面。

7　确认［追踪参数设置］设定为“0”。
8　按下［结束］。
主程序创建。
※　［组掩码］　　：[1,*,*,*,*,*,*,*]
　［追踪参数设置］　　：0
※　关于程序，请参阅后述的程序示例。



追踪程序创建

创建追踪程序。追踪程序创建“拾取程序”和“放置程序”。

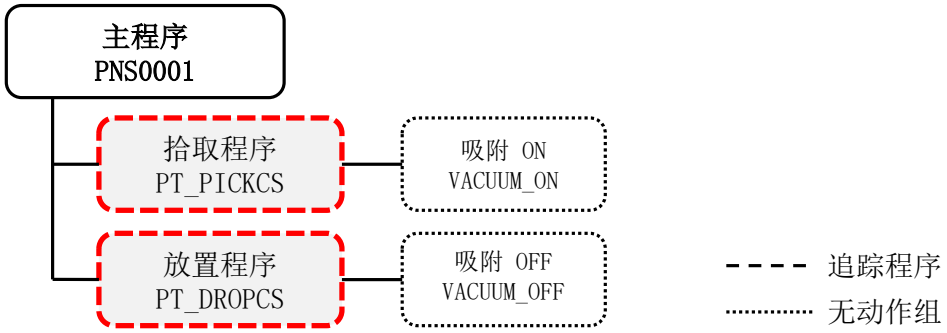


图 3.1.5(b)追踪程序的位置关系

1　在机器人控制装置的示教器上按下［SELECT］键。
　显示程序目录画面。
2　按下 F2［创建］。
　显示程序创建画面。
　如果未显示 F2［创建］、按下［＞（下一页）］切换画面。

3　将光标对准［程序名］，输入程序名称。
程序名称的输入方法可以从弹出显示的［单词］、［大写］、［小写］、[**其他**／键盘］选择。
例：PT_PICKCS, PT_DROPCS
4　在机器人控制装置的示教器上按下［ENTER］键。

5　按下 F2［详细］。

显示程序详细画面。

6　确认在［下一步］的第 1 组中输入了“1”，再按下 F3［组掩码］。

显示以下追踪程序的详细画面。

7　［追踪参数设置］中设定“1”或“2”。
　　拾取（PICK）时　　：1
　　放置（DROP）时　　：2

8　按下 F1［结束］。
创建追踪程序.
※［组掩码］　　：[1,*,*,*,*,*,*,*]
　［追踪参数设置］　　：拾取（PICK）　：1
　　放置（DROP）　：2

※关于程序，请参阅后述的程序示例。

吸附 ON / OFF 用程序创建

创建吸附 ON / OFF 用程序。在吸附 ON / OFF 用程序中，以吸附 ON 设定“拾取程序”以“吸附 OFF”设定“放置程序”。

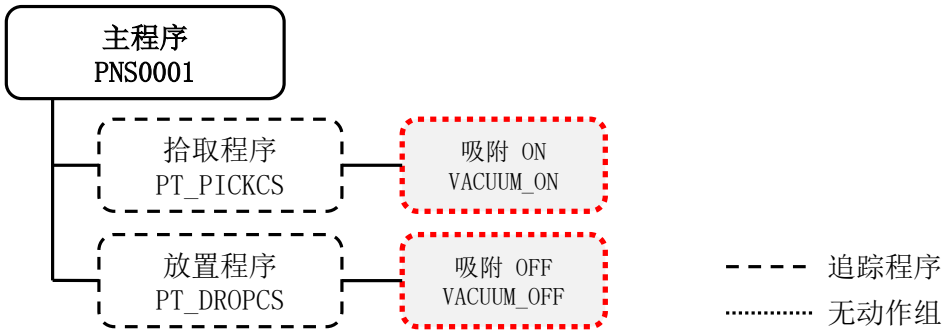


图 3.1.5（c）吸附 ON / OFF 用程序的位置关系

1　在机器人控制装置的示教器上按下［SELECT］键。
　显示程序目录画面。
2　按下 F2［创建］。
　显示程序创建画面。
　如果未显示 F2［创建］、按下［>（下一页）］切换画面。

3　将光标对准［程序名］，输入程序名称。
　　程序名称的输入方法可以从弹出显示的［单词］、［大写］、［小写］、[**其他**／键盘］选择。
　　例：VACUUM_ON、VACUUM_OFF
4　在机器人控制装置的示教器上按下［ENTER］键。

5　按下 F2［详细］。
　　显示程序详细画面。

6　将光标对准［组掩码］，将组掩码全部更改成“*”。
　　将光标对准要更改的位置，按下 F5［*］就可以更改。
7　按下 F1［结束］。
　　吸附 ON / OFF 用程序被创建。
　　※［组掩码］　　：[*,*,*,*,*,*,*,*]
　　［追踪参数设置］　　：0
　　※ 关于程序，请参阅后述的程序示例。

程序示例

表示本项说明程序的各种示例。
关于 KAREL 程序，请参阅后述的“KAREL 程序的使用”。
主程序的示例（PNS0001）如下所示。

PNS0001

```
 1:  !PKCSGETID
 2:  ! (Get conveyor station ID)
 3:  !*1:Conveyor name
 4:  !*2:Register number
 5:  !     for conveyor station ID
 6:  !*3:Group number (Can be omitted)
 7:  CALL PKCSGETID('CONV1',“垢恬嫋 ID Reg”=11)
 8:  CALL PKCSGETID('CONV2',“垢恬嫋 ID Reg”=12)
 9:
10:  UFRAME_NUM=0
11:  UTOOL_NUM=1
12:  PAYLOAD[1]
13:L PR[1:HOME POS] 500mm/sec FINE Wjnt
14:
15:  !--- System startup ---
16:  !PKWCSTART (System startup,
17:  ! including check and clear)
18:  CALL PKWCSTART
19:
20:  LBL[101]
21:  !--- Pick program ---
22:  CALL PT_PICKCS
23:  IF DI[1:CYCLE STOP]=ON,JMP LBL[901]
24:  !--- Placement program ---
25:  CALL PT_DROPCS
26:  IF DI[1:CYCLE STOP]=ON,JMP LBL[901]
27:  JMP LBL[101]
28:
29:  LBL[901]
30:  UFRAME_NUM=0
31:  UTOOL_NUM=1
32:L PR[1:HOME POS] 500mm/sec FINE Wjnt
33:
34:  !PKWCEND (System end)
35:  CALL PKWCEND
[END]
```

KAREL 程序
KAREL 程序
KAREL 程序
KAREL 程序

追踪程序（拾取程序）的示例（PT_PICKCS）如下所示。

PT_PICKCS

```
 1:  UFRAME_NUM=0
 2:  UTOOL_NUM=1
 3:
 4:  LBL[101:QUE check]
 5:  STOP_TRACKING
 6:  !PKCSGETQUE(Get part information)
 7:  !*1:Register number
 8:  !      for conveyor station ID
 9:  !*2:Usually 1.
10:  !      2 or more in the case of
11:  !      consecutive acquisition)
12:  !*3:Waiting time msec
```

```
13:   !*4:Vision register number
14:   !      for part information
15:   !*5:Process status
16:   !      register number
17:   !*6:Model ID (Can be omitted)
18:   CALL PKCSGETQUE
      (“勧僕揮垢恬嫺 ID”=R[11:CONV1 ID],“銭偬議炎芝”=1,
      “階扮峙(ms)”=100,“温灰喘 VR”=1,“彝蓑 Reg”=2)
19:   IF R[2]=0,JMP LBL[111]
20:   IF DI[1:CYCLE STOP]=ON,JMP LBL[901]
21:   JMP LBL[101]
22:
23:   LBL[111:Pick operation]
24:   !ADJ_OFS (Adjust offset)
25:   !*1:Usually 1 (vision register)
26:   !*2:Vision register number
27:   !      before adjustment
28:   !*3:Position register number
29:   !      for adjustment amount
30:   !*4:Vision register number
31:   !      after adjustment
32:   CALL ADJ_OFS(1,1,21,1)
33:L PR[7] R[26]mm/sec CNT100 VOFFSET,VR[1] Offset,PR[11]
34:L PR[7] R[27]mm/sec CNT R[24] VOFFSET,VR[1] AP_LDR[21] TB R[25]sec,
    CALL VACUUM_ON
35:   WAIT R[23]
36:   PAYLOAD[2]
37:L PR[7:Pick Pos] R[28]mm/sec
    CNT100 VOFFSET,VR[1] Offset,PR[11] RT_LDR[22]
38:
39:   !PKCSACKQUE (Part information
40:   ! process notification)
41:   !*1:Register number
42:   !      for conveyor station ID
43:   !*2:Processing results.
44:   !      Normally 1
45:   CALL PKCSACKQUE(“勧僕揮垢恬嫺 ID”=R[11:CONV1 ID],“撹孔”=1)
46:
47:   LBL[901:Pick finished]
[END]
```

KAREL 程序

KAREL 程序

拾取接近位置

拾取位置

放置缩回位置

KAREL 程序

追踪程序（放置程序）的示例（PT_DROPCS）如下所示。

PT_DROPCS

```
1:   UFRAME_NUM=0
2:   UTOOL_NUM=1
3:
4:   LBL[101:QUE check]
5:   STOP_TRACKING
6:   !PKCSGETQUE(Get part information)
7:   !*1:Register number
8:   !      for conveyor station ID
9:   !*2:Usually 1.
10:  !      2 or more in the case of
11:  !      consecutive acquisition)
12:  !*3:Waiting time msec
13:  !*4:Vision register number
14:  !      for part information
```

```
15:   !*5:Process status
16:   !      register number
17:   !*6:Model ID (Can be omitted)
18:   CALL PKCSGETQUE
      (“勧僕揮垢恬嫺 ID”=R[12:CONV2 ID],
       “銭偬議炎芝”=1,“階扮峙(ms)”=100,
       “温灰喘 VR”=2,“彜蓑 Reg”=3)
19:   IF R[3]=0,JMP LBL[111]
20:   IF DI[1:CYCLE STOP]=ON,JMP LBL[901]
21:   JMP LBL[101]
22:
23:   LBL[111:Place operation]
24:L PR[8] R[36]mm/sec
   CNT100 VOFFSET,VR[2] Offset,PR[12] Wjnt
25:L PR[8] R[37]mm/sec
   CNT R[34] VOFFSET,
   VR[2] AP_LDR[31] TB R[35]sec,
   CALL VACUUM_OFF
26:   WAIT R[33]
27:   PAYLOAD[1]
28:L PR[8] R[38]mm/sec
      CNT100 VOFFSET,
      VR[2] Offset,PR[12] RT_LDR[32]
29:
30:   !PKCSACKQUE (Part information
31:   ! process notification)
32:   !*1:Register number
33:   !      for conveyor station ID
34:   !*2:Processing results.
35:   !      Normally 1
36:   CALL PKCSACKQUE
      (“勧僕揮垢恬嫺 ID”=R[12:CONV2 ID],“撹孔”=1)
37:
38:   LBL[901:Place finished]
[END]
```

KAREL 程序

放置接近位置

放置位置

放置缩回位置

KAREL 程序

吸附 ON 程序示例（VACUUM_ON）如下所示。

VACUUM_ON

```
1:    DO[2:Vacuum burst command]=OFF
2:    DO[1:Vacuum command]=ON
[END]
```

吸附 OFF 程序示例（VACUUM_OFF）如下所示。

VACUUM_OFF

```
1:    DO[1:Vacuum command]=OFF
2:    DO[2:Vacuum burst command]=PULSE,0.3sec
[END]
```

数值寄存器一览

数值寄存器一览画面按以下步骤显示。

1　在机器人控制装置的示教器上按下［DATA］键。
2　按下 F1［画面］
　显示菜单。
3　从菜单选择［数值寄存器］。
　显示寄存器目录画面。

表 3.1.5(a)寄存器目录

R[2:GETQ Status Pick]=0	拾取工件处理状态
R[3:GETQ Status Drop]=0	放置工件处理状态
R[11:CONV1 ID]=1	传送带 1(拾取）ID
R[12:CONV2 ID]=2	传送带 2(放置）ID
R[21:Pick AP_LD]=30	拾取接近距离指定“mm”
R[22:Pick RT_LD]=30	拾取缩回距离指定 “mm”
R[23:Pick Wait]=.05	在拾取位置的待机时间 “sec”
R[24:Pick CNT]=0	拾取位置的平滑指定
R[25:Pick TB]=.1	先执行时间(VACUUM_ON) “sec”
R[26:Pick Speed1]=4000	拾取程序速度 1 “mm/sec”
R[27:Pick Speed2]=4000	拾取程序速度 2 “mm/sec”
R[28:Pick Speed3]=2000	拾取程序速度 3 “mm/sec”
R[31:Drop AP_LD]=30	放置接近距离指定“mm”
R[32:Drop RT_LD]=30	放置缩回距离指定 “mm”
R[33:Drop Wait]=.05	在放置位置的待机时间 “sec”
R[34:Drop CNT]=0	放置位置的平滑指定
R[35:Drop TB]=.1	先执行时间(VACUUM_OFF) “sec”
R[36:Drop Speed1]=2000	放置程序速度 1 “mm/sec”
R[37:Drop Speed2]=2000	放置程序速度 2 “mm/sec”
R[38:Drop Speed3]=4000	放置程序速度 3 “mm/sec”

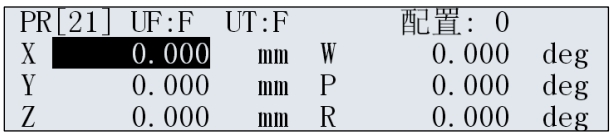
位置寄存器一览画面按以下步骤显示。

1 在机器人控制装置的示教器上按下［DATA］键。
2 按下［画面］。
显示菜单。
3 从菜单选择［位置寄存器］。
显示位置寄存器目录画面。

表 3.1.5(b) 位置寄存器目录

PR[1:HOME POS]=R	原位置
PR[7:Pick Pos]=R	拾取位置
PR[8:Drop Pos]=R	放置位置
PR[11:Pick Apr Z]=R	从拾取位置偏移
PR[12:Drop Apr Z]=R	从放置位置偏移
PR[21:ADJ_OFS]=R	拾取位置补偿量调整用

拾取位置补偿量的调整量初始值全部设成“0”。



机器人动作

机器人的拾取 / 放置动作如下图。

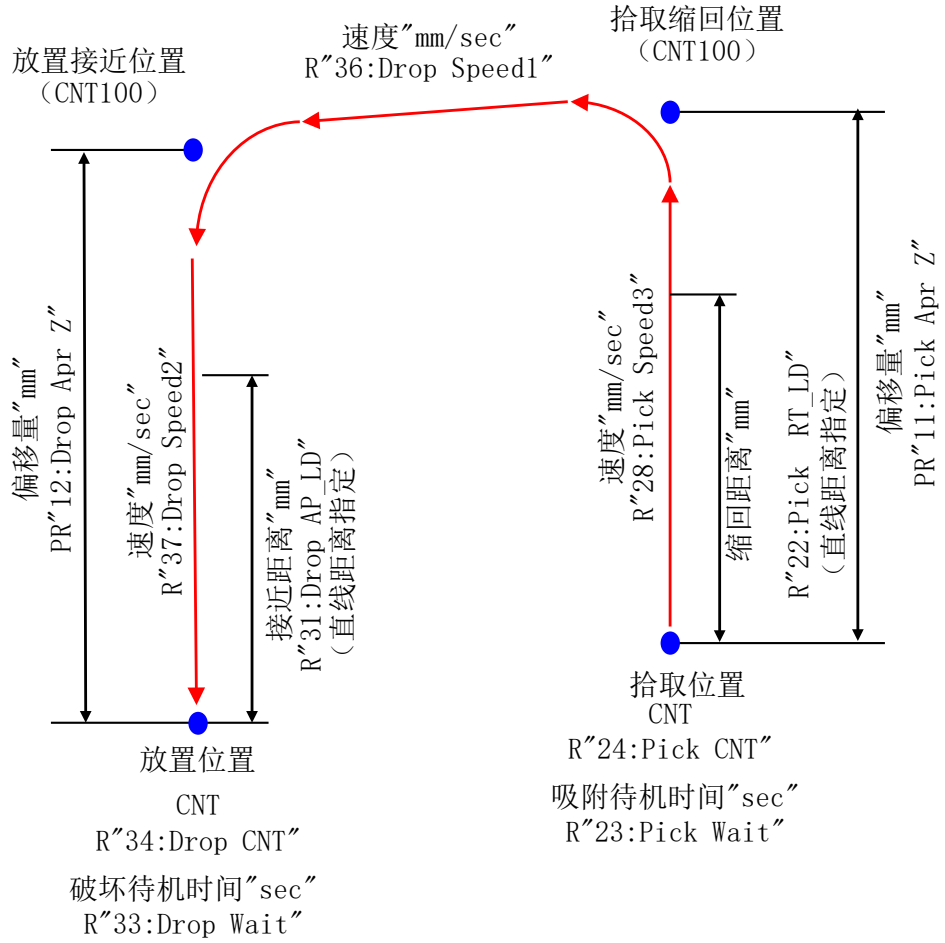


图 3.1.5(d) 机器人动作(拾取→放置)

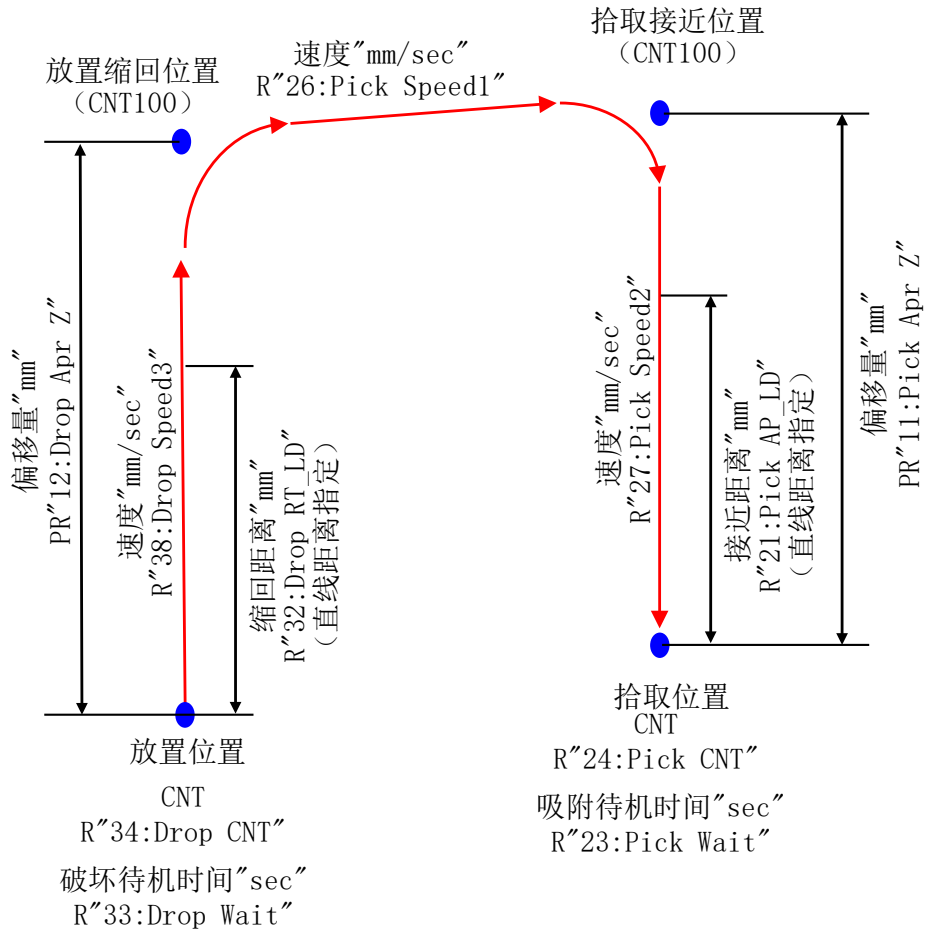


图 3.1.5(e) 机器人动作(放置→拾取)

KAREL 程序的使用

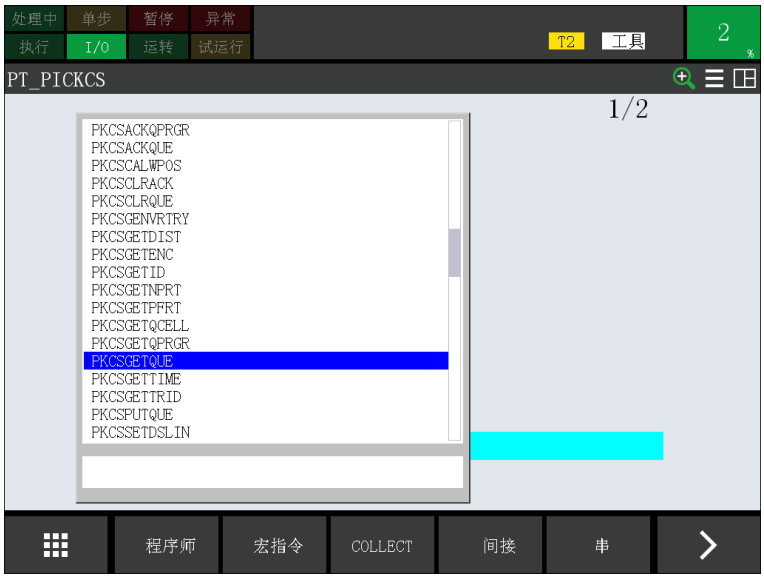
为了使用 KAREL 程序，按以下步骤添加。

1　在程序编辑画面按下 F1［指令］。
2　从菜单选择［调用］。

3　从菜单选择［调用程序］。

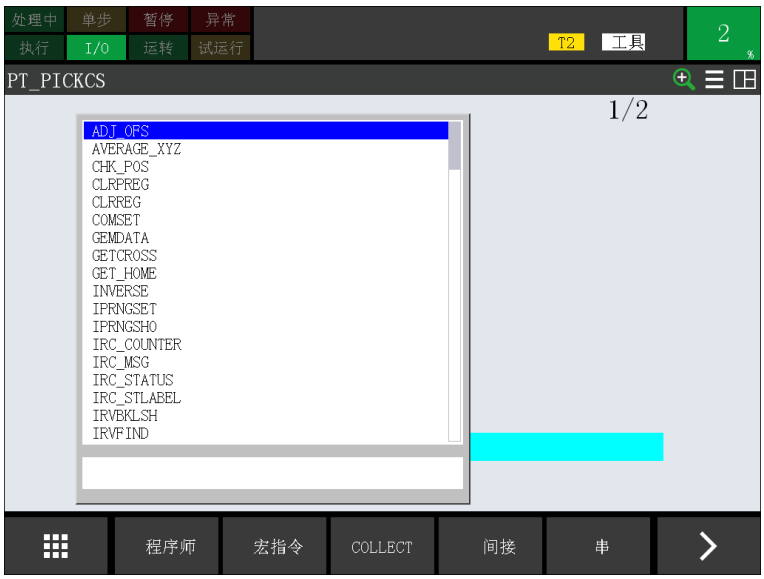
4 按下 F3［COLLECT］。
在目录中显示“KAREL 程序”。
5 选择“KAREL 程序”，在机器人控制装置的示教器上按下［ENTER］键。
显示 KAREL 程序目录画面。

6 选择调用的 KAREL 程序。
在程序编辑画面添加 KAREL 程序。
以下画面为调用以“PK”开始的 KAREL 程序时。



如果是以“PK”开始的 KAREL 程序，参数将自动显示。
更改参数时，将光标对准参数，输入需要的数值。
这样就完成了操作。

以下画面为调用开头为“PK”之外的 KAREL 程序时。

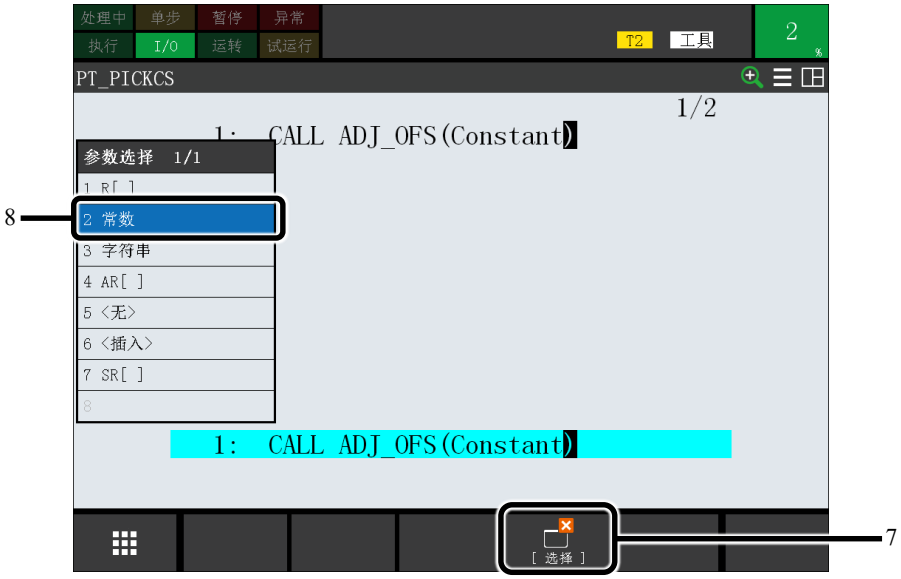


以下步骤只有在调用开头为“PK”之外的 KAREL 程序时需要。
以调用“ADJ_OFS”时的情况为例进行说明。

7　将光标对准添加到程序编辑画面的“ADJ_OFS”右侧，选择 F4［选择］。

添加参数时，将光标对照右侧的“ ）”，按下 F4［选择］。

8　从菜单选择［常数］。



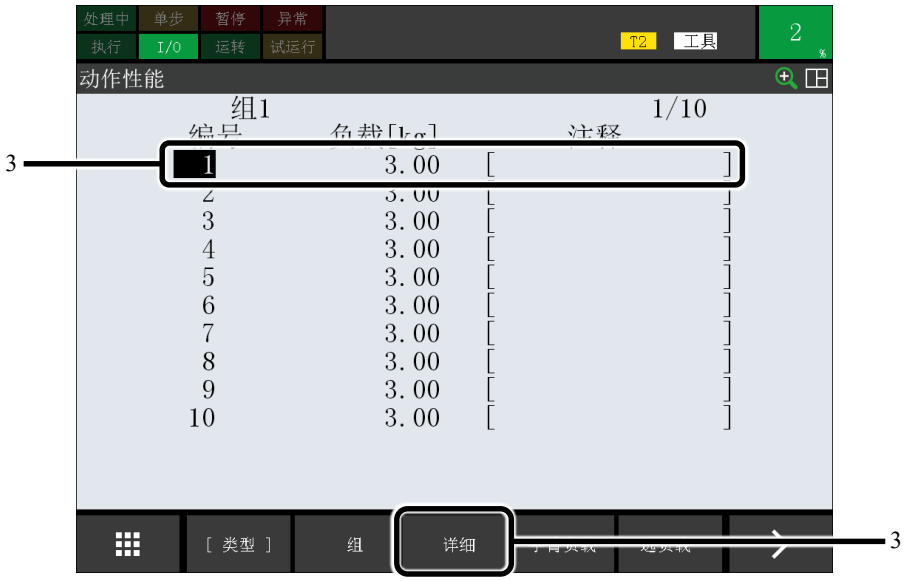
9　添加所需数量的参数后，在各自的常数中输入数值。

3.1.6　负载设定（动作性能）

在［动作性能］画面设定机器人的负载。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2　从菜单选择［系统］→［动作］。

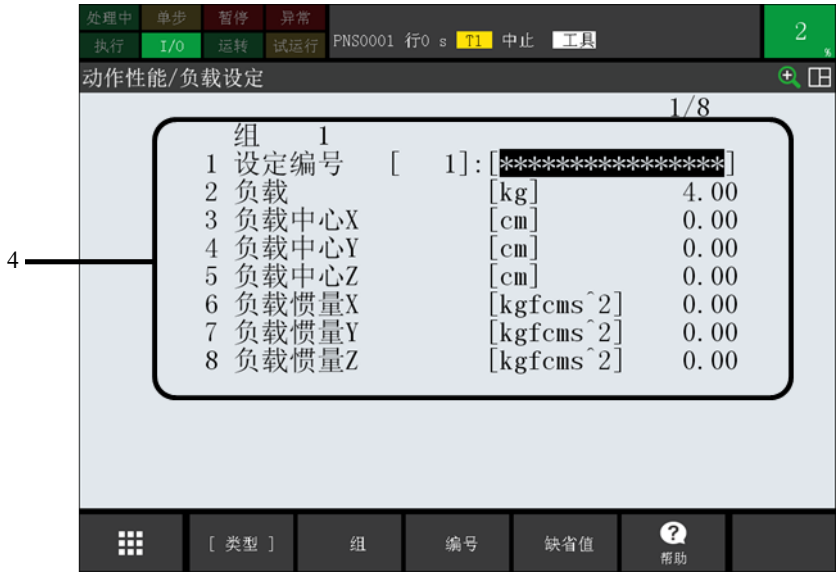
3 将光标对准［1］，按下 F3［详细］。



显示负载条件 No.1 的设定画面。

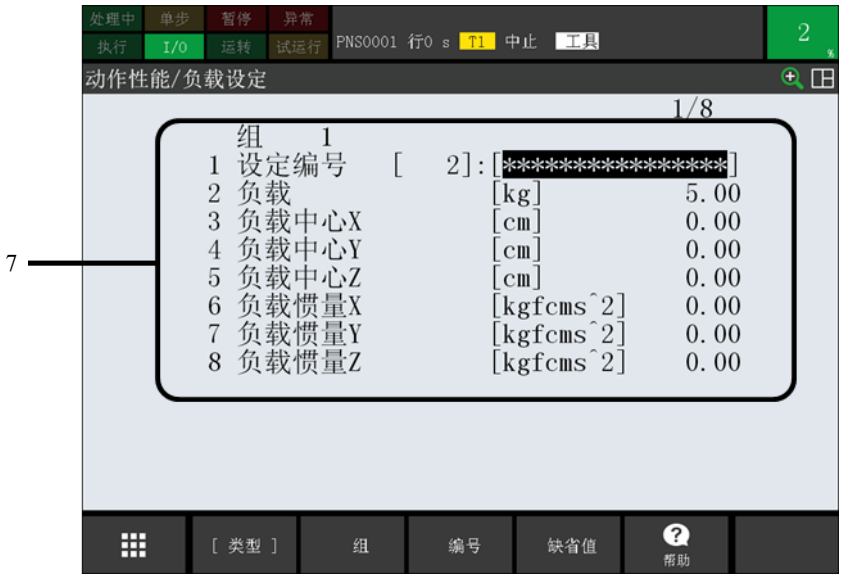
4 至少输入［负载］。并不一定要输入所有项目。

※ 拾取 / 放置程序没有像前述样本程序(PT_PICKCS、PT_DROPCS)保持工件时，如果使用负载设定［1］，请在这里设定“只有机械手”的负载。



5　在机器人控制装置的示教器上按下［PREV］键。
返回动作性能一览画面。
6　将光标对准［2］，按下 F3［详细］。

显示负载条件 No.1 的设定画面。
7　至少输入［负载］。并不一定要输入所有项目。
※　拾取 / 放置程序像前述样本程序(PT_PICKCS、PT_DROPCS)保持工件时，如果使用负载设定［2］，请在这里设定“机械手+工件”的负载。



标准程序中使用负载“*”命令切换负载。

3.2 应用程序的设定

本节中针对工作单元及传感器设定等 *i*RPickTool 的应用程序设定进行说明。

3.2.1 工作单元设置

设定工作单元。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［*i*RPickTool］。

显示以下 *i*RPickTool 设置画面。

3　选择［机器人］，按下树形图上面的［新建］按钮。

4　在［名称］中输入机器人名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。

5　按下 F4［确定］。
创建对象。

6　在［控制装置］处从菜单选择［此控制装置］（机器人为 1 台时）。
7　在［组编号］处从菜单选择［1］。

8　首先创建供给传送带的对象。选择［传送带］，按下树形图上面的［新建］按钮。

9　在［名称］中输入供给传送带的名称。
　　推荐直接使用初始值显示的名称。
　　以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
10　按下 F4［确定］。
　　创建对象。

新建传送带后，就会自动创建传感器和传送带工作站。

11 选择按步骤 10 自动创建的传送带工作站［CSTN1］。
12 在［机器人］中从菜单选择机器人。
选择按步骤 4 输入的机器人名称。

13 从菜单选择［追踪参数设置］。选择“1”。

14 从菜单选择［编码器编号］。选择“1”。

15 然后创建排出传送带的对象。选择［传送带］，按下树形图上面的［新建］按钮。

16 在［名称］中输入排出传送带的名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
17 按下 F4［确定］。
创建对象。

新建传送带后，就会自动创建传感器和传送带工作站。

18 选择按步骤 17 自动创建的传送带工作站［CSTN2］。
19 在［机器人］中从菜单选择机器人。
选择按步骤 4 输入的机器人名称。

20 从菜单选择［追踪参数设置］。选择“2”。

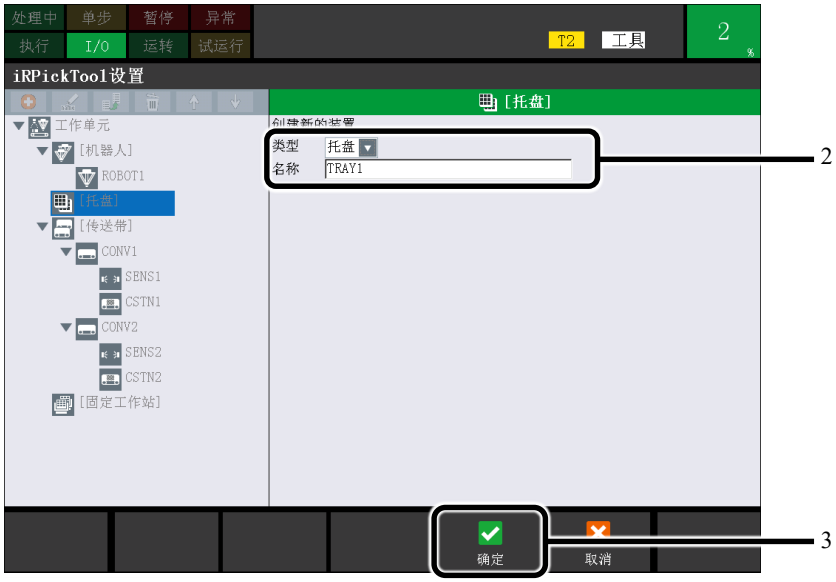
21 从菜单选择［编码器编号］。
选择“2”。

3.2.2 托盘设定

设定在排出传送带上使用的托盘。

1 选择［托盘］，按下树形图上面的［新建］按钮。

2 在［名称］中输入托盘名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
3 按下 F4［确定］。
创建对象。



显示以下画面。

4 确定托盘的坐标系。
原点为任意位置。
如果使传送带的传送方向与托盘的坐标系的 X 方向一致将比较容易明白（其他方向也没问题）。

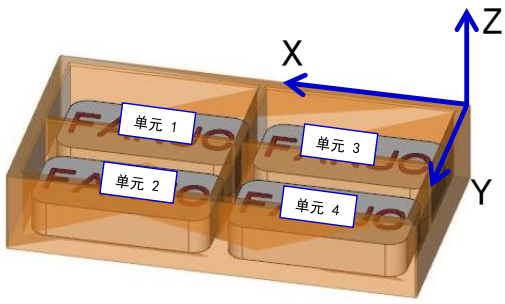


图 3.2.2(a) 确定托盘的原点

5 按下 F3［单元］。
6 从菜单选择［追加单元］。

7 追加第 1 层的单元。

8　在单元编辑画面输入各单元的坐标值。
选择要编辑的单元行，输入到画面下面的框内。
关于单元的坐标值，输入托盘坐标系上的坐标值。请参照详细“6.3 托盘设定”。

托盘堆 2 层时，进行以下操作。
9　按下 F4［层操作］。
10　从弹出菜单选择［追加层］。

11　在［单元数］中输入第 2 层的单元数。
12　在［默认 Z(mm)］中输入 Z 值(mm)。
13　按下 F4［确定］。

显示以下画面。
在［层编号］中创建层“2”。

14 在单元编辑画面输入层 2 的各单元坐标值。
选择要编辑的单元行，输入到画面下面的框内。
※此时，基准单元保持未示教也没有问题。按“3.2.10 基准位置的设定、机器人的位置示教”自动教示。

3.2.3 传送带设定

进行传送带设定。根据有无使用托盘，步骤不同。

3.2.3.1 供给传送带的设定

设定供给传送带。不使用托盘时的设定。

1 选择传送带［CONV1］。

显示传送带的设定画面。
传送带的设定可直接使用初始值所显示的以下数据。
［追踪类型］ ：直线
［伺服传送带］ ：没有勾选［已使用］
［托盘］ ：不使用

［同一物检查的阈值］ ：10.0mm
［负载平衡］ ：进度一定，所有模型 ID 是共通的

3.2.3.2 排出传送带的设定

设定排出传送带。即使用托盘时的设定。

1 选择传送带［CONV2］。

显示传送带的设定画面。

2 ［托盘］选择［TRAY1］，［初始状态］选择［空］。
传送带之外的设定可直接使用初始值所显示的以下数据。
［追踪类型］ ：直线
［伺服传送带］ ：没有勾选［已使用］
［同一物检查的阈值］ ：10.0mm（默认）
［负载平衡］ ：进度一定，所有模型 ID 是共通的

3.2.4 追踪坐标系设定

设定追踪坐标系

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。
显示工具坐标系一览画面。

5 按下 F5［切换］。
6 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

7 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
8 从菜单选择［设置］→［*i*RPickTool］。

9　首先设定供给传送带的追踪坐标系。
　　选择传送带［CONV1］。
10　按下 F4［设定追踪坐标系］。

显示以下画面。

11　根据画面所显示的步骤，将点阵板安装在传送带上游侧。
　　为防止偏差，用胶带等固定。

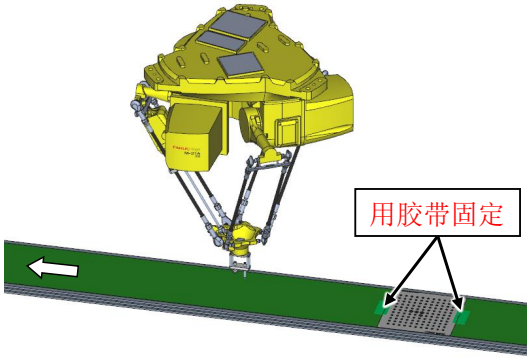


图 3.2.4(a)　将点阵板安装在上游侧

12　安装方向应使传送带的传送方向与点阵板的 X 方向保持同一方向。

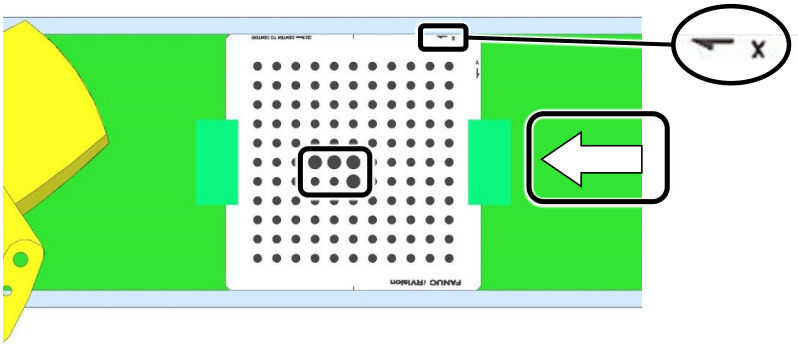
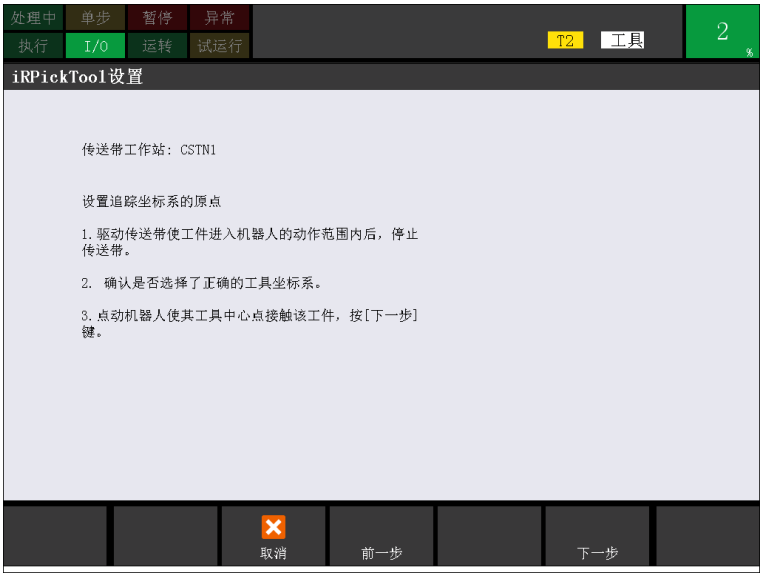


图 3.2.4(b)对齐安装方向

13　按下 F5［下一步］。

显示以下画面。



14　运行传送带，点阵板在机器人动作范围内到达上游后关停。

15 触碰点阵板的原点。

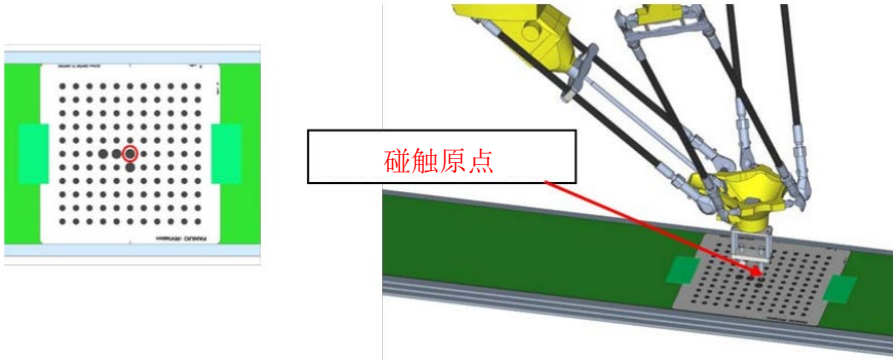


图 3.2.4(c) 触碰点阵板的原点

16 按下 F5［下一步］。

显示以下画面。

17 使机器人向上空方向移动。

18 运行传送带，点阵板在机器人动作范围内到达下游后关停。

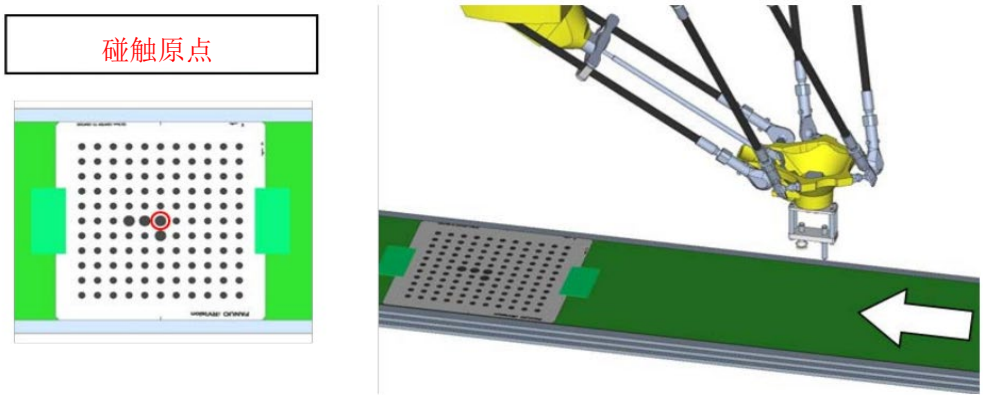


图 3.2.4(d) 停止传送带

19 触碰点阵板的原点。

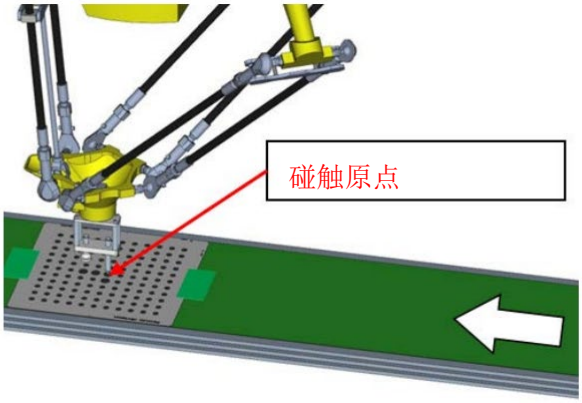


图 3.2.4(e) 触碰点阵板的原点

20 按下 F5［下一步］。

显示以下画面。

21 触碰点阵板的 Y 方向。
此时，传送带不运行。
不是机器人的 Y 方向。对于传送带前进方向，为左侧。

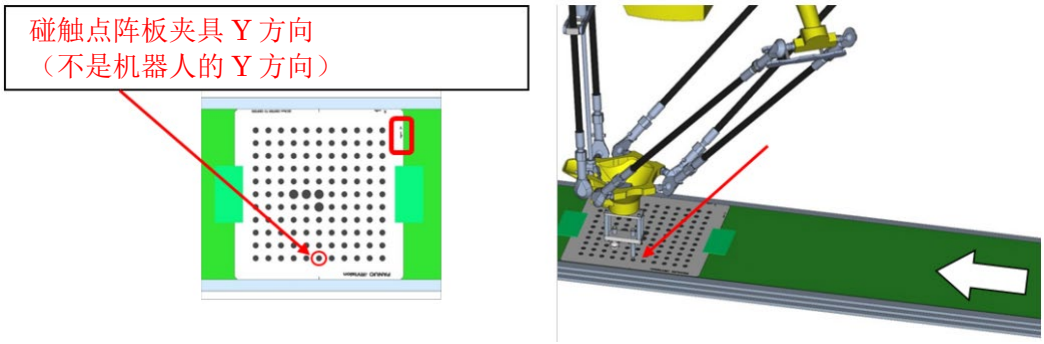


图 3.2.4(f) 触碰点阵板的 Y 方向

22 按下 F5［下一步］。

23 按下 F5［下一步］。

24 按下 F4［确定］。

设定供给传送带的追踪坐标系。

25 然后设定排出传送带的追踪坐标系。
选择传送带［CONV2］。

26 通过与步骤 10 到 24 相同的步骤，设定排出传送带的追踪坐标系。

3.2.5 传感器设定

进行传感器设定。

3.2.5.1 供给传送带传感器的设定

设定供给传送带的传感器。即［距离］＋［通过视觉检出］时的设定。

1 选择传感器的［SENS1］。

2 设定以下项目。

［本传感器］ ：勾选［有效］
［控制装置］ ：［此控制装置］
［触发条件］ ：［距离］
［触发］ ：［通过视觉检出］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

3 设定以下项目。

［触发距离］ ：输入传送带每前进多少 mm 就会进行视觉检出。
※ 建议在相机视野内将可以检测工件 2 次的距离设为触发距离。

［视觉程序］ ：此时不需要设定。
※ 在即将要设定基准位置前设定。关于设定请参阅“3.2.10.1 供给基准位置的设定、机器人的位置示教”。

3.2.5.2 排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时）

设定排出传送带的传感器。［DI］/［RI］＋［放入工件］时的传感器设定。［HDI］＋［放入工件］时，请跳过该项，进入下一项“3.2.5.3 排出传送带传感器的设定（［HDI］＋［投入工件］时）”。

1 选择传感器的［SENS2］。

2 设定以下项目。

［本传感器］	：勾选［有效］
［控制装置］	：［此控制装置］
［触发条件］	：［DI］
［触发］	：［放入工件］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

3 设定以下项目。

［DI 编号］	：输入分配传感器输入的 DI 编号
［再输入 DI 无效时间］	：输入要禁用的时间（用于防止颤动）
［传感器位置］	：此时不需要设定 ※ 使用传感器位置的设定进行设定。关于设定请参阅“3.2.6 传感器位置的设定”。

跳过下一项“3.2.5.3 排出传送带传感器的设定（［HDI］＋［投入工件］时）”，请进入“3.2.6 传感器位置的设定”项。

3.2.5.3 排出传送带传感器的设定（［HDI］＋［放入工件］时）

设定排出传送带的传感器。［HDI］＋［放入工件］时的传感器设定。［DI］／［RI］＋［放入工件］时，请参阅前项“3.2.5.2 排出传送带传感器的设定（［DI］／［RI］＋［投入工件］时）”，而不是本项。

1　选择传感器的［SENS2］。

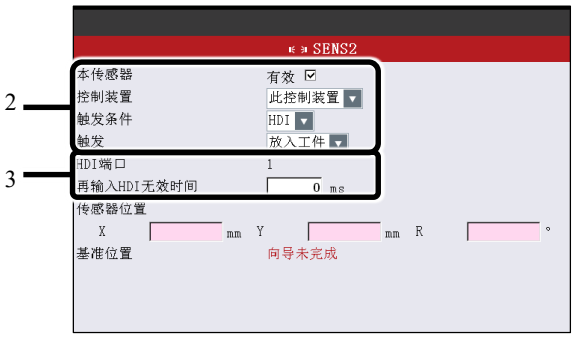
2　设定以下项目。

［本传感器］　：勾选［有效］
［控制装置］　：［此控制装置］
［触发条件］　：［HDI］
［触发］　：［放入工件］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

3　设定以下项目。

［HDI 端口编号］　：显示通过编码器设定指定的编号
［再输入 HDI 无效时间］　：输入要禁用的时间（用于防止颤动）
［传感器位置］　：该步骤中不需要设定
※ 使用传感器位置的设定进行设定。关于设定请参阅“3.2.6 传感器位置的设定”。



3.2.6 传感器位置设定

设定排出传送带的传感器位置。即使用托盘时的设定。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。

显示工具坐标系一览画面。

5 按下 F5［切换］。

6 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

7 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

8 从菜单选择［设置］→［*i*RPickTool］。

9　选择传感器［SENS2］，按下 F4［传感器位置］。

显示以下画面。

10　将托盘安装到传送带传感器的上游。

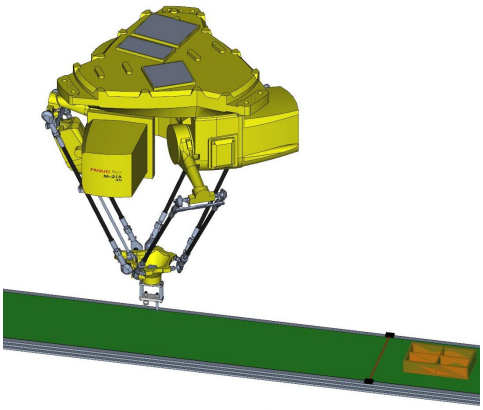


图 3.2.6(a)将托盘安装到传感器上游

11 运行传送带，托盘到达机器人动作范围内关停。
显示传感器检测工件瞬间的触发计数。

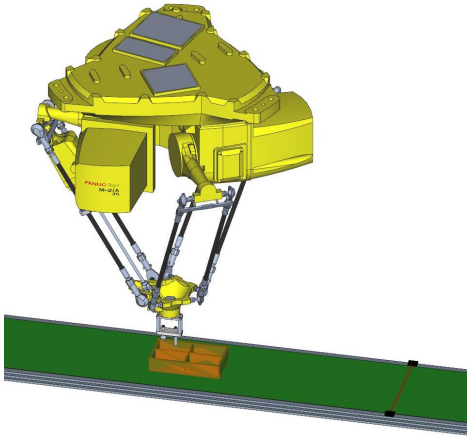


图 3.2.6(b)到达机器人动作范围内后停止

12 按下 F5［下一步］。

显示以下画面。

13 触碰托盘的原点。

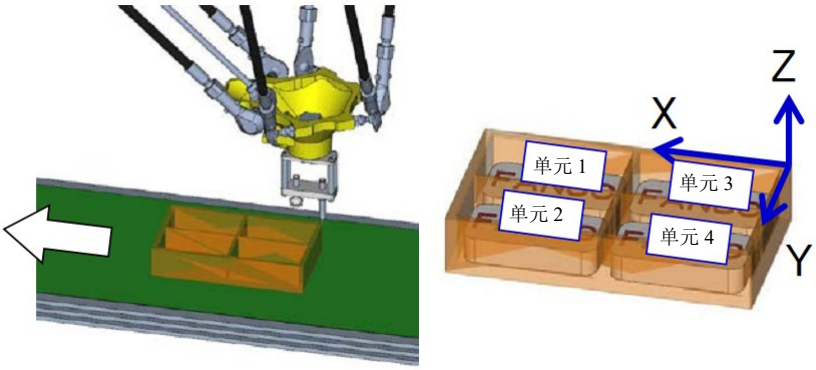


图 3.2.6(c)触碰托盘的原点

14　按下 F4［记录］。
　　显示当前的编码器计数、触碰位置。
15　按下 F5［下一步］。

显示以下画面。

16　触碰托盘的 X 轴方向。

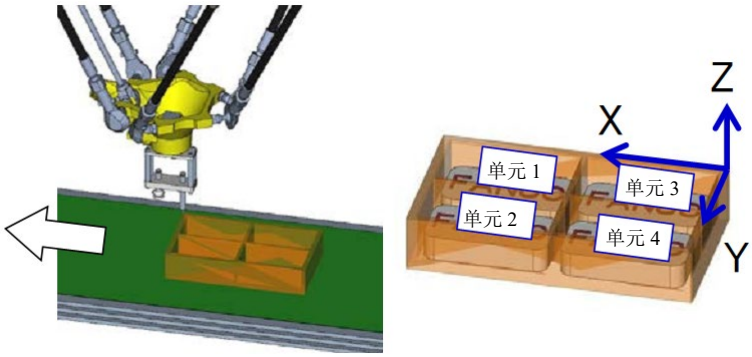


图 3.2.6(d)　触碰托盘的 X 轴方向

17 按下 F4［记录］。
显示当前的编码器计数、触碰位置。
18 按下 F5［结束］，

返回原来的设置画面。在传感器位置 X、Y、R 显示数值。

3.2.7 视觉设定

设定供给传送带的视觉。

3.2.7.1 相机校准

进行相机校准。

1 准备与相机视野相符的点阵板。
确认右下方的格子间距。

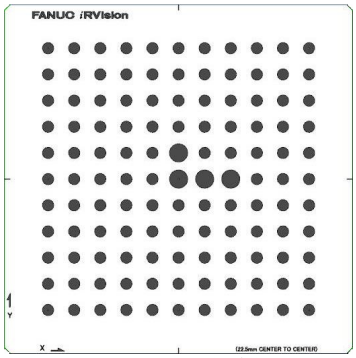


图 3.2.7.1(a)点阵板

> **⚠ 注意**
> **无需在图像上显示出整个点阵。**
> **有几个点超出图像范围也没关系，要使点阵分布在图像中检出工件的整个区域。点阵只能在图像中检出工件的部分区域反映出的状态下，无法进行高精度校对。**

注释
点阵板固定夹具也可以使用金属板。请根据客户的环境准备。本公司准备了几种标准件。

2 在相机正下方安装点阵板。
为防止偏差，用胶带等固定。

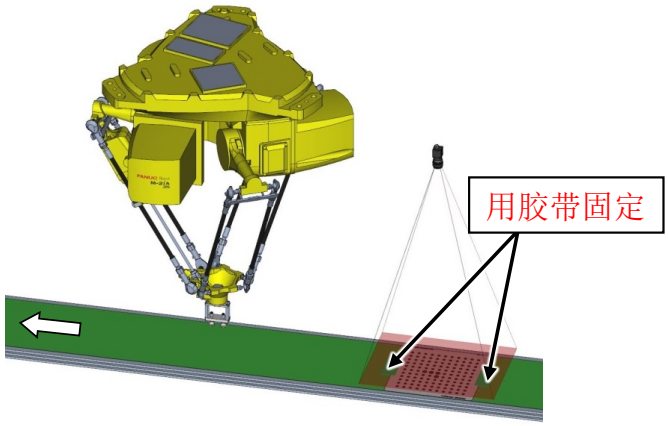


图 3.2.7.1(b) 安装点阵板

3 点阵板的方向可以任意方向。
例如，拾取追踪时，应使传送带的传送方向与点阵板的 X 方向保持同一方向。

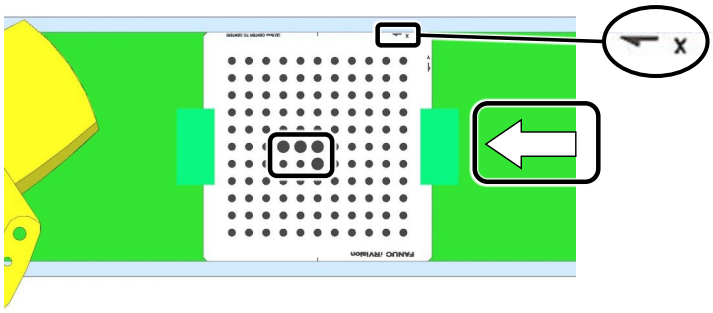


图 3.2.7.1(c)对齐安装方向

4 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
5 从菜单选择［设置］→［坐标系］。

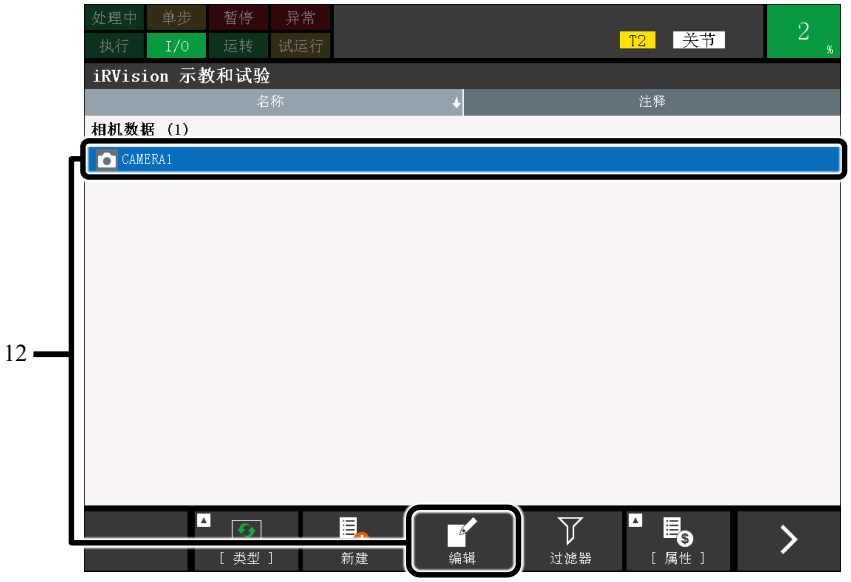
6 按下 F3［坐标］。
7 从菜单选择［工具］。
显示工具坐标系一览画面。

8 按下 F5［切换］。
9 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

10 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

11 从菜单选择［*i*RVision］→［示教和试验］。显示［*i*RVision 示教和试验］画面。

12 选择按“3.1.2 相机的连接确认”创建的相机数据，按下 F3［编辑］。



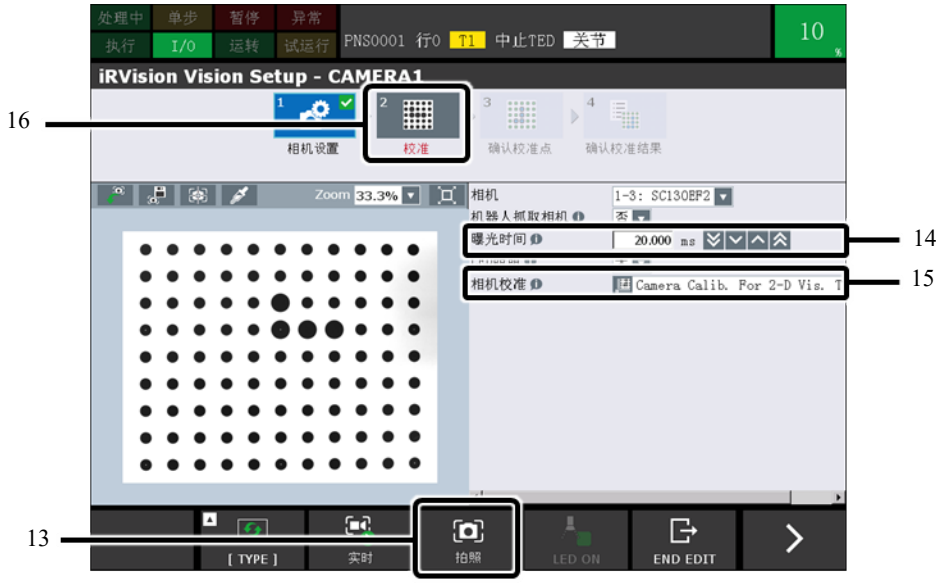
显示相机的设定画面。

13 按下 F3［拍照］，确认在画面左侧的框内拍摄到点阵板。

14 输入［曝光时间］。

15 在［相机校准］处从菜单选择［Camera Calib. For Vis. Track］。

16 按下画面上方的［校准］。



显示校准的设定画面。

17 选择［传送带名］。

18 输入点阵的［格子间距］。

19 在［投影方法］中选择［透视投影］。
用初始值选择［透视投影］。

> **注意**
> **［正投影］只能在检出的工件高度一定，点阵面与检出的工件上表面高度相同时选择。**

20 在［焦点距离］中输入使用的镜头焦距。

> **注意**
> **选择［自动］时，由于传送带上的点阵板与相机的拍摄面接近平行，从原理上来说，［焦点距离］的计算方法有可能无法正确地计算焦距。**

注释

斜放点阵板，可以用［自动］计算出正确的［焦点距离］。

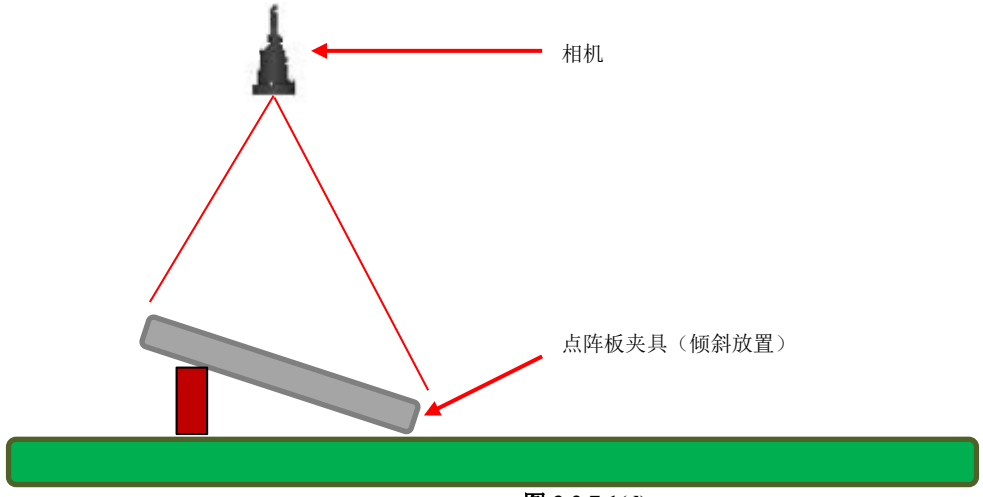
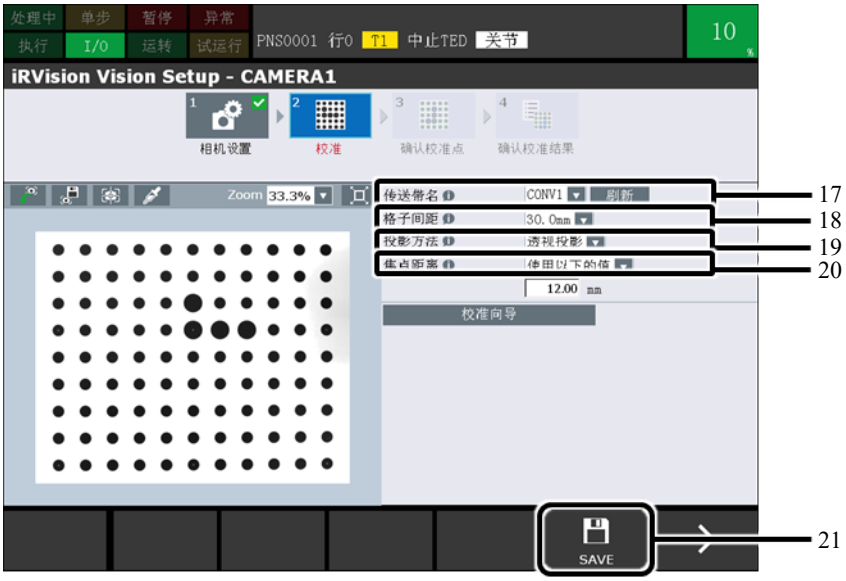
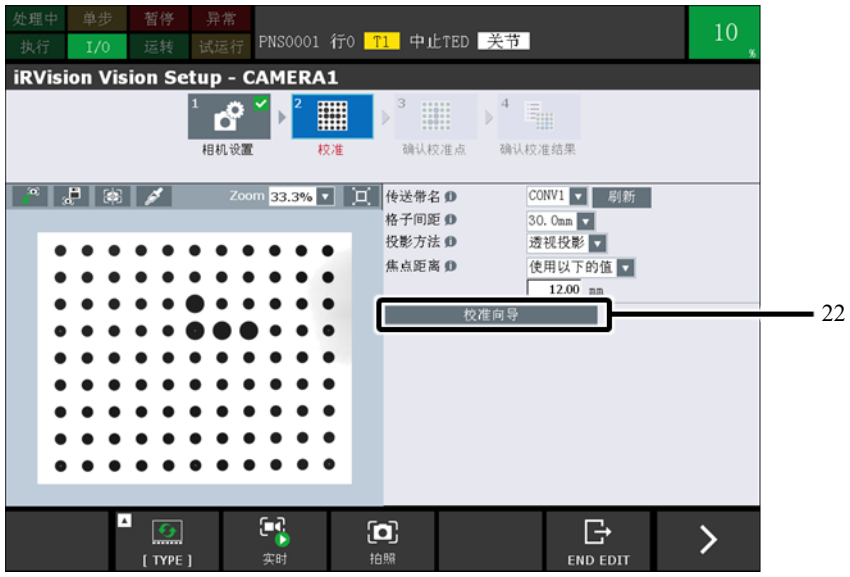


图 3.2.7.1(d)

21 按下［＞（下一页）］，再按下 F5［保存］。

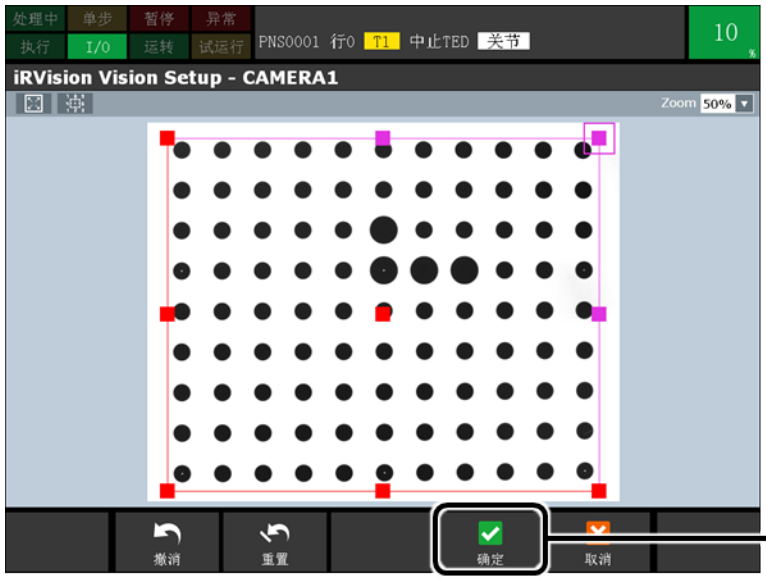


22　按下［校准向导］



23　确认在画面上已显示点阵板，按下 F4［确定］。

24 用紫红色的长方形窗口圈出画面显示的点阵。
25 按下 F4［确定］。



26 运行传送带，当点阵板在机器人到达动作范围内中央附近后关停。

27 触碰点阵板的原点。

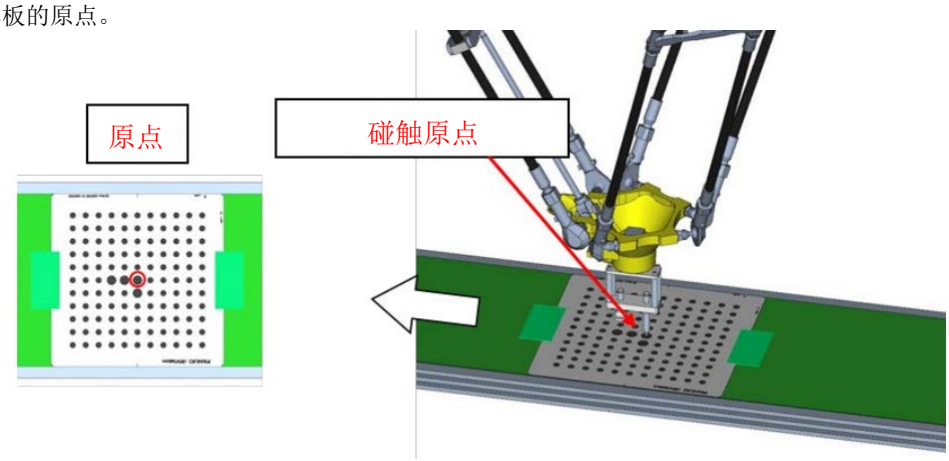


图 3.2.7.1(e) 触碰点阵板的原点

28　按下 F4［确定］。

29　在传送带停止的状态下触碰点阵板的 X 轴方向。

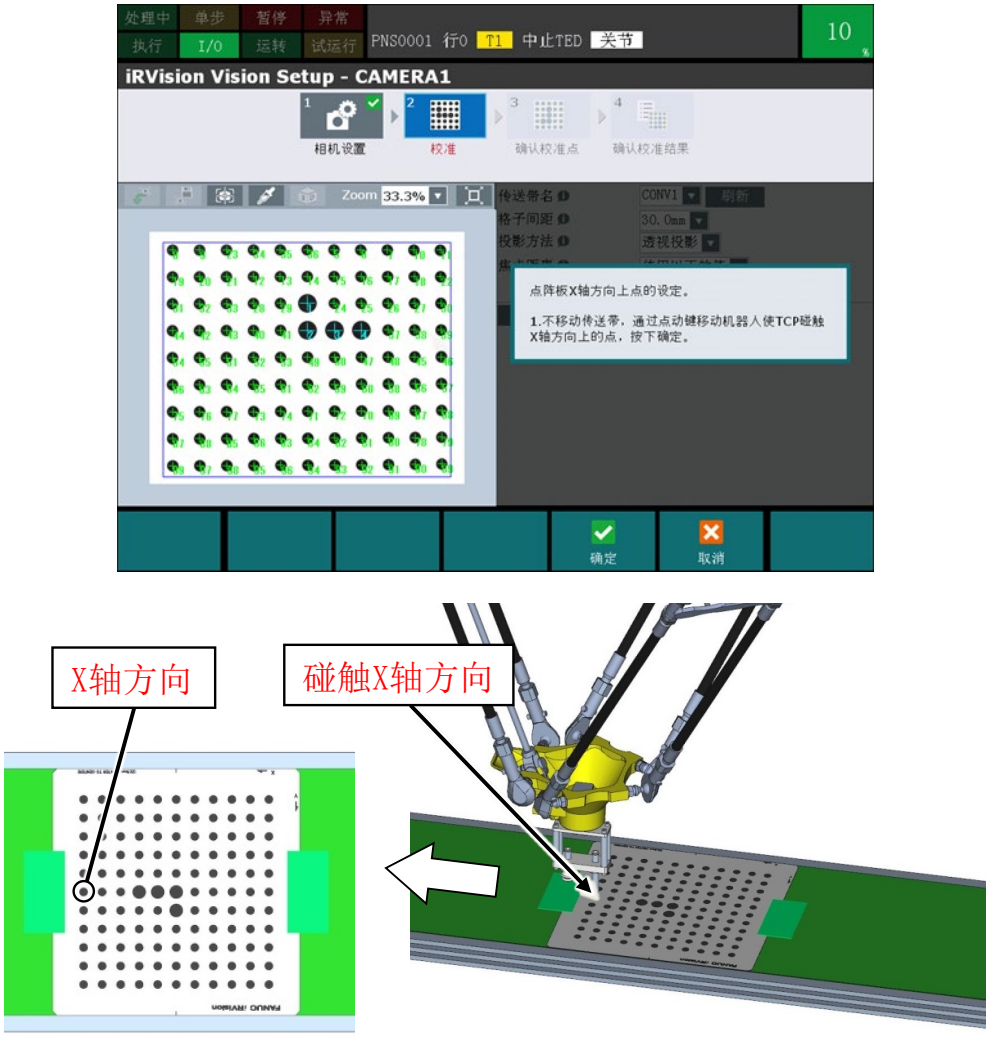


图 3.2.7.1(f)　触碰点阵板的 Y 轴方向

30 按下 F4［确定］。

31 在传送带停止的状态下触碰点阵板的 Y 轴方向。

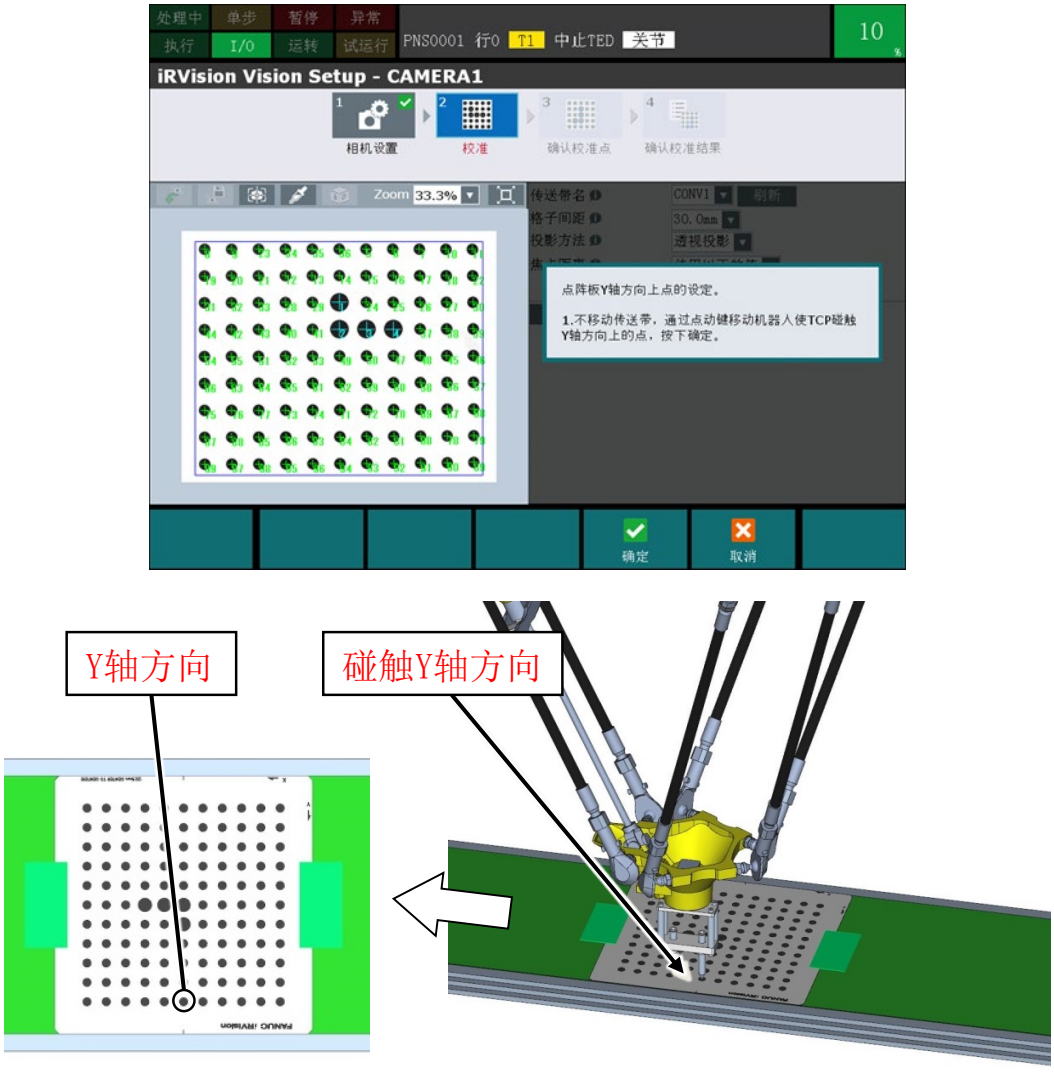


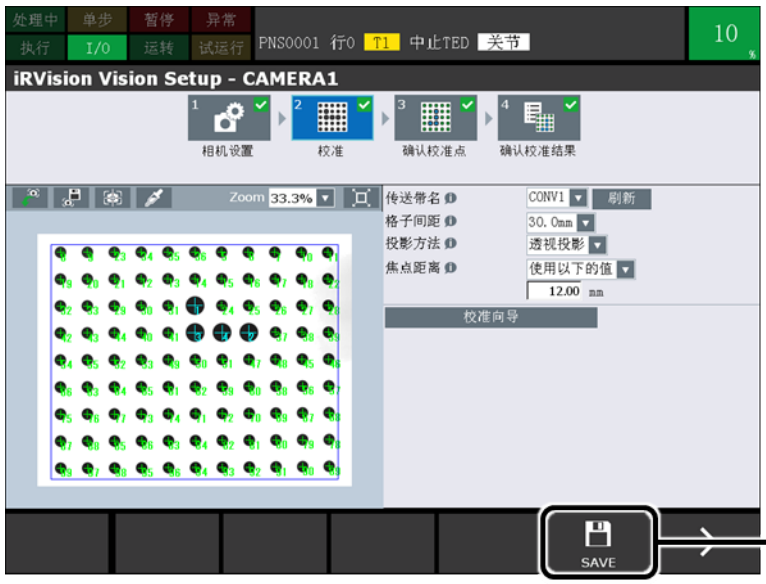
图 3.2.7.1(g) 触碰点阵板的 Y 轴方向

32 按下 F4［确定］。

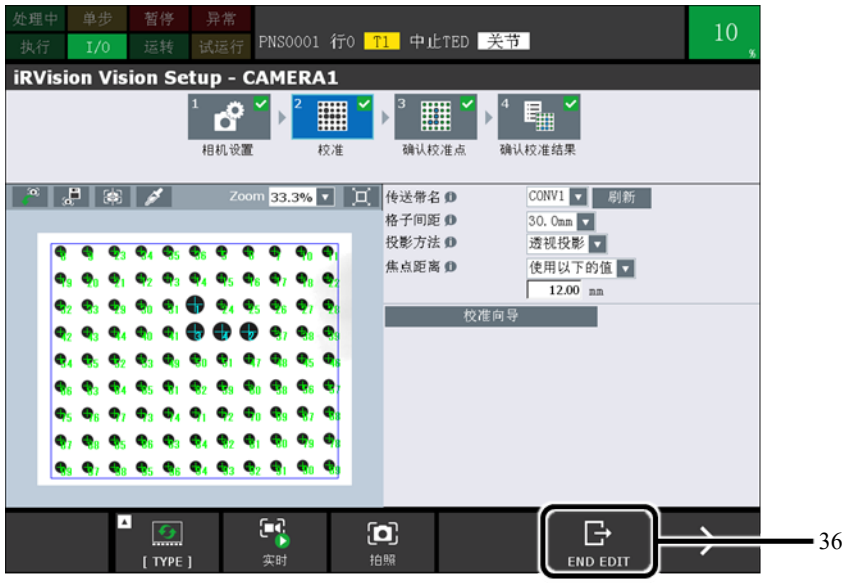
33 显示计算的基准距离。确认实际基准距离处的值妥当后按下 F4［确定］。如果值不妥当，请确认设定步骤及设定的参数是否正确。

34 按下 F4［确定］。

35 按下［＞（下一页）］，再按下 F5［保存］。



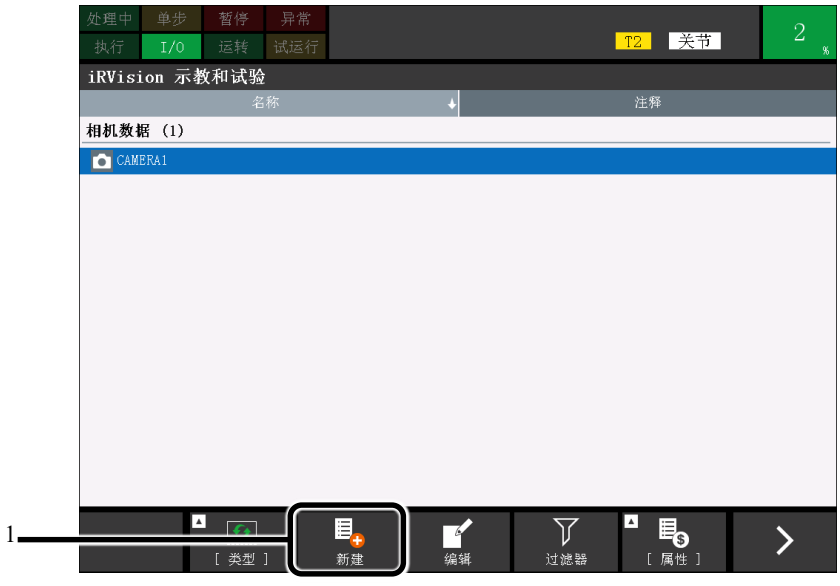
36　再次按下［>（下一页）］，按下 F5［结束编辑］。



3.2.7.2　视觉程序

设定视觉程序。

1　按下 F2［新建］。



显示“创建新的视觉数据”的画面。

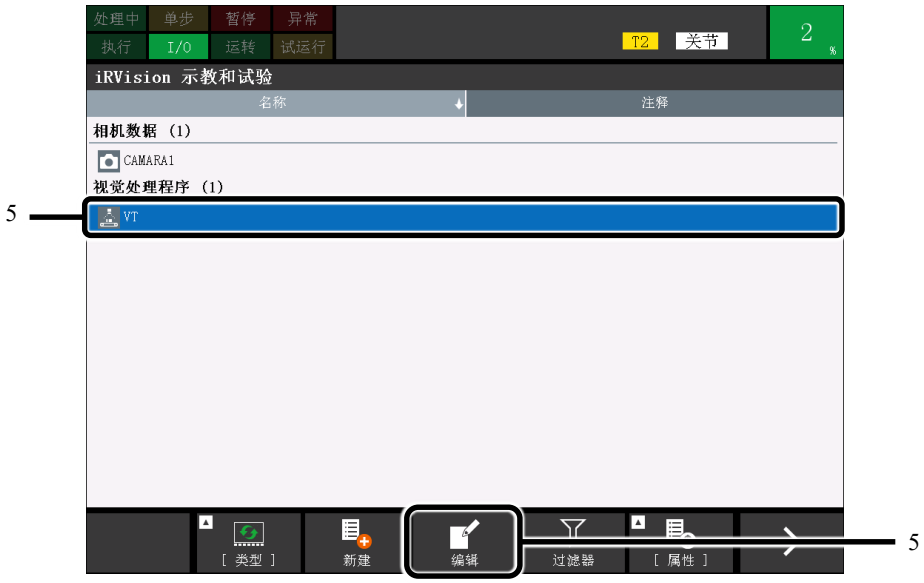
2　选择［2-D Single-View Visual Tracking］，按下 F4［下一个］。

显示以下画面。

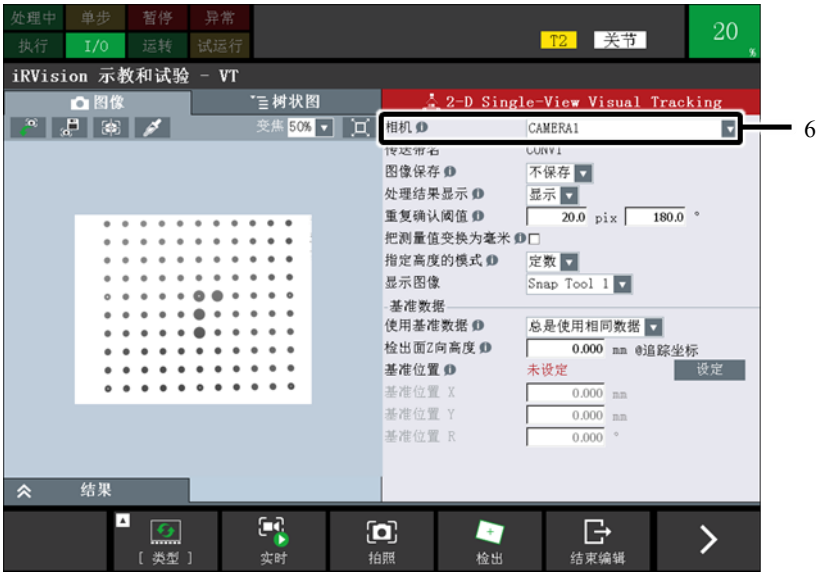
3　在［名称］中输入视觉程序名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例　：VT

4　按下 F4［确定］。

5　选择创建的视觉程序，按下 F3［编辑］。



6　从菜单选择［相机］。



7　测量工件的厚度。
实测从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。

8 在［检出面 Z 向高度］中输入工件厚度。
输入从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。

注释

［检出面 Z 高度］从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。请用尺子等测量。系统与“检出面 Z 高度”的关系如下图。

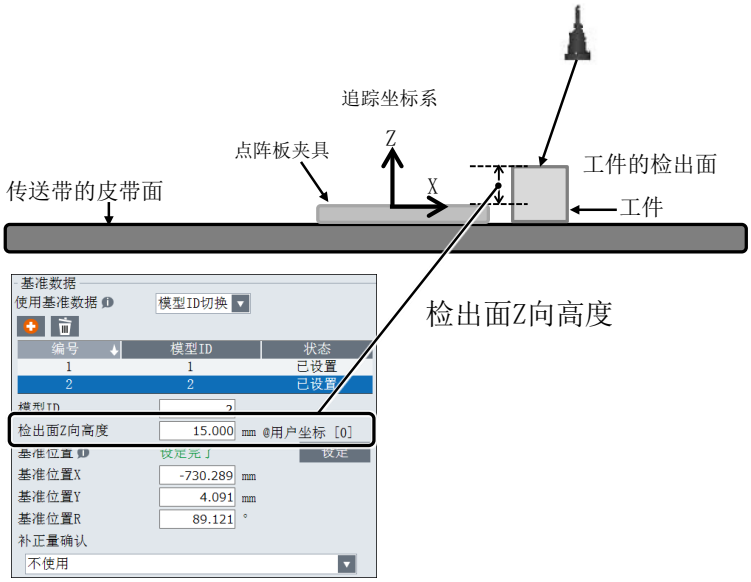


图 3.2.7.2(a)系统与“检出面 Z 向高度”的关系

9　按下树状图标签，选择［Snap Tool 1］。

10　输入［曝光时间］。

在设定曝光时间时，要使曝光中移动的工件图像偏差尽量小。

例如，考虑图像的偏差，传送带速度为 100 mm/s 时，为了将视觉检出时的位置偏差控制在 0.1 mm 以下，就要设定低于 1 ms 的值作为曝光时间。

0.1(mm) ÷ 100(mm/s) = 1(ms)

关于根据传送带速度计算合适的曝光时间，请参阅“2.5 适用探讨”中的“曝光时间”。

11 将工件安装在相机视野内。
如果已供给方向已确定，将调整方向再安装。
供给姿势为±180 度时，调整成放置侧的姿势，使示教变得容易。

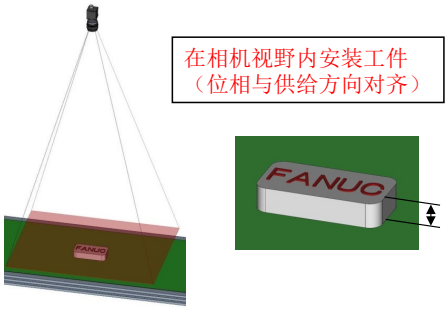


图 3.2.7.2(b) 安装工件

12 按下树形图标签，选择［GPM Locator Tool 1］。

13 按下 F3［拍照］。
确认工件已显示在画面。

14 按下［模型示教］。

15 用紫红色的长方形窗口圈出工件。

16 按下 F4［确定］。

17 按下［遮蔽］中的［编辑］。

18 选择画面上方的工具，将不要匹配的部分涂成红色。
19 按下 F4［确定］。

20 根据需要设定搜索范围。
关于详细设定请参阅“iRVision 操作说明书 (参考篇)”中的“命令工具”章“图形匹配工具(GPM Locator Tool)”节。

21 按下［＞（下一页）］，再按下 F5［保存］。

22　再次按下［>（下一页）］，按下 F5［结束编辑］。

3.2.8 传送带工作站设定

设定传送带工作站。

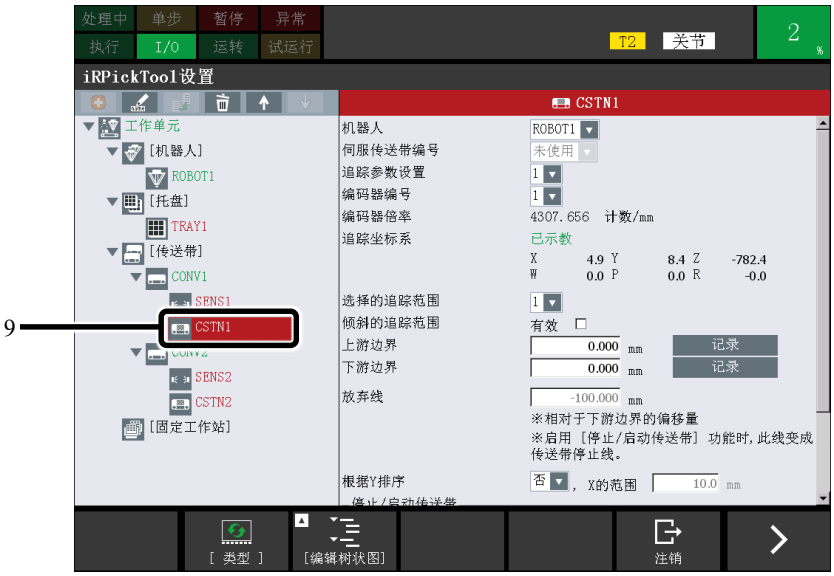
1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。
显示工具坐标系一览画面。

5 按下 F5［切换］。
6 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

7 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
8 从菜单选择［设置］→［*i*RPickTool］。

9　首先设定供给传送带的传送带工作站。选择传送带工作站［CSTN1］。



10　在作业区上游点动移动机器人。

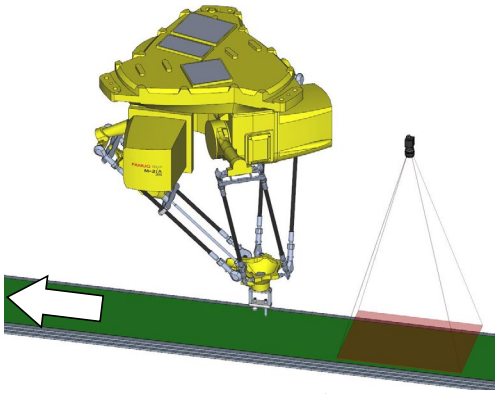


图 3.2.8(a)　点动移动机器人

11　按下［上游边界］中的［记录］。
记录追踪坐标系上的当前位置。

12 在作业区下游点动移动机器人。

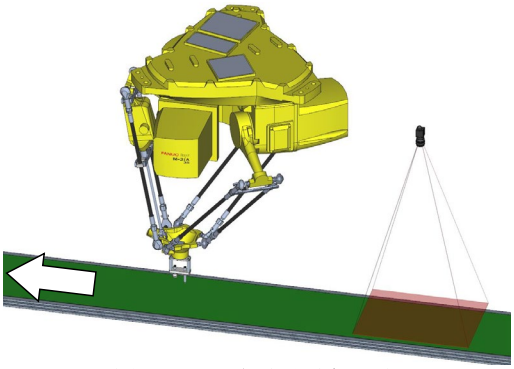


图 3.2.8(b) 点动移动机器人

13 按下［下游边界］中的［记录］。
记录追踪坐标系上的当前位置。

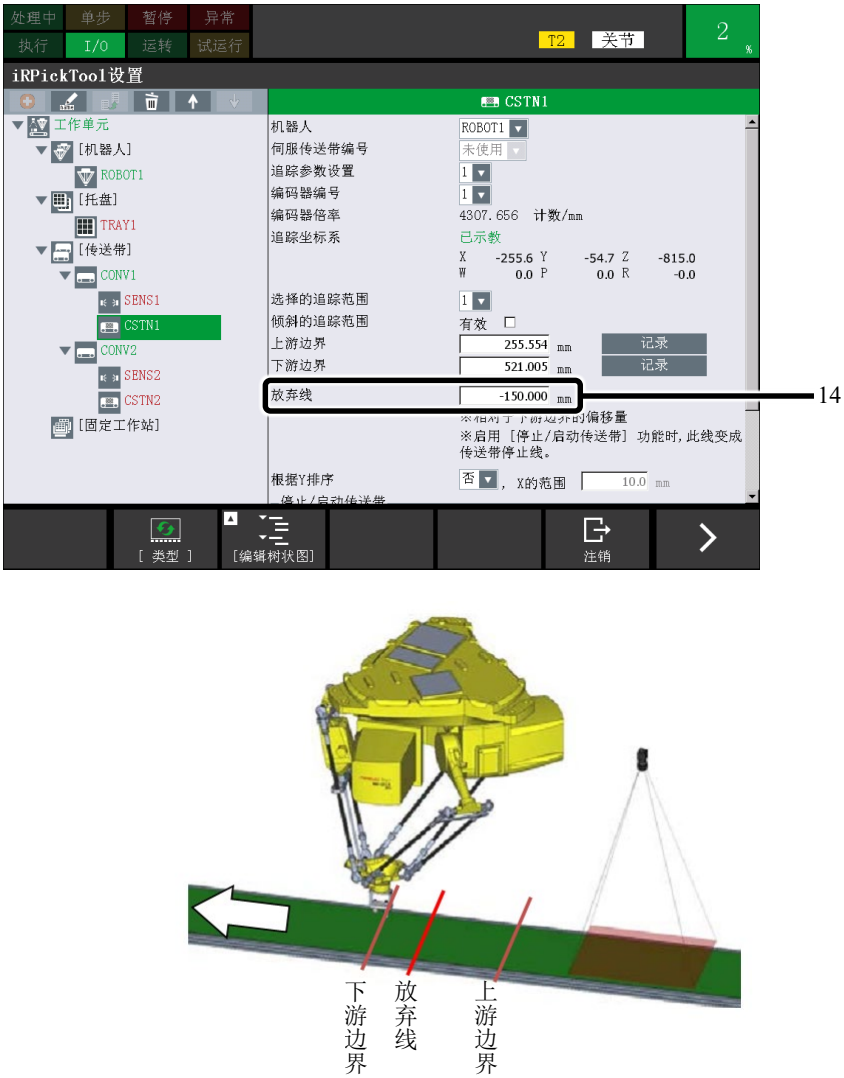
14 输入［放弃线］。
跳过距离值需要设成大于以下计算数值的值。

追踪所需时间（sec）×传送带速度（mm/sec）
※ 追踪所需时间是指机器人从开始对某个工件进行追踪动作到结束对该工件的作业（取出等）的时间。

例如，机器人的动作时间约为 0.8 sec、传送带速度为 150 mm/sec 时，传送带自走距离为
0.8（sec） × 150（mm/sec） ＝ 120（mm）。

因此，要留一些余量，将跳过距离设成－150 mm。

与放弃线有关的详细内容请参阅“6.6 传送带工作站设定”设定。



放弃线(mm) >
追踪所需时间(sec)× 传送带速度(mm/sec)

图 3.2.8(c) 放弃线的设定

［上游边界］、［下游边界］可通过直接输入进行修正。

15　然后设定排出传送带的传送带工作站。选择传送带工作站［CSTN2］。
16　通过与步骤 10 到 14 相同的步骤，设定排出传送带的传送带工作站［上游边界］、［下游边界］、［放弃线］。

3.2.9　机械手工具坐标系的设定

设定机械手工具坐标系。

此处例子为 4 轴拳头机器人的设定。
在 4 轴拳头机器人中，即使机械接口坐标系中的机械手工具中心点(X, Y) 不是(0, 0)，也建议将工具坐标系的 X, Y 设成 0。原因是为防止从传送带抓取工件时，基本轴（J1～J3 轴）通过巨大运动，使循环时间延迟，或是超出机器人动作范围。
此外，为将工具的 Z 轴向上，直接输入工具的姿势。直接输入机械接口坐标系绕 X、Y、Z 轴的工具坐标系旋转角 W、P、R 的数值。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。
　显示菜单。
2　从菜单选择［设置］→［坐标系］。

3　按下 F3［坐标］。

4　从菜单选择［工具］。

显示工具坐标系一览画面。

5　将光标对准设定的工具坐标系编号行，按下 F2［详细］。

画面为利用工具 1 设定时的例子。

显示选中坐标系编号的工具坐标系设定画面。

6　按下 F2［方法］。
7　从菜单选择［直接输入法］。

8　将光标对准［Z］及［W］、［P］、［R］，输入各自的值。
［Z］输入用规尺等测量的数值或是机械手的设计值。
［W］、［P］、［R］直接输入。
※推荐（W、P、R）=（180、0、0）。

9　在机器人控制装置的示教器上按下［PREV］键。
显示工具坐标系一览画面。
可以确认所有工具坐标系的设定值。

10　按下 F5［切换］。

11　输入坐标系编号。
　　启用工具坐标系。
　　※如不进行步骤 10、11，设定的坐标系就不会有效。

3.2.10 基准位置的设定、机器人的位置示教

设定基准位置，示教机器人位置。

3.2.10.1 供给基准位置的设定、机器人的位置示教

设定供给传感带上的基准位置，示教机器人位置。［距离］＋［通过视觉检出］时的步骤。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［*i*RPickTool］。

3 选择传感器［SENS1］，在［视觉程序］中选择按“3.2.7.2 视觉程序”创建的视觉程序名称（VT）。
4 按下 F3［基准位置］。

显示如下［基准位置向导　步骤 1/2］。

注释

如果可以使用［存储检出位置到 PR］，通过使用该功能可以简化后述的机器人位置示教。详细请参阅后述的“在［存储检出位置到 PR］使用位置寄存器的编号时”。

5 将工件放置在相机视野内的中央附近。

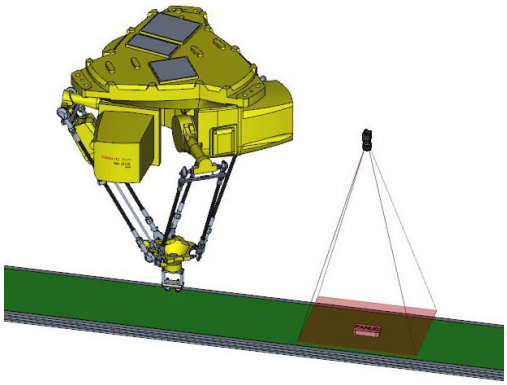


图 3.2.10.1(a) 将工件放置在相机视野内

6 按下 F4［检出］。
如果检出成功，显示［检出位置］。
※ 检出不成功时
- 打开视觉程序，进行检出测试。
- 利用检出测试进行检出时，在传感器画面（步骤 3）确认视觉程序的选择是否正确。

检出成功后，F5［下一步］有效。

7 按下 F5［下一步］。

8 按下 F5［结束］，

显示以下画面。
确认在［基准位置］中显示“向导成功”。

9　运行传送带，工件到达机器人动作范围内后关停。
　※停止传送带时，请注意传送带上工件不要偏离。
10　将机器人点动移动到工件拾取位置。

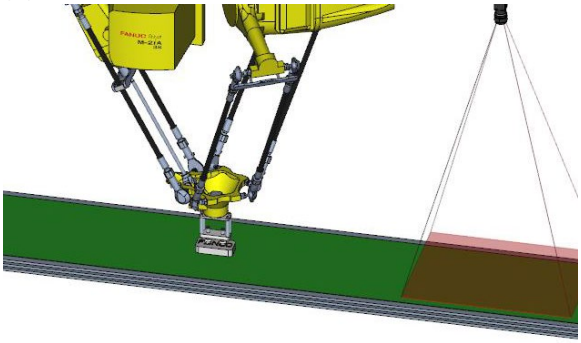


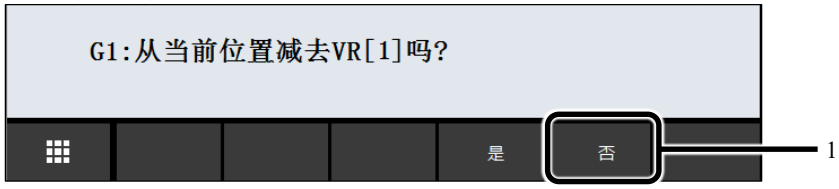
图 3.2.10.1(b)　将机器人移动到工件拾取位置

11 显示拾取程序（PT_PICKCS)，将光标对准示教拾取位置行。
12 ［SHIFT］键的同时按下 F5［TOUCHUP］。
如果未显示 F5［TOUCHUP］、按下［＞（下一页）］切换显示。

13 显示“VR 未初始化，是否继续？”的消息后，按下 F4［是］，示教位置。

注释

1 通过重新设定在视觉寄存器中存在数值时，会显示“从当前位置减去 VR 吗？”，按下 F5［否］。



2 通过重新设定在视觉寄存器中存在数值时，为防止错误，按以下步骤清除视觉寄存器的值后再进行位置示教，对于“从当前位置减去 VR 吗？”，不论是选择 F4［是］还是 F5［否］，都会示教相同的位置。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

2 从菜单选择［下页］。

3 从菜单选择［数据］→［视觉寄存器］。

4　将光标对准［VR］中的“1”，按下 F4［详细］。

5　在［偏移］中所有的［X］、［Y］、［Z］、［W］、［P］、［R］值中输入“0”。

在［存储检出位置到 PR］使用位置寄存器的编号时

如果利用步骤 4 所显示的［基准位置向导 步骤 1/2］在［存储检出位置到 PR］中输入位置寄存器编号，就会在位置寄存器中自动设定检出位置作为基准位置，从而可以简化机器人的位置示教。

该功能可使用的条件如下。

- 视觉的型号原点位置正确
- 检出面 Z 向高度正确
- 机械手的工具坐标系设定正确

使用位置寄存器的基准位置设定步骤如下。

1 ［在基准位置向导 步骤 1/2］中勾选［存储检出位置到 PR］。在［PR］中输入被用作拾取位置的位置寄存器编码。拾取程序像后述的样本程序(PT_PICKCS)一样，使用位置寄存器［7］时，输入 7。

2 在［选择工具坐标］中输入工具坐标系编号。

⚠ 注意

选中工具的当前位置决定了存储于位置寄存器的机器人姿势及形态。如果将不同于按“3.2.9 机械手工具坐标系的设定”设定的工具坐标系的编号指定成［选择工具坐标］，将有可能无法出现想要的姿势或形势，存在危险。关于基准位置的设定，请在操作时要十分注意。

3 通过点动操作移动机器人，调整工具的位置，以形成取出工件时的样子。

⚠ 注意

圆弧追踪时，在检出工件时以及工件被传送到追踪领域机器人进行取出时，工件的相位将发生变化。取出姿势与相对于工件被传送到追踪领域机器人进行取出时的工件相位不符，而是变成相对于工件检出时相位的取出姿势。

4　按下 F4［检出］。

如果检测成功，在［检出位置］显示数值。

5　用执行时显示器确认检出是否顺利。

6　如果已在执行显示器上显示，在示教器上一边按［i］键，一边按多次［DISP］键，就会在［iRVision］的执行时显示器上显示检出结果的画面。如果没有显示执行时监视，请让［iRVision］的执行时监视显示出来。

确认检出结果后，一边重新按下［i］键，一边多次按［DISP］键，就会返回基准位置向导的画面。

注释

如果检出不成功，在 *i*RVision 示教和试验画面显示视觉程序，执行检出测试。在 *i*RVision 示教和试验画面，不论检测测试是否成功，如果无法在执行时显示器上确认检测结果，请确认视觉程序是否正确。

如果检出成功，显示［检出位置］。

7　按下 F5［下一步］。

显示确认按步骤 2 设定的位置寄存器消息。

8　按下 F4［确定］。

在位置寄存器中注册检测位置，显示［基准位置向导 步骤 2/2］。

9　在［位置寄存器编号］中输入作为拾取位置偏移使用的位置寄存器编号。拾取程序像后述的样本程序(PT_PICKCS)一样，使用位置寄存器［11］时，输入 11。

⚠ 注意
假设［位置寄存器编号］作为针对追踪坐标系的位置补偿量使用。

10 在［Z］处输入从追踪坐标系看到的 Z 向接近补偿量。生产线传送带通常为正值。

运转传送带，将工件移动到机器人的正面。
一边按下［SHIFT］键，一边按下 F4［移至］。

⚠ 注意
请将画面右上角显示的速度倍率设成 10%以下，小心作业。

11 吸附盘的顶端点经由接近点碰触检测位置。

注释
为了触碰正确位置，需要正确设定视觉程序的型号原点位置与工具坐标系。
如果碰触位置不正确，请用点动操作将机器人移动到合适位置，并按下 F3［记录］。

12 按下 F5［结束］。

使用位置寄存器的基准位置设定完成。
［iRPickTool 设定］画面显示［向导成功］。

3.2.10.2 排出基准位置的设定、机器人的位置示教

设定排出传送带上的基准位置，示教机器人位置。即［DI］/［RI］/［HDI］＋［投入工件］时的步骤。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［*i*RPickTool］。

3 选择传感器［SENS2］，按下 F3［基准位置］。

显示以下画面。

4 将托盘放在触发传感器上游。

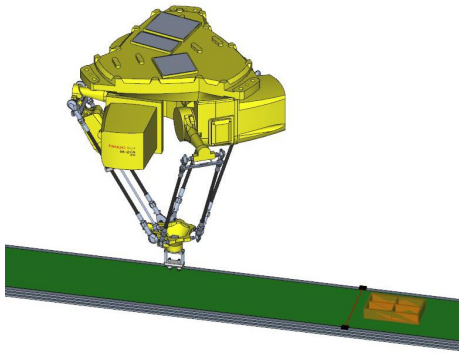


图 3.2.10.2(a) 将托盘放在触发传感器上游

5 运行传送带，托盘到达机器人动作范围内关停。
如果检出成功，显示［触发数］。

6 按下 F5［下一步］。
检出成功后，F5［下一步］有效。

7　如果按下 F5［结束］，

8　显示以下消息。按下 F5［确定］。

显示以下画面。
确认在［基准位置］中显示［向导成功］。

9　运行传送带，托盘到达机器人动作范围内关停。
※请注意不要让托盘偏离。

10　将机器人点动移动到托盘单元 1 的放置位置。

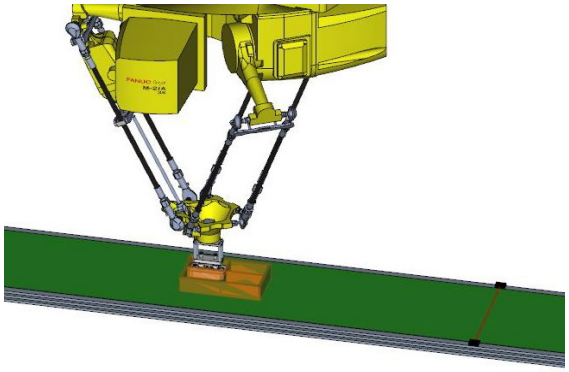


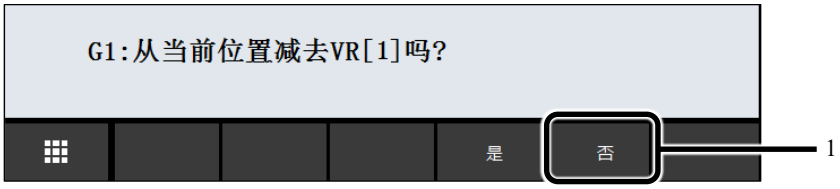
图 3.2.10.2(b)　将机器人移动到托盘单元 1 的放置位置。

11 显示放置程序（PT_DROPCS)，将光标对准示教放置位置行。
12 ［SHIFT］键的同时按下 F5［TOUCHUP］。
如果未显示 F5［TOUCHUP］、按下［＞（下一页）］切换显示。

13 显示“VR 未初始化，是否继续？”的消息后，按下 F4［是］，示教位置。

注释

1 通过重新设定在视觉寄存器中存在数值时，会显示“从当前位置减去 VR 吗？”，按下 F5［否］。



2 通过重新设定在视觉寄存器中存在数值时，为防止错误，按以下步骤清除视觉寄存器的值后再进行位置示教，对于“从当前位置减去 VR 吗？”，不论是选择 F4［是］还是 F5［否］，都会示教相同的位置。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

2 从菜单选择［下页］。

3 从菜单选择［数据］→［视觉寄存器］。

4　将光标对准［VR］中的“2”，按下 F4［详细］。

5　在［偏移］中所有的［X］、［Y］、［Z］、［W］、［P］、［R］值中输入“0”。

3.2.11　追踪动作的微调整

在需要精度的系统中微调追踪动作。请参照“6.12 追踪动作的微调整”。

4 *i*RPickTool（基本功能+Plug & Play）的启动调试步骤

本章针对 *i*RPickTool（基本功能+ Plug & Play）的启动调试步骤进行说明。
按本章步骤执行，完成使用下图 4 轴拳头机器人的系统启动。

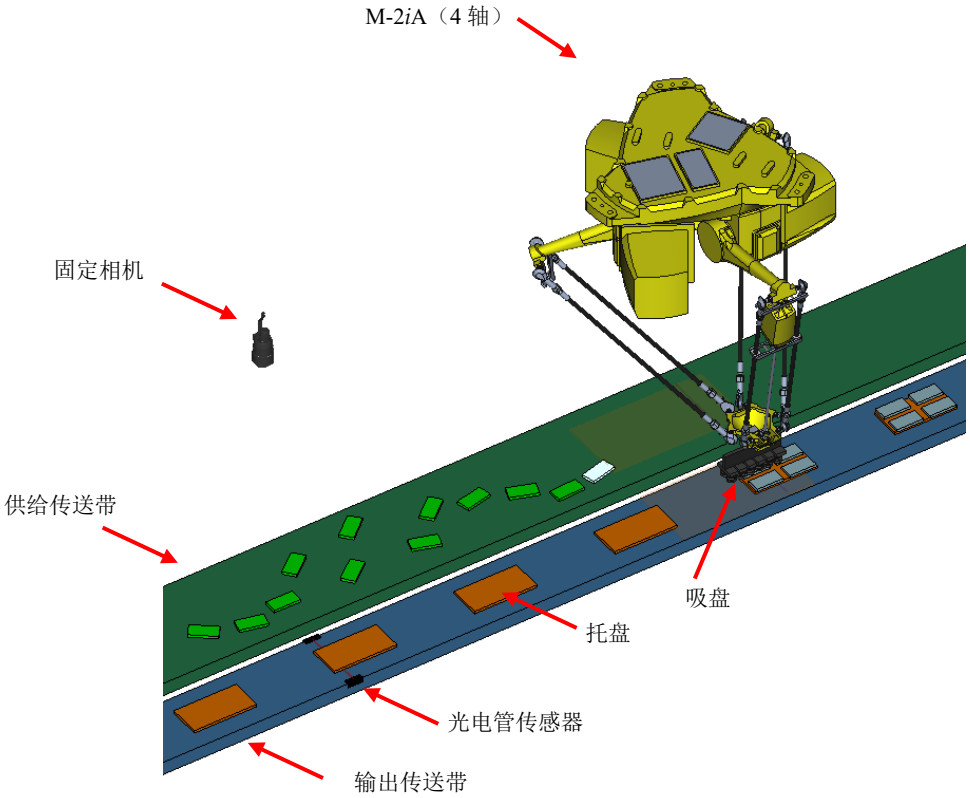


图 4（a） 系统构成

传送带的启动调试步骤内容会因检测的［触发条件］与满足触发条件时执行的［触发］组合以及有无使用托盘而异。
请根据用途选择组合。

本章将以系统构成为例对以下组合进行说明。

供给：［距离］＋［通过视觉检出］
排出：［DI］/［RI］/［HDI］＋［放入工件］（使用托盘）

关于使用其他组合的传送带及固定工作站时的启动调试步骤，各树形图项目详细内容请参阅“6 基本功能参考”。但是，关于标准程序及 Plug & Play 用选项（夹爪/区域/选项）请参阅“7 Plug & Play 参考”。

由于 Plug & Play 提供了标准程序，因此在 *i*RPickTool（基本功能+ Plug & Play）的启动调试步骤中不需要编写程序。

4.1 设定应用程序前的准备工作

本项将针对硬件连接等使用 *i*RPickTool 的追踪系统启动调试准备工作进行说明。

4.1.1 脉冲编码器的连接与设定

安装脉冲编码器，与机器人控制装置连接。
连接后，在机器人控制装置的示教器上设定参数。

注释
本说明书中，将检测传送带行程的装置称为“脉冲编码器”。
在示教器及 *i*RPickTool 设置画面当中，“编码器”这一术语作为“脉冲编码器”的同义词使用。

可以使用 α A1000S 脉冲编码器与量增脉冲编码器。

注释
使用 1 个或 3 个脉冲编码器时，可以将 α A1000S 脉冲编码器连接到主板使用。连接请参阅“附录 D α A1000S 脉冲编码器”，设定请参阅“5.2.3 设定连接到主板的脉冲编码器”。

4.1.1.1 脉冲编码器的连接

将脉冲编码器连接到机器人控制装置。

连接至生产线追踪用接口板时

参考下图进行连接，使安装在供给传送带的脉冲编码器的编码器编号为 1，安装在排出传送带上的编码器编号为 2。

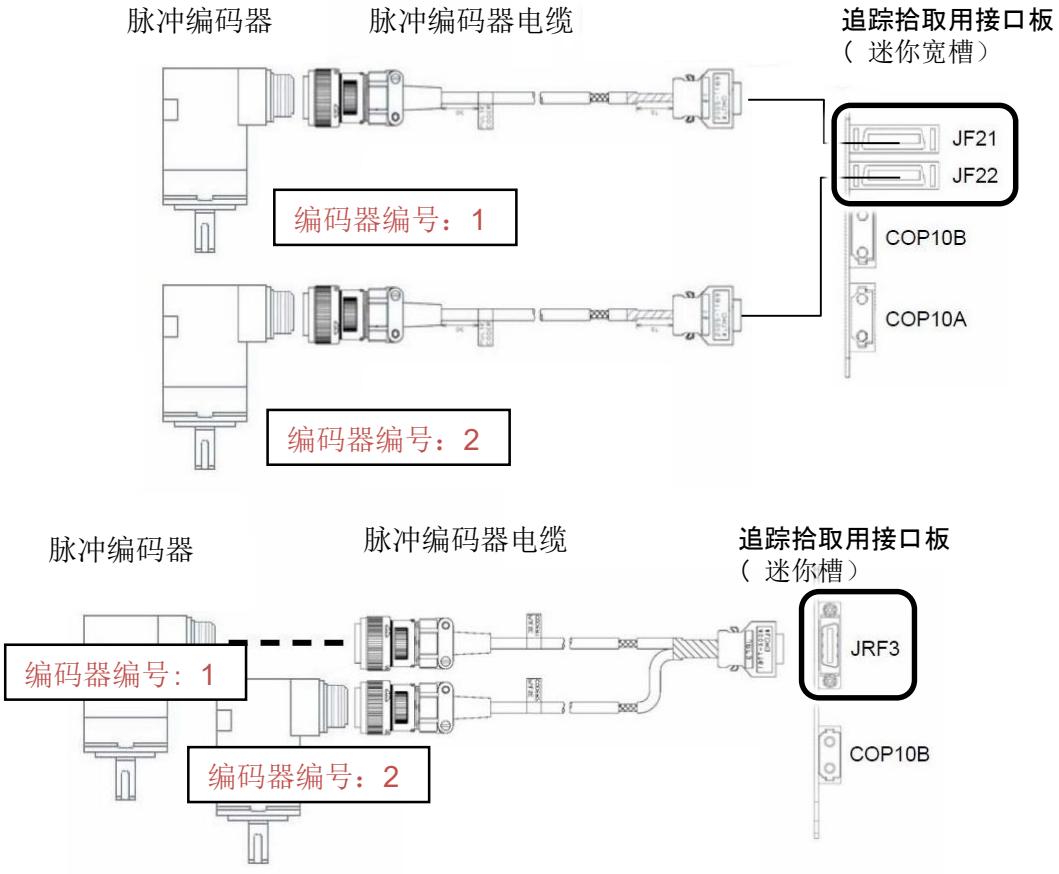


图 4.1.1.1（a） 生产线追踪用接口板与脉冲编码器的连接

4.1.1.2 脉冲编码器的设定

设定脉冲编码器参数。设定以下项目。

- ［编码器轴］
- ［编码器类型］
- ［平均（更新次数）］

1 在机器人控制装置的示教器上按下［MENU］键。
显示菜单。
2 从菜单选择［设置］→［编码器］。

显示以下画面。
设定项目编号［9］以后为可选功能的菜单。此处不设定。

3 首先设定安装在供给传送带上的脉冲编码器参数。确认［编码器编号］为 1。如果数值不是 1，按下 F3［编码器］，输入 1。

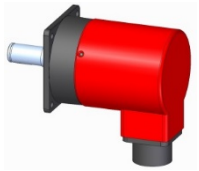
4 把光标对准［编码器轴］，输入 1。但是，在连接到主板的情况下，R-30*i*B Compact Plus / R-30*i*B Mini Plus 时保持初始值 0。

5　将光标对准［编码器类型］，按下 F4［选择］。
　显示菜单。
6　选择与连接脉冲编码器对应的编码器类型。

编码器类型

编码器有以下类型。

- α A1000S 脉冲编码器（红色脉冲编码器）：串行增量式
- 量增脉冲编码器（黑色脉冲编码器）：增加值



α A1000S（A860-0372-T001）
编码器类型：串行增量式

※在连接到主板的情况下，R-30*i*B Compact Plus / R-30*i*B Mini Plus 时为主串行增量式

A860-0301-T001～T004
编码器类型：增加值

※不可连接到主板

如果更改编码器轴或编码器类型，就会显示类似以下要求冷启动的警告。

7　将光标对准［平均（更新次数）］，输入瞬时速度的累积个数（移动平均宽度）。
通过设定瞬时速度的累积个数，即使传送带在机器人追踪中发生突然停止，机器人也不会紧急停止，而是平稳停下。
通常输入［10］。

8　然后设定安装在排出传送带上的脉冲编码器参数。按下 F3［编码器］，输入 2。
9　把光标对准［编码器轴］，输入 2。
10　把光标对准［编码器类型］，按下 F4［选择］，选择与连接脉冲编码器对应的编码器类型。
11　将光标对准［平均（更新次数）］，输入数值。与步骤 7 相同。通常输入［10］。

※　如果在选择中使用 HDI 信号，会显示［9 HDI 端口］，因此输入与使用 HDI 信号相对应的端口编号。

12　最后冷启动机器人控制装置。

4.1.1.3 脉冲编码器的设定确认

1 打开［编码器］画面，将光标对准［编码器启用］，按下 F4［ON］。
2 运转传送带，确认在［当前计数值（计数）］处显示脉冲计数。

3 对于连接至排出传送带的脉冲编码器，也选择编码器编号：2，进行同样确认。

4.1.2 相机的连接确认

确认安装在供给传送带上的相机连接。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［*i*RVision］→［示教和试验］。

显示［*i*RVision 示教和试验］画面。

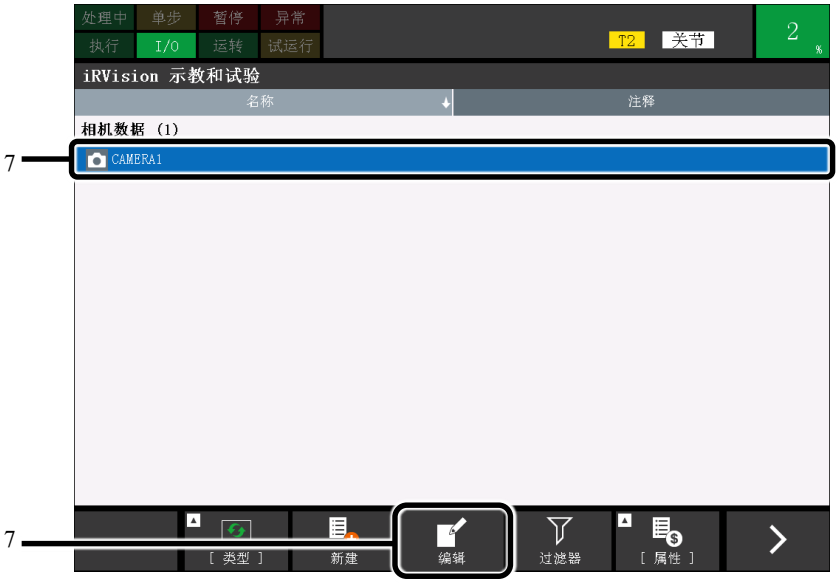
3　按下 F2［新建］。

4　选择［二维相机］，按下 F4［下一个>］。

显示以下画面。

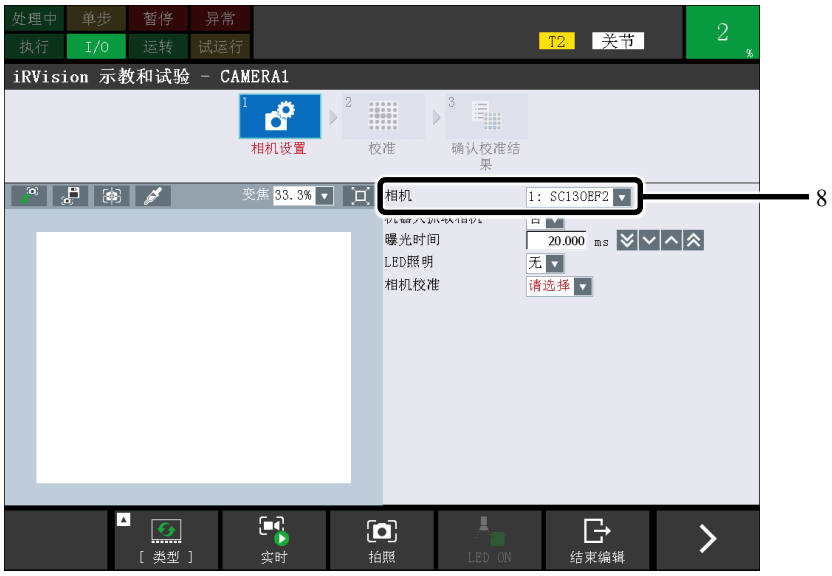
5 在［名称］中输入相机名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例 ：CAMERA1
6 按下 F4[确定]。

在列表中显示创建的相机数据。
7 选择创建的相机数据，按下 F3［编辑］。



显示以下画面。

8 从菜单选择［相机］。



9 为进行镜头的焦点调整，将写有文字的物品放在相机视野内，按下 F2［实时］。
确认在画面左侧的框内可以拍摄出放置的物品。

※ 图像较暗时，调整以下。

- 打开镜头光圈。
- 更改照明。
- 更改曝光时间。

10 确认相机视野。
视野过大或过小时，调整相机位置。

11 相机位置确定后，调整镜头焦点。

图 4.1.2（a） 调焦

12 按下 F2［停止］。
实景停止。
13 按下 F5［结束编辑］。
编辑画面关闭。

4.1.3 触发传感器的输入确认

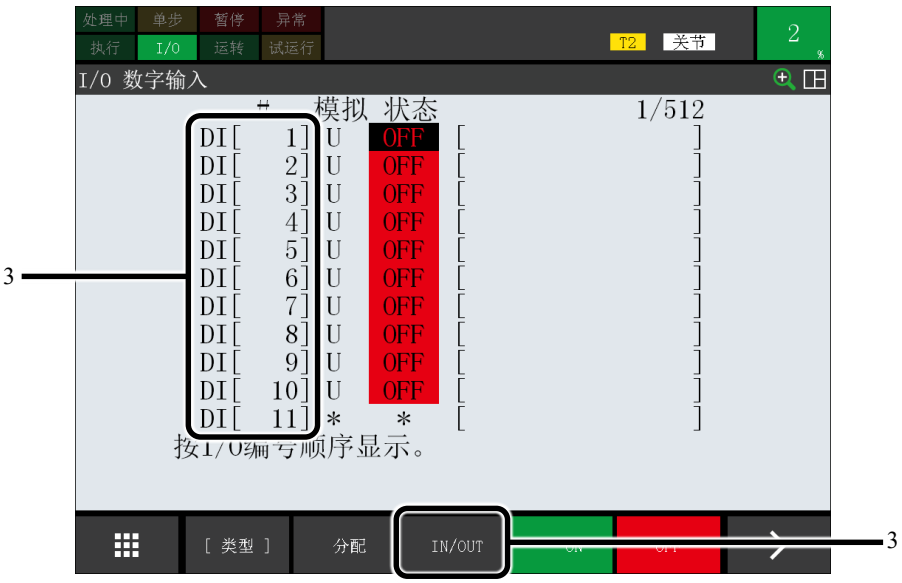
确认触发传感器的输入

4.1.3.1 使用 ［DI］/［RI］

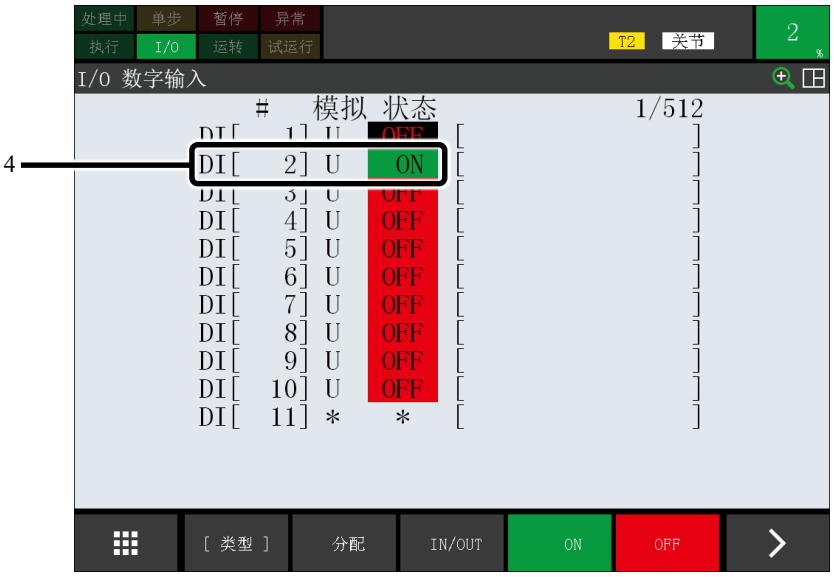
确认使用 DI/RI 时的触发传感器输入

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 使用 DI 时，从菜单选择［I/O］→［数字］。使用 RI 时，从菜单选择［I/O］→［机器人］。
画面为使用 DI 时的样子。

3 按下 F3［IN/OUT］，确认显示为 DI（RI）。



4　通过从传感器的输入确认 DI（RI）是否变成 ON。

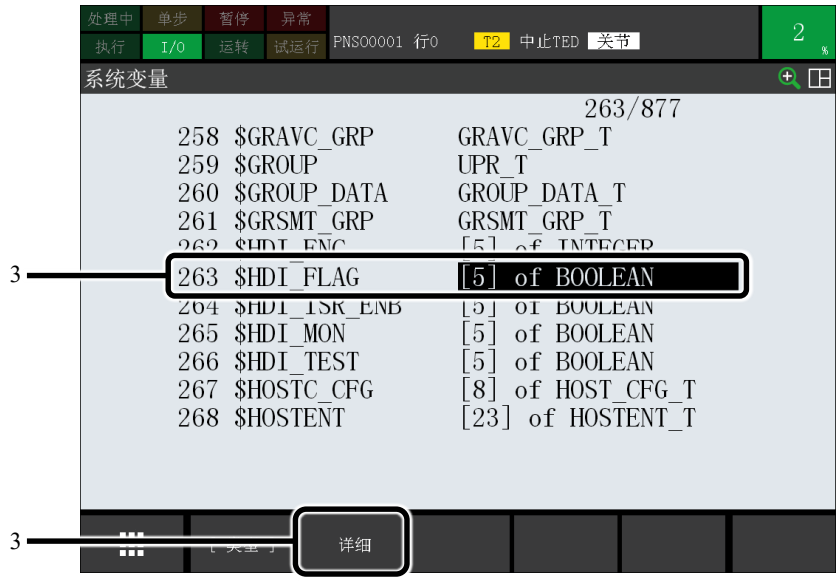


4.1.3.2　使用［HDI］

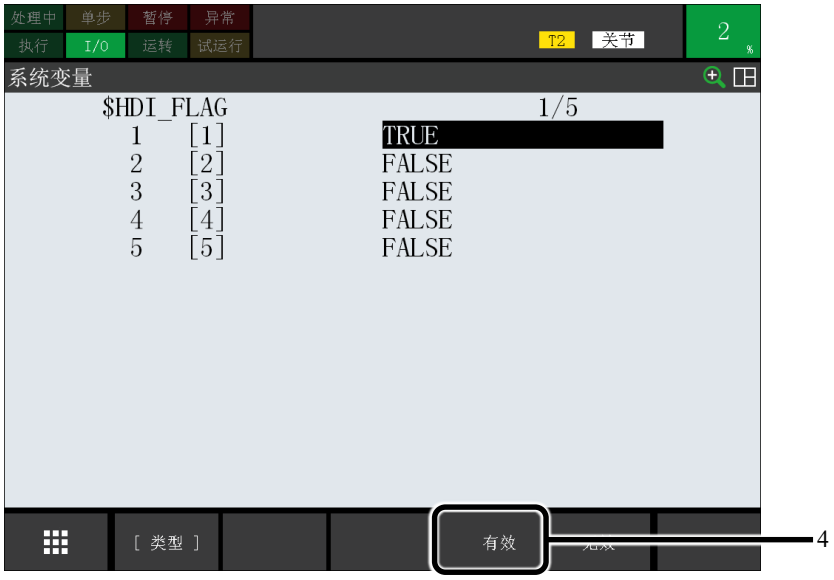
确认使用 HDI 时的触发传感器输入。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。
　显示菜单。
2　从菜单选择［下页］→［系统］→［变量］。

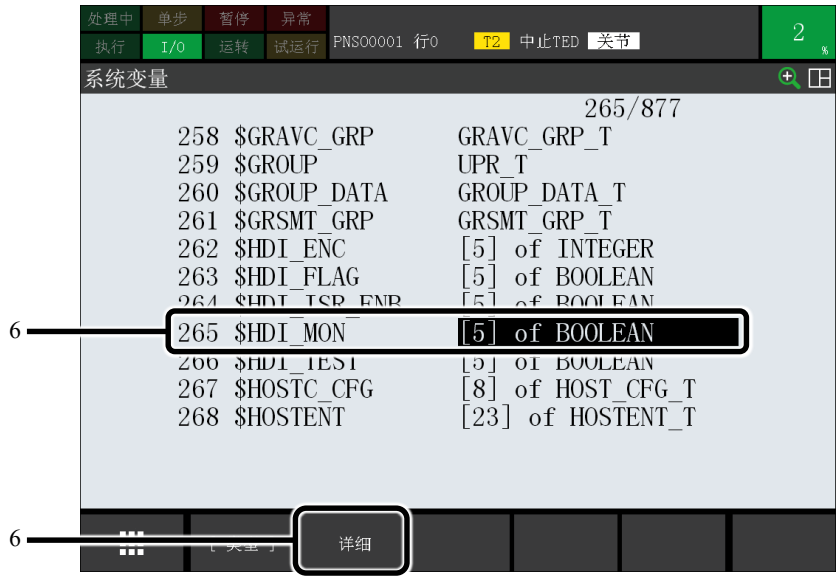
3　将光标对准“$HDI_FLAG”，按下 F2［详细］。



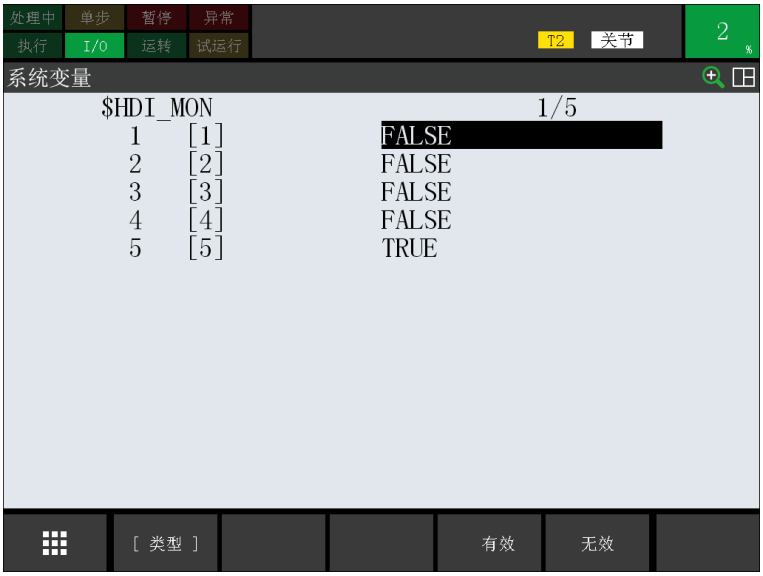
4　将光标对准确认的 HDI 端口号，按下 F4［有效］。



5　在机器人控制装置的示教器上按下［PREV］键。
　显示［系统变量］画面。
6　将光标对准“$HDI_MON”，按下 F2［详细］。

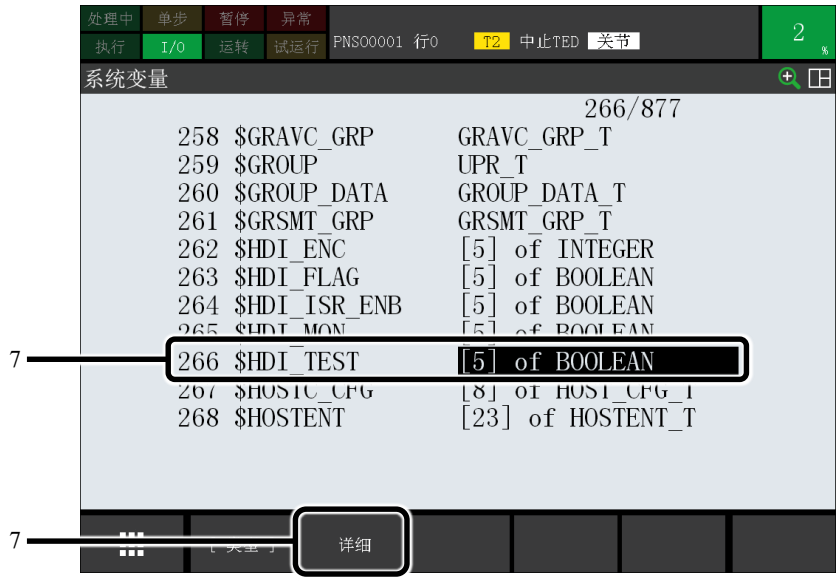


确认与连接到传感器端口号相对应的变量。
输入 ON　：TRUE
输入 OFF　：FALSE

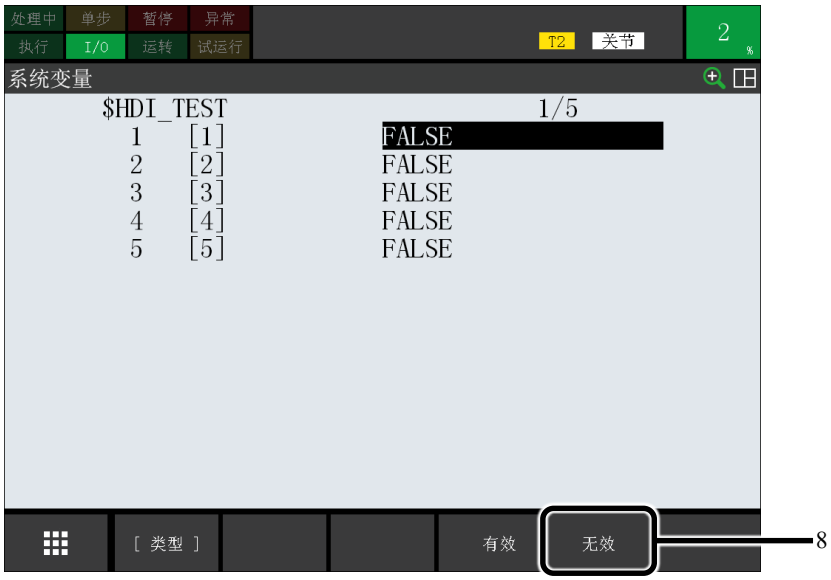


输入时间短，无法按步骤 7 的方法确认时，按以下方法确认。

7　将光标对准“$HDI_TEST”，按下 F2［详细］。



8　将光标对准确认的 HDI 端口号，按下 F5［无效］。
输入一设成 ON 就会变成 TRUE，即使输入变成 OFF，也保持 TURE 状态。



4.1.4 设定碰触销的工具坐标系

设定碰触销的工具坐标系。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 显示工具坐标系之外时，按下 F3［坐标］，从菜单选择［工具］。

4 将光标对准设定的工具坐标系编号行，按下 F2［详细］。
显示选中坐标系编号的工具坐标系设定画面。
画面为利用工具 9 设定时的例子。

⚠ 注意
在 Plug & Play 标准程序中使用多个机械手时，可从 1 号开始按顺序使用只可抓住的工件个数的工具坐标系。为防止工具坐标系被覆盖，在未使用的工具坐标系编号处设定触碰销的工具坐标系。

5 按下 F2［方法］。
显示菜单。
6 从菜单选择［二点+Z］。

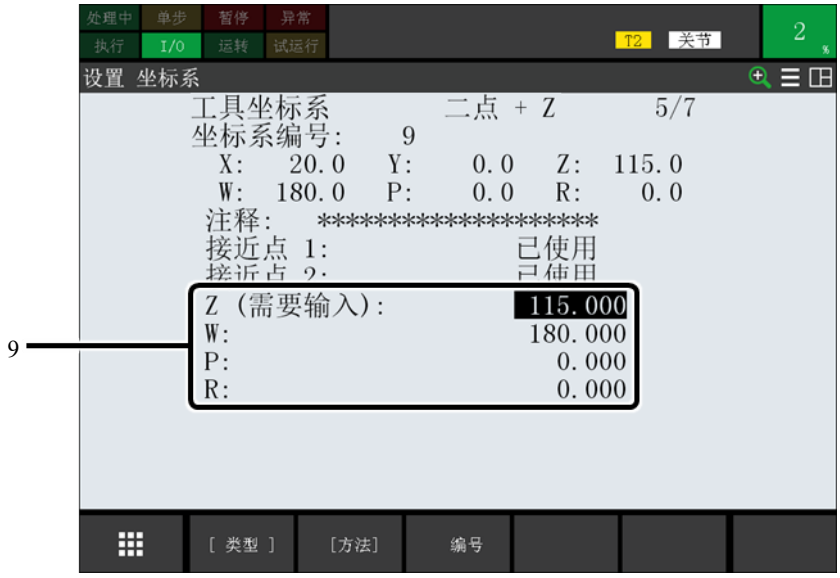
显示以下画面。

7 将光标对准［接近点 1］，输入接近点 1。
通过点动移动到应记录机器人的点。
在按下示教器［SHIFT］键的同时，按下 F5［记录］，将现值的数据作为接近点输入。
显示“已记录”。

8 将光标对准［接近点 2］，输入接近点 2。
通过点动移动到应记录机器人的点。
在按下示教器［SHIFT］键的同时，按下 F5［记录］，将现值的数据作为接近点输入。
输入全部的接近点后，显示“已使用”，设定工具坐标系的 X、Y。

⚠ 注意
1 输入接近点时，请将法兰盘面朝下。
2 请在不同的位置输入接近点 1 和 2 。
3 如果没有满足以上条件，将显示“输入点设置无效”的消息。

9 将光标对准［Z］及［W］、［P］、［R］，输入各自的值。
［Z］用输入规尺等测量的数值。
［W］、［P］、［R］直接输入。
※推荐（W、P、R）=（180、0、0）。



10 在机器人控制装置的示教器上按下［PREV］键。
显示工具坐标系一览画面。
可以确认所有工具坐标系的设定值。

11 按下 F5［切换］。
12 输入设定的工具坐标系的坐标系编号。

启用工具坐标系。
※ 如不进行步骤 11、12，设定的坐标系就不会有效。
13 将机器人的点动进给设定成［工具］。
在机器人控制装置的示教器上，在将点动进给切换成“工具”之前按下［手动进给］键。

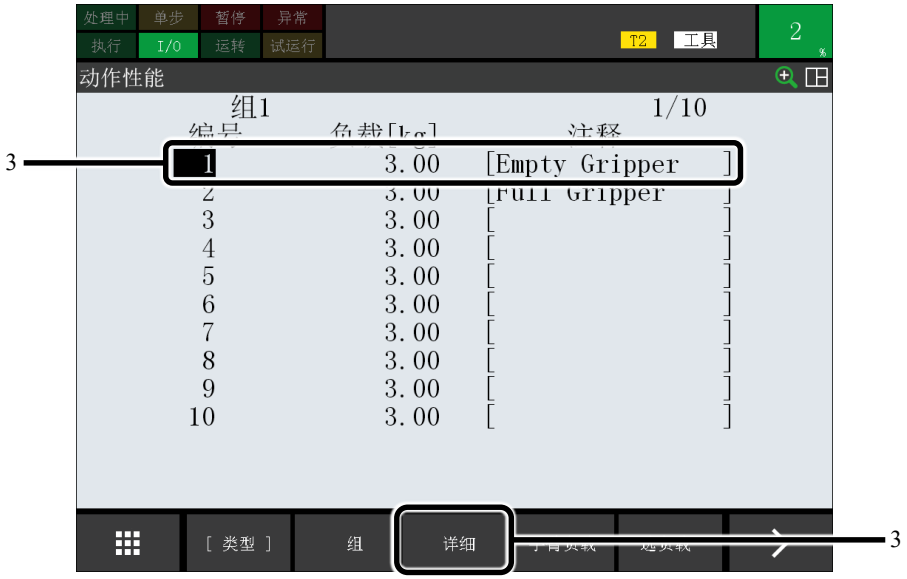
14 在机器人控制装置的示教器中按下 -Z (J6) 键或 +Z (J6) 键。
Z 轴旋转进行点动动作。
确认工具坐标系（TCP）设定正确。

4.1.5 负载设定（动作性能）

在［动作性能］画面设定机器人的负载。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［系统］→［动作］。

3 将光标对准［1］，按下 F3［详细］。



显示负载条件 No.1 的设定画面。

4 至少输入［负载］。并不一定要输入所有项目。
※ 请设定“仅机械手”的负载。

5 在机器人控制装置的示教器上按下［PREV］键。
返回动作性能一览画面。

6 将光标对准［2］，按下 F3［详细］。

显示负载条件 No.2 的设定画面。

7 至少输入［负载］。并不一定要输入所有项目。
※ 请设定“机械手+工件”的负载。

标准程序中使用各设定［*］命令切换负载。

4.2 应用程序的设定

本项中针对工作单元及传感器设定等 *i*RPickTool 的应用程序设定进行说明。

4.2.1 工作单元设置

设定工作单元。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［*i*RPickTool］。

显示以下“*i*RPickTool 设置”画面。

3　选择［机器人］，按下树形图上面的［新建］按钮。

4　在［名称］中输入机器人名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。

5　按下 F4[确定]。
创建对象。

6　在［控制装置］处从菜单选择［此控制装置］（机器人为 1 台时）。
7　在［组编号］处从菜单选择［1］。

8　首先创建供给传送带的对象。选择［传送带］，按下树形图上面的［新建］按钮。

9 在［名称］中输入供给传送带的名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
10 按下 F4[确定]。
创建对象。

新建传送带后，就会自动创建传感器和传送带工作站。

11 选择按步骤 10 自动创建的传送带工作站［CSTN1］。
12 在［机器人］中从菜单选择机器人。
选择按步骤 4 输入的机器人名称。

选择机器人名称后，传送带工作站操作［OP_CS_CSTN1］就会自动添加到选定的机器人下面。

关于传送带工作站操作的设定请参阅“4.2.10 操作的设定”。

13 从菜单选择［追踪参数设置］。选择“1”。

14 从菜单选择［编码器编号］。选择“1”。

15 然后创建排出传送带的对象。选择［传送带］，按下树形图上面的［新建］按钮。

16 在［名称］中输入排出传送带的名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。

17 按下 F4[确定]。
创建对象。

新建传送带后，就会自动创建传感器和传送带工作站。

18 选择按步骤 17 自动创建的传送带工作站［CSTN2］。

19 在［机器人］中从菜单选择机器人。
选择按步骤 4 输入的机器人名称。

选择机器人名称后，传送带工作站操作“OP_CS_CSTN2”就会自动添加到选定的机器人下面。

20　从菜单选择［追踪参数设置］。选择“2”。

21 从菜单选择［编码器编号］。
选择“2”。

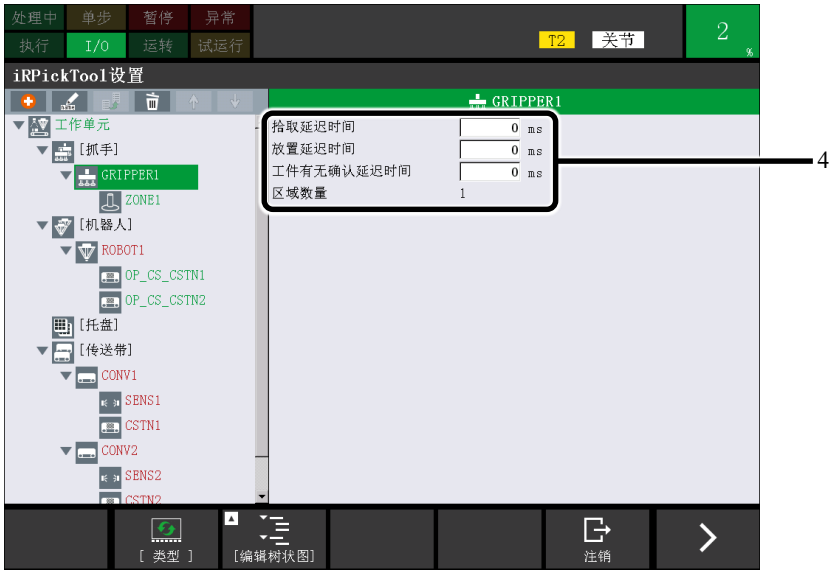
4.2.2 抓手的设定

设定夹爪和区域。
在卡爪中输入适合拾取/放置动作的待机时间。卡爪的子对象的区域设定中，输入工具坐标系、抓住输出信号、抓住确认输入信号等。

1 选择树形视图中的［夹爪］，按下树形图上的［新建］ 按钮。

2 在［名称］中输入抓手名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
3 按下 F4[确定]。

添加抓手。在树形视图［抓手］的下面自动添加区域。
4 设定［拾取延迟时间］、［放置延迟时间］、［工件有无确认延迟时间］。
- 拾取延迟时间：机器人到达拾取位置后设定待机的时间。
- 放置延迟时间：机器人到达放置位置后设定待机的时间。
- 工件有无确认延迟时间：设定从发出区域激活 I/O 到执行工件有无确认的时间。



5　选择［ZONE］。
显示区域的设定画面。

6　设定［区域 工具坐标系］、［区域激活 I/O］、［区域工件有无 I/O］。

- 区域 工具坐标系：设定与卡爪各区域对应的工具坐标系。
  ※此时，区域 工具坐标系可以保持未设定状态。按“4.2.11 机械手工具坐标系的设定”设定。
- 区域激活 I/O：输出值为“ON”、“OFF”或是［脉冲］。指定［脉冲］为输出值时，以秒为单位指定脉冲输出时间。
- 区域工件有无 I/O：输入信号的值为［ON］或者是［OFF］。

区域的添加

使用多个拾取机械手时，按以下步骤将区域添加到卡爪。

1　选择希望在树形视图上添加的［GRIPPER］，按下［新建］按钮。

在新建画面显示［ZONE 2］。
在［名称］中自动输入标准名称（例：ZONE2）。请根据需要进行更改。

2　按下 F4[确定]。

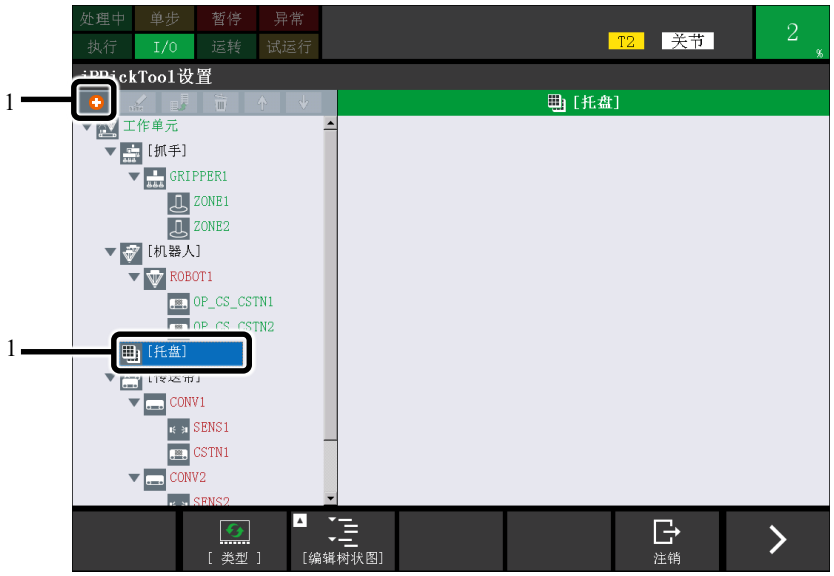
在树形视图中添加［ZONE 2］。

3　如果要在这个卡爪中再创建多个区域，按区域反复进行步骤 1、2。

4.2.3 托盘设定

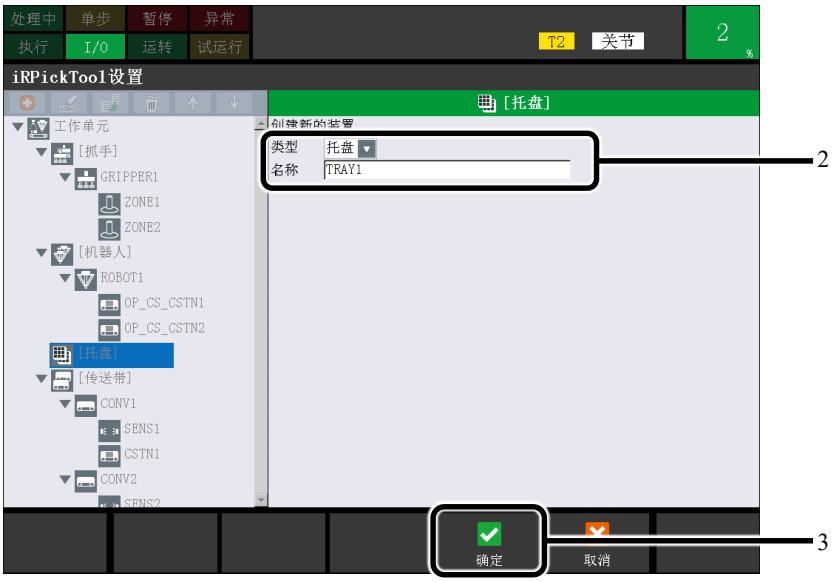
设定在排出传送带上使用的托盘。

1 选择［托盘］，按下树形图上面的［新建］按钮。



2 在［名称］中输入托盘名称。
推荐直接使用初始值显示的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。

3 按下 F4[确定]。
创建对象。



显示以下画面。

4 确定托盘的坐标系。
原点为任意位置。
如果使传送带的传送方向与托盘坐标系的 X 方向一致将比较容易明白（其他方向也没问题）。

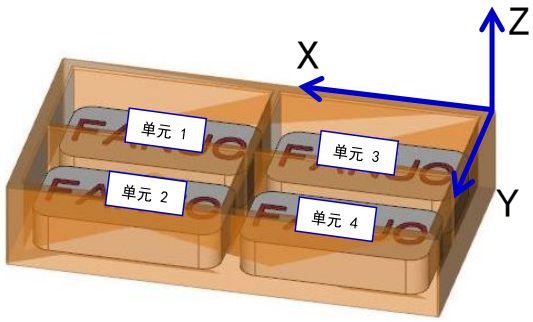


图 4.2.3（a） 确定托盘的原点

5 按下［单元］。
6 从菜单选择［追加单元］。

7　追加第 1 层的单元。

8　在单元编辑画面输入各单元的坐标值。
选择要编辑的单元行，输入到画面下面的框内。
关于单元的坐标值，输入托盘坐标系上的坐标值。请参照详细“6.3 托盘设定”。

托盘堆 2 层时，进行以下操作。

9　按下［层操作］。

10　从菜单选择［追加层］。

11　在［单元数］中输入第 2 层的单元数。

12　在［默认 Z（mm）］中输入 Z 值（mm）。

13　按下 F4［确定］。

显示以下画面。
在［层编号］中创建层“2”。

14 在单元编辑画面输入层 2 的各单元坐标值。
选择要编辑的单元行，输入到画面下面的框内。
※此时，基准单元保持未示教也没有问题。按“4.2.12 基准位置的设定、机器人的位置示教”自动示教。

4.2.4 传送带设定

进行传送带设定。根据有无使用托盘，步骤不同。

4.2.4.1 供给传送带的设定

设定供给传送带。不使用托盘时的设定。

1 选择传送带［CONV1］。

显示传送带的设定画面。
传送带的设定可直接使用初始值所显示的以下数据。

［追踪类型］ ：直线
［伺服传送带］ ：没有勾选［已使用］
［托盘］ ：不使用
［同一物检查的阈值］ ：10.0mm
［负载平衡］ ：进度一定，所有模型 ID 是共通的

4.2.4.2 排出传送带的设定

设定排出传送带。即使用托盘时的设定。

1 选择传送带［CONV2］。

显示传送带的设定画面。

2 ［托盘］选择［TRAY1］，［初始状态］选择［空］。

传送带之外的设定可直接使用初始值所显示的以下数据。

［追踪类型］ ：直线

［伺服传送带］ ：没有勾选［已使用］

［同一物检查的阈值］ ：10.0mm

［负载平衡］ ：进度一定，所有模型 ID 是共通的

4.2.5 追踪坐标系设定

设定追踪坐标系

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。
显示工具坐标系一览画面。

5 按下 F5［切换］。
6 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

7 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
8 从菜单选择［设置］→［*i*RPickTool］。

9　首先设定供给传送带的追踪坐标系。
　选择传送带［CONV1］。
10　按下 F4［设定追踪坐标系］。

显示以下画面。

11 根据画面所显示的步骤，将点阵板安装在传送带上游。
为防止偏差，用胶带等固定。

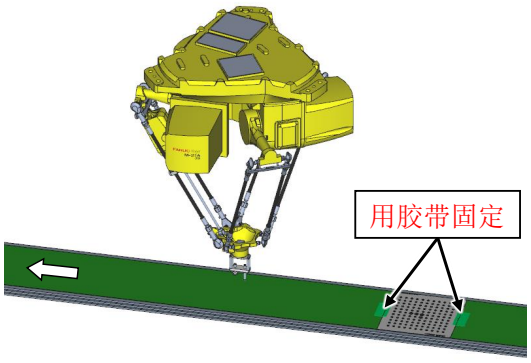


图 4.2.5（a） 将点阵板安装在上游侧

12 安装方向应使传送带的传送方向与点阵板的 X 方向保持同一方向。

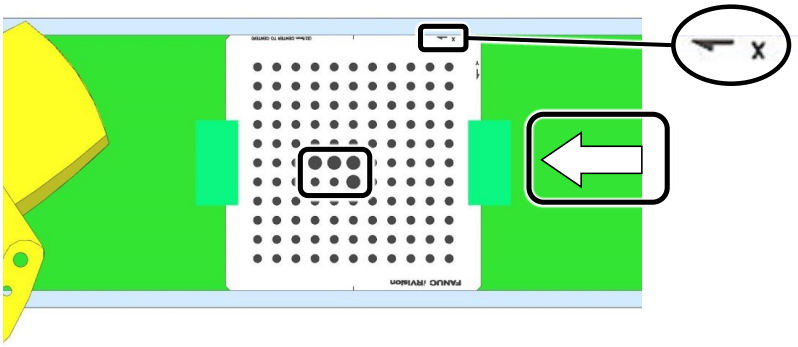


图 4.2.5（b） 对齐安装方向

13 按下 F5［下一步］。

显示以下画面。

14 运行传送带，点阵板在机器人动作范围内到达上游后关停。
15 触碰点阵板的原点。

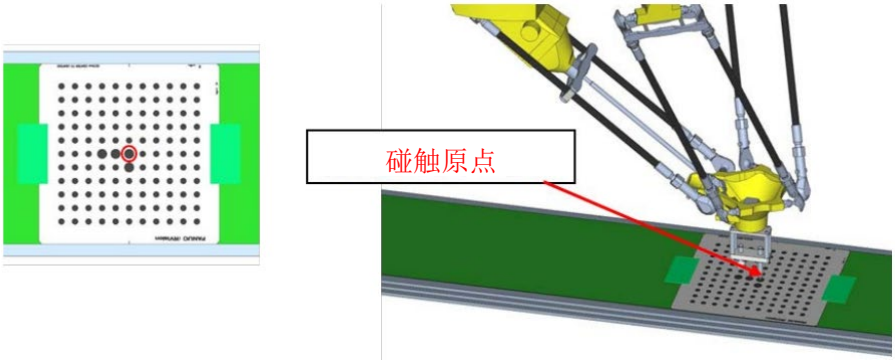


图 4.2.5（c） 触碰点阵板的原点。

16 按下 F5［下一步］。

显示以下画面。

17 使机器人向上空方向移动。
18 运行传送带，点阵板在机器人动作范围内到达下游后关停。

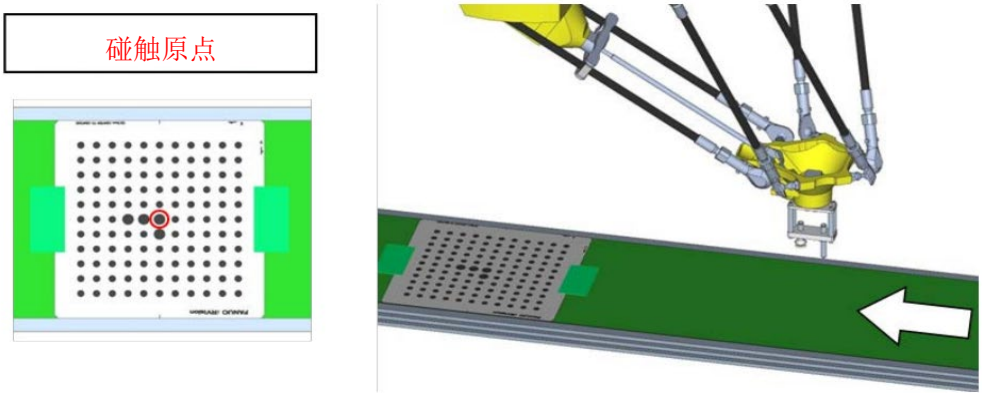


图 4.2.5（d） 停止传送带

19 触碰点阵板的原点。

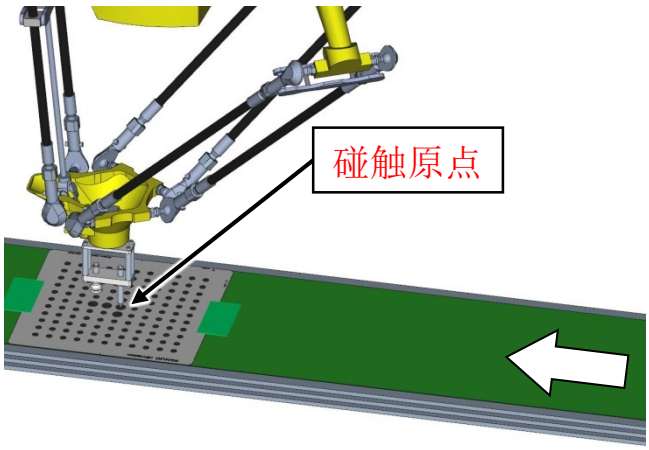


图 4.2.5（e） 触碰点阵板的原点

20 按下 F5［下一步］。

显示以下画面。

21 触碰点阵板的 Y 方向。
此时，传送带不运行。
不是机器人的 Y 方向。对于传送带前进方向，为左侧。

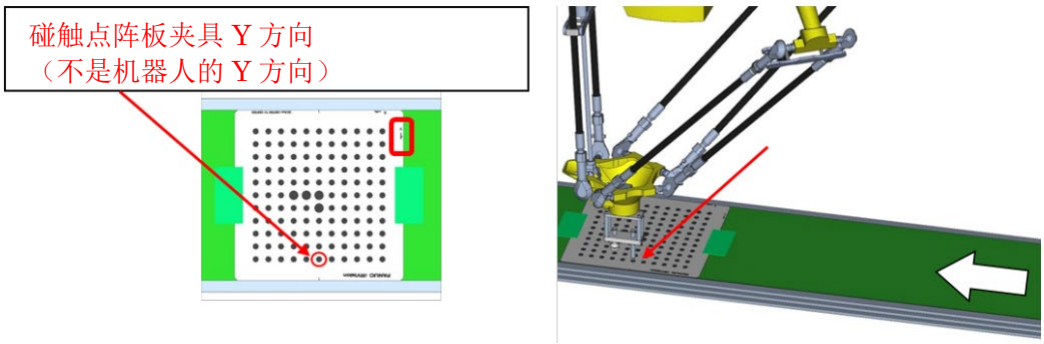


图 4.2.5（f） 向点阵板的 Y 方向触碰

22 按下 F5［下一步］。

23 按下 F5［下一步］。

24 按下 F4［确定］。

设定供给传送带的追踪坐标系。

25 然后设定排出传送带的追踪坐标系。
选择传送带［CONV2］。

26 通过与步骤 10 到 24 相同的步骤，设定排出传送带的追踪坐标系。

4.2.6 传感器设定

进行传感器设定。

4.2.6.1 供给传送带传感器的设定

设定供给传送带的传感器。即［距离］＋［通过视觉检出］时的设定。

1 选择传感器的［SENS1］。

2 设定以下项目。

［本传感器］ ：勾选［有效］
［控制装置］ ：［此控制装置］
［触发条件］ ：［距离］
［触发］ ：［通过视觉检出］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

3 设定以下项目。

［触发距离］ ：输入传送带每前进多少 mm 就会进行视觉检出。
※ 建议在相机视野内将可以检测工件 2 次的距离设为触发距离。

［视觉程序］ ：此时不需要设定。
※ 在即将要设定基准位置前设定。关于设定请参阅“4.2.12.1 供给基准位置的设定、机器人的位置示教”。

4.2.6.2 排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时）

设定排出传送带的传感器。［DI］/［RI］＋［放入工件］时的传感器设定。［HDI］＋［放入工件］时，请跳过该项，进入下一项“4.2.6.3 排出传送带传感器的设定（［HDI］＋［放入工件］时）”。

1 选择传感器的［SENS2］。

2 设定以下项目。

［本传感器］ ：勾选［有效］
［控制装置］ ：［此控制装置］
［触发条件］ ：［DI］
［触发］ ：［放入工件］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

3 设定以下项目。

［DI 编号］ ：输入分配传感器输入的 DI 编号
［再输入 DI 无效时间］ ：输入要禁用的时间（用于防止颤动）
［传感器位置］ ：此时不需要设定
※ 使用传感器位置的设定进行设定。关于设定请参阅“4.2.7 传感器位置的设定”。

跳过下一项“4.2.6.3 排出传送带传感器的设定（［HDI］＋［放入工件］）”，请进入“4.2.7 传感器位置的设定”项。

4.2.6.3 排出传送带传感器的设定（［HDI］＋［放入工件］时）

设定排出传送带的传感器。［HDI］＋［放入工件］时的传感器设定。［DI］/［RI］＋［放入工件］时，请参阅前项“4.2.6.2 排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时）”，而不是本项。

1　选择传感器的［SENS2］。

2　设定以下项目。

［本传感器］　：勾选［有效］
［控制装置］　：［此控制装置］
［触发条件］　：［HDI］
［触发］　：［放入工件］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

3　设定以下项目。

［HDI 端口编号］　：显示通过编码器设定指定的编号
［再输入 HDI 无效时间］　：输入要禁用的时间（用于防止颤动）
［传感器位置］　：该步骤中不需要设定
※ 使用传感器位置的设定进行设定。关于设定请参阅“4.2.7 传感器位置的设定”。

4.2.7 传感器位置设定

设定排出传送带的传感器位置。即使用托盘时的设定。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。

显示工具坐标系一览画面。

5 按下 F5［切换］。

6 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

7 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

8 从菜单选择［设置］→［*i*RPickTool］。

9　选择传感器［SENS2］，按下 F4［传感器位置］。

显示以下画面。

10　将托盘安装到传送带传感器的上游。

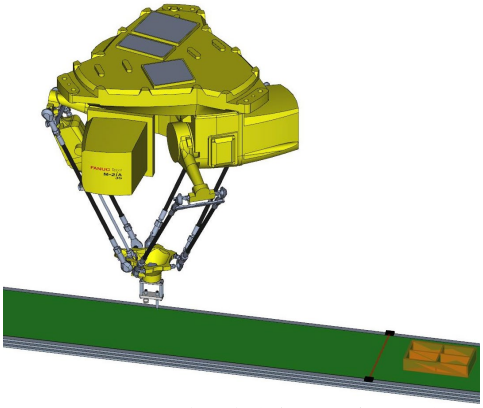


图 4.2.7（a）将托盘安装到传感器上游

11 运行传送带，托盘到达机器人动作范围内关停。
显示传感器检测工件瞬间的触发计数。

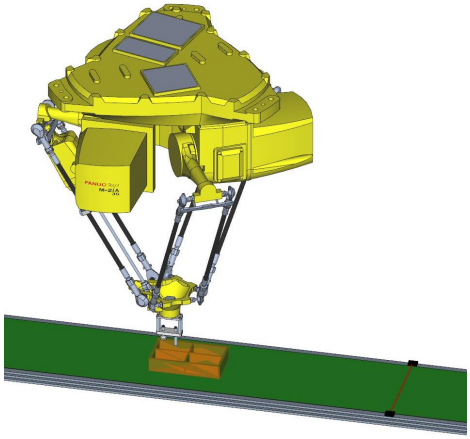


图 4.2.7（b）到达机器人动作范围内后停止

12 按下 F5［下一步］。

显示以下画面。

13 触碰托盘的原点。

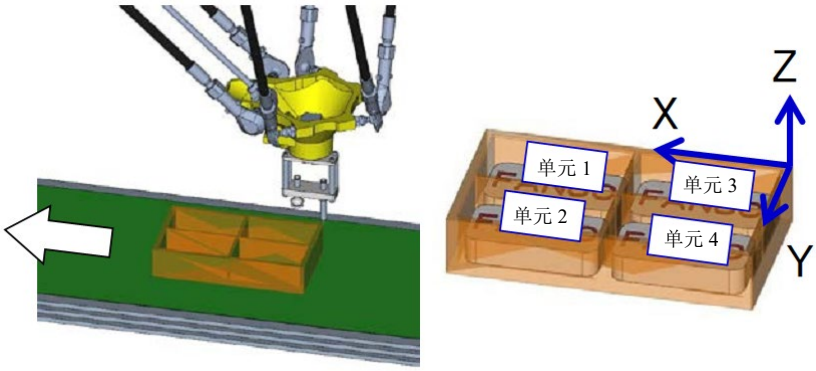


图 4.2.7（c）触碰托盘的原点

14　按下 F4［记录］。
　　显示当前的编码器计数、触碰位置。
15　按下 F5［下一步］。

显示以下画面。

16　触碰托盘的 X 轴方向。

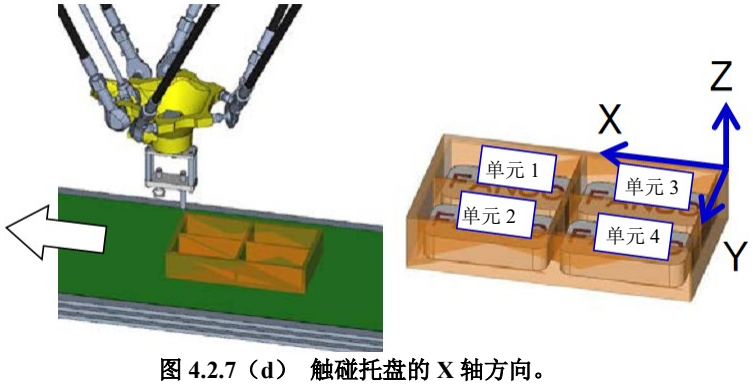


图 4.2.7（d） 触碰托盘的 X 轴方向。

17 按下 F4［记录］。
显示当前的编码器计数、触碰位置。
18 按下 F5［结束］。

返回原来的设置画面。在传感器位置 X、Y、R 输入数值。

4.2.8 视觉设定

设定供给传送带的视觉。

4.2.8.1 相机校准

进行相机校准。

1 准备与相机视野相符的点阵板。
确认右下方的格子间距。

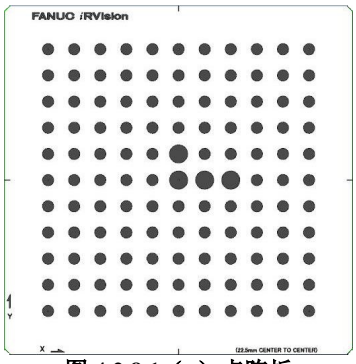


图 4.2.8.1（a）点阵板

> **⚠ 注意**
> **无需在图像上显示出整个点阵。**
> **有几个点超出图像范围也没关系，要使点阵分布在图像中检出工件的整个区域。点阵只反映在图像中检出工件的部分区域的状态下，无法进行高精度校对。**

注释
点阵板固定夹具也可以使用金属板。请根据客户的环境准备。本公司准备了几种标准件。

2 在相机正下方安装点阵板。
为防止偏差，用胶带等固定。

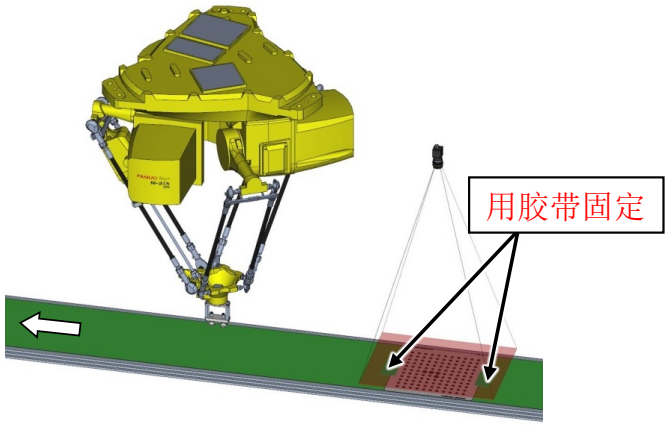


图 4.2.8.1（b） 安装点阵板

3 点阵板的方向可以任意方向。
例如，拾取追踪时，应使传送带的传送方向与点阵板的 X 方向保持同一方向。

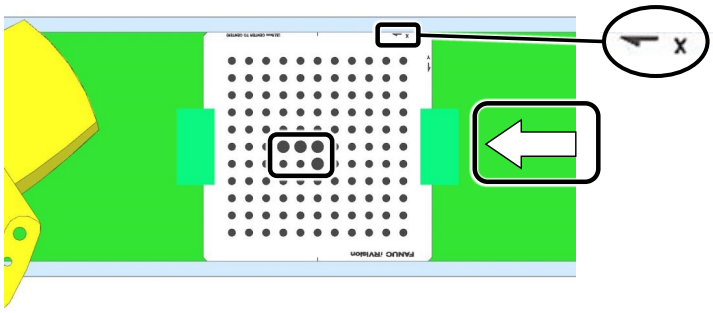


图 4.2.8.1（c）对齐安装方向

4 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
5 从菜单选择［设置］→［坐标系］。

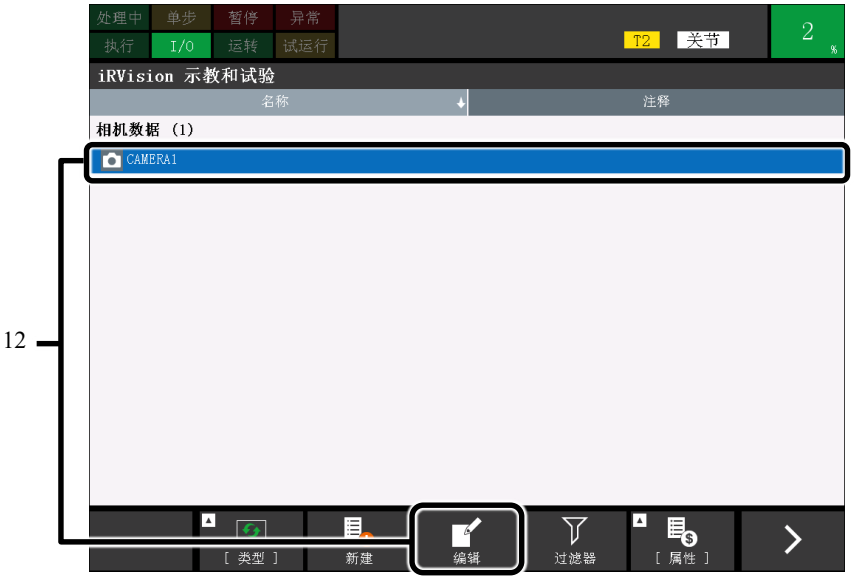
6 按下 F3［坐标］。
7 从菜单选择［工具］。
显示工具坐标系一览画面。

8 按下 F5［切换］键。
9 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

10 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

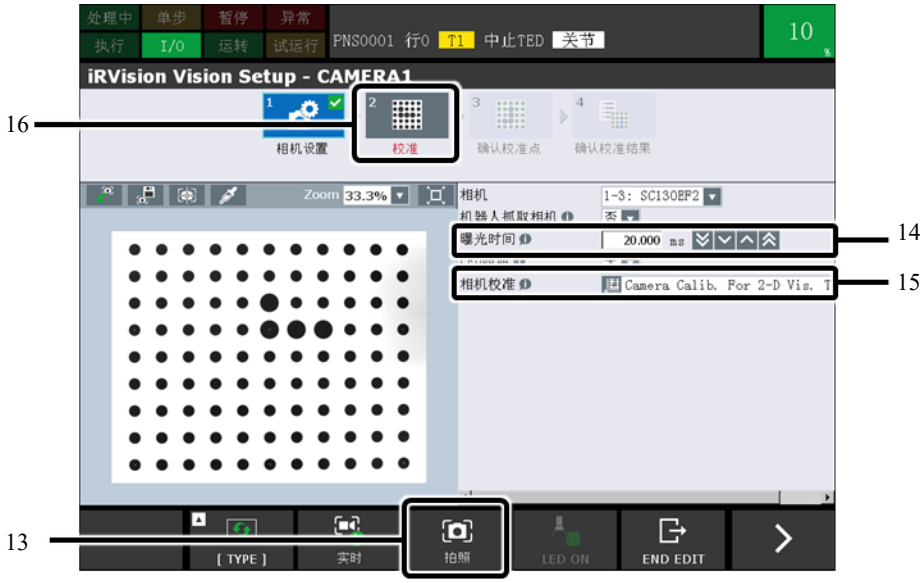
11 从菜单选择［*i*RVision］→［示教和试验］。显示［*i*RVision 示教和试验］画面。

12 选择按“4.1.2 相机的连接确认”创建的相机数据，按下 F3［编辑］。



显示相机的设定画面。

13 按下 F3［拍照］，确认在画面左侧的框内拍摄到点阵板。
14 输入［曝光时间］。
15 在［相机校准］处从菜单选择［Camera Calib. For Vis. Track］。
16 按下画面上方的［校准］。



显示校准的设定画面。

17 选择［传送带名］。
18 输入点阵的［格子间距］。
19 在［投影方法］中选择［透视投影］。
用初始值选择［透视投影］。

> **注意**
> **［正投影］只能在检出的工件高度一定，点阵面与检出的工件上表面高度相同时选择。**

20 在［焦点距离］中输入使用的镜头焦距。

> **注意**
> **选择［自动］时，由于传送带上的点阵板与相机的拍摄面接近平行，从原理上来说，［焦点距离］的计算方法有可能无法正确地计算焦距。**

注释

斜放点阵板，可以用［自动］计算出正确的［焦点距离］。

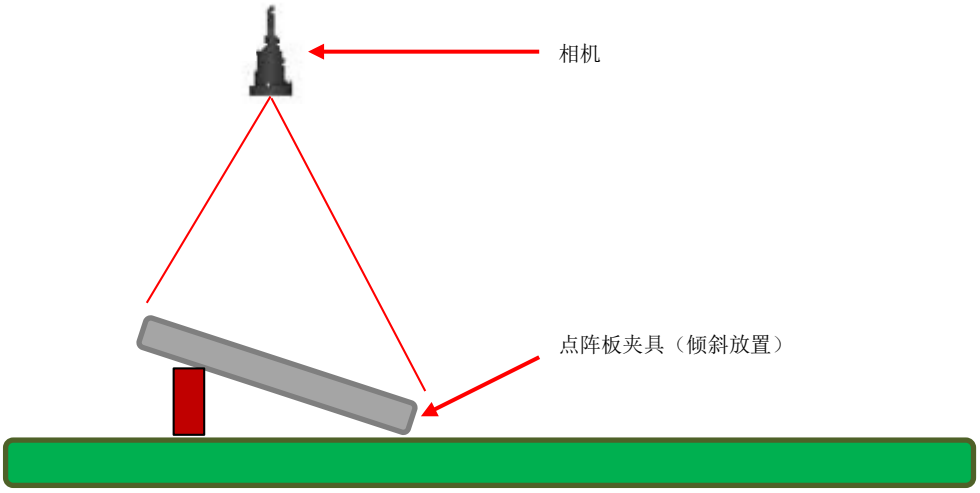
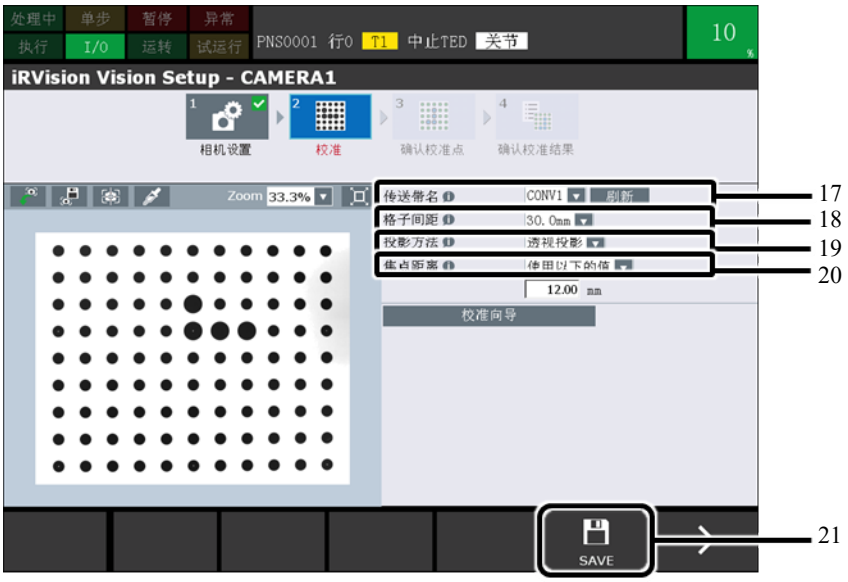
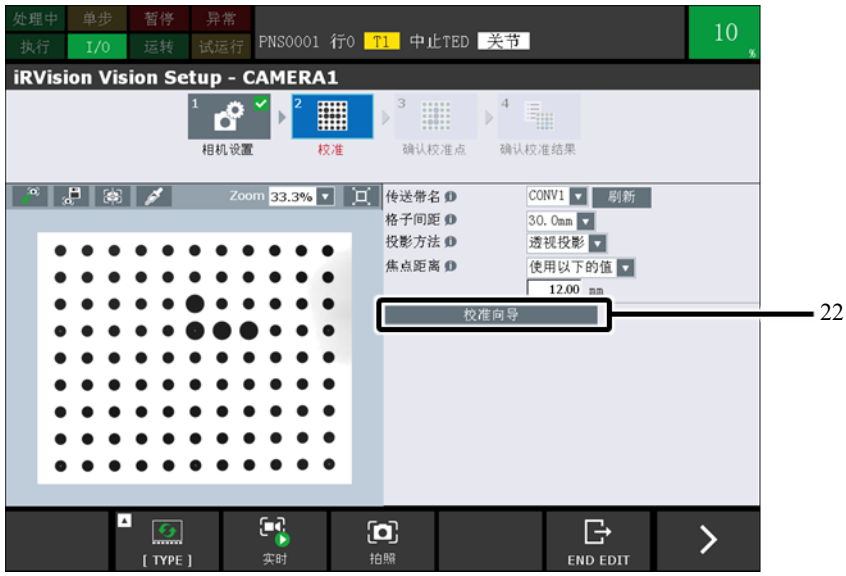


图 4.2.8.1（d）

21 按下［＞（下一页）］，再按下 F5［保存］。



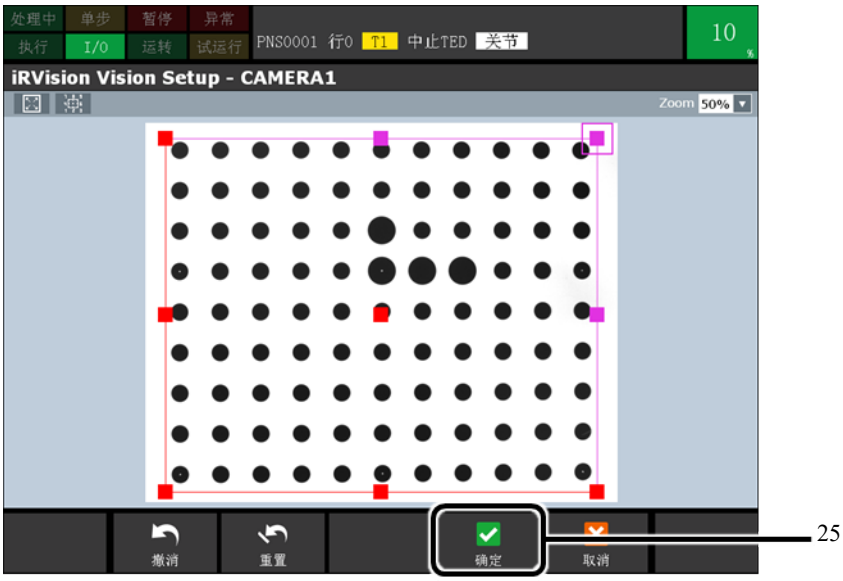
22　按下［校准向导］



23　确认在画面上已显示点阵板，按下 F4［确定］。

24　用紫红色的长方形窗口圈出画面显示的点阵。

25　按下 F4[确定]。



26　运行传送带，当点阵板在机器人到达动作范围内中央附近后关停。

27　触碰点阵板的原点。

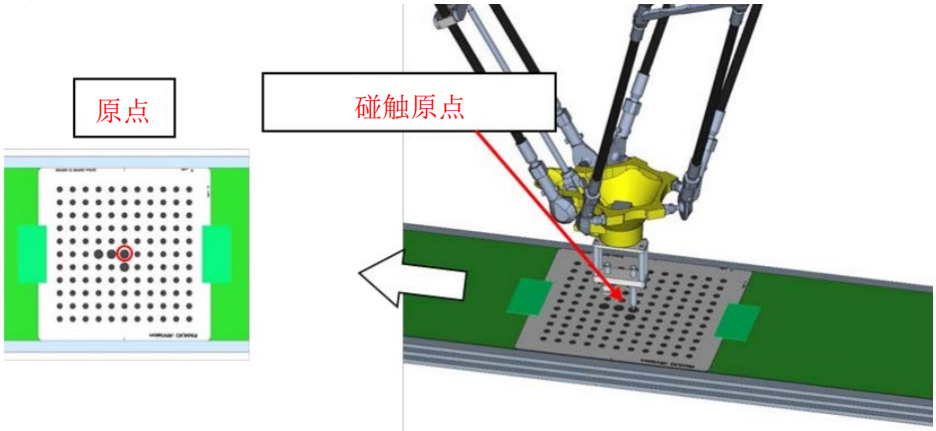


图 4.2.8.1（e）　触碰点阵板的原点

28　按下 F4[确定]。

29　在传送带停止的状态下触碰点阵板的 X 轴方向。

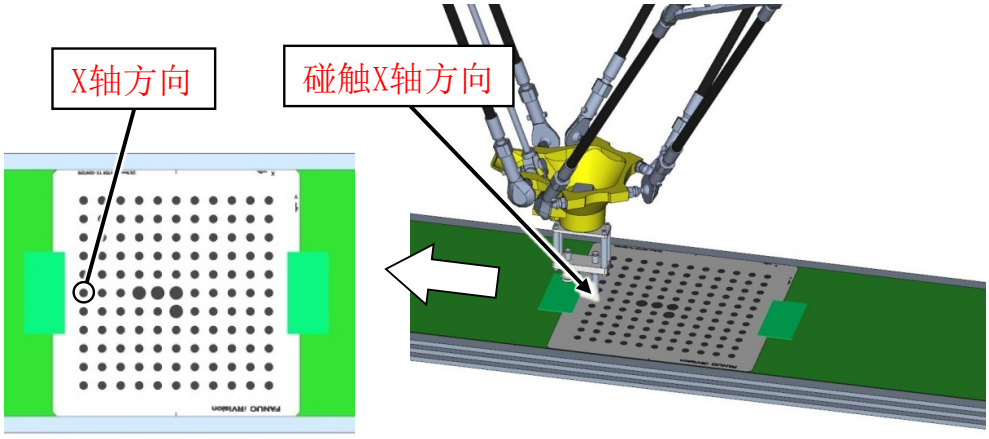


图 4.2.8.1（f） 触碰点阵板的 Y 轴方向

30 按下 F4[确定]。

31 在传送带停止的状态下触碰点阵板的 Y 轴方向。

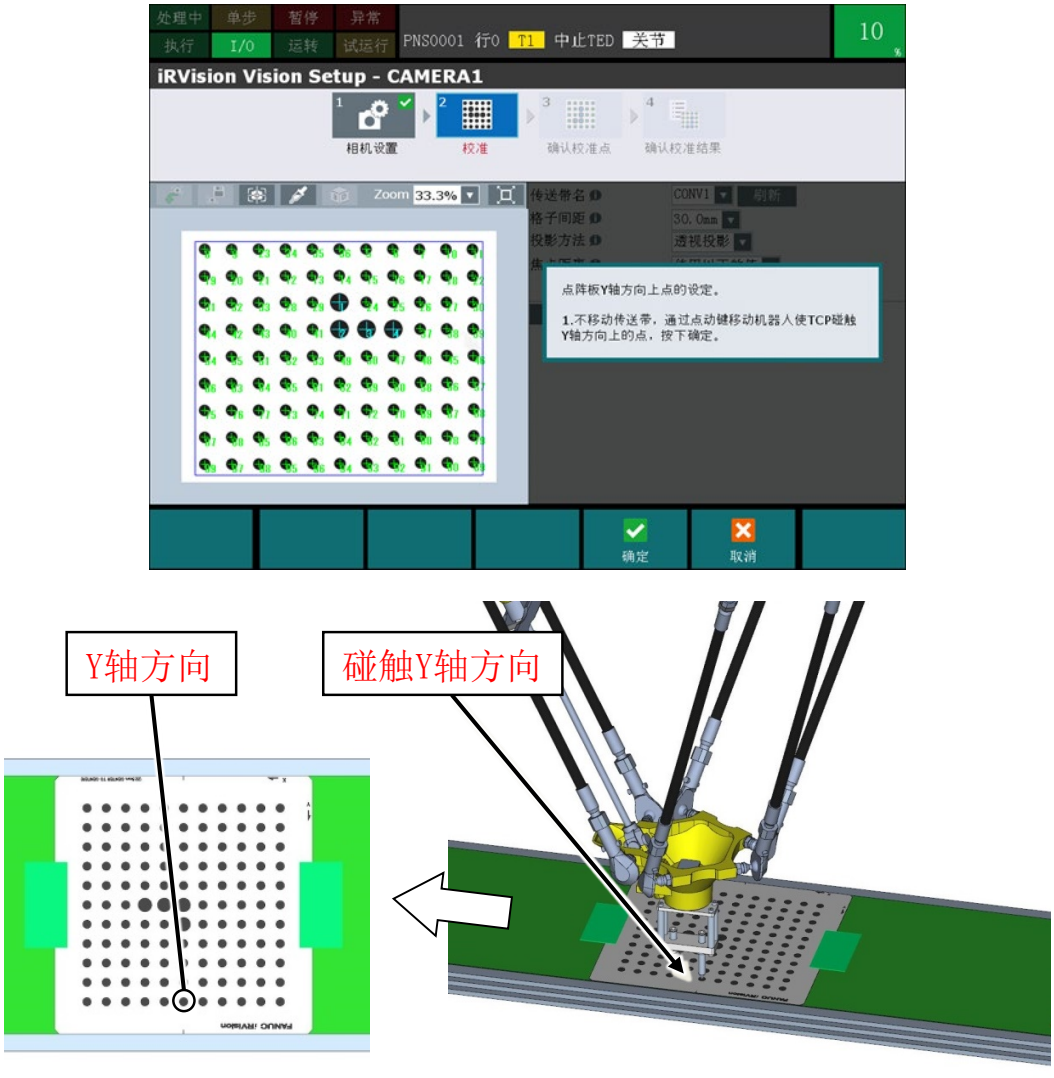


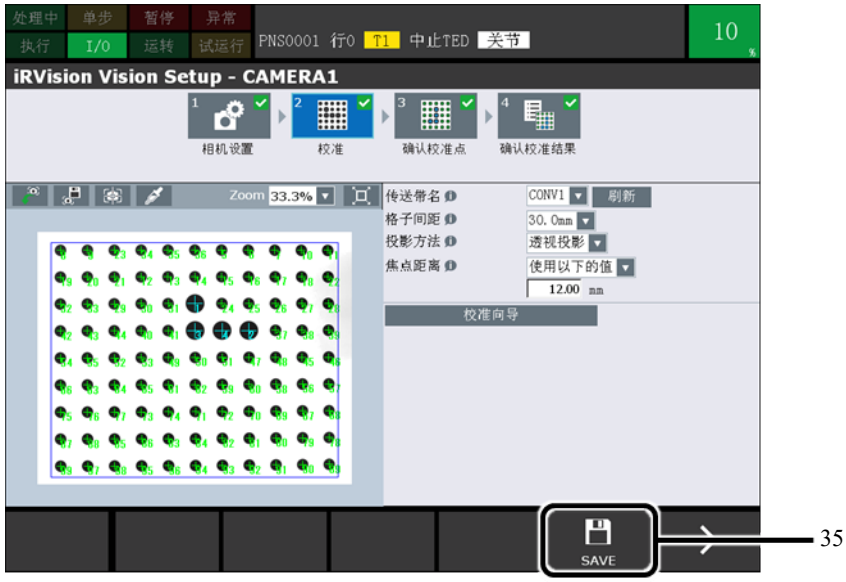
图 4.2.8.1（g） 触碰点阵板的 Y 轴方向

32 按下 F4[确定]。

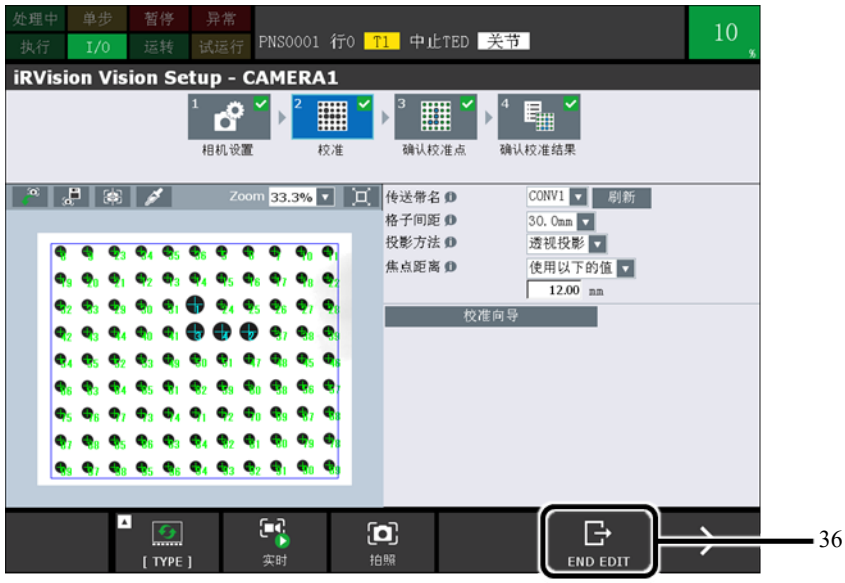
33 显示计算的基准距离。确认实际基准距离处的值妥当后按下 F4［确定］。如果值不妥当，请确认设定步骤及设定的参数是否正确。

34　按下 F4[确定]。

35　按下［＞（下一页）］，再按下 F5［保存］ 。



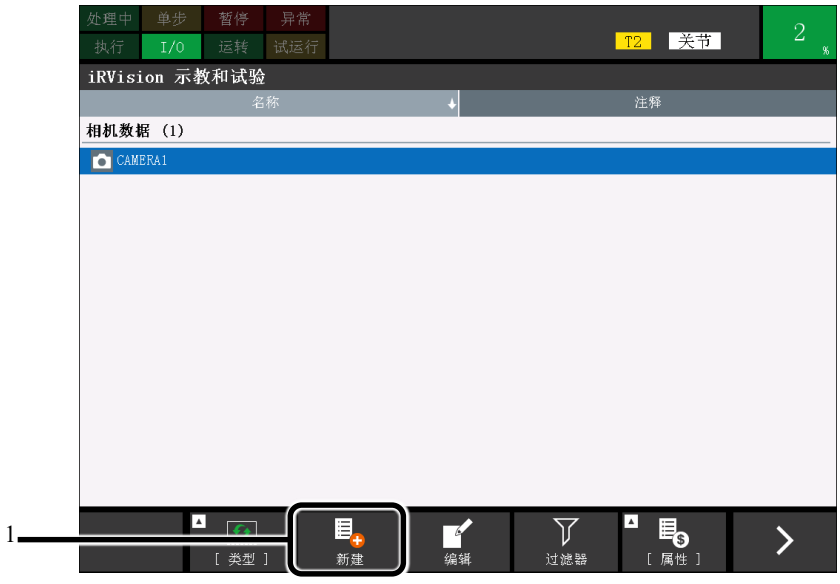
36　再次按下［＞（下一页）］，按下 F5［结束编辑］。



4.2.8.2　视觉程序

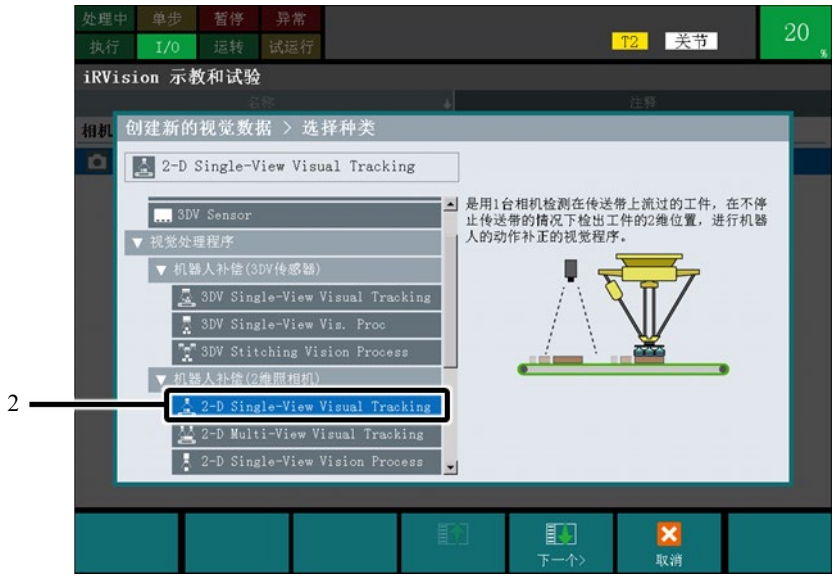
设定视觉程序。

1　按下 F2［新建］。



显示［创建新的视觉数据］的画面。

2 选择［2-D Single-View Visual Tracking］。

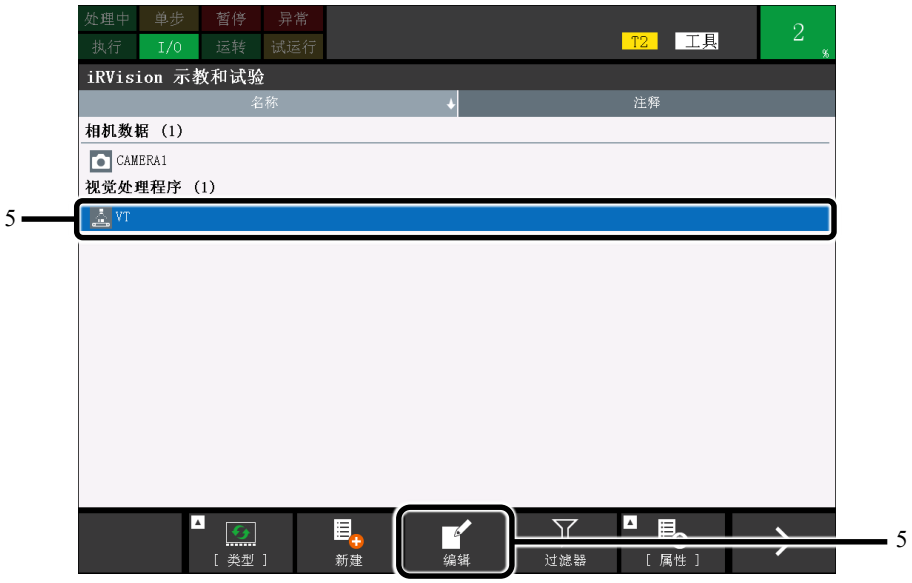


显示以下画面。

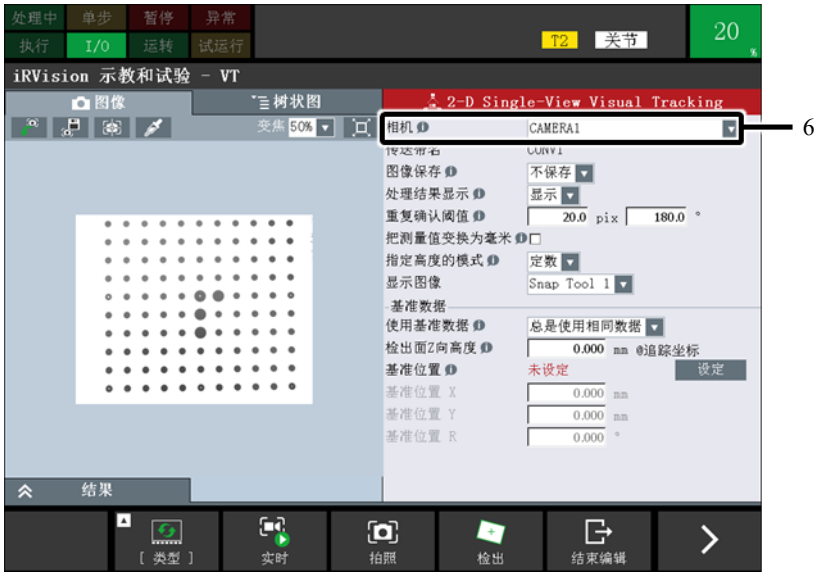
3 在［名称］中输入视觉程序名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例 ：VT

4 按下 F4［确定］。

5　选择创建的视觉程序，按下 F3［编辑］。



6　从菜单选择［相机］。



7 测量工件的厚度。
实测从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。

8 在［检出面 Z 向高度］中输入工件厚度。
输入从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。

注释

［检出面 Z 高度］从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。请用尺子等测量。系统与“检出面 Z 高度”的关系如下图。

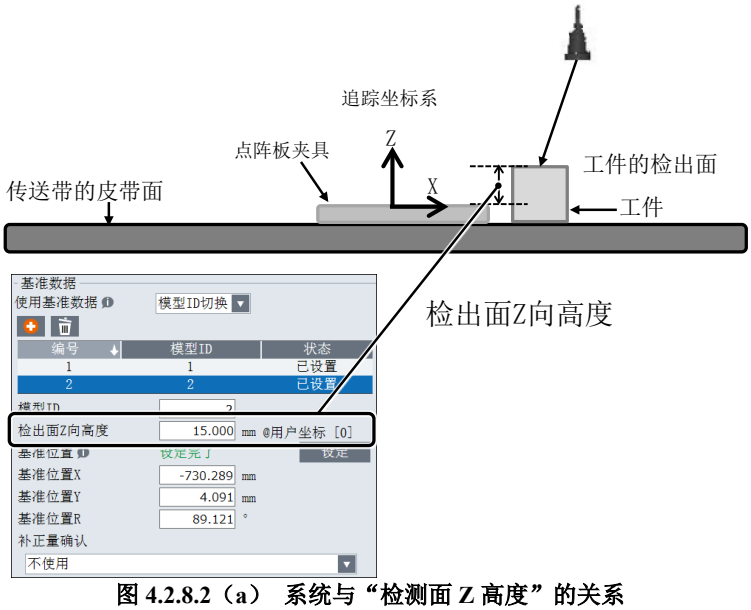


图 4.2.8.2（a） 系统与“检测面 Z 高度”的关系

9　按下树状图标签，选择［Snap Tool 1］。

10　输入［曝光时间］。

在设定曝光时间时，要使曝光中移动的工件图像偏差尽量小。

例如，考虑图像的偏差，传送带速度为 100 mm/s 时，为了将视觉检出时的位置偏差控制在 0.1 mm 以下，就要设定低于 1 ms 的值作为曝光时间。

0.1(mm) ÷ 100(mm/s) = 1(ms)

关于根据传送带速度计算合适的曝光时间，请参阅“2.5 适用探讨”中的“曝光时间”。

11 将工件安装在相机视野内。
如果已供给方向已确定，将调整方向再安装。
供给姿势为±180 度时，调整成放置侧的姿势，使示教变得容易。

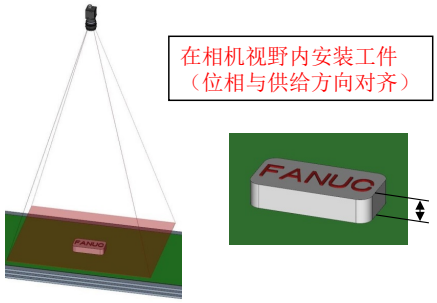


图 4.2.8.2（b）安装工件

12 按下树形图标签，选择［GPM Locator Tool 1］。

13 按下 F3［拍照］。
确认工件已显示在画面。

14 按下［模型示教］。

15 用紫红色的长方形窗口圈出工件。
16 按下 F4[确定]。

17 按下［遮蔽］中的［编辑］。
将画面向下滚动后显示。

18 选择画面上方的工具，将不要匹配的部分涂成红色。
19 按下 F4[确定]。

20 根据需要设定搜索范围。
关于详细设定请参阅“iRVision 操作说明书（参考篇）”中的“命令工具”章“图形匹配工具（GPM Locator Tool）”节。

21 按下［＞（下一页）］，再按下 F5［保存］。

22 再次按下［>（下一页）］，按下 F5［结束编辑］。

4.2.9 传送带工作站设定

设定传送带工作站。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。
显示工具坐标系一览画面。

5 按下 F5［切换］。
6 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

7 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

8　从菜单选择［设置］→［*i*RPickTool］。

9　首先设定供给传送带的传送带工作站。选择传送带工作站［CSTN1］。

10　在作业区上游点动移动机器人。

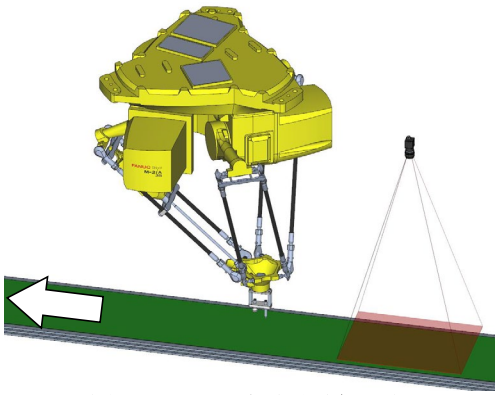


图 4.2.9（a）　点动移动机器人

11 点击［上游边界］中的［记录］。
追踪坐标系上的当前位置被记录在［上游边界］。

12 在作业区下游点动移动机器人。

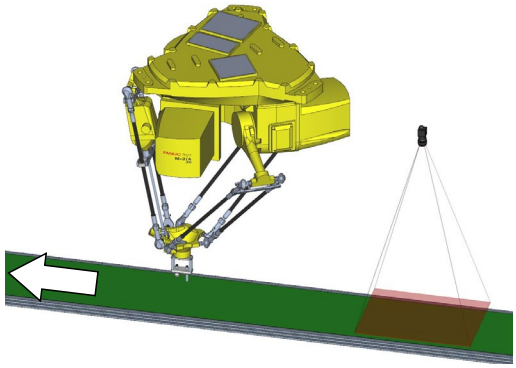


图 4.2.9（b） 点动移动机器人

13 点击［下游边界］中的［记录］。
追踪坐标系上的当前位置被记录在［下游边界］。

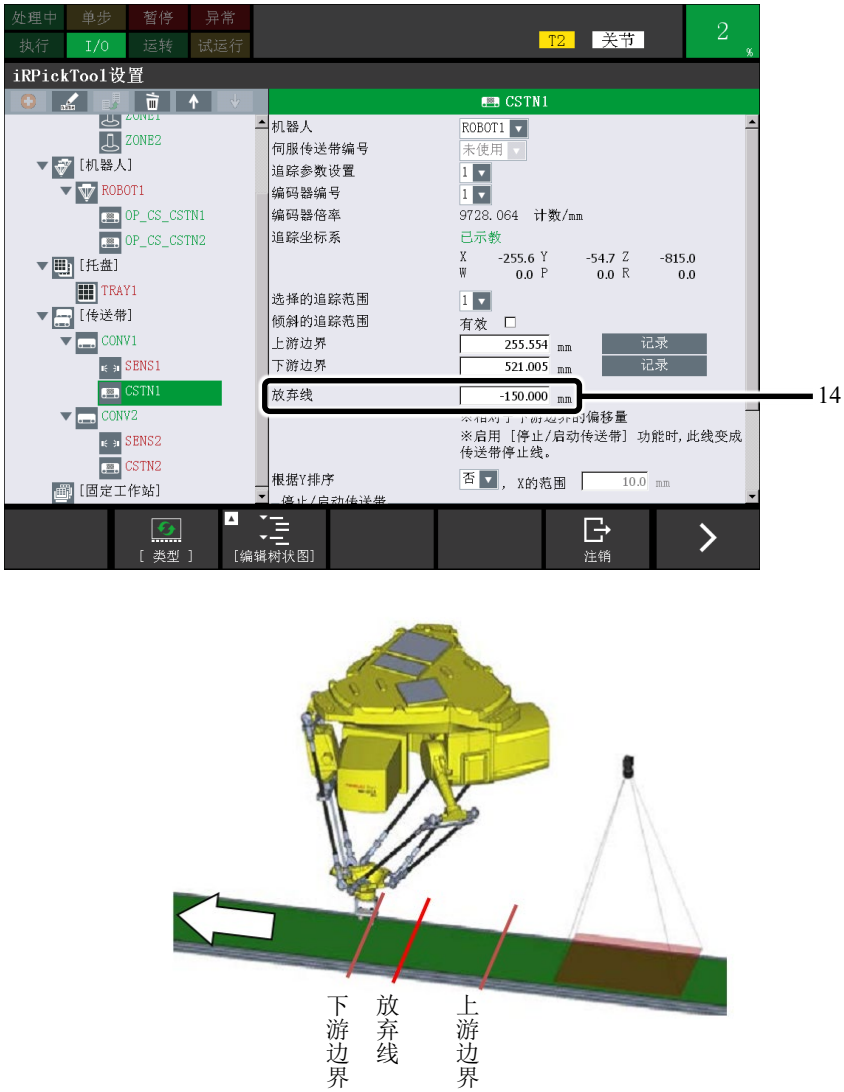
14 输入［放弃线］。
跳过距离值需要设成大于以下计算数值的值。

追踪所需时间※（sec）×传送带速度（mm/sec）
※ 追踪所需时间是指机器人从开始对某个工件进行追踪动作到结束对该工件的作业（取出等）的时间。

例如，机器人的动作时间约为 0.8 sec、传送带速度为 150 mm/sec 时，传送带自走距离为
0.8（sec） × 150（mm/sec）＝ 120（mm）。

因此，要留一些余量，将跳过距离设成－150 mm。

与放弃线有关的详细内容请参阅“6.6 传送带工作站设定”设定。



放弃线(mm) >
追踪所需时间(sec)× 传送带速度(mm/sec)

图 4.2.9（c）放弃线的设定

［上游边界］、［下游边界］可通过直接输入进行修正。

15 然后设定排出传 送带的传送带工作站。选择传送带工作站［CSTN2］。
16 通过与步骤 10 到 14 相同的步骤，设定排出传送带的传送带工作站［上游边界］、［下游边界］、［放弃线］。

4.2.10 操作的设定

设定操作。

传送带工作站设定以下项目。

项目	设定内容
操作	在传送带工作站选择机器人执行的作业为［拾取］或［放置］。
多数拾取和放置	启用［多数拾取和放置］后，只有添加到卡爪的区域数同时拾取/放置工件。如果不使其有效，则机器人个别拾取各个工件。
接近时偏移量	用工具坐标系的值输入以［拾取］或［放置］位置为基准的接近时偏移量。几乎所有的应用程序中，设定的偏移量只有 Z。

项目		设定内容
待机位置		启用待机位置时，如果机器人的拾取/放置的工件不存在，按之后说明的待机位置 X、Y 的设定，移动到计算出的待机位置，等待工件。 标准待机位置为加上拾取/放置位置与接近补偿量 Z 值的高度，是传送带工作站在追踪区域上游边界。 如果移动上游侧边界，待机位置也将随之自动移动。 如果不使用待机位置，当机器人的拾取/放置动作结束时，机器人在最近的周期的逃避位置待机。
	X	以毫米单位设定待机位置的偏移量。对应传送带工作站在追踪坐标系 X 方向的移动量。 如果输入正值，则沿着传送带的传送方向，待机位置向下游方向偏移。 如果输入负值，待机位置向上游移动。
	Y	以毫米单位输入待机位置的偏移量值。 对应传送带工作站在追踪坐标系 Y 方向的移动量。
工件有无检查		在对应的传送带工作站设定启用/禁用机器人拾取或/放置工件时的工件有无检查。
基准位置 PR		无法更改该值。
接近偏移量 PR		无法更改该值。
工件数据 VR		无法更改该值。

4

4.2.11 机械手工具坐标系的设定

设定机械手工具坐标系。

此处例子为 4 轴拳头机器人的设定。
在 4 轴拳头机器人中，即使机械接口坐标系中的机械手工具中心点(X, Y) 不是(0, 0)，也建议将工具坐标系的 X, Y 设成 0。原因是为防止从传送带抓取工件时，基本轴（J1～J3 轴）通过巨大运动，使循环时间延迟，或是超出机器人动作范围。
此外，为将工具的 Z 轴向上，直接输入工具的姿势。直接输入机械接口坐标系绕 X、Y、Z 轴的工具坐标系旋转角 W、P、R 的数值。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。
　显示菜单。
2　从菜单选择［设置］→［坐标系］。

3 按下 F3［坐标］。
4 从菜单选择［工具］。
显示工具坐标系一览画面。
5 将光标对准工具 1，按下 F2［详细］。
显示坐标系编号 1 的工具坐标系设定画面。

6 按下 F2［方法］。
7 从菜单选择［直接输入法］。

8　将光标对准［Z］及［W］、［P］、［R］，输入各自的值。
［Z］用输入规尺等测量的数值。
［W］、［P］、［R］直接输入。
※推荐（W、P、R）=（180、0、0）。

9　将按步骤 8 输入的［Z］及［W］、［P］、［R］做好备注。
10　在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
11　从菜单选择［设置］→［*i*RPickTool］。

12　选择区域的［ZONE1］。

13 在［区域工具坐标系］的［Z］及［W］、［P］、［R］中输入与按步骤 8 输入值（按步骤 9 备注的数值）相同的值。

14 使用多个拾取机械手时，与步骤 1 到 14 相同，设定工具坐标系 2 和［ZONE2］。

15 同样，只反复工具坐标系 3 与［ZONE3］、工具坐标系 4 与［ZONE4］、…使用的区域数。

16 选择机器人［ROBOT1］。

17 在［抓手］中选择［GRIPPER1］。

18 与步骤 1 到 4 相同，显示工具坐标系一览画面。
可以确认所有工具坐标系的设定值。
19 按下 F5［切换］。

20 输入坐标系编号 1。
启用工具坐标系 1。
※如不进行步骤 19、20，设定的工具坐标系 1 就不会有效。

4.2.12 基准位置的设定、机器人的位置示教

设定基准位置，示教机器人位置。

4.2.12.1 供给基准位置的设定、机器人的位置示教

设定供给传感带上的基准位置，示教机器人位置。［距离］＋［通过视觉检出］时的步骤。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［*i*RPickTool］。

3 选择传感器［SENS1］，在［视觉程序］中选择按“4.2.8.2 视觉程序”创建的视觉程序名称（VT）。
4 按下 F3［基准位置］。

显示如下［基准位置向导 步骤 1/2］。

⚠ 注意

在 Plug & Play 中无法使用[存储检出位置到 PR]。这是由于在［存储检出位置到 PR］中，假定接近时偏移量作为位置补正量使用，但是在标准程序中，将接近时偏移量作为了工具补偿量。

5 将工件放置在相机视野内的中央附近。

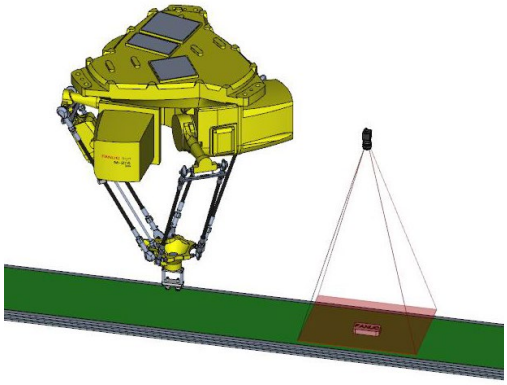


图 4.2.12.1（a） 将工件放置在相机视野内

6　按下 F4［检出］。
如果检出成功，显示［检出位置］。
※　检出不成功时
- 打开视觉程序，进行检出测试。
- 利用检出测试进行检出时，在传感器画面（步骤 3）确认视觉程序的选择是否正确。

检出成功后，F5［下一步］有效。

7　按下 F5[下一步]。

8　按下 F5［下一步］。

显示以下画面。
确认在［基准位置］中显示“向导成功”。

9　运行传送带，工件到达机器人动作范围内后关停。
　※停止传送带时，请注意传送带上工件不要偏离。
10　将机器人点动移动到工件拾取位置。

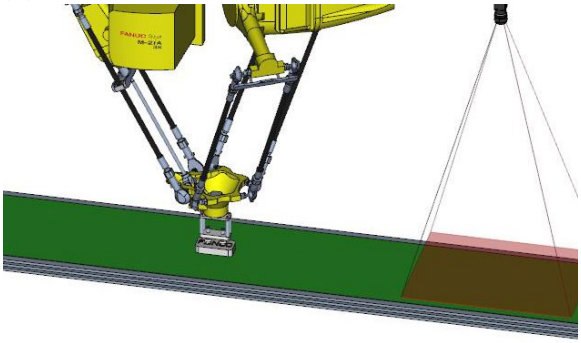
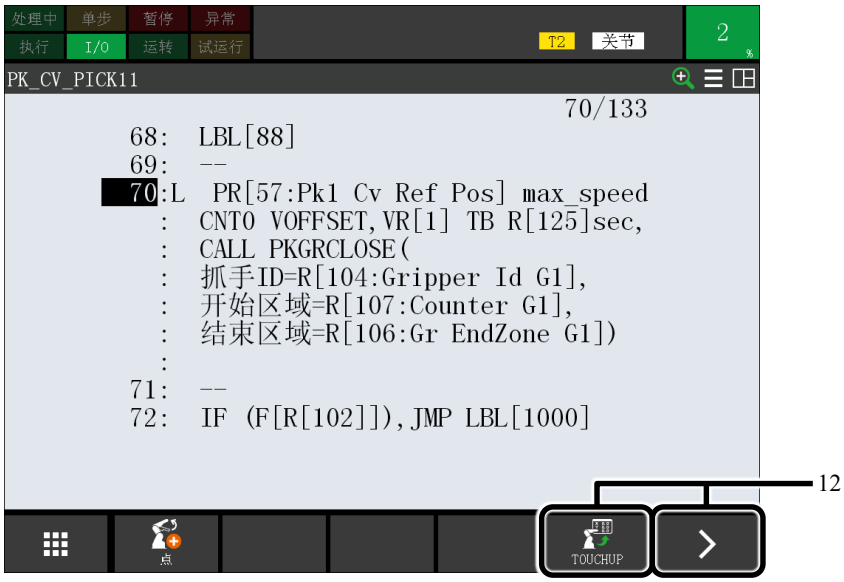


图 4.2.12.1（b） 将机器人移动到工件拾取位置

11 显示拾取程序 PK_CV_PICK11，将光标对准示教拾取位置的下一行。

70:L PR[57:Pk1 Cv Ref Pos] max speed CNT0 VOFFSET,VR[1] TB R[125]sec,CALL PKGRCLOSE(抓手[ID]=R[104:Gripper Id G1],开始区域=[107:Counter G1], 结束区域=R[106:Gr EndZone G1])

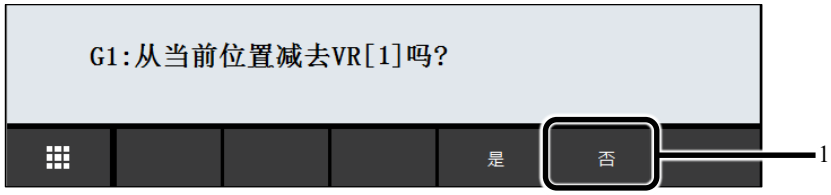
12 ［SHIFT］键的同时按下 F5［TOUCHUP］。
如果未显示 F5［TOUCHUP］、按下［>（下一页）］切换显示。



13 显示“VR 未初始化是否继续？”的消息后，按下 F4［是］，示教位置。

注释

1 通过重新设定在视觉寄存器中存在数值时，会显示“从当前位置减去 VR 吗？”，按下 F5[否]。



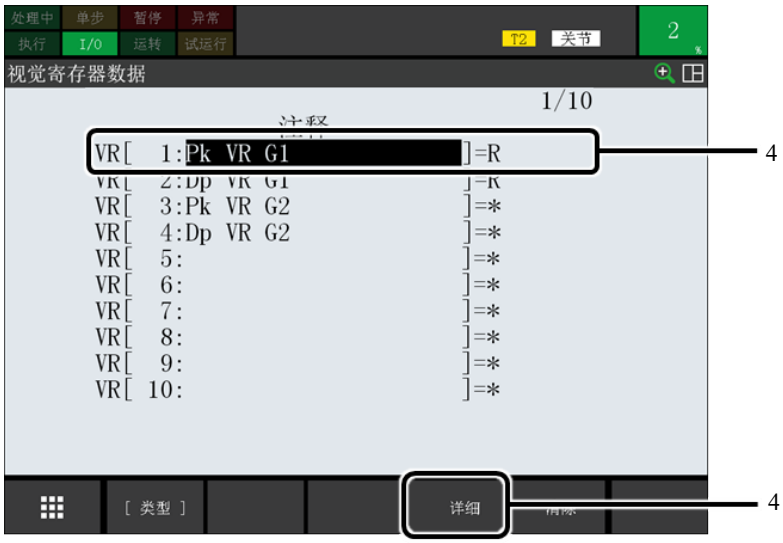
2 通过重新设定在视觉寄存器中存在数值时，为防止错误，按以下步骤清除视觉寄存器的值后再进行位置示教，对于“从当前位置减去 VR 吗？”，不论是选择 F4［是］还是 F5［否］，都会示教相同的位置。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

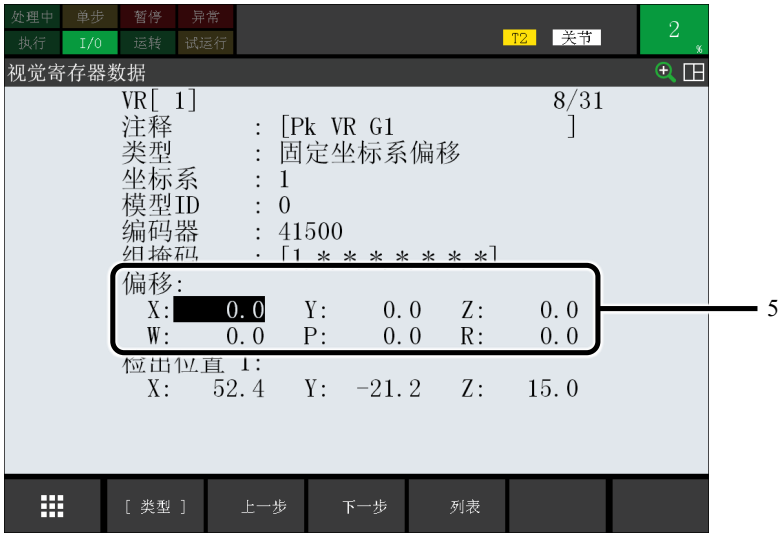
2 从菜单选择［下页］。

3 从菜单选择［数据］→［视觉寄存器］。

4　将光标对准［VR］中的“1”，按下 F4［详细］。



5　在［偏移］中所有的［X］、［Y］、［Z］、［W］、［P］、［R］值中输入“0”。



4.2.12.2 排出基准位置的设定、机器人的位置示教

设定排出传送带上的基准位置，示教机器人位置。即［DI］/［RI］/［HDI］＋［投入工件］时的步骤。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［*i*RPickTool］。

3 选择传感器［SENS2］，按下 F3［基准位置］。

显示以下画面。

4 将托盘放在触发传感器上游。

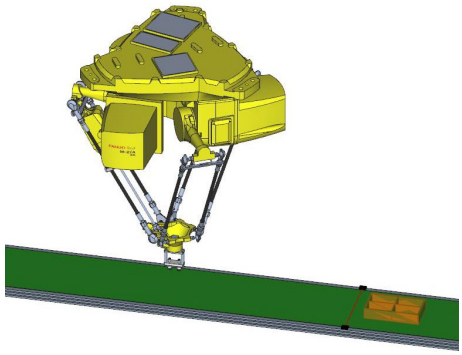


图 4.2.12.2（a）将托盘放在触发传感器上游

5 运行传送带，托盘到达机器人动作范围内关停。
如果检出成功，显示［触发数］。

6 按下 F5[下一步]。
检出成功后，F5［下一步］有效。

7　按下 F5［结束］。

8　显示以下消息。按下 F5［确定］。

显示以下画面。
确认在［基准位置］中显示“向导成功”。

9　运行传送带，托盘到达机器人动作范围内关停。
※请注意不要让托盘偏离。

10　将机器人点动移动到托盘单元 1 的放置位置。

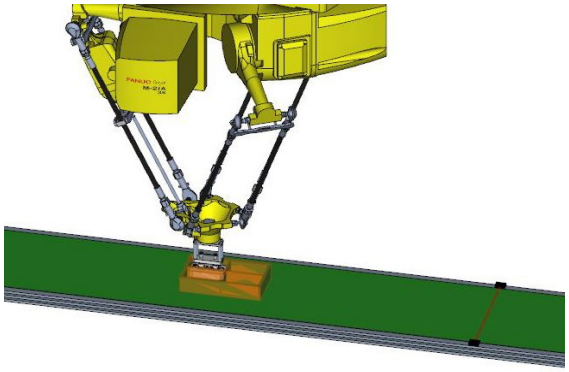
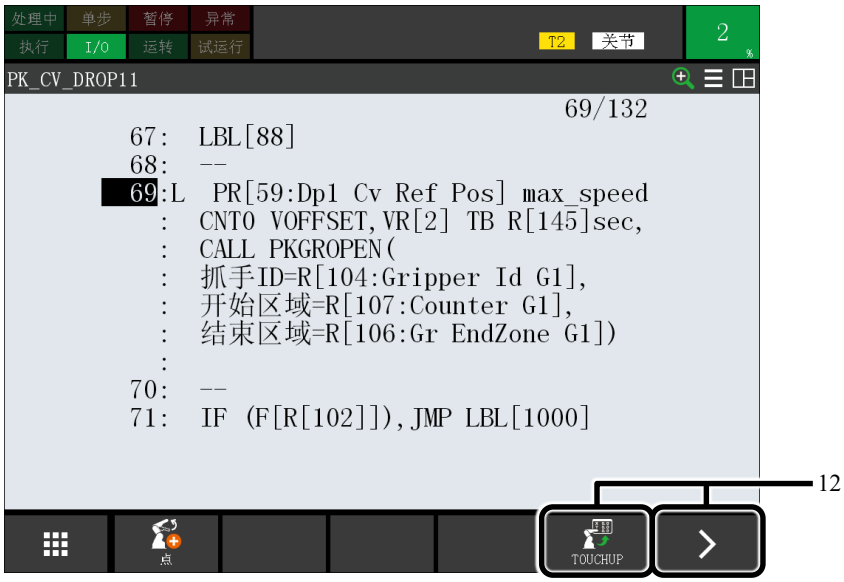


图 4.2.12.2（b） 将机器人移动到托盘单元 1 的放置位置

11 显示放置程序，将光标对准示教放置位置的下一行。

69:L PR[59:Dp1 Cv Ref Pos] max speed CNT0 VOFFSET,V[2] TB R[145]sec,CALL PKGROPEN(抓手ID=R[104:Gripper Id G1], 开始区域=R[107:Counter G1], 结束区域=R[106:Gr EndZone G1])

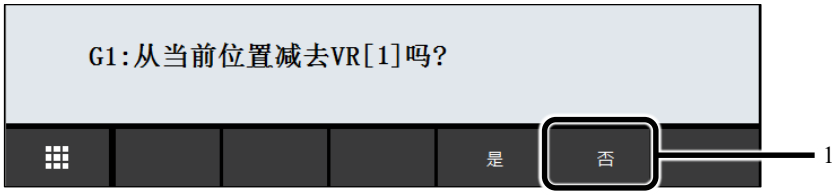
12 ［SHIFT］键的同时按下 F5［TOUCHUP］。
如果未显示 F5［TOUCHUP］、按下［>（下一页）］切换显示。



13 显示“VR 未初始化是否继续？”的消息后，按下 F4［是］，示教位置。

注释

1　通过重新设定在视觉寄存器中存在数值时，会显示“从当前位置减去 VR 吗？”，按下 F5[否]。



2　通过重新设定在视觉寄存器中存在数值时，为防止错误，按以下步骤清除视觉寄存器的值后再进行位置示教，对于“从当前位置减去 VR 吗？”，不论是选择 F4［是］还是 F5［否］，都会示教相同的位置。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。

2　从菜单选择［下页］。

3　从菜单选择［数据］→［视觉寄存器］。

4　将光标对准［VR］中的“2”，按下 F4［详细］。

5　在［偏移］中所有的［X］、［Y］、［Z］、［W］、［P］、［R］值中输入“0”。

4.2.12.3 机器人的原位置示教

示教机器人的原位置。

1 显示出主程序 PK_MAIN1。
2 将倍率设成小于 5%。
3 在在示教模式（T1,T2）下从头开始执行 PK_MAIN1。
在 PK_PERCH1 的第 15 行，由于位置［1］未示教，通过报警停止。

4 不更改选择的用户坐标系、工具坐标系，通过点动使机器人移到原位置。
5 ［SHIFT］键的同时按下 F5［TOUCHUP］。
当前位置示教成位置［1］。

4.2.13 追踪动作的微调整

在需要精度的系统中微调追踪动作。请参照“6.12 追踪动作的微调整”。

5 设定前的其他准备工作

本章针对构筑机器人网络，对使用 *i*RPickTool 的设定及注意事项等第 3、4 章启动调试步骤中未说明的其他准备工作进行说明。
此外，还对多台机器人控制装置构筑机器人网络，使用 *i*RPickTool 时的准备工作进行说明。

5.1 网络的构筑

如果系统中有多个机器人控制装置，请使用网络进行连接。

利用电脑安装 *i*RPickTool 时，即使系统内只有 1 台机器人控制装置，也需要设定网络。

5.1.1 网络电缆的连接

使用 *i*RPickTool 的追踪，在机器人控制装置之间进行众多的信息交换。如果发生通信延迟，会影响系统的性能。因此，机器人间的通信，形成仅连接机器人控制装置的本地的网络，不会受其他的通信的通信量影响。

R-30*i*B Plus / R-30*i*B Compact Plus / R-30*i*B Mini Plus 控制装置中，有 2 个以太网端口，1 台机器人控制装置可以隶属于 2 个不同的网络。需要连接工厂网络等外部网络时，如下图所示，工厂的网络使用端口 1 连接，机器人间的通信使用端口 2。这种情况，在端口 2 构筑的本地的网络上，不连接视觉追踪相关的机器人控制装置，以及 *i*RPickTool 的示教用电脑以外的装置。

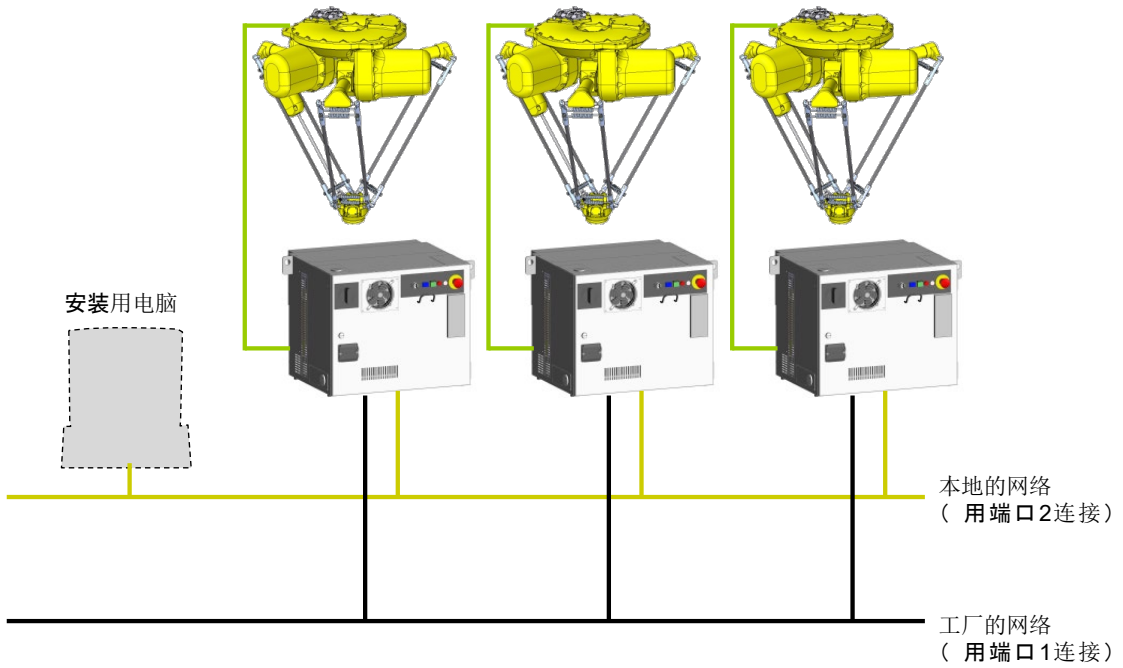


图 5.1.1 (a) 两个不同网络的连接例子

使用端口 1 的情况，以太网电缆连接主板 CD38A 端口，使用端口 2 时，连接 CD38B 端口。

注释
1 机器人控制装置不与外部网络连接时，可使用端口 1、端口 2 中的任意一个。
2 后面叙述的 *i*RPickTool 的示教用电脑，与机器人间通信的本地的网络连接。

5.1.2 构造网络时的注意事项

构造网络时的注意事项如下。请遵守相关事项，进行以下设定。

- 如同“5.1.1 网络电缆的连接”所说明的，有 2 个端口。设定在“5.1.3IP 地址的设定”中使用的端口的 IP 地址、子网掩码。
- 使用端口 1 和端口 2 时，需要对端口 1 和端口 2 设定不同的网络地址。详细信息，在“5.1.3IP 地址的设定”中进行说明。

以下是系统中有多个机器人控制装置时的注意事项。仅 1 台机器人的系统，无需阅读以下内容。

- IP 地址的设定中，需要对各机器人控制装置取不同的机器人名称。
- IP 地址的设定中，需要使机器人间的通信所使用的端口的 IP 地址为连号。例如，系统中有 4 台机器人控制装置，在使用端口 2 时，使各机器人控制装置中的端口 2 的 IP 地址分别为 172.16.0.1、172.16.0.2、172.16.0.3、172.16.0.4。
- 使用以太网编码器时，必须使连接脉冲编码器的机器人控制装置的 IP 地址在前头。刚才的例子中，使连接脉冲编码器的机器人控制装置的端口 2 的 IP 地址为 172.16.0.1。

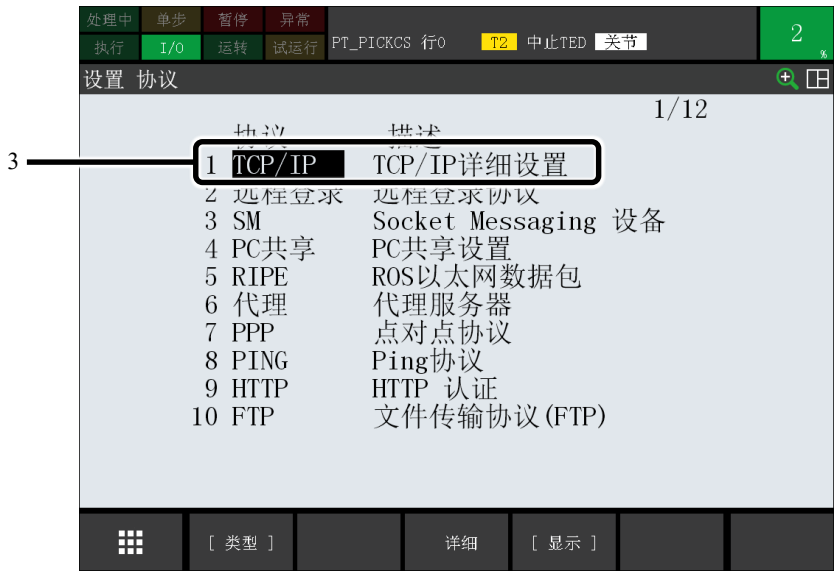
5.1.3 IP 地址的设定

通过以下步骤，对各机器人控制装置设定 IP 地址。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［主机通讯］。

显示［设置协议］画面。

3 将光标对准［TCP/IP］，按下［ENTER］键。



显示［设置主机通信］画面。

4 在［机器人名称］中输入该机器人控制装置的名称。

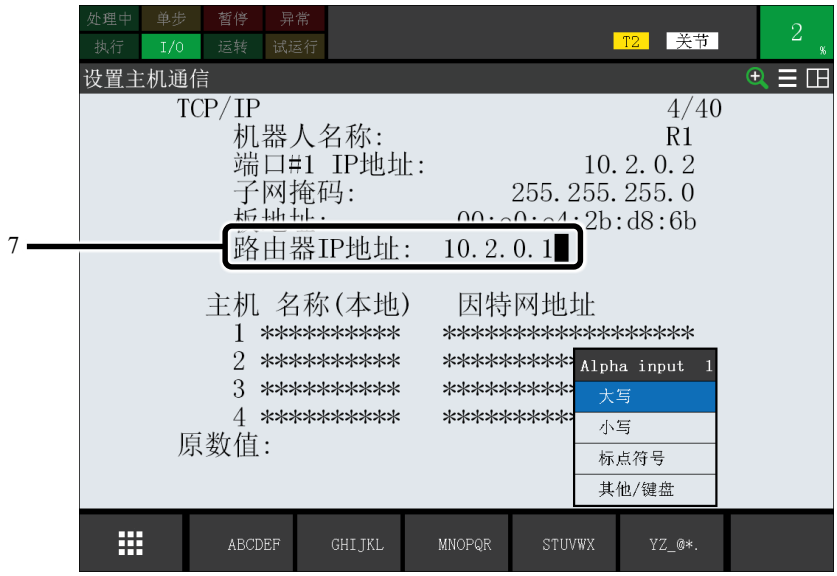
⚠ 注意
机器人名只能使用字母、数字、“—”号。打头文字只能使用字母。最后的字母不能使用“—”号。请不要插入多余的空格。

5 在［端口#1IP 地址］中输入机器人控制装置的 IP 地址。

⚠ 注意
请不要在 IP 地址中插入多余的空格、0。如果有多余的空格或 0，将无法正常通信。

6 在［子网掩码］中输入子网掩码。

7 在［路由器 IP 地址］中输入默认网关的 IP 地址。



8 按下 F3［端口］，选择端口。

9 以与步骤 5，6 相同的要领，设定端口 2 的 IP 地址，子网掩码。

10 再次接通机器人控制装置的电源。

> ⚠ **注意**
> **如果不再次投入机器人控制装置的电源，则无法正确进行以后的设定。请务必再次投入机器人控制装置的电源。**

端口 1 的设定（一例）

端口 2 的设定（一例）

⚠ 注意

使用端口 1 和端口 2 时，需要对端口 1 和端口 2 设定不同的网络地址。例如，子网掩码为 255.255.0.0，IP 地址为 172.16.0.1 时，网络地址为 172.16，主机地址为 0.1。对端口 1 和端口 2 设定网络地址相同的 IP 地址时，在投入电源时，会发生“HOST-179IP 地址配置错误”的报警，仅端口 1 有效，端口 2 无效。

5.1.4 机器人间的通信设定

如果系统中有多个机器人控制装置，则进行机器人间的通信设定。仅 1 台机器人的系统，不需要进行机器人间通信的设定。无需阅读此项，请跳过。

机器人控制装置间的通信中，内部使用名为“ROS 以太网接口包（RIPE）”的通信功能。
关于 RIPE 的详细内容，请参照“Ethernet Function OPERATOR'S MANUAL（控制装置以太网功能操作说明书）B-82974EN”中的“ROS INTERFACE PACKETS OVER Ethernet (RIPE) （ROS 以太网接口包（RIPE））”这一章。

主控设备和从控设备

机器人间通信，由 1 台主控制装置以及另一台从控制装置构成。*i*RPickTool 中，任何机器人控制装置都可以成为主控。但是，使用以太网编码器时，建议使连接脉冲编码器的控制装置成为主控制装置。

机器人间通信的设定，首先从从控制装置进行。用所有的从控制装置结束设定后，进行使用主控制装置的设定。结束用主控制装置的设定后，所有控制装置自动再起动，设定结束。详细步骤，在下面进行说明。

使用从控设备的设定

通过以下步骤，进行使用从控制装置的设定。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。

2 从菜单选择［设置］→［主机通讯］。

3 将光标对准［RIPE］，按下［ENTER］。

显示如下画面。

4 按下 F2［从控］。
如果从控设备画面已打开，F2 为执行，则不需要该步骤。

切换成如下 ROS 以太网数据包（从控设备）画面

5　对［主控设备 IP 地址］设定在机器人间通信中所使用的端口的主控制装置的 IP 地址。

6　设定［成员索引］。
使“5.1.3IP 地址的设定”中机器人间通信所使用的端口的 IP 地址作为主控制装置的前头，连号设定。输入从该 IP 地址的前头的顺序。
根据该设定，自动设定［从控设备 IP 地址］。
请确认显示的 IP 地址与“5.1.3IP 地址的设定”中该控制装置中设定的 IP 地址一致。

7　以上设定结束后，按下 F3［自动］。

显示如下画面。

8 会询问在接收数据后是否再起动，请按下 F3［是］。

9 使用全部的从控制装置进行以上步骤。如下面画面所示，使所有的从控设备处于等待接收来自主控设备的信息的状态。

使用主控设备的设定

用所有的从控制装置结束设定后，以下面的步骤进行使用主控制装置的设定。

1　在机器人控制装置的示教器上按下［MENU（菜单）］键。

2　从菜单选择［设置］→［主机通讯］。

3　将光标对准［RIPE］，按下［ENTER］。

切换成如下 ROS 以太网数据包（主控设备）画面

注释

在打开 ROS 以太网数据包（从控设备）画面的情形下，按下 F2［主控设备］，切换到 ROS 以太网数据包（主控设备）画面。

4 对［主控设备 IP 地址］设定在机器人间通信中所使用的端口的主控制装置的 IP 地址。该控制装置的 IP 地址。

5　在［成员数量］中，设定参加机器人间通信的控制装置的数量。

6　根据需要，设定［更新间隔］。是用于确认机器人是否在线的周期。

7　以上设定结束后，按下 F3［自动］。

显示如下画面。

8 显示“将所有从控设备置为自动模式”的信息。
通过之前说明的“使用从控设备的设定”步骤，所有的从控制装置处于等待接收来自主控制装置的信息的状态，按下 F4［继续］。

显示如下画面。

9 ［主机名称］中显示参加机器人间通信的所有控制装置的机器人名和 IP 地址。为了反映设定，会询问是否重新启动，请按 F4［是］。

所有控制装置重新启动，设定结束。

⚠ 注意
机器人间通信的设定结束后，请不要变更机器人名称、主机名称。需要使机器人名称和主机名称相同。如果进行变更，请从头开始重新进行机器人间通信的设定。

5.2 脉冲编码器的设置和设定

设置脉冲编码器，与机器人控制装置连接。
本节针对脉冲编码器的安装及连接到主板时，以及在多台机器人上使用时的设定进行说明。
关于脉冲编码器与机器人控制装置的连接及设定，请参阅“3 *i*RPickTool 基本（基本功能）启动调试步骤”及“4 *i*RPickTool（基本功能+ Plug & Play）的启动调试步骤”。

注释
本说明书中将检测传送带行程的装置称为“脉冲编码器”。在示教器及 *i*RPickTool 设置画面当中，“编码器”这一术语作为“脉冲编码器”的同义词使用。

*i*RPickTool 中，可使用αA1000S 脉冲编码器和量增脉冲编码器。
有关各个脉冲编码器在 R-30*i*B Plus / R-30*i*B Mate Plus / R-30*i*B Compact Plus / R-30*i*B Mini Plus 控制装置中使用时的构成，请参照下表。

表 5.2(a) 各机器人控制装置中的必要构成

脉冲编码器	R-30iB Plus 控制装置	R-30iB Mate Plus 控制装置	R-30iB Compact Plus / R-30iB Mini Plus 控制装置
αA1000S 脉冲编码器（A860-0372-T001）	“附录 D αA1000S 脉冲编码器”表 D.1(a)	“附录 D αA1000S 脉冲编码器”表 D.1(b)	“附录 D αA1000S 脉冲编码器”表 D.1(c)
量增脉冲编码器（A860-0301-T001～T004）	“附录 E 量增脉冲编码器”表 E.1(a)	“附录 E 量增脉冲编码器”表 E.1(b)	不能使用。

5.2.1 设置脉冲编码器

在传送带上设置脉冲编码器。

脉冲编码器

脉冲编码器设置于机器人动作范围外，使其在机器人动作中不发生干涉。此外，使用相机检出工件时，将脉冲编码器设置于不会进入相机视野的位置。

相比将脉冲编码器直接安装于传送带的驱动轴、从动轴，在脉冲编码器前端安装旋转盘，使旋转盘与传送带皮带直接接触进行旋转的方式，能更确实地测量传送带的移动量。

请注意旋转盘与传送带之间不能发生打滑（传送带动作，但是旋转盘不旋转的状态）。如果发生打滑，将无法正确计算传送带的移动量，机器人的搬运精度下降。使用弹簧，将脉冲编码器按在传送带上固定，较为有效。

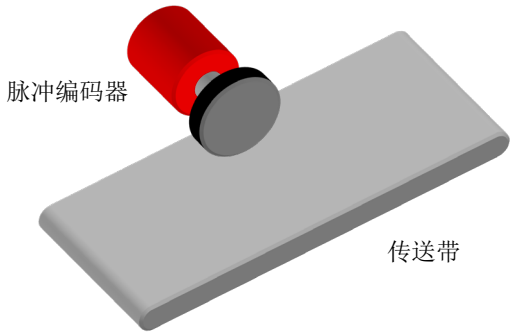


图 5.2.1(a) 脉冲编码器的安装位置

5.2.2 连接脉冲编码器

将脉冲编码器连接到机器人控制装置。

> **⚠ 注意**
> **以与电源电缆并行的方式，一起对连接脉冲编码器的电缆进行布线时，将会受到电源电缆引起的噪声的影响，有时会无法正确读取脉冲编码器的信号。请注意，要使得电源电缆与其它电缆分开布线。**

关于脉冲编码器的连接方法，请参阅“附录 E αA1000S 脉冲编码器”或“附录 F 量增脉冲编码器”。

5.2.3 设定连接到主板的脉冲编码器

本项针对将αA1000S 脉冲编码器连接到主板使用时的设定进行说明。

设定以下项目。

- ［编码器类型］
- ［平均（更新次数）］

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
显示菜单。
2 从菜单选择［设置］→［编码器］。
3 按下 F3［编码器］，输入［编码器编号］。未将其他脉冲编码器连接到接口端口时选择编码器编号 1，将其他脉冲编码器连接到接口端口时，选择编码器编号。将两个αA1000S 脉冲编码器连接到主 CPU 板时（仅 R-30iB Compact Plus / R-30iB Mini Plus），如果连接到 PULSECODER1 选择 1，连接到 PULSECODER2 选择 2。
4 将光标对准［编码器类型］，按下 F4［选择］。
5 选择［主串行增量式］。
6 将光标对准［平均（更新次数）］，输入瞬时速度的累积个数（移动平均宽度）。
通过设定瞬时速度的累积个数，即使传送带在机器人追踪中发生突然停止，机器人也不会紧急停止，而是平稳停下。
通常输入 10。

> **注释**
> **连接至主板时，［编码器轴］保持初始值 0。**

7 冷启动机器人控制装置。

5.2.4 在多台机器人上设定脉冲编码器

本项针对利用多台机器人使用以太网编码器时的设定进行说明。
1 台机器人时或是在多台机器人使用脉冲分配器时，请参阅“3.1.1.2 脉冲编码器的设定”。

多台机器人使用以太网编码器时的设定如下。

使用主控设备的设定

使用连接脉冲编码器的机器人（主控设备）的设定方法。

1 设定步骤与“3.1.1.2 脉冲编码器的设定”或者“4.1.1.2 脉冲编码器的设定”步骤 1～7 相同。
编码器的设置画面如下所示。
机器人控制装置中，如果有以太网编码器选项，则追加［以太网主 RIPE 编号］和［以太网主编码器编号］的项目。

2 将光标对准［以太网主 RIPE 编号］，输入该控制装置的 RIPE 编号。如果是机器人间通信的主控制装置，则 RIPE 编号是 1。如果是从控控制装置，则是［成员索引］的值。［成员索引］的确认方法请参照“5.1.4 机器人间的通信设定”中的“使用从控设备的设定”。

3 将光标对准［以太网主编码器编号］，输入选择的编码器编号。
输入的编号就是设定的编码器编号。
使用追踪拾取用接口板（迷你宽槽），脉冲编码器与追踪拾取用接口板（迷你宽槽）的 JF21 的连接器连接时是 1，与 JF22 连接时是 2。使用追踪拾取用接口板（迷你槽），使用 1 个脉冲编码器时是 1，使用 2 个脉冲编码器时，为对应各自电缆标签的编号。

4 再次接通机器人的电源。

> **⚠ 注意**
> **主板上所连接的脉冲编码器上使用以太网编码器功能时，需要将主编码器的编码器编号设定为比从编码器的编码器编号小的数字。**

使用从控设备的设定

1 打开编码器的设置画面。可以选择任意的［编码器编号］。
2 在［以太网主 RIPE 编号］和［以太网主编码器编号］中，输入与在主控设备中设定的值相同的值。
3 除此以外的值不更改。
4 再次接通机器人控制装置的电源。

> **⚠ 注意**
> **使用以太网编码器时，以机器人间通信连接的控制装置的软件版次必须完全相同。**

5.3 相机的设置和连接

设置相机，与机器人控制装置连接。
如果是带队列管理的追踪，请跳过此节。

选择使用的相机

视觉追踪中，可使用数码相机（白黑、彩色）、模拟相机。

连接相机

将相机与机器人控制装置连接。详细信息请参照“传感器机构部/控制部操作说明书 B-83984CM”的“2. 结成”章。

> **⚠ 注意**
> **使用以太网编码器时，为了取得检出工件瞬间的正确编码器计数，在机器人间通信的主控制装置上连接相机。**

设置相机

在相机上安装镜头，设置于传送带上方。请使相机光轴与传送带面垂直。

此外，设置相机时，请使传送带的传送方向为相机图像尺寸较大侧。通常为横向。这样，可确保传送带传送方向的视野宽敞。

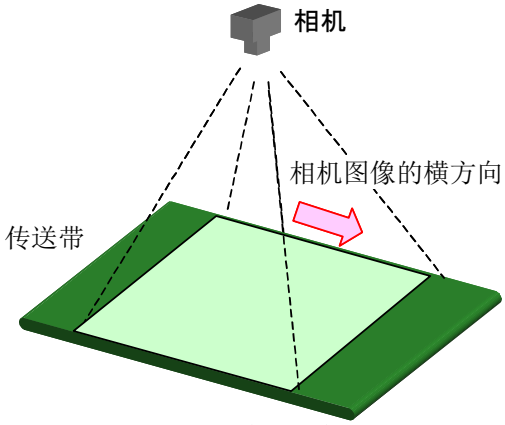


图 5.3 (a) 相机的安装例子

设置相机和光电管传感器

在输入光电管传感器的信号后，使用相机进行拍摄的情况下，传感器检测 DI 信号，在输入 DI 信号后，等待传送带移动指定距离后，进行相机的拍摄。
从该 DI 信号的输入，至视觉拍摄的时间，根据使用的硬件构成而变化，一般情况为约 0.1 秒左右。因此，拍摄时机可能在期待的瞬间之后。
于是，在距离传送带移动速度 x 0.1 秒长度的下游侧位置设置相机。
也可以在更下游侧设置相机。
这种情况，可通过调整传感器的设定参数［触发偏移］，在希望的位置进行拍摄。关于［触发偏移］，请参照“6.5 传感器的设定”。

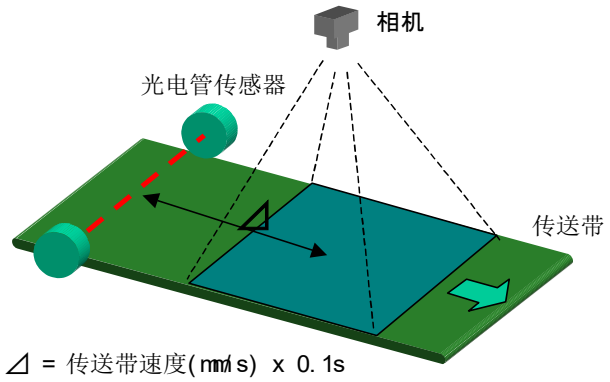


图 5.3 (b) 相机与光电管传感器的安装例子

5.4 工具坐标系的设定

工具坐标系的设定为追踪坐标系的设定，或者是利用相机校准中使用的碰触销定义工具中心点（TCP）位置与工具姿势。
如果有多台机器人，需要所有的机器人都进行该作业。

在机器人的指尖部固定触针，在销子前端设定工具坐标系。
将 1 个工具坐标系数据分配给触针，进行设定，使工具坐标系的原点与销子前端一致。
具体方法有：3 点示教法/6 点示教法/直接示教法/2 点＋Z 值示教法（4 轴机器人的情形）等。
关于详细操作，请参照“操作说明书（基本操作篇）B-83284CM”中的“机器人系统的设定”这一章的“设置坐标系”节。

触针在追踪坐标系的设定、相机的校准、传感器位置/托盘位置的设定等中使用。需要的操作结束后，可卸下触针。
但是一旦卸下触针后，如果无法确保其机构上正确地重新安装的再现性，则设定的 TCP 无效，需要重新进行 TCP 设定。
因此，考虑到今后万一发生时有触针的作业的情况，需要对触针在设计上确保其安装的再现性。
此外，建议依照“6.11 机器人的位置示教”（或“7.7 机器人的位置示教”）进行机器人的程序的位置示教之前，一直保持安装着触针的状态。

5.5 点阵板夹具

视觉追踪的相机校准中，使用点阵板夹具。有关点阵板制作的注意点等，在“*i*RVision 操作说明书（参考篇）”的点阵板夹具章有记载，请参考。
此外，本公司还提供标准的点阵板夹具，敬请利用。

6 基本功能参考

本章针对您在使用 *i*RPickTool Basic（基本功能）或 *i*RPickTool（基本功能+ Plug & Play）时参考的菜单项目详情及设定方法进行说明。
此外，还针对机器人程序及 KAREL 程序等具体程序设定例子进行说明。

关于 *i*RPickTool Basic（基本功能）的启动调试步骤，请参阅“3 *i*RPickTool Basic（基本功能）启动调试步骤”。关于 *i*RPickTool（基本功能 + Plug & Play）的启动调试步骤，请参阅“4 *i*RPickTool（基本功能+Plug & Play）的启动调试步骤”。

*i*RPickTool Basic 的菜单与本说明书相关项目如下图所示。

图 6(a) 与 *i*RPickTool Basic 菜单相关的项目

除了图 6(a)中的 *i*RPickTool Basic 的菜单项目，作为基本功能的参考，还在“6.8 KAREL 程序”、“6.9 机器人的程序”根据具体例子对程序进行了说明，而在“6.10 基准位置设定”、“6.11 机器人的位置示教”中根据具体例子对机器人位置的示教进行了说明。

此外，在“6.12 追踪动作的微调整”对完成示教基本动作后，进一步优化机器人的方法进行说明。

在“6.13 启动多台机器人时的注意事项”中对利用多台机器人开始生产时的注意事项进行说明。
“6.14 预整列设定”中，说明预整列功能中特化的内容。

基本操作

*i*RPickTool 对机器人、传送带等系统的各个构成要素进行设定。该构成要素称为“对象”。
这里对进行各个对象的设定而准备的对象的生成方法等进行说明。
而 *i*RPickTool 的设定可在工作单元内的任意控制装置上进行。
但是，通过机器人间通信所连接的机器人不可以同时用 2 台以上的机器人控制装置设定 *i*RPickTool。

创建新的对象

在机器人控制装置的示教器上按压［MENU］键，从菜单选择［设置］→［*i*RPickTool］后，就会显示如下 *i*RPickTool 设定画面。

在 *i*RPickTool 的设定画面中，在左侧树状图上选择要创建的对象修正。
例如，新建机器人时，按以下步骤进行。

1　选择［机器人］。
2　按下树状图上的新建 按钮，或是选择 F2［编辑树状图］中的［新建］。

显示对象的新建画面。

3 在新建对象的画面中输入不与其他对象重复的名称。也可以直接使用初始显示中输入的名称。
输入名称时，请注意以下事项。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。

4 按下 F4［确定］，创建对象。
如果取消新建时，按下 F5［取消］。

新建对象后，创建的对象就会添加到树状图，并显示该设定画面。
新建机器人时，显示如下设定画面。

在树状图上显示的对象颜色分别为：红色表示未示教状态，绿色表示已示教状态。

新建传感器、传送带工作站

如果新建传送带，将分别自动生成如下图所示的一个传感器和一个传送带工作站。

如果多台机器人在传送带上作业，添加传送带工作站（在树状图上显示为［CSTN1］）。
此外，如果传送带上有多个传感器，则添加传感器（在树状图上显示为［SENS1］。）。
新建对象时，按以下步骤进行。

1　在传送带选中的状态下按下树状图上的新建 按钮，或是选择 F2［编辑树状图］中的［新建］。
显示如下画面。

2　［类型］中追加的对象，指定传送带工作站或传感器。
与新建其他的对象时相同，［名称］中显示默认的名称，因此与其他的对象一样新建对象。

进行各对象的设定

在 *i*RPickTool 设置画面左侧的树状图上，选择需设定的对象。然后在右侧的设置画面进行各项目的设定。
在树状图上显示的对象颜色分别为：红色表示未示教状态，绿色表示已示教状态。
各对象的详细请参阅“6.1 工作单元设定”到“6.7 固定工作站设定”，Plug & Play 时，请进一步参阅“7.1 工作单元设定”到“7.4 固定工作站设定”。

删除对象

如果删除对象，按以下步骤进行。

1 在 *i*RPickTool 设定画面的左侧树状图上选择要删除的对象。
2 按下树状图上的删除 按钮，或是选择 F2［编辑树状图］中的［删除］。

显示如下画面。

3 按下 F4［确定］删除对象。
要取消删除时，按下 F5［取消］。

更改对象的名称

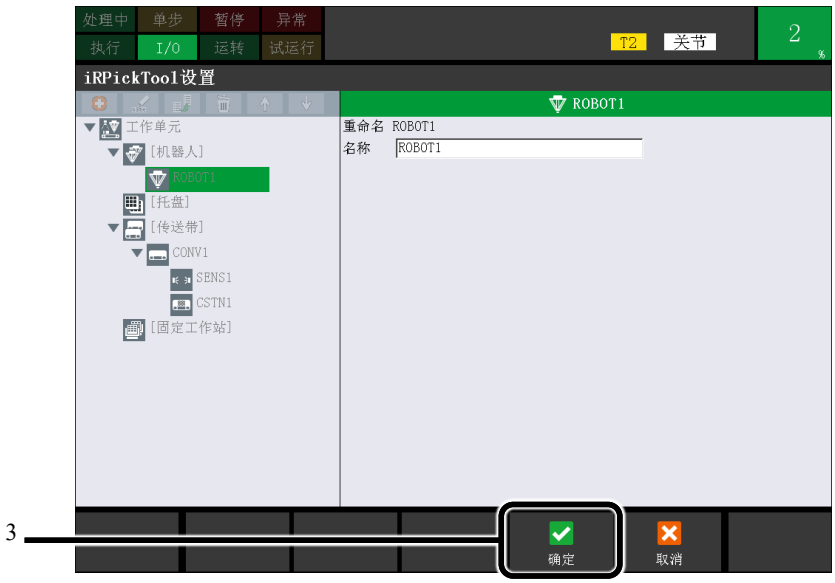
更改对象名称时，按以下步骤进行。

1 在 *i*RPickTool 设定画面的左侧树状图上选择要更改名称的对象。

2 按下树状图上的更改名称 按钮，或是选择 F2［编辑树状图］中的［更改名称］。

显示如下画面。

3 输入对象的新名称，按下 F4［确定］。
要取消更改名称时，按下 F5［取消］。



更换对象顺序

替换传送带工作站时，替换对象传感带工作站的顺序。

> **注意**
> **传送带工作站的配置，在画面树状图上表示的顺序，与从传送带上游至下游的顺序相同。**

在 *i*RPickTool 设置画面左侧的树状图上，选择需变更顺序的对象。
要向上移动，请按下 F2［编辑树状图］的［上移］，或树状图上的 按钮。
要向下移动，请按下 F2［编辑树状图］的［下移］，或树状图上的 按钮。

同步数据

*i*RPickTool 的设定，使用存储在机器人控制装置的系统变量中的数据，使系统内所有机器人控制装置都被设定为相同内容的数据。如果进行了机器人间通信的设定，通过本章介绍的 *i*RPickTool 设定画面进行设定后，自动向各机器人传送并保存。

而对于系统内未切入电源的机器人控制装置，也可以进行 *i*RPickTool 的设定。
但是，在这种情况下，仅离线的机器人控制装置处于无数据一贯性的状态。
这种情况，开启系统内所有的机器人控制装置的电源，用以下方法进行数据同步，使所有控制装置的 *i*RPickTool 的设定相同。

1 如果在无数据一贯性的状态下打开 *i*RPickTool 的设置画面，将显示以下画面。
2 按下 F4［确定］。

于是如下图所示，进入无法更改设定项目的状态。

3 这里，确认各对象的设定，确认其设定是否也可传送至其他控制装置。
根据其他控制装置的设定进行同步时，按下 F5［注销］关闭 *i*RPickTool 的设置画面，使用其他控制装置打开 *i*RPickTool 的设置画面，进行同样的确认。

> ⚠ **注意**
> **如果通过以下步骤，根据非本意的设定进行同步，可能导致设定无法恢复原状。请务必全面确认该控制装置的设定是否真的可传送至其他控制装置。**

4 确认设定可传送至其他控制装置后，在 *i*RPickTool 的任意设置画面按下［＞（下一页）］按钮，显示以下的功能键。

5 按下 F1［同步数据］。
显示如下画面。
6 按下 F4［确定］。

7 同步数据结束后，显示如下画面。
8 按下 F4［确定］。

更新树状图

*i*RPickTool 的设置画面，不自动反映现在的状况，因此需通过［更新树状图］明确将现在状况反映于设置画面。具体在以下情况下使用。

- 在打开 *i*RPickTool 的设置画面的状态下，离线状态的控制装置成为在线状态时，其状态变化不自动反映于设置画面。

这种情况下，在 *i*RPickTool 的任意设置画面按下［＞（下一页）］按钮，显示以下的功能键，按下 F2［更新树状图］。

6.1 工作单元设置

工作单元的设定画面如下。

各项目的内容如下。

[动态负载平衡]

如果其有效，在负载平衡有效的传送带上，某机器人成为离线时，该机器人应负责的工件成为其他机器人负责的工件。如果机器人成为在线，则对象工件再次成为该机器人的工件。

[方案管理]

指定保存方案的机器人。所谓方案，是用于总结 *i*RPickTool 的工作单元整体，通过使用不同方案，可更换工作单元整体。方案的详细信息，请参照“9.1 使用方案变更工作单元”。所有的方案只放置在已被作为方案管理所指定的机器人控制装置中。

> **注**
> **方案管理在默认情况下无法变更。**

6.1.1 对象数量设定

在工作单元的设定画面可更改各个可设定对象的最大数量。
最大数量的更改步骤如下。

1 按下 F4［设置对象］。
显示如下画面。

2 在要添加对象的［最新最大数量］中输入添加后的新对象的最大数量。

3 如按下 F4［确定］，则显示以下信息。

4 如按下 F4［确定］，自动重新启动设定 RIPE 的全部控制装置。
如果启动后重新打开对象设定，如下图所示，对象的当前最大数量已成为设定的数量。

注
1 无论此处设定的传感器的最大数如何，1 台机器人最多可同时执行 4 台传感器。
2 有关各对象可变更的最大数量，请参照表 6.1.1(a)。

表 6.1.1(a) :各对象可变更的最大数量

对象	最大数量
机器人	32
托盘	16
传感器	32
传送带	32
传送带工作站	84
固定工作站	84

6.1.2 方案

将 *i*RPickTool 的整个工作单元汇总在一起称为方案。
通过使用不同的方案，可以更换整个工作单元。例如，在以下情形有效。

- 从某一个产品更改成另一产品
- 用不同配套的夹爪、托盘、机器人、传感器、传送带工作站、固定工作站生产相同产品

> **⚠ 注意**
> **使用方案时，需要符合通过 ROS 以太网接口包（RIPE）连接的所有控制装置的软件版本。**

使用方案时，选定一个方案（称为活动方案。），执行与该内容相对应的工作单元。

关于详情，请参阅“9.1 使用方案变更工作单元”。

6.2 机器人设定

机器人的设定画面如下。

各项目的内容如下。

[控制装置]

选择对应机器人的控制装置。如果未设定 RIPE，仅显示“此控制装置”。

[组编号]

选择对应机器人的组编号。如果未能以［控制装置］选择的控制装置安装多组时，只显示 1。

[动态追踪误差调整]

此项目在基本动作的示教完成后进行设定。在基本动作的示教完成之前，无需进行设定。详情请参照“6.12 追踪动作的微调整”。
所谓动态追踪误差，是引起机器人对传送带的追踪延迟的误差。请依照“6.12.3 机器人的动态误差的调整”进行值的设定。这里的设定值，对伺服传送带以外的传送带有效。

使用伺服传送带时，请对各个伺服传送带调整以下参数。

$SLTK_GRP［伺服传送带的组编号］.$SRVO_DELAY

［跳过动作实施范围］

在基本动作的示教结束后进行这些项目的设定。在基本动作的示教完成之前，无需进行设定。详情请参照“9.4 超出领域时，跳过追踪动作”。

［圆形的追踪范围］

要设定圆形追踪区域启用，在 0~5000 的范围内设定半径。
只有用选定的机器人启用圆形追踪区域时，才会在传送带工作站的设定画面显示［圆形追踪区域］项目。

另外，圆形的追踪范围有效时，预整列功能以及追踪坐标系的根据 Y 排序的功能不能使用。

6.3 托盘设定

进行托盘设定。
托盘的设定画面如下。

不使用固定工作站，且不在传送带上使用托盘的情况，不需设定托盘。请跳过此节。
各项目的内容如下。

［基准单元］

根据各单元的 X、Y、Z、R 与基准单元的 X、Y、Z、R 计算针对各单元的补偿量。

按下 F3［基准单元］，将单元 1 的值设定为基准单元。
此外，如果是使用托盘的传送带，在传感器设定画面设定基准位置后，将单元 1 的值作为基准单元设定。

固定工作站时，按下 F3［基准单元］，设定基准单元后，示教单元 1 的机器人位置。
如果是使用托盘的传送带，在传感器设定画面设定基准位置后，示教单元 1 的机器人位置。如果是使用托盘的传送带，可以省略在托盘设定画面设定基准单元的步骤。

示教单元 1 的机器人位置后，如果只有在特定单元微调机器人位置，更改各单元的坐标值。

更改基准单元时，需要重新设定传感器的基准位置并重新示教机器人的位置。

［层编号］

设定多个层时，切换显示单元列表的层。

注
1 在传送带上使用托盘时，追加单元的顺序，与执行时工件放置于托盘上(或从托盘上取下)的顺序无关。根据各个状况，从下游侧的单元依次进行。需要特意控制顺序时，使用“6.3.2 层操作”所述的“层”。
2 1 个托盘上可追加单元的最大数量为 160 个。

［编辑单元 1］～［编辑单元 160］

编辑单元。根据选中的单元，项目名称变化为［编辑单元 1］～［编辑单元 160］。

［X（mm）］，［Y（mm）］，［Z（mm）］，［R（deg）］

输入加亮单元的托盘坐标系上的坐标值。

关于坐标值的设定，以利用托盘操作 4 个工件为例进行说明。

首先，决定将托盘坐标系放在哪个位置。
例如，如下图 6.3（a）设成箱上面的角。

注
在传送带上使用托盘的情况下，托盘坐标系的姿势决定为如下所示。
托盘坐标系的 Z 轴方向与追踪坐标系的 Z 轴方向相同。此外，托盘坐标系的 XY 平面与追踪坐标系的 XY 平面平行。于是，只要决定托盘坐标系的 X 轴方向，也就定下了所有其他轴的方向。

托盘坐标系可在托盘上自由定义，请定于容易碰触的位置。向机器人指导传送带上的托盘坐标系时，必须用机器人的 TCP 碰触托盘坐标系的原点和 X 轴方向点。
此外，固定工作站上的托盘坐标系设定为用户坐标系。

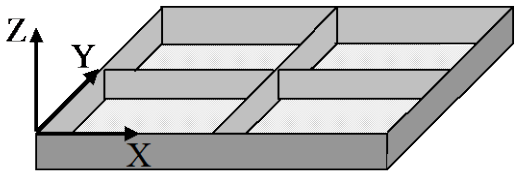


图 6.3(a)

对托盘追加单元，输入各单元的坐标值。
例如，俯视图 6.3(a)的托盘，如下图所示。

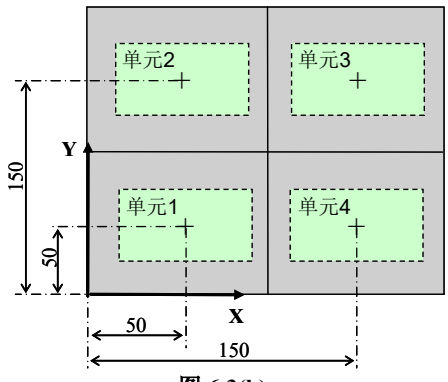


图 6.3(b)

“+”标记是各单元的中心位置。各单元的坐标值如下所示。

工件高度不同时的托盘设定

处理的工件高度不同时，利用单元的 Z 进行应对。

例如，如下图所示有 2 个单元，各个放置的工件高度不同。

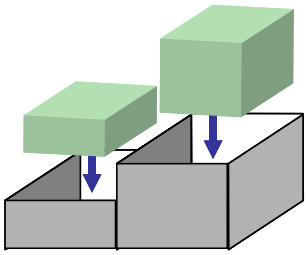


图 6.3(c)工件高度不同的例子

这种情况，根据工件的高度，变更各单元的 Z 值。例如下面所示。这种情况，放置的工件种类不同，因此设定不同模型 ID。

［模型 ID］
可对单元设定模型 ID。在需要根据被分配的单元，改变机器人 TP 程序上的处理等时使用。
如果并非必要，标准值可以保持为 1。

使用模型 ID 的例子，在“8 不同用途的设定例子”中介绍。

6.3.1 单元

在托盘上追加单元时，选择［追加单元］。要删除加亮的单元时，选择［删除单元］。

注
1 在传送带上使用托盘时，追加单元的顺序，与执行时工件放置于托盘上（或从托盘上取下）的顺序无关。根据各个状况，从下游侧的单元依次进行。需要特意控制顺序时，使用下面所述的“层”。
2 1 个托盘上可追加单元的最大数量为 160 个。

取多个的机械手的托盘设定

使用取多个的机械手时，机器人的一次动作可处理多个工件。

例如，考虑以取 3 个机械手，使用如下图所示的可收藏 12 个工件的托盘的情况。
虚线圆表示工件。

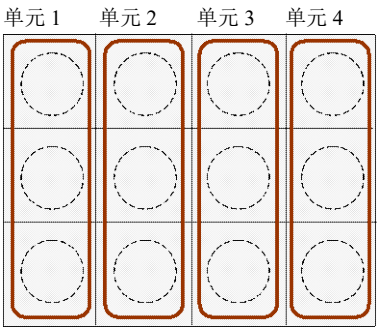


图 6.3.1(a) 拾取 3 个工件的机械手托盘例子

机器人每次处理 3 个工件。这种情况下，将每次可处理的 3 个工件作为一个单元来考虑。
因此，托盘具有 4 个单元。
下面示出具有 4 个单元时的托盘的一个设定例。

6.3.2 层操作

如果在托盘内放置有多段工件，则生成对应各段的层。追加层时，选择［1.追加层］，要删除用［层编号］选择的层时，选择［删除层］。
通常情况，在 1 个层内生成全部单元。托盘内有多个相同层内的单元时，根据情况决定处理顺序。例如，某机器人从其追踪范围内的下游侧的单元开始，依次处理。
在例如托盘以段为组，如果放置下面的段有剩余，也不能放置于上面的段等需要控制托盘内放置工件顺序的情况下，使用多个层。工件放置于托盘的情况，后述的传送带、固定工作站的［初始状态］设定为［空］。于是，残留有层编号较小的下层单元的期间，层编号较大上层的单元上不能放置工件。

以下是使用层的例子。箱子中 3 段层叠堆放工件。这种情况，单元是 3 个托盘，其优先顺序是：如果第 1 段不放置，则第 2 段、第 3 段都无法放置，如果第 2 段不放置，则第 3 段无法放置。

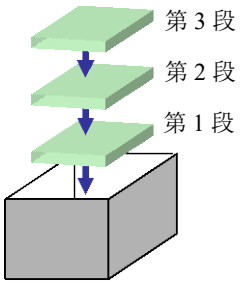


图 6.3.2(a)在托盘内多层放置工件的例子

这种情况，托盘做如下设定。

1　首先，第 1 段的单元 1 做以下设定。

2　然后，为追加第 2 层，从 F4［层操作］选择［追加层］。
　　于是，显示如下画面，输入第 2 层的单元数 1、以及第 2 层的 Z =–50 mm，按下 F4［确定］。

3　选择了追加的层 2，与单元 1 同样，对单元 2 的 X 和 Y 输入 X = 25.0 mm, Y = 25.0 mm。

4　同样，指定层 3 的 Z 并追加，设定单元 3 的 X、Y。

如以上所述，将单元 1 设定为层 1(最下面)、单元 2 设定为层 2(中间)、单元 3 设定为层 3(最上面)。各单元的 X 和 Y 的值相同，Z 是如上面层的单元的较大值。

并且，在处理该托盘的传送带、固定工作站中，后述的［初始状态］设定为［空］。这样设定后，在层编号最小的单元 1 上放置工件之前，机器人上不分配单元 2 和 3 的信息，这些位置上不会放置工件。

6.3.3 基准单元的设定

设定基准单元。基准单元的设置，在“6.11 机器人的位置示教”之前进行。

在托盘的设定画面按下 F3［基准单元］，显示如下提醒重新设定传感器基准位置和重新示教机器人位置的消息。

按下 F4［确定］后，“未示教（红色文字）”从［基准单元］消失，显示用［编辑单元 1］设定的 X、Y、Z、R 值。

6.4 传送带设定

进行传送带设定。
有关传送带的子对象的传感器、传送带工作站的详细设定，分别在“6.5 传感器设定”、“6.6 传送带工作站设定”中进行说明。

传送带的设定画面如下。

[追踪类型]

选择线追踪或圆弧追踪。线追踪时，选择［直线］，圆弧追踪的情况，选择［圆弧］。设定后，子对象的传感器、传送带工作站的设定将全部恢复到初始化设定。

[再循环]

勾选只有在［追踪类型］选择［圆弧］时显示的［再循环］后，未完成处理的托盘从最下游工作站被传到最上游工作站。
例如，圆弧传送带上的托盘处于缺齿状态时，进行再循环就可以填上缺齿的托盘。

可设定［再循环］的条件如下。

- 在［追踪类型］中选择［圆弧］
- 使用［托盘］
- 在子对象中添加 2 个以上的传送带工作站

到托盘完成，機器人在圆弧传送带上循環

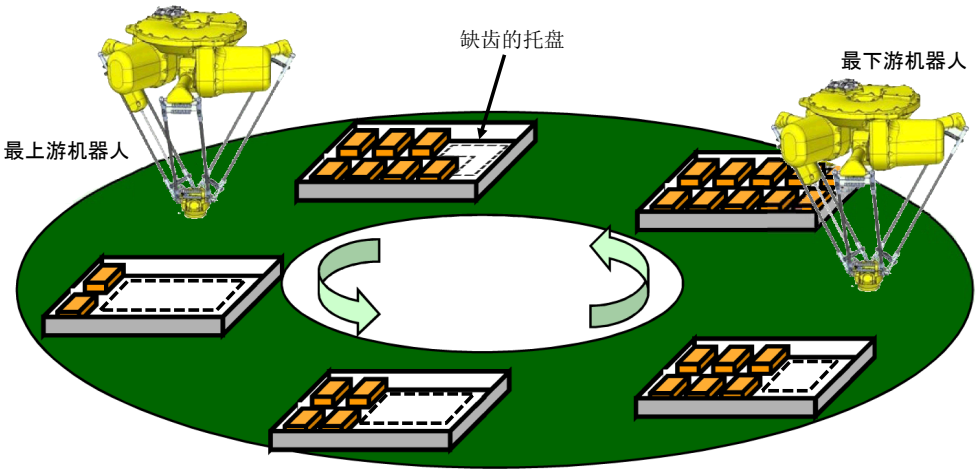


图 6.4(a)缺齿托盘的再循环示意图

［伺服传送带］

仅在追踪类型被设定为［直线］时，可进行伺服传送带的设定。传送带为伺服传送带时，请选中［已使用］。设定后，子对象的传感器、传送带工作站的设定将全部恢复到初始化设定。
使用伺服传送带时，需事先另行进行伺服传送带的设定。详细信息，请参照“9.7 伺服传送带追踪拾取功能”。

［托盘］

在传送带上使用托盘时，选择使用的托盘。不使用托盘时，选择［未使用］。

［初始状态］

在传送带上传送的空托盘上放置工件时，指定为［空］。从传送带上传送的装有工件的托盘中拾取工件时，指定为［满］。

［同一物检查的阈值］

视觉多次检出相同工件时，用于判断工件是否相同的阈值。通常情况，多次检出相同工件时，由于检测误差等，各个工件的现在位置不会完全一致。在这里设定阈值，仅留下被多次检出的工件中的其中之一。
如果判断为同一工件，则根据［同一物检查的模式］的设定，只留下一个结果。通常设定为 10 mm 左右。

［同一物检查的模式］框

通过同一物检查判断为同一工件时，指定保留哪个结果。

［模型 ID］

从［不考虑］、［保留最大］、［保留最小］、［只以最大的凋果覆盖模型 ID］、［只以最小的凋果覆盖模型 ID］中选择。

［测量值］

［模型 ID］为［不考虑］时，从［不考虑］、［保留最大］、［保留最小］中选择。
［模型 ID］为［保留最大］或［保留最小］时，从［不考虑］、［要是模型 ID 相同，保留最大］、［要是模型 ID 相同，保留最小］中选择。
［模型 ID］为［只以最大的凋果覆盖模型 ID］或［只以最小的凋果覆盖模型 ID］时，不能选择。自动更改为［不考虑］。
选择了［不考虑］以外的项目时，从［值 1］到［值 10］中选择使用哪个测量值。

［第一个/最后一个］

［模型 ID］为［不考虑］，［测量值］为［不考虑］时，从［保留最初］、［保留最后］中选择。

［模型 ID］为［不考虑］，［测量值］为［保留最大］或［保留最小］时，从［要是测量值相同，保留最初］、［要是测量值相同，保留最后］中选择。
［模型 ID］为［保留最大］或［保留最小］，［测量值］为［不考虑］时，从［要是模型 ID 相同，保留最初］、［要是模型 ID 相同，保留最后］中选择。
［模型 ID］为［保留最大］或［保留最小］，［测量值］为［要是模型 ID 相同，保留最大］或［要是模型 ID 相同，保留最小］时，从［要是模型 ID、测量值相同，保留最初］、［要是模型 ID、测量值相同，保留最后］中选择。
［模型 ID］为［只以最大的潤果覆盖模型 ID］或［只以最小的潤果覆盖模型 ID］时，不能选择。自动更改为［保留最初（非模型 ID）］。

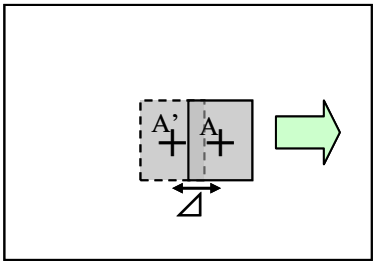
分别出现以下举动。

- ［模型 ID］为［不考虑］，［测量值］为［不考虑］，［第一个/最后一个］为［保留最初］时，保留最初的结果。
- ［模型 ID］为［不考虑］，［测量值］为［不考虑］，［第一个/最后一个］为［保留最后］时，保留最后的结果。
- ［模型 ID］为［不考虑］，［测量值］为［保留最大］，［第一个/最后一个］为［要是测量值相同，保留最初］时，保留指定的测量值较大一方的结果。测量值相同时，保留最初的结果。
- ［模型 ID］为［不考虑］，［测量值］为［保留最大］，［第一个/最后一个］为［要是测量值相同，保留最后］时，保留指定的测量值较大一方的结果。测量值相同时，保留最后的结果。
- ［模型 ID］为［不考虑］，［测量值］为［保留最小］，［第一个/最后一个］为［要是测量值相同，保留最初］时，保留指定的测量值较小一方的结果。测量值相同时，保留最初的结果。
- ［模型 ID］为［不考虑］，［测量值］为［保留最小］，［第一个/最后一个］为［要是测量值相同，保留最后］时，保留指定的测量值较小一方的结果。测量值相同时，保留最后的结果。
- ［模型 ID］为［保留最大］，［测量值］为［不考虑］，［第一个/最后一个］为［要是模型 ID 相同，保留最初］时，保留模型 ID 较大一方的结果。模型 ID 相同时，保留最初的结果。
- ［模型 ID］为［保留最大］，［测量值］为［不考虑］，［第一个/最后一个］为［要是模型 ID 相同，保留最后］时，保留模型 ID 较大一方的结果。模型 ID 相同时，保留最后的结果。
- ［模型 ID］为［保留最大］，［测量值］为［要是模型 ID 相同，保留最大］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最初］时，保留模型 ID 较大一方的结果。模型 ID 相同时，保留指定的测量值较大一方的结果。模型 ID 和测量值都相同时，保留最初的结果。
- ［模型 ID］为［保留最大］，［测量值］为［要是模型 ID 相同，保留最大］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最后］时，保留模型 ID 较大一方的结果。模型 ID 相同时，保留指定的测量值较大一方的结果。模型 ID 和测量值都相同时，保留最后的结果。
- ［模型 ID］为［保留最大］，［测量值］为［要是模型 ID 相同，保留最小］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最初］时，保留模型 ID 较大一方的结果。模型 ID 相同时，保留指定的测量值较小一方的结果。模型 ID 和测量值都相同时，保留最初的结果。
- ［模型 ID］为［保留最大］，［测量值］为［要是模型 ID 相同，保留最小］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最后］时，保留模型 ID 较大一方的结果。模型 ID 相同时，保留指定的测量值较小一方的结果。模型 ID 和测量值都相同时，保留最后的结果。
- ［模型 ID］为［保留最小］，［测量值］为［不考虑］，［第一个/最后一个］为［要是模型 ID 相同，保留最初］时，保留模型 ID 较小一方的结果。模型 ID 相同时，保留最初的结果。
- ［模型 ID］为［保留最小］，［测量值］为［不考虑］，［第一个/最后一个］为［要是模型 ID 相同，保留最后］时，保留模型 ID 较小一方的结果。模型 ID 相同时，保留最后的结果。
- ［模型 ID］为［保留最小］，［测量值］为［要是模型 ID 相同，保留最大］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最初］时，保留模型 ID 较小一方的结果。模型 ID 相同时，保留指定的测量值较大一方的结果。模型 ID 和测量值都相同时，保留最初的结果。
- ［模型 ID］为［保留最小］，［测量值］为［要是模型 ID 相同，保留最大］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最后］时，保留模型 ID 较小一方的结果。模型 ID 相同时，保留指定的测量值较大一方的结果。模型 ID 和测量值都相同时，保留最后的结果。
- ［模型 ID］为［保留最小］，［测量值］为［要是模型 ID 相同，保留最小］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最初］时，保留模型 ID 较小一方的结果。模型 ID 相同时，保留指定的测量值较小一方的结果。模型 ID 和测量值都相同时，保留最初的结果。
- ［模型 ID］为［保留最小］，［测量值］为［要是模型 ID 相同，保留最小］，［第一个/最后一个］为［要是模型 ID、测量值相同，保留最后］时，保留模型 ID 较小一方的结果。模型 ID 相同时，保留指定的测量值较小一方的结果。模型 ID 和测量值都相同时，保留最后的结果。
- ［模型 ID］为［只以最大的潤果覆盖模型 ID］，［测量值］中［不考虑］、［第一个/最后一个］为［保留最初（非模型 ID）］时，保留最初的结果。但是，模型 ID 按照对同一工件的结果的最大值进行覆盖保存。

- ［模型 ID］为［只以最小的潤果覆盖模型 ID］，［测量值］中［不考虑］、［第一个/最后一个］为［保留最初（非模型 ID）］时，保留最初的结果。但是，模型 ID 按照对同一工件的结果的最小值进行覆盖保存。

同一物检查的阈值以下时

⊿为同一物检查的阈值以下时，［模型 ID］中［不考虑］被选择，［测量值］中［不考虑］被选择，［第一个/最后一个］中［保留最初］被选择的默认设定中，只保留第 n–1 个检出的 A’工件。



A’：第（n−1）个检出对象物的现在位置
A ：第n个检出对象物的现在位置
⊿ ：AA’间的距离

图 6.4(b)低于同一物检查阈值的例子

> ⚠ **注意**
> **同一物检查，可能省略多次检出的相同工件以外的工件。如果先检出的工件的现在位置与新检出工件的现在位置之间的距离为阈值以下，即使是不同的模型 ID，被判断为同一工件后一方也会被省略。**

> **注**
> **7DF1 系列 14 版以前的软件时，［同一物检查的阈值］被显示为［重叠的阈值］。此外，不支持［同一物检查的模式］。7DF1 系列 14 版以前的软件时，与［模型 ID］中［不考虑］被选择，［测量值］中［不考虑］被选择，［第一个/最后一个］中［保留最初］被选择的默认设定的情况具有相同的举动。**

［负载平衡］

指定在该传送带上分配工件的方法。

［无效］

各机器人尽量多地处理工件。

［进度一定］

进行分配，使各机器人以指定的比率处理工件。但是，即使因供给过多发生跳过，也不进行补充。结果可能导致各机器人处理的工件数量不等。

［保持个数比例］

进行分配，使各机器人以指定的比率处理工件。最终，各机器人处理的工件数量相等。

并且，除负载平衡无效的情况以外，选择指定平衡(各机器人处理工件的量的比率)的方法。

［所有模型 ID 共通］

可指定仅 1 组的平衡。无论工件的模型 ID 如何，使用指定的平衡进行分配。

［针对每个模型 ID］

最多可指定 8 组的平衡。根据工件的模型 ID，切换至使用的平衡。

> **注**
> **有关利用模型 ID 的负载平衡的指定方法，在“8 不同用途的设定例子”中介绍几个具体使用例。**

［负载平衡］框

［负载平衡］框指［工作单元设定］内的下面部分。

在负载平衡非无效的情况下，边更改模型 ID，边指定各传送带工作站的机器人处理工件的量的比率。

指定［所有模型 ID 共通］时，如左边项目显示。例如，传送带上有 3 个传送带工作站，需进行等分时，如果所有传送带工作站指定为 1，则以 1:1:1 的比例进行分配。1:1:1 和 2:2:2 是完全相同的意思。

指定［针对每个模型 ID］时，如右边项目显示。边选择模型 ID，边对全部模型 ID 指定比率。以［指定需处理的模型 ID］指定模型 ID。对 2 个以上的模型 ID 指定平衡时，按下［添加］按钮，可追加模型 ID 的 ID。可指定最多 8 个模型 ID。可通过按下［删除］按钮，删除不要的模型 ID 的 ID。

如果［跳过］设定为 0 以外的值，该传送带将跳过指定值部分。物理性相同传送带上定义有多个传送带对象，需要在其他传送带对象上传送一定量的工件时，设定为 0 以外的值。

［设置预整列］

在相同传送带上使用整列功能(预整列功能)时的设定项目。
详细信息请参照“6.14 预整列设定”。

［预整列］

在相同传送带上使用整列功能(预整列功能)时，选中［已使用］。选中后，显示详细的设定项目。有关各个项目的详细信息，请参照“6.14 预整列设定”。

［日志设置］

如果日志有效，则放入队列的工件的信息会被记录于暂时存储器。并能在执行中输出至指定的输出装置。这里获得的日志，在 ROBOGUIDE 图像回放工件的供给时使用。

［日志］

日志有效时选中。从选中的瞬间起开始记录。记录的工件数为 1000 个，如果超过 1000 个，则从老的工件依次删除。

［输出设备］

选择输出日志的装置。可选择 UD1 或 MC。通过按下输出按钮，输出之前的日志。在机器人稼动中也能输出。

［输出控制装置］

显示输出日志的控制装置。在该传送带上，输出至在线的最上游机器人的控制装置。

6.4.1 追踪坐标系设定

设定追踪坐标系和编码器的倍率。

追踪坐标系、编码器的倍率的设定，也可在机器人示教器的追踪参数的设置画面上进行，但是在特别是以多个机器人进行追踪的情况，需要进行全部机器人共通的追踪坐标系设定，建议根据以下说明进行。

⚠ 注意
对于 3 轴～5 轴机器人，如果在示教器的追踪参数设置画面中进行追踪坐标系的设定，追踪坐标系的 W, P 中可能为 0 以外的值。另一方面，如果依照以下说明步骤进行设定，3 轴～5 轴机器人的追踪坐标系的 W, P 必须是 0。3 轴～5 轴机器人在追踪过程中，法兰盘面只能进行一直向着正下方(或正上方)的动作，请依照以下说明进行追踪坐标系的设定。
另一方面，以下说明的步骤，不对应 3 轴～5 轴机器人中世界坐标系的 Z 值变化的追踪。这种情况下，必须在示教器的追踪参数的设置画面中进行设定。但是，这种情况下，追踪过程中的法兰盘面也必须一直向着正下方(或正上方)，因此需要通过直接输入坐标系的成分进行微调整。

追踪坐标系设定，通过在传送带上传送夹具进行。只要是具有机器人能够通过 TCP 高精度碰触的位置(1 个位置)的物品，都可作为夹具。由本公司购买校准用板时，也可使用校准用板上的 1 个点。
这里对使用校准用板时的情况进行说明。

⚠ 注意
1 追踪坐标系的设定或重设的情况，必须重新进行传送带工作站的设定、相机的校准、基准位置的设定以及机器人的位置示教。
2 当有多台机器人时，如果传送带上的各控制装置，编码器的计数不一致，则无法进行追踪坐标系的设定。请使全部机器人控制装置的电源关闭 / 开启。
3 如果有多台机器人，请不要在传送带的皮带上移动夹具。如果夹具在传送带上移动时，接触到机器人等而偏离，则需从头开始追踪坐标系的设定。
4 追踪坐标系会影响最终的处理精度。请使用正确设定的 TCP，碰触夹具。
5 在追踪坐标系的设定作业中，如果机器人的手腕姿势(W, P, R)变化，则无法进行正确设定。请仅在 X, Y, Z 方向上点动机器人，使机器人的手腕姿势无变化。

根据追踪类型不同，设定内容不同。
关于生产线传送带的设定，请参阅“3 *i*RPickTool Basic（基本功能）启动调试步骤”。同一传送带上有多台机器人时，在完成设定第 1 台机器人的追踪坐标系后，就会开始设定第 2 台机器人的追踪坐标系。

此外，对于已设定追踪坐标系的传送带，如果只在特定传送带工作站修改追踪坐标系的设定，请参阅“6.6.1 追踪坐标系的追加”。

6.4.1.1 圆弧传送带的情况

圆弧传送带的设定步骤如下。

安装圆弧追踪

针对使用圆弧追踪时的 *i*RPickTool 安装方法进行说明。作为圆弧追踪时的特别事项，请注意以下几点。

请务必在后述“圆弧传送带设定”前将［追踪类型］选择为［圆弧］。

追踪坐标系的 Z 轴方向

圆弧追踪中，根据传送带的旋转方向，追踪坐标系的 Z 轴方向有的情况下会从传送带面变为朝下。具体情形如下所示。

- 传送带沿着逆时针方向运动时，Z 轴变为朝上。

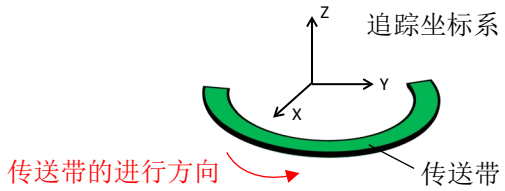


图 6.4.1.1(a)传送带：逆时针、Z 轴：朝上

- 传送带沿着顺时针方向运动时，Z 轴变为朝下。

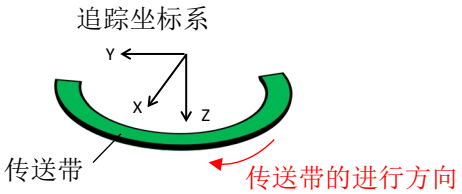


图 6.4.1.1(b)传送带：顺时针、Z 轴：朝下

托盘坐标系的 Z 轴方向成为与追踪坐标系的 Z 轴方向相同的方向。请在仔细确认追踪坐标系的 Z 轴方向后，进行“6.3 托盘设定”的 X, Y, Z, R 的设定。
此外，对于“3.2.7.2 视觉程序”中说明的［基准数据］的［检出面 Z 向高度］，也请在仔细确认 Z 轴方向后进行设定。

> **⚠ 注意**
> **在进行追踪坐标系的设定时，请尽可能将传送带设定为低速后进行。**

圆弧传送带的设定

1 仅追加以传送带操作的机器人数量的传送带工作站，在各传送带工作站的设置画面选择机器人和追踪参数设定表的编号、编码器编号。
详细信息请参照“6.6 传送带工作站设定”。

2 在传送带的设置画面按下［设定追踪坐标系］。
显示如下画面。

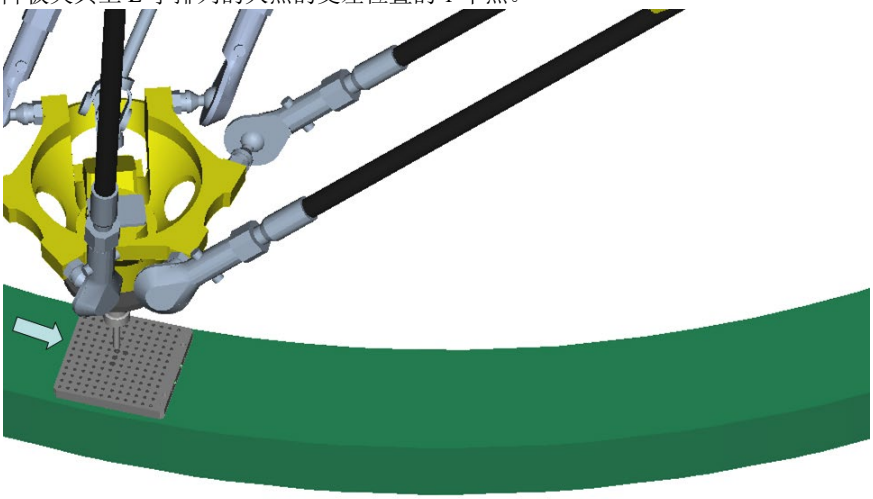
3 将可以机器人的 TCP 碰触的夹具放置于传送带的最上游部，按下 F5［下一步］。
不自动计算编码器的倍率，直接使用从实测值计算所得的值的情况，在文本框内输入倍率值，按下［指定倍率］。
无论是哪种选择，都显示以下画面。
作为碰触夹具使用视觉的校准用板时，由于对 1 个点碰触 3 次，大点面向任何方向都没有问题。

> ⚠ **注意**
> 碰触时，要尽量在传送带的外周进行。这是为了提高追踪的精度。

4 “5.4 工具坐标系的设定”中设定的工具坐标系作为手动进给坐标系被选择。

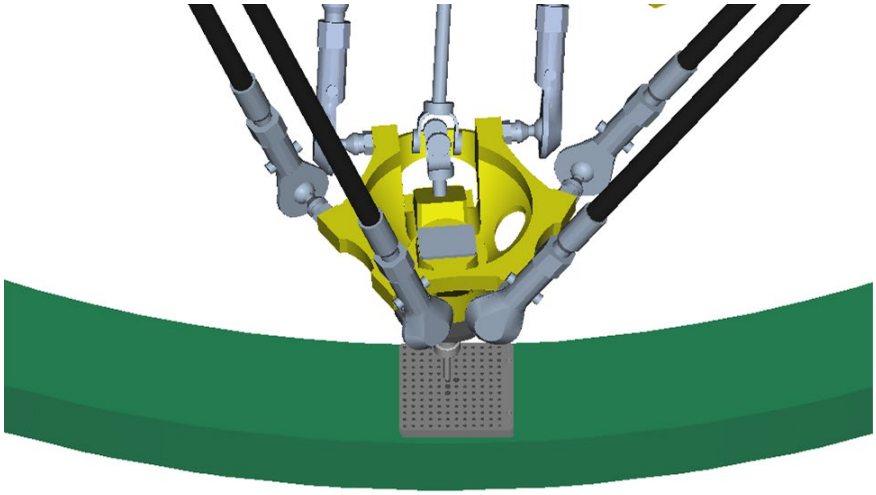
> ⚠ **注意**
> **在夹具中使用碰触机器人选择工具坐标系。并不仅限于打开着 *i*RPickTool 的设置画面的机器人控制装置。**

5 移动并固定传送带，尽量使夹具进入机器人动作领域内的上游部。
6 点动移动机器人，用 TCP 触碰夹具。
此处碰触在点阵板夹具上 L 字排列的大点的交差位置的 1 个点。



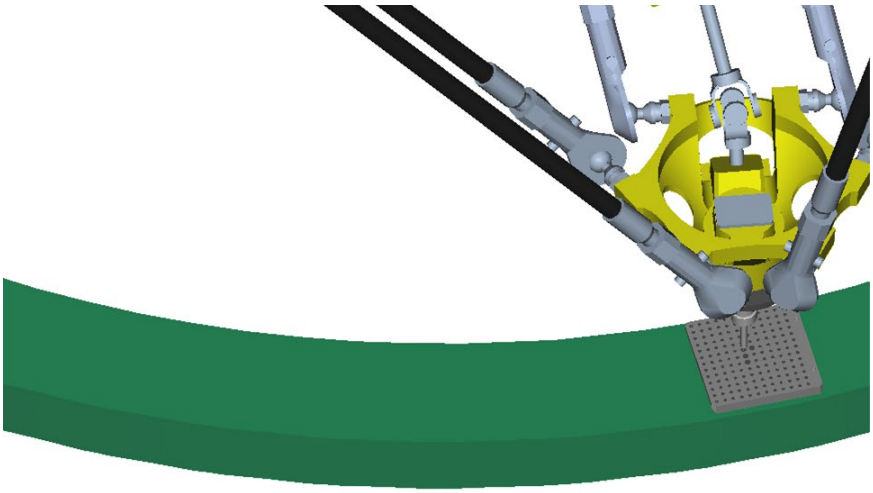
7 在碰触后的状态下，按下 F5［下一步］。
显示如下画面。

8 移动机器人至即使传送带动作，TCP 也不干涉的位置，移动并固定传送带，使夹具进入机器人动作领域内的中游部。
9 并且再次点动机器人，碰触之前以 TCP 碰触的夹具的相同位置。



10 在碰触后的状态下，按下 F5［下一步］。
显示如下画面。

11 移动机器人至即使传送带动作，TCP 也不干涉的位置，移动并固定传送带，尽量使夹具进入机器人动作领域内的下游部。
12 并且再次点动机器人，碰触之前以 TCP 碰触的夹具的相同位置。



13 在碰触后的状态下，按下 F5［下一步］。
显示如下画面。

14 通过以上操作，第 1 台机器人的追踪坐标系设定操作结束，设定了追踪坐标系和倍率。
如果相同传送带上有多台机器人，按下 F5［下一步］，开始第 2 台机器人的追踪坐标系的设定。
15 在各机器人反复进行步骤 4～12。
全部机器人结束操作后，会显示以下画面。

16 按下 F4［确定］，结束追踪坐标系的设定。

注
需要重新进行机器人的碰触时，可通过［前一步］返回前一个设定页。

6.5 传感器设定

进行传感器设定。

> **注意**
> 1 如果传送带设定中的追踪坐标系的设定不结束，则无法进行传感器设定。
> 2 即使已结束这里的设定，传感器的示教还尚未完成。通过进行基准位置的设定“6.10 基准位置设定”，传感器就会成为示教状态(绿色)。

6.5.1 传感器详细设定

在树状图上选择传感器［SENS 1］后，显示如下画面。

［本传感器］

指定是否使用画面中选择的传感器。如果使用，则选中有效。

［控制装置］

设定机器人之间通信的情况，选择连接于本传感器的控制装置名。未设定机器人之间通信的情况(即 1 台控制装置的情况)，只能选择［此控制装置］。

> **注**
> **在［控制装置］中选择的控制装置的机器人需要在相同传送带上的任意一个传送带工作站上被选择。**

［触发条件］

从以下选择触发条件。

［距离］

传送带每次进行指定距离移动时，执行动作。

［DI］

检测到 DI 信号的跳起(关闭→开启)，执行动作。

［HDI］

检测到 HDI 信号的输入，执行动作。比 DI 更高精度。

［RI］
检测 RI 信号的输入，执行动作。

［标记］
检测到指定标记的输入，执行动作。使用伺服传送带时，自动选择。

［触发］
满足触发条件时，从以下选择执行的动作。

［放入工件］
将 1 个工件放入队列。
有以下限制。
- 工件的模型 ID 固定为 0。
- 不可在传送带中为［负载平衡］选择［针对每个种类编号］。

［通过视觉检出］
通过视觉检出工件，将检出的工件放入队列。触发条件为 HDI 或者标记时将无法进行选择。

［程序的执行］
执行指定的 TP 程序。
有以下制约。
- 触发条件为 HDI 或者标记时将无法进行选择。
- 只有在直线传送带上可以使用。
- 使用托盘的传送带不能使用。
- 预整列的输入传送带不能使用。
- 只在 3rd Party 接口加载项中可以使用。详细内容请参照“I 3rd Party 接口加载项”。

注
7DF1 系列 16 版以前的软件时，［执行程序］不能使用。
此外，在 ROBOGUIDE 中［程序的执行］不能使用。

可能的触发条件和动作的组合如下所示。

表 6.5.1 可能的触发条件和动作的组合

	距离	DI	HDI	RI	标记
放入工件	○	○	○	○	○
通过视觉检出	○	○	×	○	×
程序的执行	○	○	×	○	×

关于用触发条件与动作的组合切换的设定画面，请参阅“6.5.1.1 ［距离］ ＋ ［放入工件］的情况”～“6.5.1.11 ［RI］＋［执行程序］的情况”。

6.5.1.1 ［距离］ ＋ ［放入工件］的情况

传送带每次进行指定距离移动时，队列中放入 1 个工件。
需要在传送带上等间隔排列工件时，原来是通过等间隔粘贴胶带，并检测该胶带进行排列，现在只要使用该组合，无需粘贴胶带就能等间隔排列工件。
此外，由于不实际检出工件也能在队列中放入工件，因此在需要进行拾取动作的模拟时也有用。

[触发距离]

指定传送带每移动多少 mm 放入工件。
指定圆弧传送带的情况下的角度。

[目标位置]

每次触发距离传送带行进所生成的工件位置。用后述目标位置的设定向导进行设定。

[基准位置]

结束后述目标位置的设定向导后，绿色显示［向导成功］。

6.5.1.2 ［距离］＋［通过视觉检出］的情况

传送带每次进行指定距离移动时，执行视觉。

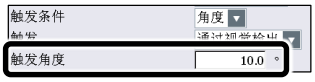
[触发距离]

指定传送带每移动多少毫米执行视觉。建议在相机视野内将可以检测工件 2 次的距离设为触发距离。

> ⚠ **注意**
> **如果触发距离设定得过短，前一次的视觉检出还未结束，就执行下一个视觉的状况持续，则可能发出“INTP-302 堆栈溢出”报警。这种情况，请加长该距离，在前一个视觉检出结束后进行下一个视觉的执行。具体请参照“9.2.2 确认传感器的状态”的［动作时间］的数值。**

圆弧传送带的情况

圆弧传送带时，关于［距离］＋［通过视觉检出］时的［触发距离］项目指定角度（单位：度）。



［视觉程序名］

输入执行的视觉程序名。

［托盘位置］

托盘的检出位置和托盘原点的相对位置。用后述托盘位置的设定向导进行设定。

［基准位置］

结束后述基准位置的设定向导后，绿色显示［向导成功］。

6.5.1.3 ［距离］+［程序的执行］

指定的距离传送带每次移动时执行指定的 TP 程序。

［触发距离］

指定传送带每移动多少 mm 执行 TP 程序。

［编码器取得用 DI/RI 编号］

用于将 3rd Party 视觉的拍摄时机传达给控制装置的输入信号的编号。该信号被输入时，控制装置取得编码数。已经在其他传感器中被指定的编码器取得用 DI/RI 编号不能指定。

［程序名］

指定执行的 TP 程序的名称。使用动作组的程序不能使用。已经在其他传感器中指定的程序名不能指定。

6.5.1.4 ［DI］＋［放入工件］的情况

检测到 DI 信号的跳起(关闭→开启)，在队列中放入 1 个工件。

［DI 编号］

指定检知的数字输入信号的编号。

> **注意**
> **与机器人示教器追踪参数的设置画面中显示的［触发输入编号］不同。**

［再输入 DI 无效时间］

一旦检知 DI 的输入后，如果比这里设定的时间更快检知下一个 DI 的输入，则该 DI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致双重检知工件的现象。

［传感器位置］

如果检知工件，该位置将被作为检知位置被放入队列。用后述传感器位置的设定向导进行设定。

［基准位置］

结束后述基准位置的设定向导后，绿色显示［向导成功］。

6.5.1.5 ［DI］＋［通过视觉检出］的情况

检知到 DI 信号的跳起(关闭→开启)，执行视觉，将检出的工件放入队列。

［DI 编号］

指定检知的数字输入信号的编号。

> **注意**
> **与机器人示教器追踪参数的设置画面中显示的［触发输入编号］不同。**

［再输入 DI 无效时间］

一旦检知 DI 的输入后，如果比这里设定的时间更快检知下一个 DI 的输入，则该 DI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致视觉多次回走，CPU 负载无意识增加的现象。

［触发偏移］

检知 DI 信号的跳起后，传送带仅移动这里指定的距离后，执行视觉。在相机与光电管传感器分离配置的情况下使用。

圆弧传送带的情况

圆弧传送带的情况关于［DI］/［RI］＋［通过视觉检出］时的［触发偏移］项目指定角度（单位：度）。

［视觉程序］

指定执行的视觉程序名。

［托盘位置］

托盘的检出位置和托盘原点的相对位置。用后述托盘位置的设定向导进行设定。

［基准位置］

结束后述基准位置的设定向导后，绿色显示［向导成功］。

6.5.1.6 ［DI］+［程序的执行］

检测到 DI 信号的跳起（关闭→开启）执行指定的 TP 程序。

［DI 编号］

指定检知的数字输入信号的编号。

> **注意**
> **与机器人示教器追踪参数的设置画面中显示的［触发输入编号］不同。**

［再输入 DI 无效时间］

一旦检知 DI 的输入后，如果比这里设定的时间更快检知下一个 DI 的输入，则该 DI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致视觉多次回走，CPU 负载无意识增加的现象。

［触发偏移］

检知 DI 信号的跳起后，传送带仅移动这里指定的距离后，执行 TP 程序。

［编码器取得用 DI/RI 编号］

用于将 3rd Party 视觉的拍摄时机传达给控制装置的输入信号的编号。该信号被输入时，控制装置取得编码数。与 DI 编号相同的输入信号不能指定。已经在其他传感器中被指定的编码器取得用 DI/RI 编号不能指定。

［程序名］

指定执行的 TP 程序的名称。具有动作组的程序不能使用。已经在其他传感器中指定的程序名不能指定。

6.5.1.7 ［HDI］＋［放入工件］的情况

检知到 HDI 信号的输入，在队列中放入 1 个工件。

［HDI 端口］

显示检知的 HDI 端口编号。显示的端口编号，是传送带工作站指定的编码器编号的参数设定表中设定的编号。

［再输入 HDI 无效时间］

一旦检知 HDI 的输入后，如果比这里设定的时间更快检知下一个 HDI 的输入，则该 HDI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致双重检知工件的现象。

［传感器位置］

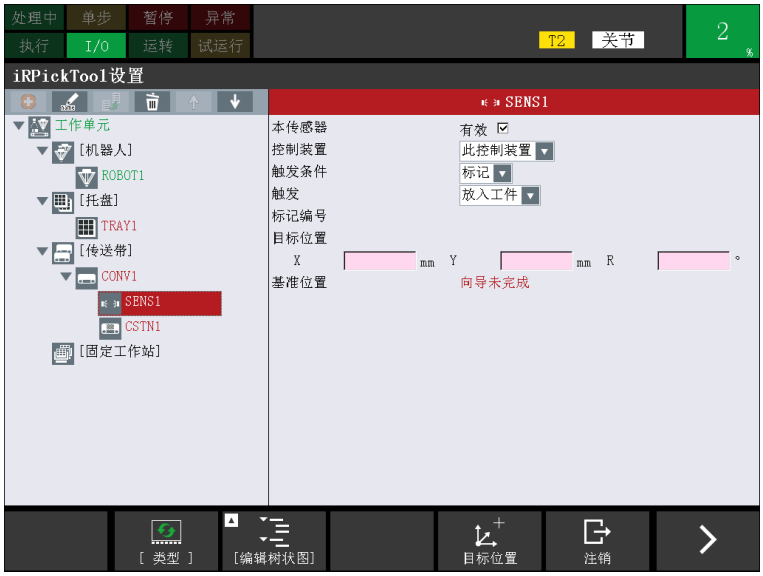
如果检知工件，该位置将被作为检知位置被放入队列。用后述传感器位置的设定向导进行设定。

［基准位置］

结束后述基准位置的设定向导后，显示［向导成功］。

6.5.1.8 ［标记］＋［放入工件］的情况

仅在伺服传送带的情况下使用。
检知到标记的跳起（关闭→开启），在队列中放入 1 个工件。由伺服传送带驱动时自动输入标记。



［标记编号］

显示检知的标记编号。在伺服传送带的设定中，设定标记编号。在该画面中无法进行更改。

［目标位置］

检知标记的输入时生成的工件位置。用后述目标位置的设定向导进行设定。

［基准位置］

结束后述目标位置的设定向导后，显示［向导成功］。

6.5.1.9 ［RI］＋［放入工件］的情况

检测到 RI 信号的跳起(关闭→开启)，在队列中放入 1 个工件。

启动调试［RI］+［放入工件］组合的步骤与［DI］+［放入工件］组合相同。
详情请参阅“3 *i*RPickToolBasic（基本功能）的启动调试步骤”或者是“4 *i*RPickTool（基本功能+ Plug & Play）的启动调试步骤”。

［RI 编号］

指定检知的数字输入信号的编号。

> **注意**
> **与机器人示教器追踪参数的设置画面中显示的［触发输入编号］不同。**

［再输入 RI 无效时间］

一旦检知 RI 的输入后，如果比这里设定的时间更快检知下一个 RI 的输入，则该 RI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致双重检知工件的现象。

［传感器位置］

如果检知工件，该位置将被作为检知位置被放入队列。用后述传感器位置的设定向导进行设定。

［基准位置］

结束后述基准位置的设定向导后，绿色显示［向导成功］。

6.5.1.10 ［RI］＋［通过视觉检出］的情况

检知到 RI 信号的跳起（关闭→开启），执行视觉，将检出的工件放入队列。

［RI 编号］

指定检知的数字输入信号的编号。

> **注意**
> **与机器人示教器追踪参数的设置画面中显示的［触发输入编号］不同。**

［再输入 RI 无效时间］

一旦检知 RI 的输入后，如果比这里设定的时间更快检知下一个 RI 的输入，则该 RI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致视觉多次回走，CPU 负载无意识增加的现象。

［触发偏移］

检知 RI 信号的跳起后，传送带仅移动这里指定的距离后，执行视觉。在相机与光电管传感器分离配置的情况下使用。

圆弧传送带的情况

圆弧传送带的情况关于［DI］/［RI］＋［通过视觉检出］时的［触发偏移］项目指定角度（单位：度）。

［视觉程序］

指定执行的视觉程序名。

［托盘位置］

托盘的检出位置和托盘原点的相对位置。用后述托盘位置的设定向导进行设定。

［基准位置］

结束后述基准位置的设定向导后，绿色显示［向导成功］。

6.5.1.11 ［RI］+［程序的执行］时

执行检测 RI 信号的输入指定的 TP 程序。

［RI 编号］

指定检知的数字输入信号的编号。

> **注意**
> **与机器人示教器追踪参数的设置画面中显示的［触发输入编号］不同。**

［再输入 RI 无效时间］

一旦检知 RI 的输入后，如果比这里设定的时间更快检知下一个 RI 的输入，则该 RI 会被作为误输入而被忽略。通过恰当设定该值，可防止因外部传感器的输入的触点振动而导致视觉多次回走，CPU 负载无意识增加的现象。

［触发偏移］

检知 DI 信号的跳起后，传送带仅移动这里指定的距离后，执行 TP 程序。

［编码器取得用 DI/RI 编号］

用于将 3rd Party 视觉的拍摄时机传达给控制装置的输入信号的编号。该信号被输入时，控制装置取得编码数。与 RI 编号相同的输入信号不能指定。已经在其他传感器中被指定的编码器取得用 DI/RI 编号不能指定。

［程序名］

指定执行的 TP 程序的名称。具有动作组的程序不能使用。已经在其他传感器中指定的程序名不能指定。

6.5.2 每次触发条件和动作组合的必要设定

所需的设定因触发条件与动作组合而异。

表 6.5.2 每个触发条件和动作组合所需的设定

触发条件	［触发］	有无使用托盘使用	6.5.3 传感器位置设定	6.5.4 视觉设定	6.5.5 托盘位置设定	6.5.6 目标位置设定
［距离］	［放入工件］	使用	×	×	×	O
		不使用	×	×	×	O
	［通过视觉检出］	使用	×	O	O	×
		不使用※1	×	O	×	×
	［程序的执行］	不使用※2	×	×	×	×
［DI］	［放入工件］	使用※1	O	×	×	×
		不使用	O	×	×	×
	［通过视觉检出］	使用	×	O	O	×
		不使用	×	O	×	×
	［程序的执行］	不使用※2	×	×	×	×
［HDI］	［放入工件］	使用※1	O	×	×	×
		不使用	O	×	×	×
［标记］	［放入工件］	使用	×	×	×	O
		不使用	×	×	×	O
［RI］	［放入工件］	使用※1	O	×	×	×
		不使用	O	×	×	×
	［通过视觉检出］	使用	×	O	O	×
		不使用	×	O	×	×
	［程序的执行］	不使用※2	×	×	×	×

※1 关于设定及启动调试步骤，请参阅“3 *i*RPickTool Basic（基本功能）的启动调试步骤”或者是“4 *i*RPickTool（基本功能+ Plug & Play）的启动调试步骤”。

※2 对触发使用［程序的执行］时，这些设定不是在 iRPickTool 的设定画面进行，而是通过 TP 程序进行。详细内容请参照“I 3rd Party 接口加载项”。

6.5.3 传感器位置设定

示教安装有光电管传感器的位置。
以触发条件选择［DI］/［HDI］/［RI］的任意一个，动作为［放入工件］时所需的设定。
工件横穿光电管传感器时，在队列中放入该位置。在传送带上使用与不使用托盘的情况，在步骤上略有不同。

注
需要在通过此传感器选择的控制装置中打开 iRPickTool 的设定画面，进行传感器位置设定。

使用托盘时的传感器位置设定步骤，请参阅“3 *i*RPickTool Basic（基本功能）的启动调试步骤”或者是“4 *i*RPickTool（基本功能+ Plug & Play）的启动调试步骤”。

6.5.3.1 不使用托盘的情况

1 使传送带停止。
2 按下 F4［传感器位置］。
则显示以下画面。

3 将工件从光电管传感器放到上游侧，在工件来到机器人前面之前运转传送带。

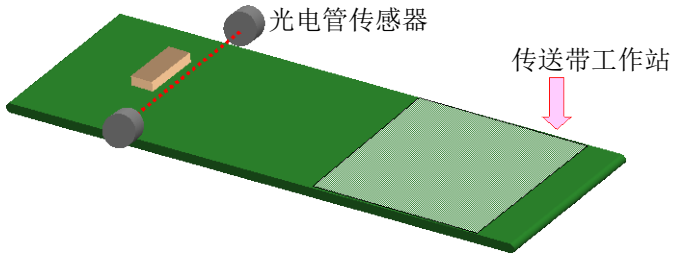


图 6.5.3.1(a)

如果传感器检知工件，则显示检知时的编码器值，F5［下一步］有效。
4 按下 F5［下一步］。

显示以下画面。

5 以点动形式运行机器人，碰触工件的下游侧的边界（通过光电管传感器检测到的地方）。

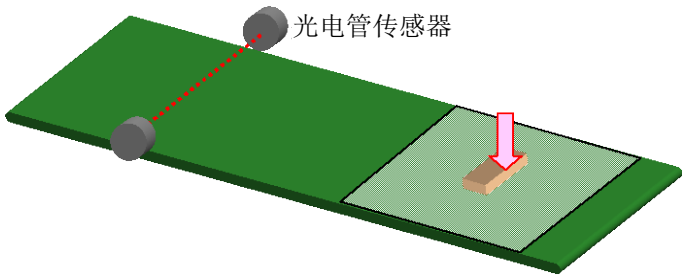


图 6.5.3.1(b)

6 按下 F4［记录］。
显示碰触时的编码器值、碰触位置、计算的位置，F5［结束］有效。

7 按下 F5［结束］，

返回原来的设置画面。仅在传感器位置的 X 中输入计算所得的值(Y 和 R 保持为 0)。

6.5.4 视觉设定

进行用于工件检出的相机设定、相机的校准、视觉程序的示教。
通过动作选择［通过视觉检出］时所必要的设定。

> **⚠ 注意**
> **视觉的设定必须由相机连接的控制装置进行。**

关于使用 1 台二维相机时的视觉的设定步骤，请参照“3 *i*RPickTool 基本（基本功能）的启动调试步骤”或“4 *i*RPickTool（基本功能 + Plug & Play）的启动调试步骤”。

注

设定圆弧传送带时，关于［基准数据］中的［检出面 Z 向高度］项目，在设定时要注意追踪坐标系的 Z 轴方向。

- **传送带沿着逆时针方向运动时，Z 轴变为朝上。**

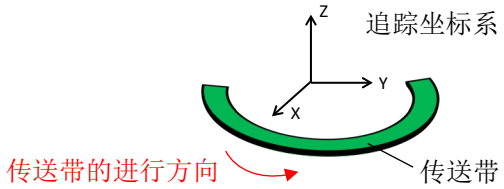


图 6.5.4(a)传送带为逆时针方向运动时的 Z 轴

- **传送带沿着顺时针方向运动时，Z 轴变为朝下。**

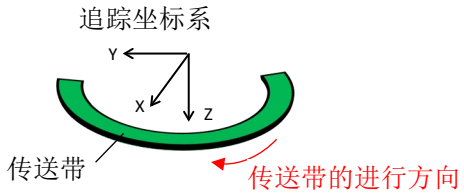


图 6.5.4(b)传送带为顺时针方向运动时的 Z 轴

关于使用 1 台 3DV 传感器时的视觉的设定步骤，请参照“9.9 3D 视觉传感器”。
关于使用多台二维相机（多视图）时的视觉的设定步骤，请参照“9.8 多视图”。

6.5.5 托盘位置设定

作为动作选择［通过视觉检出］，并使用托盘的情况，设定托盘的原点和检出位置的相对位置。需要实现结束相机的校准、视觉程序的示教。

注
需要在通过此传感器选择的控制装置中打开 iRPickTool 的设定画面，进行托盘位置设定。

6.5.5.1 ［距离］＋［通过视觉检出］的情况。

1 使传送带停止。
2 按下 F5［托盘位置］。
则显示以下画面。

3 托盘放置于相机视野内，按下 F4［检出］。

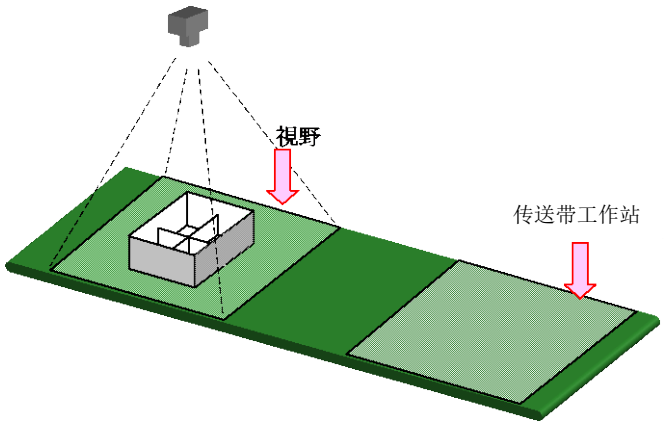


图 6.5.5.1(a)

如果成功检出托盘，则显示检出位置，F5［下一步］变得有效。

4 按下 F5［下一步］。

则显示以下画面。

5 运转传送带，直至托盘来到机器人的前面。

6 点动机器人，碰触托盘坐标系的原点。

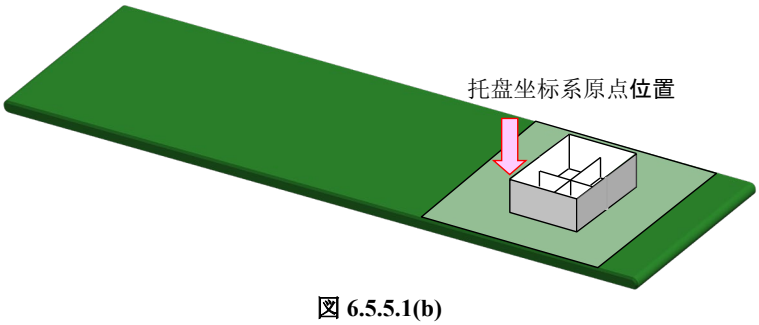


图 6.5.5.1(b)

7 按下 F4［记录］。
显示碰触时的编码器值、碰触位置，F5［下一步］有效。
8 按下 F5［下一步］。

则显示以下画面。

9 点动机器人，碰触托盘坐标系的 X 方向点。

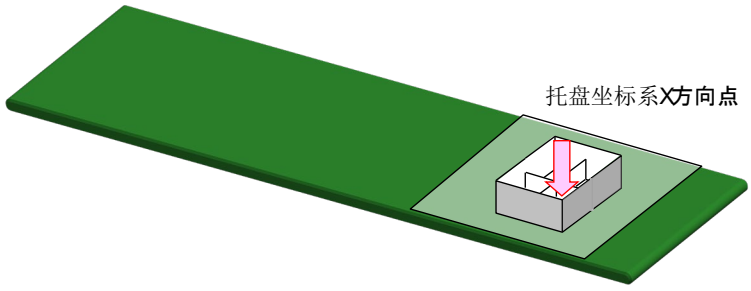


图 6.5.5.1(c)

10 按下 F4［记录］。
显示碰触位置、计算的位置，F5［结束］有效。

11 按下 F5［结束］，

返回原来的设置画面。在托盘位置的 X、Y、R 中输入值。

6.5.5.2 ［DI］＋［通过视觉检出］的情况

1 使传送带停止。
2 按下 F5［托盘位置］。
则显示以下画面。

3 将托盘放在传送带最上游部，在托盘来到机器人前运转传送带。
光电管传感器检出托盘后，传送带只有触发偏移后，*i*RVision 自动检出工件。*i*RVision 检出托盘时无需停止传送带。如果 *i*RVision 检出成功，则显示检出位置，F5［下一步］有效。

4 按下 F5［下一步］。
则显示以下画面。

5 点动机器人，碰触托盘坐标系的原点。

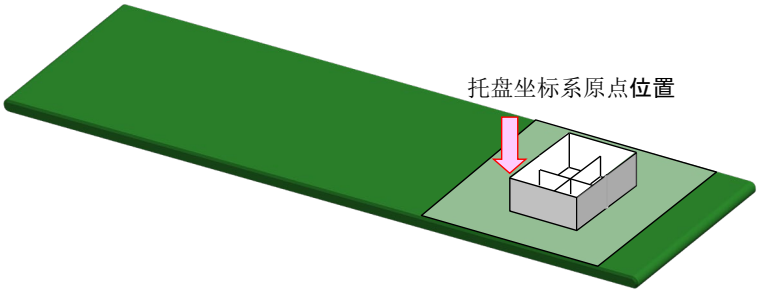


图 6.5.5.2(a)

6 按下 F4［记录］。
显示碰触时的编码器值、碰触位置，F5［下一步］有效。

7 按下 F5［下一步］。

则显示以下画面。

8 点动机器人，碰触托盘坐标系的 X 方向点。

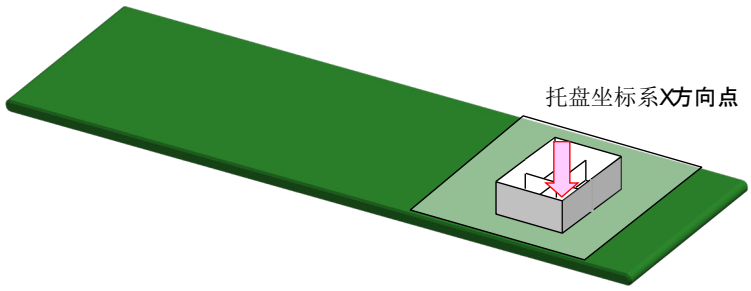


图 6.5.5.2(b)

9 按下 F4［记录］。
显示碰触位置、计算的位置，F5［结束］有效。

10 按下 F5［结束］，

返回原来的设置画面。在托盘位置的 X、Y、R 中输入值。

6.5.6 目标位置设定

触发条件和动作的组合，仅在［距离］+［放入工件］或［标记］+［放入工件］的情况下需要进行设定。
对生成工件的位置进行示教。每次传送带移动一定距离，或输入伺服传送带的驱动信号，该位置会被放入队列。这种组合的情况，由于不实际检出工件，在目标位置设定的同时，也进行机器人位置的示教。在传送带上使用与不使用托盘的情况，在步骤上略有不同。

注
需要在通过此传感器选择的控制装置中打开 iRPickTool 的设定画面，进行目标位置设定。

6.5.6.1 不使用托盘的情况

1 使传送带停止。
2 按下 F4［目标位置］。则显示以下画面。

3 在需要生成的位置放置工件。
利用全部的机器人，将工件放置于用于机器人位置的示教的最上游机器人动作领域的上游。
4 按下 F4［触发设置］。如果按下 F4［触发设置］，则显示现在的编码器值，F5［下一步］有效。

5 按下 F5［下一步］。

则显示以下画面。

6 使传送带动作，从最上游的机器人依次示教机器人的位置。详细内容请参照“6.11 机器人的位置示教”
但是如果工件来到该机器人前面，保持示教机器人时的位置不变，按下 F4［记录］。

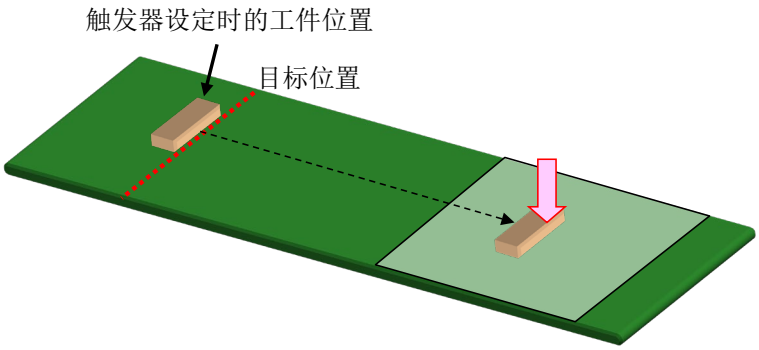


图 6.5.6.1(a)

显示碰触时的编码器值、碰触位置、计算的位置，F5［结束］有效。

7 按下 F5［结束］，

返回原来的设置画面。仅在目标位置的 X 输入计算所得的值(Y 和 R 保持为 0)。

8 进行该机器人的位置示教。

9 并使传送带动作，进行下游机器人的位置示教。

6.5.6.2 使用托盘的情况

1 使传送带停止。
2 按下 F4［目标位置］。则显示以下画面。

3 在需要生成的位置放置托盘。
利用全部的机器人，将托盘放置于用于机器人位置的设定的最上游机器人动作领域的上游。
4 按下 F4［触发设置］。如果按下 F4［触发设置］，则显示现在的编码器值，F5［下一步］有效。
5 按下 F5［下一步］。

则显示以下画面。

6 使传送带动作，从最上游的机器人依次示教机器人的位置。详细内容请参照“6.11 机器人的位置示教”
但是如果工件来到该机器人前面，首先点动机器人，碰触托盘坐标系的原点位置，按下 F4［记录］。

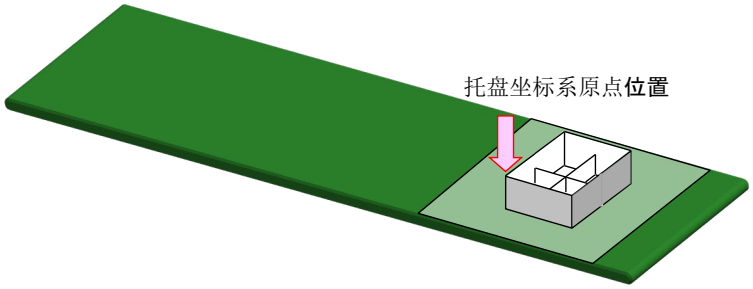


图 6.5.6.2(a)

7 显示碰触时的编码器值、碰触位置，F5［下一步］有效。

8　按下 F5［下一步］。

则显示以下画面。

9　点动机器人，碰触托盘坐标系的 X 方向点。

10　按下 F4［记录］。

显示碰触位置、计算的位置，F5［结束］有效。

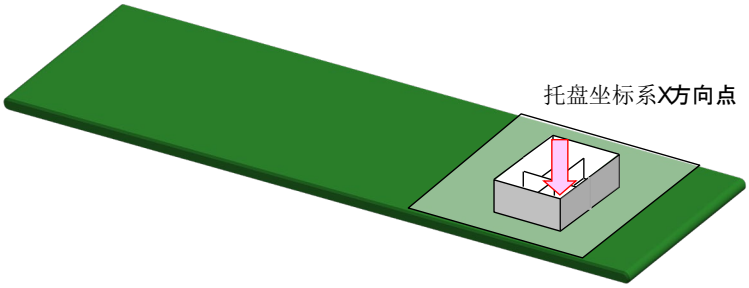


图 6.5.6.2(b)

11 按下 F5［结束］，

返回原来的设置画面。在目标位置的 X、Y、R 中输入值。

12 进行该机器人的位置示教。
13 并使传送带动作，进行下游机器人的位置示教。

6.5.7 可同时执行的传感器的限制

1 台机器人最多可同时执行 4 台传感器。
此外，存在满足以下所有条件的传感器的组时，不能启动传感器。请进行更改连接传感器的控制装置等处理。

- 连接各个传感器的控制装置相同
- 双方的传感器使用高精度触发 ※1
- 在各个传感器对应的传送带工作台上使用的编码器相同 ※2

※1 关于使用高精度触发的触发条件和动作组合，请参照表 6.5.7(a)。

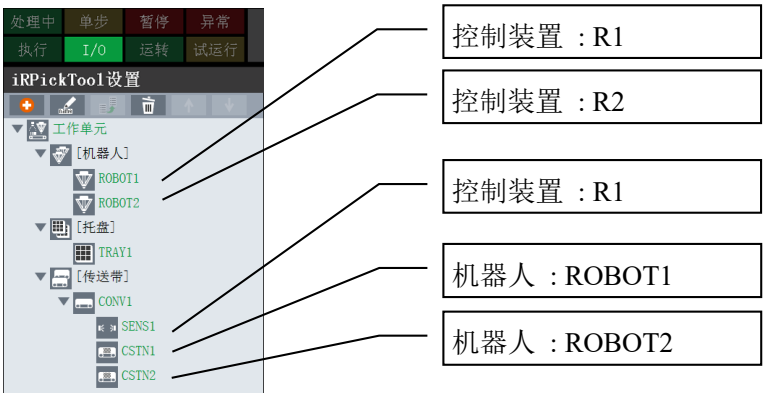
表 6.5.7(a) 使用高精度触发的触发条件和动作组合

［触发条件］	［触发］	使用高精度触发
［距离］	放入工件	×
	通过视觉检出	×
	程序的执行	○
［DI］	放入工件	○
	通过视觉检出	×
	程序的执行	○
［HDI］	放入工件	×
［标记］	放入工件	×
［RI］	放入工件	○
	通过视觉检出	×
	程序的执行	○

※2 传感器对应的传送带工作站是指满足以下所有条件的工作站。

- 属于与传感器相同的传送带
- 传送带工作站的机器人对应的控制装置和连接有传感器的控制装置相同

如果是下图的构成，SENS1 对应的传送带工作站为 CSTN1。



6.6 传送带工作站设定

进行各传送带工作站设定。
传送带工作站的设定画面如下。

⚠ 注意
如果追踪坐标系的设定不结束，则无法进行传送带工作站设置画面中的几乎所有设定项目。请先进行追踪坐标系的设定，再进行各传送带工作站的设定。

［机器人］

选择在该传送带工作站进行操作的机器人。

［伺服传送带编号］

用编号选择在该传送带工作站使用的伺服传送带设定。

伺服传送带的设定，不是通过 *i*RPickTool 的设置画面，而是从［MENU］键选择［设置］的［伺服传送带］，从伺服传送带的设置画面进行。详细信息，请参照“9.7 伺服传送带追踪拾取功能”。

［追踪参数设置］

选择在该传送带工作站使用的追踪参数设定表的编号。通常对每个传送带指定不同的追踪参数设定表的编号。

[编码器编号]

选择在该传送带工作站使用的编码器编号。

[编码器倍率]

显示设定的编码器的倍率。

[追踪坐标系]

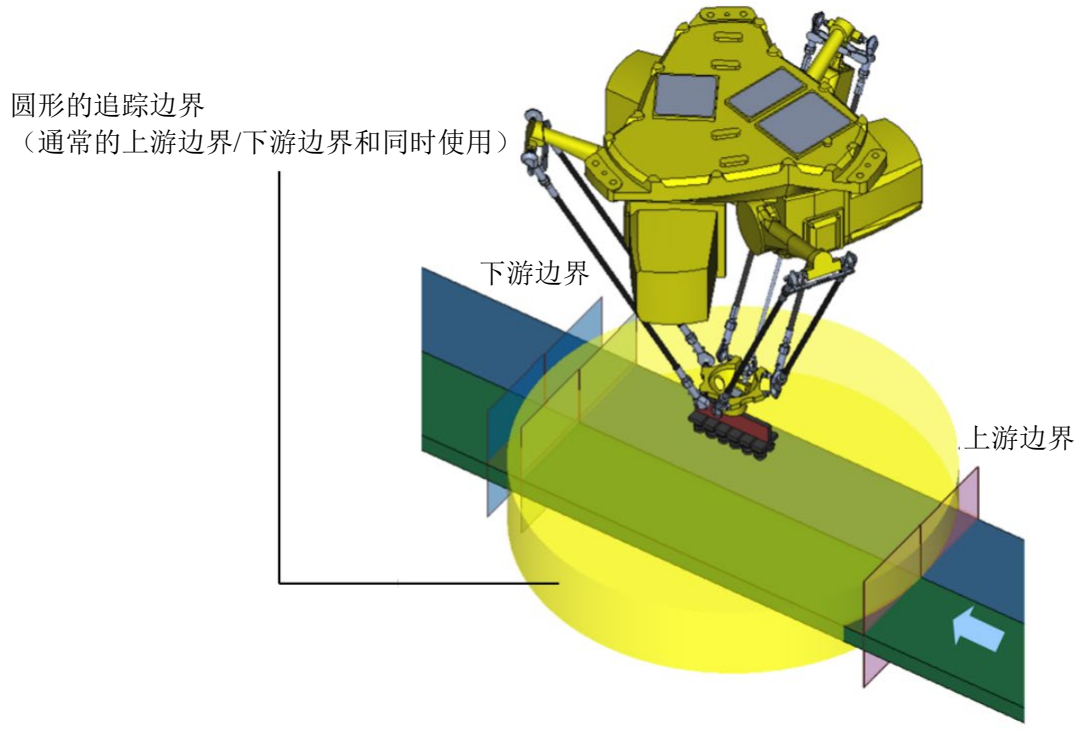
显示设定的追踪坐标系的状态和坐标值。

[选择的追踪范围]

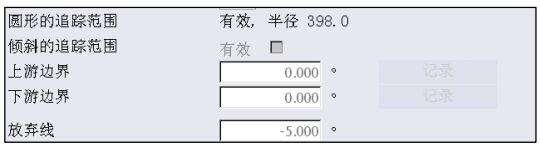
选择使用的追踪范围的编号。通常选择 1。

> **注**
>
> **通过 TP 程序更改使用的追踪范围时，使用追踪命令的 SELBOUND。详细内容请参照“Line Tracking OPERATOR'S MANUAL B-83474EN”中的“TRACKING INSTRUCTIONS”的项目。**
> **另外，追踪命令的 SELBOUND 导致的更改将自动反映到 *i*RPickTool 的设定画面。请暂时先关闭设定画面后再重新打开，或者更新树状图。**

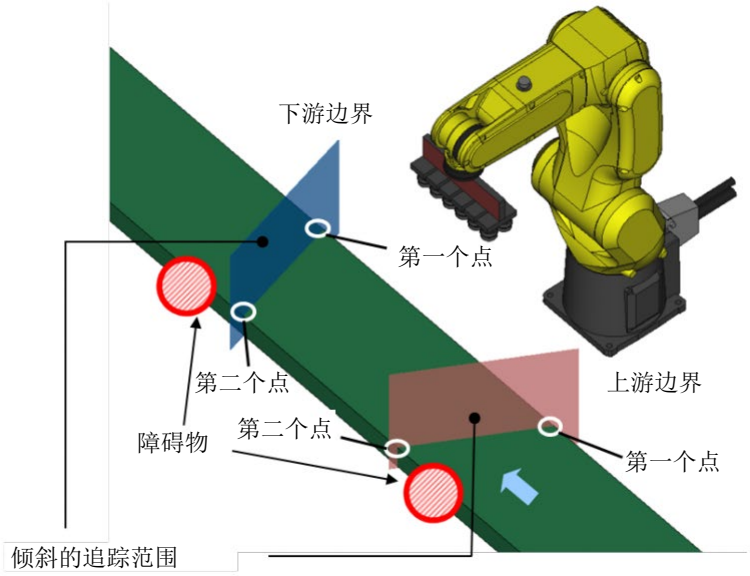
[圆形的追踪范围]



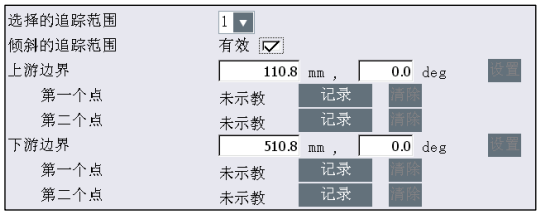
只有用选定的机器人启用［圆形追踪范围］时，才会显示台下［有效］与［半径］。



[倾斜的追踪范围]



设定倾斜的追踪范围时启用。
启用倾斜的追踪范围后，就可记录以下上游侧与下游侧的倾斜位置。



另外，启用圆弧追踪时将无法选择。

[第一个点]

碰触上游边界的第一个点及下游边界的第一个点，设定位置记录。

[第二个点]

碰触上游边界的第二个点及下游边界的第二个点，设定位置记录。

[记录]按钮
记录第一个点及第二个点的设定位置。按下[记录]按钮后，“未示教”变成“已示教”。

[清除]按钮
清除第一个点及第二个点的设定位置。

[上游边界]

指定追踪范围的上游的边线位置。[上游边界]是该机器人可能到达的最上游侧的位置。
上游边界、下游边界的设定，在选择工具坐标系的状态下进行，该工具坐标系是利用随着传送带进行动作的程序使用的。请参照“6.9.2.2 进行拾取动作的程序”。
在上游边界的边界位置点动 TCP，通过按下[记录]按钮，设定追踪坐标系的 X 坐标值。圆弧传送带的情况，设定旋转角度。也可通过手动输入更改值。

启用倾斜的追踪范围后，就可以设定第一点的倾斜范围与第二点的倾斜范围。

［下游边界］

指定追踪范围的下游的边线位置。［下游边界］是该机器人可能到达的最下游侧的位置。
在下游边界的边界位置点动 TCP，通过按下［记录］按钮，设定追踪坐标系的 X 坐标值。圆弧传送带的情况，设定旋转角度。也可通过手动输入更改值。

启用倾斜的追踪范围后，就可以设定第一点的倾斜范围与第二点的倾斜范围。

［设定］按钮

设定追踪范围（上游侧）及追踪范围（下游侧）的 mm 值与 deg 值。

［放弃线］

指定是否放弃传送带上送来的工件的判断位置。由于从追踪范围的下游侧边线(下游边界)的偏移量，通常设定为较小负数。这样，判断认为工件即将离开追踪范围时无法拾取。不能设定正数。

根据机器人对某工件开始追踪动作起，到对该工件的操作(拾取等)结束的时间平均和传送带的移动速度，通过以下要领决定放弃线的值。

线传送带的情况

机器人的动作时间约 0.8 s、传送带速度 100 mm/s 的情况下传送带的自走距离如下所示。0.8(s) ×100(mm/s) = 80(mm)
因此，要留一些余量，将跳过距离设为 –100 mm 左右。

圆弧传送带的情况

机器人的动作时间约 0.8 s、传送带速度 5 deg/s 的情况，
0.8(s) ×5(deg/s)= 4(deg)。
因此，留一些余量，将跳过距离设为 –5 度左右。

⚠ 注意
后述的停止/启动传送带有效时，即使工件越过放弃线，机器人在不超过下游侧追踪范围内进行取工件的操作。如上所述，传送带超过放弃线后停止，因此要在传送带停止后取工件，需机器人超过放弃线进行操作。

注
如果上游边界与放弃线间隔窄，则容易发生没有可分配至传送带工作站的工件的状况。如果没有可分配至传送带工作站的工件，则机器人等待至下一个工件越过上游边界。如果使上游边界与放弃线间始终至少有 1 个工件，则机器人可连续取工件。例如，以基本等间隔供给工件的情况，如果上游边界与放弃线的间隔设定得比工件的供给间隔更宽，则方便机器人连续取工件。

各项目的位置关系

［上游边界］、［下游边界］、［放弃线］的位置关系，总结如下图。

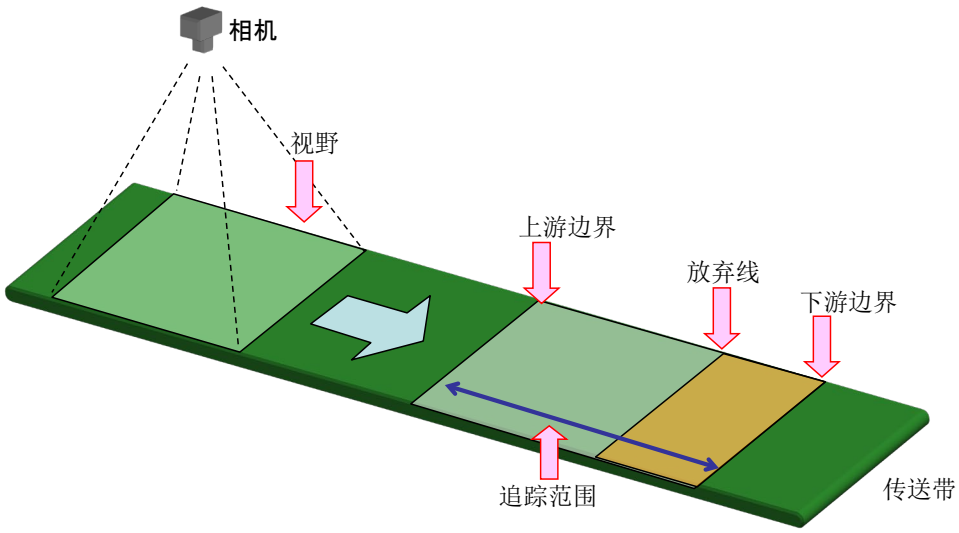


图 6.6(a)追踪范围与放弃线的位置关系

［根据 Y 排序］

沿着追踪坐标系的 Y 方向，即与传送带传送方向正交的方向依次排序有关传送带上传送的工件的信息，依次处理从传送带一端传送的工件的功能。
在［X 的范围］中，指定被作为相同列的传送带传送方向的范围。例如，如果在［X 的范围］指定 100mm，在距离最下游的工件 100mm 以内的工件中，找出 Y 在最小(或最大)位置的工件，根据该工件的信息，优先分配至机器人。

> **⚠ 注意**
> **以下情形，不能根据 Y 进行排序。**
> - **在传送带上设置托盘时**
> - **倾斜的追踪范围有效时**
> - **在机器人上，圆形的追踪范围有效时**

［停止/启动传送带］

工件越过放弃线时，自动使传送带停止的功能。如果机器人处理该工件，则传送带重新开始工作。

使用该功能时，指定用于停止/启动传送带的 DO 信号的编号，以及表示以信号开启停止还是信号关闭停止的极性。

如果该功能有效，放弃线的设定要领改变。指定从发出传送带停止信号起，到传送带实际停止的惰走距离。例如，传送带停止前惰走 150mm，则放弃线设定为－150 mm 左右的值。

［传送带重启线］

工件越过放弃线，传送带暂停，即使拾取相应的工件，在其它工件还处于此线的下游时，传送带不会重启。
因此，如果在放弃线附过有大量工件，即使重启，传送带也不会立即停止。为了防止出现这种状态，在传送带上设定重启线。
下例中，工件 A 越过传送带停止线(放弃线)，传送带停止，但是即使拾取工件 A，由于传送带重启线的下游还有工件 B，所以传送带不会重启。

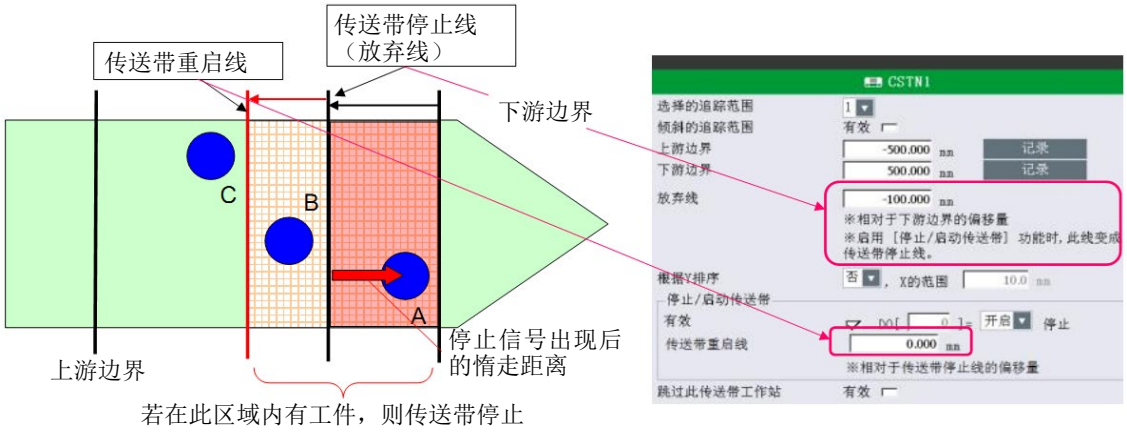


图 6.6(b)传送带重启线与放弃线的设定项目

> **⚠ 注意**
> **1 如果在相同机器人控制装置上定义的多个传送带工作站上，对停止/启动传送带指定相同 DO 编号，且反向设定 DO 的极性(即一个是以 DO＝开启停止传送带，另一个以 DO＝关闭停止传送带)，则停止/启动传送带不正确动作。**
> **2 停止/启动传送带有效时，请通过编码器的设定，使平均(更新次数)值为 10 以上。如果保持标准值为 1，在传送带停止或重新动作时，机器人会敏感地跟随传送带，可能发生“移动时误差过大”。**

［跳过此传送带工作站］

其有效后，该传送带工作站内不再输入工件的信息。在特地需要令对象机器人休息时使该功能有效。

［设置预整列］

根据条件不同，可能显示有关预整列功能的设定项目。

详细信息请参考“6.14 预整列设定”。

6.6.1 追踪坐标系的追加

对于已经在传送带的设定页完成追踪坐标系的设定的传送带，只对特定的传送带工作站重新设定追踪坐标系的设定。在以特定的机器人进行追踪坐标系设定时，所选择的工具坐标系错误的情况下，或新追加机器人的情况下使用。不能重新设定最上游的传送带工作站。重新设定的传送带工作站以外的传送带工作站中，追踪坐标系不变化。

根据传送带的追踪类型不同，设定内容也不同。

6.6.1.1 线传送带的情况

线传送带时的步骤。圆弧传送带的情况，请参照“6.6.1.2 圆弧传送带的情况”。

1 在传送带工作站的设置画面按下［追加追踪坐标系］。

则显示以下画面。

2 将可以机器人的 TCP 碰触的夹具放置于传送带的最上游部，按下 F5［下一步］。

注
1 在最上游的传送带工作站的设定页中，不显示［追加追踪坐标系］。
2 如果未设定最上游的传送带工作站的追踪坐标系，则［追加追踪坐标系］无效。这种情况，在传送带的设定页进行全部的传送带工作站的追踪坐标系设定。

则显示以下画面。

3 将以最上游的机器人在“5.4 工具坐标系的设定”设定的工具坐标系作为手动进给坐标系进行选择。
4 移动并固定传送带，使夹具进入最上游的机器人动作领域内。
5 点动移动机器人，用 TCP 触碰夹具。
6 在碰触后的状态下，按下 F5［下一步］。

则显示以下画面。第 1 行中，显示重新设定追踪坐标系的传送带工作站名。

7 在修改追踪坐标系设定的机器人中，将按“5.4 工具坐标系的设定”设定的工具坐标系选择为手动进给坐标系。
8 在重新设定追踪坐标系的传送带工作站中，与“3.2.4 追踪坐标系设定”中说明的 14～22 一样，碰触夹具。
于是，会显示如下画面，直至 Y 轴方向点的设定结束。
通过以上操作，追加追踪坐标系的作业结束，设定了与最上游机器人相同的编码器倍率以及新计算所得的追踪坐标系。

9　按下 F5［下一步］。

显示如下画面。

6.6.1.2 圆弧传送带的情况

圆弧传送带的设定步骤如下。

1 在传送带工作站的设置画面按下［追加追踪坐标系］。

则显示以下画面。

2 将可以机器人的 TCP 碰触的夹具放置于传送带的最上游部，按下 F5［下一步］。

注
1 在最上游的传送带工作站的设定页中，不显示［追加追踪坐标系］。
2 如果未设定最上游的传送带工作站的追踪坐标系，则［追加追踪坐标系］无效。这种情况，在传送带的设定页进行全部的传送带工作站的追踪坐标系设定。

则显示以下画面。

3 将以最上游的机器人在“5.4 工具坐标系的设定”设定的工具坐标系作为手动进给坐标系进行选择。

4 移动并固定传送带，使夹具进入最上游的机器人动作领域内。

5　点动移动机器人，用 TCP 触碰夹具。
6　在碰触后的状态下，按下 F5［下一步］。

则显示以下画面。第 1 行中，显示重新设定追踪坐标系的传送带工作站名。

7　在重新设定追踪坐标系的传送带工作站中，与“6.4.1 追踪坐标系设定”的“6.4.1.1 圆弧传送带的情况”中说明的步骤 4～13 一样，碰触夹具。
于是，会显示如下画面，直至辅助点的设定结束。
通过以上操作，追加追踪坐标系的作业结束，设定了与最上游机器人相同的编码器倍率以及新计算所得的追踪坐标系。

8　按下 F5［下一步］。

显示如下画面。

6.7 固定工作站设定

进行固定工作站的设定。

[机器人]

选择在该固定工作站进行操作的机器人。

[托盘]

选择放置于该固定工作站的托盘。此外，从以下选择初始状态。初始状态，具有在托盘内指定层的意思。

> **注意**
> **固定工作站中，即使只有 1 个放置位置，也需要指定托盘。**

空：

托盘为空状态下，投入队列。这种情况，机器人从下面的层开始放置工件。

满：

托盘已满的状态下，投入队列。这种情况，机器人从上面的层开始拾取工件。

并且，指定表示托盘坐标系的用户坐标系编号。需要对于该用户坐标系定义托盘。
用户坐标系的设定，不是通过 *i*RPickTool 的设置画面，而是从［MENU］键选择［设置］的［坐标系］，从坐标系的设置画面进行。详细信息请参照“操作说明书（基本操作篇）B-83284CM”的“机器人系统的设定”章、“设置坐标系”节、“设定用户坐标系”项。

> **注意**
> **可以在固定工作站上选择的用户坐标系编号为 1 至 9。**

[缓冲区]

作为缓冲区使用托盘时，选中［有效］。如果使之有效，将工件暂时放置于固定工作站，可以再次从固定工作站拾取该工件。如果供给量有差异，可能通过使用缓冲区，使生产量稳定。

[拾取顺序]

指定从缓冲区拾取工件的顺序。仅缓冲区有效时可以指定。

先进先出：

拾取缓冲区内最先放入的工件。

后进先出：

拾取缓冲区内最后放入的工件。

6.8 KAREL 程序

*i*RPickTool 的基本功能，由以下的 KAREL 程序提供。用 CALL 指令调出这些 KAREL 程序，实现各种功能。
本节对提供的 KAREL 程序的规格进行说明。

*i*RPickTool 中使用的 KAREL 程序，有以下规定。
- 开头的 2 个字符以［PK］打头。
- 之后的 2 个字符表示相关对象。
- 第 5 个字符起，表示该程序的功能。

6.8.1 有关工作单元的程序

工作单元相关程序打头的 4 个字符是［PKWC］。［PK］后面的 2 个字符是［WC］。(但是，部分除外。)

6.8.1.1 PKWCSTART.PC

系统启动时调出，并进行以下内容。最初的 2 个，进行与后面叙述的 PKWCCHECK 相同的处理。
- 确认工作单元内的控制装置间，无设定矛盾。
- 确认没有无效的设定。
- 清除对应调出的控制装置的所有传送带工作站内的工件信息。对于对应调出的控制装置的全部传送带工作站，进行与后面叙述的 PKCSCLRQUE 相同的处理。详细信息，请参照“6.8.5.2 PKCSCLRQUE.PC”。
- 开始对调出的控制装置具有的各传感器中所设定的触发条件的监视。进行与后面叙述的 PKSNSTART 相同的处理。详细信息，请参照“6.8.4.5 PKSNSTART.PC”。

无参数。

使用例：

```
CALL PKWCSTART
```

6.8.1.2 PKWCEND.PC

系统结束时调出，并进行以下内容。
- 停止对调出的控制装置具有的各传感器中所设定的触发条件的监视。进行与后面叙述的 PKSNSTOP 相同的处理。

无参数。

使用例：

```
CALL PKWCEND
```

6.8.1.3 PKWCCHECK.PC

确认工作单元内的控制装置间无设定矛盾，且没有无效设定。单独调出 PKCSCLRQUE 和 PKSNSTART 时，是代替对 PKWCSTART 的调出。

无参数。

使用例：

```
CALL PKWCCHECK
```

6.8.1.4 PKWCRSTPRDST.PC

复位 *i*RPickTool 的生产状况画面的显示。生产状况画面的详情，请参阅“9.2 确认生产状况”。

无参数。

使用例：

```
CALL PKWCRSTPRDST
```

6.8.1.5 PKGETVAR.PC

从数值寄存器取出指定的 KAREL 变量或系统变量的值。变量的指定，如下所示，可指定至 3 层(包括排列)。
var1_name
var1_name[var1_idx]
var1_name[var1_idx].var2_name[var2_idx].var3_name[var3_idx].

例如，系统变量 $pkcs[1].$n2pick[2]中，根据以上例，进行如下设定。
var1_name = “$ pkcs”
var1_idx = 1
var2_name = “$ n2pick”
var2_idx = 2
var3_name = “...”
var3_idx = 0

如果使用该程序，可取得客户自行生成的 KAREL 的变量。
此外，在无法通过字符数的限制进行证明时，也可使用 TP 程序的参数指令(例如：R[1] =$pkcs[1].$n2pick[2])。

参数 1：
以字符串指定具有需要取得的变量的程序名。
需要取得系统变量时，设定“*SYSTEM*”。

参数 2：
指定需要取得的变量的类型。
可以指定以下 3 种类型。
1: 整数型
2: 实数型
3: 布尔值(逻辑型)

参数 3：
指定最上位层的变量名(相当于上述例中的 var1_name)。
系统变量的情况，请注意以“$”打头。

参数 4：
设定最上位层的变量的排列编号(相当于上述例中的 var1_index)。
变量为非排列时，请指定 0。

参数 5：
指定第 2 层的变量名(相当于上述例中的 var2_name)。
如果没有第 2 层，则请指定“…”(空白)。
系统变量的情况，请注意以“$”打头。

参数 6：
设定第 2 层的变量的排列编号(相当于上述例中的 var2_index)。
变量为非排列时，或参数 5 为空白时，请指定 0。

参数 7：
指定第 3 层的变量名(相当于上述例中的 var3_name)。
如果没有第 3 层，则请指定“…”(空白)。
系统变量的情况，请注意以“$”打头。

参数 8：
设定第 3 层的变量的排列编号(相当于上述例中的 var3_index)。
变量为非排列时，或参数 7 为空白时，请指定 0。

参数 9：
指定存储指定变量的值的数值寄存器序号。

使用例 1：
将名为“PROGA”的 KAREL 程序的整数型变量“cycle[1].infeed[2].box_orient”的值存储于 R[95]。

```
CALL PKGETVAR(程序名称='PROGA',整数,变量名称 1='CYCLE',排列编号 1=1,变量名称 2='INFEED',
排列编号 2=2,变量名称 3='BOX_ORIENT',排列编号 3=0,数值 Reg=95)
```

使用例 2：
将名为“PROGA”的 KAREL 程序的实数型变量“box_rot”的值存储于 R[95]。

```
CALL PKGETVAR(程序名称='PROGA',实数,变量名称 1='BOX_ROT',排列编号 1=0,变量名称 2='...',
排列编号 2=0,变量名称 3='...',排列编号 3=0,数值 Reg=95)
```

使用例 3：
将整数型的系统变量“$pkcs[1].$n2pick[2]”的值存储于 R[95]。

```
CALL PKGETVAR(程序名称='*SYSTEM*',整数,变量名称 1='$PKCS',排列编号 1=1,变量名称 2='$N2PICK',
排列编号 2=2,变量名称 3='...',排列编号 3=0,数值 Reg=95)
```

6.8.1.6 PKSETVAR.PC

变更 KAREL 变量或系统变量被指定的值。除执行该程序的控制装置外，也可以变更以机器人间通信(RIPE)连接的控制装置的变量。

参数 1：
指定存在有需变更变量的控制装置名。
以机器人间通信的设定，定义控制装置名。变更执行该程序的控制装置的变量时，请指定“…”(空白)。

参数 2：
以字符串指定具有需要变更的变量的程序名。
需要取得系统变量时，设定“*SYSTEM*”。

参数 3：
指定需变更的变量名。（例：cycle[1].box_rot、$pkcs[1].$n2pick[2]、$pkclrq）
请注意系统变量时开头为“$”。
参数 4：
指定需变更的值。
变量类型为布尔值(逻辑型)时，OFF 或假时指定为 0，ON 或真时指定为 1。

参数 5：
指定变量类型是否是布尔值(逻辑型)。
如果是布尔值(逻辑型)，则指定为 1，不是时(整数型或实数型)，则指定为 0。

使用例 1：
将执行该程序的控制装置中的，名为“PROGA”的 KAREL 程序的实数型变量“cycle[1]. box_rot”变更为 R[95]的值。

```
CALL PKSETVAR(RIPE 名称='...',程序名称='PROGA',变量名称='CYCLE[1].BOX_ROT',变量数值=R[95],
布尔型=0)
```

使用例 2：

将通过机器人间通信的设定定义为“ROBOT2”的控制装置中，整数型的系统变量“$pkcs[1].$n2pick[1]”变更为R[95]的值。

```
CALL PKSETVAR(RIPE 名称='ROBOT2',程序名称='*SYSTEM*',变量名称='$PKCS[1].$N2PICK[1]',
变量数值=R[95],布尔型=0)
```

6.8.1.7 PKPRGRGETEN.PC

返还可否在此控制装置上使用预整列功能。

参数 1：
指定存储可否使用预整列功能的值之寄存器序号。
可以使用时返还 1，不可使用时返还 0。

使用例：
将执行此程序的控制装置上可否使用预整列功能存储在 R［7］中。

```
CALL PKPRGRGETEN(预整列有效 Reg=7)
```

6.8.1.8 PKRSTWC.PC

将 *i*RPickTool 的设定数据重置为出厂时的设定。此程序只有在通过 *i*RPickTool 的树状图视图示教的传送带、传送带工作站、固定工作站、传感器等所有设定数据被删除时才被调出。

- 只删除执行本程序的机器人控制装置的数据。有多台机器人控制装置，且想要删除所有控制装置的数据时，需要在各机器人控制装置执行该程序
- 仅删除 *i*RPickTool 的设定数据。示教的追踪坐标系及位置寄存器不会被删除和重置。
- 方案不被删除。

无参数。

使用例：

```
CALL PKRSTWC
```

6.8.1.9 PKABO.PC

工作单元内存在的各机器人控制装置的所有 *i*RPickTool 程序（任务）的执行将一度中断。此程序用于应用程序中检出致命错误，要中止所有机器人的程序的情况等。此外，应用程序试运行或调试过程中也可以使用。

- 为了中断在树状图视图中定义的所有机器人控制装置的所有 *i*RPickTool 任务，仅用一台机器人控制装置调用 PKABO。
- 此程序使用以太网与其他机器人控制装置通信。因此，存在离线的机器人控制装置时，不能中断该机器人控制装置的任务。

无参数。

使用例：

```
CALL PKABO
```

6.8.1.10 PKLABO.PC

执行此程序的机器人控制装置中正在执行的任务全部中断。

无参数。

使用例：

```
CALL PKLABO
```

6.8.1.11 PKPOSTERR.PC

按照指定的轻重程度（警告、暂时停止、强制结束）定位错误。

参数 1：
指定错误编号。

参数 2：
指定错误轻重程度。（警告、暂时停止、强制结束）

注
若已经显示参数的说明，则作为示教时的选项，显示
1 警告
2 暂时停止
3 强制结束
内部设定为 0：警告/1：暂时停止/2：强制结束。
没有显示参数的说明时，“警告”时请指定为 0，“暂时停止”时请指定为 1，“强制结束”时请指定为 2。

使用例：
将 IRPK-232“不正确的位置寄存器”定位为暂时停止的程度。

```
CALL PKPOSTERR(127232,暂时停止)
```

6.8.1.12 PKCOPYPR2POS.PC

将位置寄存器的值复制到 TP 程序的位置变量。

参数 1：
指定复制目标的位置寄存器的编号。

参数 2：
指定 TP 程序名。

参数 3：
指定复制目标的位置变量的编号。

参数 4：
指定组编号。

使用例：
关于组 1，将 PR［11］的值复制到 MAIN1.TP 的 P［2］。

```
CALL PKCOPYPR2POS(位置 PR=11,程序名称='MAIN1',位置数据编号=2,组编号=1)
```

6.8.2 有关机器人的程序

机器人相关程序打头的 4 个字符是［PKRB］。［PK］后面的 2 个字符是［RB］。

6.8.2.1 PKRBGETID.PC

从组编号中取得机器人的 ID。

参数 1：
指定组编号。

参数 2：
指定保存有参数 1 指定的组编号对应的机器人的 ID 的数值寄存器。

使用例：
组 1 的机器人 ID 在 R［7］中取得。

```
CALL PKRBGETID(组编号=1,机器人 ID Reg=7)
```

6.8.2.2 PKRBGETSFLG.PC

取得在树状图视图的［机器人］中设定的［跳过动作实施范围］的［标记编号］。

参数 1：
指定机器人的 ID。

参数 2：
指定存储［跳过动作实施范围］的［标记编号］的值的数值寄存器编号。

使用例：
ID 为 1 的机器人的［跳过动作实施范围］的［标记编号］的值在 R［7］中取得。

```
CALL PKRBGETSFLG(机器人 ID=1,跳过动作 Reg=7)
```

6.8.3 有关机器人的程序

传送带相关程序打头的 4 个字符是［PKCV］。［PK］后面的 2 个字符是［CV］。

6.8.3.1 PKCVSETLBD.PC

变更传送带上各传送带工作站的负载平衡。
即时反映设定。
有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定传送带的名称。

参数 2：
指定存储有负载平衡的数值寄存器序号。
存储于 R［数值寄存器序号］的值，用于相对最上游传送带工作站的负载平衡。存储于 R［数值寄存器序号＋1］的值，用于相对第 2 传送带工作站的负载平衡。数值寄存器中可设定的值的范围为 0~255。
参数 3：
指定设定负载平衡的传送带工作站的数量。
通常情况下，与传送带上设定的传送带工作站的数量相同。例如，传送带上有 3 个传送带工作站，如果在这里指定 3，则可以对全部的传送带工作站设定负载平衡。

参数 4：
设定［跳过］的值。

参数 5（可省略）：
指定设定负载平衡的模型 ID。
负载平衡的模式是［所有模型 ID 共通］的情况，不进行指定。

使用例：

将相对传送带数据 CONV1 的负载平衡，从最上游的传送带工作站设定于 1:2:3。使［跳过］的值为 0。模型 ID 不进行指定。使用的数值寄存器是 R[11]、R[12]、R[13]。预先使 R[11]= 1、R[12]= 2、R[13]= 3。

```
CALL PKCVSETLBD('CONV1',起始 Reg=11,传送带工作站数=3,跳过=0)
```

⚠ 注意

1 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

2 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.3.2 PKCVSETTRNAM.PC

变更设定于传送带的托盘。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：

以字符串指定传送带的名称。

参数 2：

以字符串指定托盘的名称。

使用例：

将“CONV1”的托盘变更为“TRAY2”。

```
CALL PKCVSETTRNAM('CONV1','TRAY2')
```

⚠ 注意

1 要正常反映设定内容，通常需要调出在主程序开头运行的 KAREL 程序 PKWCSTART，或 PKSNSTART 之前调出这些 KAREL 程序。

2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.3.3 PKCVENBLB.PC

切换负载平衡功能的有效 / 无效。即时反映设定。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定传送带的名称。

参数 2：
为有效时指定“启用”或为无效指定“禁用”。

注
若已经显示参数的说明，则作为示教时的选项，显示
1 启用
2 禁用
内部设定为 1：有效 / 0: 无效。
没有显示参数的说明时，“启用”时指定 1，“禁用”时指定 0。

使用例：
通过传送带数据 CONV1，使负载平衡有效。

```
CALL PKCVENBLB ('CONV1',启用)
```

⚠ 注意
1 对参数 2 指定 1，将负载平衡切换为有效时，请注意调出该 KAREL 程序的时机。如果工件传送频率高，有超过放弃线的工件，需要注意。在刚将负载平衡切换为有效后，有工件超过放弃线，则判断为跳过最初的工件，会发生本来应每隔 1 个拾取，但是连续拾取 2 个的情况。为了防止这种情况，尽量在未处理工件在上游侧的时候，例如刚调出 PKCSGETQUE 之后，调出本 KAREL 程序。
2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。
3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。
- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.3.4 PKCVENBSC.PC

在指定的传送带工作站，切换停止/启动传送带功能的有效 / 无效。即时反映设定。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定传送带的名称。

参数 2：
指定存储有有效 1 或无效 0 的数值寄存器序号。
存储于 R［数值寄存器序号］的值，用于切换相对最上游传送带工作站的停止/启动传送带的有效 / 无效。存储于 R［数值寄存器序号+1］的值，用于切换相对第 2 传送带工作站的停止/启动传送带的有效 / 无效。

参数 3：
指定切换停止/ 启动传送带的有效 / 无效的传送带工作站的数量。
通常情况下，与传送带上设定的传送带工作站的数量相同。
例如，传送带上有 3 个传送带工作站，如果在这里指定 3，则可以对全部的传送带工作站进行停止/启动传送带的有效 / 无效的切换。

使用例：
对传送带数据 CONV1 的传送带工作站的停止/启动传送带进行以下切换。
最上游的传送带工作站，停止/启动传送带有效。第 2 传送带工作站，停止/启动传送带无效。第 3 传送带工作站，停止/启动传送带有效。使用的数值寄存器是 R[11]、R[12]、R[13]。预先使 R[11]= 1、R[12]= 0、R[13]= 1。

```
CALL PKCVENBSC('CONV1',起始 Reg 编号=11,传送带工作站数=3)
```

⚠ 注意

1 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

2 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.3.5 PKCVGETNAME.PC

树状图视图中定义的传送带的名称从传送带的 ID 中取得。名称被返回到字符串寄存器。

参数 1：
指定传送带的 ID。

参数 2：
指定存储传送带名称的字符串寄存器编号。

使用例：
ID 为 1 的传送带的名称在字符串寄存器［10］中取得。

```
CALL PKCVGETNAME(传送带 ID=1,传送带名称 SR=10)
```

6.8.4 有关传感器的程序

传感器相关程序打头的 4 个字符是［PKSN］。［PK］后面的 2 个字符是［SN］。

6.8.4.1 PKSNENBSENS.PC

切换传感器的有效 / 无效。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定传感器的名称。

参数 2：
为有效时指定“启用”或为无效指定“禁用”。

注
若已经显示参数的说明，则作为示教时的选项，显示
1 启用
2 禁用
内部设定为 1：有效 / 0: 无效。
没有显示参数的说明时，“启用”时指定 1，“禁用”时指定 0。

使用例：
将“SENS1”置于禁用，将“SENS2”置于启用后启动传感器。

```
CALL PKSNENBSENS('SENS1',禁用)
CALL PKSNENBSENS('SENS2',启用)

CALL PKWCSTART
```

⚠ 注意
1 通过 PKWCSTART 和 PKSNSTART 启动传感器后，切换传感器的启用/禁用时，在启动传感器的时刻全部的传感器都必须处在启用状态。
2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。
3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。
- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.4.2 PKSNSETVPNAM.PC

变更传感器中设定的视觉程序。通过［使用多视图检出］也可以更改传感器中设定的视觉程序。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定传感器的名称。

参数 2：
以字符串指定视觉程序的名称。

使用例：
将“SENS1”的视觉程序变更为“VT2”。

```
CALL PKSNSETVPNAM('SENS1',视觉程序='VT2')
```

⚠ 注意

1 要正常反映设定内容，通常需要调出在主程序开头运行的 KAREL 程序 PKWCSTART，或 PKSNSTART 之前调出这些 KAREL 程序。

2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

注

对参数 2 指定字符串寄存器时，请根据以下步骤进行。

1 给参数输入适当的字符串。

2 将光标放在输入的字符串上，按下 F4［选择］。

3 选择［字符串寄存器］。

6.8.4.3 PKSNSETVTOFS.PC

［DI］＋［通过视觉检出］时，变更传感器中所设定的触发偏移。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：

以字符串指定传感器的名称。

参数 2：

指定触发偏移值。

使用例：

将“SENS1”的触发偏移变更为 100mm。

```
CALL PKSNSETVTOFS('SENS1',触发偏移=100)
```

⚠ 注意

1 要正常反映设定内容，通常需要调出在主程序开头运行的 KAREL 程序 PKWCSTART，或 PKSNSTART 之前调出这些 KAREL 程序。

2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6

6.8.4.4 PKSNSETMTOFS.PC

［DI］＋［使用多视图检出］时，变更传感器中所设定的触发偏移。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定传感器的名称。

参数 2：
指定视图 1 用的触发偏移值。

参数 3：
指定视图 2 用的触发偏移值。

使用例：
将“SENS1”的视图 1 的触发偏移变更为 100mm，将视图 2 的触发偏移变更为 200mm。

```
CALL PKSNSETMTOFS('SENS1',触发偏移 1=100,触发偏移 2=200)
```

⚠ 注意

1 要正常反映设定内容，通常需要调出在主程序开头运行的 KAREL 程序 PKWCSTART，或 PKSNSTART 之前调出这些 KAREL 程序。

2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.4.5 PKSNSTART.PC

开始对调出该 KAREL 程序的控制装置具有的各传感器中，所设定的触发条件进行监视。如果满足触发条件，则自动执行设定的动作。
如果以没有传感器的控制装置运行该 KAREL 程序，则发生 IRPK-031 的警告，但仅是信息，不会影响动作。
调出 PKWCSTART 时，不需使用。

无参数。

使用例：

```
CALL PKSNSTART
```

6.8.4.6 PKSNSTOP.PC

停止对调出该 KAREL 程序的控制装置具有的各传感器中，所设定的触发条件进行监视。
调出 PKWCEND 时，不需使用。

无参数。

使用例：

```
CALL PKSNSTOP
```

6.8.5 有关传送带工作站的程序

传送带工作站相关程序打头的 4 个字符是［PKCS］。［PK］后面的 2 个字符是［CS］。

6.8.5.1 PKCSGETID.PC

在指定的传送带中，取得调出该 KAREL 程序的机器人所属的传送带工作站的 ID。

参数 1：
指定传送带名。

参数 2：
指定存储传送带工作站的 ID 的数值寄存器序号。

参数 3（可省略）：
指定组编号。被省略时，取得组编号为 1 的传送带工作站的 ID。

> **⚠ 注意**
> **参数 1 中，不指定传送带工作站名，而指定传送带名。**

如果这些参数无效，在运行程序时，状态窗口会显示以下信息，程序可能结束。各种原因和对策如下。

Bad Conveyor Name
在指定的传送带的子对象中，不存在调用此程序的控制装置以及选择了指定组时该组编号对应的机器人的传送带工作站。

内部错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。
使用例：
运行该 KAREL 程序的机器人所属的传送带工作站 1，属于传送带 CONV1。将传送带工作站 1 的 ID 存储于 R[11]。机器人是组 1，因此省略参数。

```
CALL PKCSGETID('CONV1',工作站 ID Reg=11)
```

6.8.5.2 PKCSCLRQUE.PC

清除指定的传送带工作站内的工件信息。此时，传送带工作站内的工件的信息全部被消除。通常，在系统启动时，仅调出 1 次。调出 PKWCSTART 时，不需使用。

参数 1：
指定传送带工作站的 ID。可通过前述的 PKCSGETID 取得 ID。

使用例：
清除传送带工作站 1 的工件信息。ID 存储于 R[11]中。

```
CALL PKCSCLRQUE(传送带工作站 ID=R[11:传送带工作站 1 ID])
```


6.8.5.3 PKCSGETQUE.PC

从传送带工作站取得 1 个工件信息。取得的工件信息存储于视觉寄存器。
此外，如果拾取的工件在追踪范围内，则将该编码器的值设定为追踪动作的触发器。仅待机指定的时间，直至取得的工件传送至追踪范围。

参数 1：
指定传送带工作站的 ID。可通过前述的 PKCSGETID 取得 ID。

参数 2：
通常指定为 1。
与对通过传送带的设定指定的传送带工作站的分配比率无关，需要连续分配工件信息时，指定 2 以上的值。详细信息，请参照“6.9.2.2 进行拾取动作的程序”。

参数 3：
以 ms 指定取得的工件传送至追踪范围之前的待机时间。
如果指定负数，则无限等待。如果指定为 0，则无需待机，进入下一个处理。

参数 4：
指定存储取得的工件信息的视觉寄存器序号。
如果在传送至追踪范围内之前超时，也是将信息存储于视觉寄存器。

参数 5：
指定存储处理状态的数值寄存器序号。
存储的状态如下。
0： 分配追踪范围内的工件的信息，设定追踪动作的触发器。
1： 取得的工件未在时间内传送至追踪范围。
2： 完全未能取得工件。

参数 6（可省略）：
指定特定模型 ID，想要取得工件的信息时，指定模型 ID。
可以省略，通常不指定。在“8 不同用途的设定例子”中介绍使用例。

如果这些参数无效，在运行程序时，状态窗口会显示“P_CSGETQUE error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。
P_CSGETQUE 错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：

取得传送带工作站 1 的未处理的工件信息。

ID 存储于 R[11]中，VR[1]中是工件的信息，R[2]中是处理状态。此外，如果工件未到达追踪范围，则等待 100ms。

```
CALL PKCSGETQUE(传送带工作站 ID=R[11:传送带工作站 1 ID],连续的标记=1,超时值(ms)=100,
补偿用 VR=1,状态 Reg=2)
```

⚠ 注意

1 如果取得的工件信息是托盘的各单元，则基于单元 1 和各单元的相对位置，计算工件信息。因此，相对托盘的机器人位置的示教，必须在单元 1 的位置保持一致。

2 处理的状态为 1 时，并非表示实际工件的信息已被分配。即使立即调出下一个 PKCSGETQUE，也有可能取得不同的工件信息。此外，
处理的状态为 1 时，不能通过后面叙述的 PKCSACKQUE，通知处理结果。

6.8.5.4 PKCSGETQCELL.PC

从传送带工作站取得 1 个指定了托盘的序号的单元信息。即使是应事先处理的单元为未处理状态的情况，也可以取得信息。取得的单元信息存储于视觉寄存器。
此外，如果拾取的单元在追踪范围内，则将该编码器的值设定为追踪动作的触发器。仅待机指定的时间，直至取得的单元传送至追踪范围。

参数 1：

指定传送带工作站的 ID。可通过前述的 PKCSGETID 取得 ID。

参数 2：

指定单元的序号。

参数 3：

以 ms 指定取得的单元传送至追踪范围之前的待机时间。
如果指定负数，则无限等待。如果指定为 0，则无需待机，进入下一个处理。

参数 4：

指定存储取得的单元信息的视觉寄存器序号。

如果在传送至追踪范围内之前超时，也是将信息存储于视觉寄存器。

参数 5：

指定存储处理状态的数值寄存器序号。

存储的状态如下。

0： 分配追踪范围内的单元的信息，设定追踪动作的触发器。
1： 取得的单元未在时间内传送至追踪范围。
2： 完全未能取得单元。

参数 6：

指定在指定序号的单元之前，存储应处理而未处理的单元数量的数值寄存器序号。

如果这些参数无效，在运行程序时，状态窗口会显示“pkcsgetqcell error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

pkcsgetqcell 错误 17030：

指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：

从传送带工作站 1 取得单元 5 的信息。ID 存储于 R[11]中，VR[1]中是工件的信息，R[2]中是处理状态，R[6]中是在处理单元 5 之前，应处理而未处理的单元的数量。此外，如果工件未到达追踪范围，则等待 100ms。

CALL PKCSGETQCELL(传送带工作站 ID=R[11:传送带工作站 1 ID],单元编号=5,超时值(ms)=100,补偿用 VR=1, 状态 Reg=2,空单元数 Reg=6)

⚠ 注意
1 取得的单元信息是基于单元 1 和各单元的相对位置计算所得。因此，相对托盘的机器人位置的示教，必须在单元 1 的位置保持一致。
2 处理的状态为 1 时，并非表示实际单元的信息已被分配。即使立即调出下一个 PKCSGETQCELL，也有可能取得不同托盘的单元信息。此外，处理的状态为 1 时，不能通过以下说明的 PKCSACKQUE，通知处理结果。

6.8.5.5 PKCSACKQUE.PC

以 PKCSGETQUE 或 PKCSGETQCELL 处理的状态为 0，分配工件信息后，将对该工件进行的处理通知传送带工作站。

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
以下面的任意值，指定处理的结果。
1： 成功
正常处理工件(拾取或放置于托盘的单元中等)的情况下进行指定。
2： 返回
不立即处理工件，暂时返回传送带工作站的情况下进行指定。
3： 发信
不依照预定处理工件的情况下进行指定。
例如，使用机器人的 TP 程序评价被分配的工件的位置、模型 ID 等，故意阻止对该工件进行处理的情况，属于这种情况。这种情况，该工件的信息被转移至下一个传送带工作站。
4： 删除
工件的处理失败的情况下，进行指定。
例如：拾取工件后，发生了跌落等，通过吸附确认，结果为 NG 的情况，属于这种情况。这种情况，由于机器人使工件移动了，即使将该工件的信息返回传送带工作站，下游机器人也不能正常处理该工件。因此，工件的信息被删除。

通过传送带工作站的设定，停止/启动传送带为有效，传送带来不及进行工件的处理而停止时，处理的结果是 2 以外，执行 PKCSACKQUE 后，传送带开始动作。

使用例：
返回工件正常处理。ID 存储于 R[11]。

CALL PKCSACKQUE(传送带工作站 ID=R[11:传送带工作站 1 ID],成功)

6.8.5.6 PKCSPUTQUE.PC

向传送带工作站放入工件信息。通常不使用。不使用 *i*RPickTool 提供的传感器功能，以独立的程序将工件信息投入传送带工作站时使用。

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
指定存储工件信息的视觉寄存器序号。

参数 3：
指定存储低于同一物检查的结果的数值寄存器序号。没有对同一工件，被投入信息时，指定的数值寄存器中存储 0，存在有同一工件，未投入信息时，则指定的数值寄存器中存储 1。

如果这些参数无效，在运行程序时，状态窗口会显示“P_CSPUTQUE error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

P_CSPUTQUE 错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：
向传送带工作站 1 投入工件信息。
ID 存储于 R[11]，投入的工件信息存储于 VR[3]。处理状态返回 R[21]。

```
CALL PKCSPUTQUE(传送带工作站 ID=R[11:传送带工作站 1 ID],补偿用 VR=3,重叠的状态 Reg=21)
```

6.8.5.7 PKCSGETTIME.PC

以指定顺序取得的工件，返回直至到达上游边界所需的预估时间。预估时间的单位为 ms。时间的计算，使用调出该函数时的传送带速度。如果工件还未超越上游边界，则存储正值，如果已超越，则存储负值。传送带停止时，如果工件位于比上游边界更上游的位置，则返回 2147483646。如果工件位于比上游边界更下游的位置，则返回-2147483647。此外，取得了按指定顺序取得的工件的信息。
从向以 PKCSGETQUE 处理的状态返回 0，直至调出 PKCSACKQUE，不能使用本 KAREL 程序。发出“GETQUE 之后未执行 ACKQUE”的错误。本 KAREL 程序不适用圆弧传送带。

注
以下情形，不可使用 PKCSGETTIME.PC。
- **在传送带上，在［追踪类型］中设置［圆弧］时**
- **在传送带上使用［间隙检查］时**

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
指定顺序。
例如，如果指定 3，则返回有关以之后的第 3 次 PKCSGETQUE 取得的工件的预估时间。由于是基于此时点的队列信息，因此当发生工件的跳过，或检出新工件时，顺序可能变化。

参数 3：
通常指定为 1。
如果使用取多个的机械手，在用机械手连续拾取能抓住的工件量时，指定下一次拾取第几个工件。例如，用能取 4 个的机械手抓住第 2 个工件时，指定为 3。

参数 4：
通常指定为 1。
如果使用取多个的机械手，在用机械手连续拾取能抓住的工件量时，指定能用机械手抓住的工件的个数。例如，对于能取 4 个的机械手，指定为 4。

参数 5：
指定存储预估时间的数值寄存器序号。

参数 6：
指定存储工件信息的视觉寄存器序号。

参数 7：
指定存储处理状态的数值寄存器序号。
存储的状态如下。
0： 传送带动作，计算工件到达的预估时间的情况。工件信息存储于指定的视觉寄存器。

1： 传送带停止，工件位于比上游边界更下游的位置时。对工件的到达预估时间返回-2147483647，工件信息存储于指定的视觉寄存器。
2： 传送带停止，工件位于比上游边界更上游的位置时。对工件的到达预估时间返回 2147483646，工件信息存储于指定的视觉寄存器。
3： 队列中不存在指定条件相应的工件的情况。对于存储预估时间的数值寄存器，以及存储工件信息的视觉寄存器，不能写入值。

参数 8（可省略）：
指定特定模型 ID，想要了解工件的预估时间时，指定模型 ID。
可以省略，通常不指定。
如果这些参数无效，在运行程序时，状态窗口会显示“pkcsgettime error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

Pkcsgettime 错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：
传送带工作站 1 的 ID 存储于 R[11]，以能取 4 个的机械手只抓住 1 个工件的状态下(也就是说，下一次拾取第 2 个工件的情况)，将第 3 个取得的工件的到达预估时间返回 R[13]，工件信息返回 VR[5]，处理的状态返回 R[12]。

CALL PKCSGETTIME(传送带工作站 ID=R[11:传送带工作站 1 ID],顺序=3,连续的标记=2,数量=4,时间 Reg=13, 补偿用 VR=5,状态 Reg=12)

6.8.5.8 PKCSGETDIST.PC

返回从以指定顺序取得的工件至上游边界的距离。距离的单位是毫米。此外，取得了按指定顺序取得的工件的信息。从向以 PKCSGETQUE 处理的状态返回 0，直至调出 PKCSACKQUE，不能使用本 KAREL 程序。发出“GETQUE 之后未执行 ACKQUE”的错误。本 KAREL 程序不适用圆弧传送带。

注
以下情形，不可使用 PKCSGETDIST.PC。
- **在传送带上，在［追踪类型］中设置［圆弧］时**
- **在传送带上使用［间隙检查］时**

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
指定顺序。
例如，如果指定 3，则返回有关以之后的第 3 次 PKCSGETQUE 取得的工件的距离。由于是基于此时点的队列信息，因此当发生工件的跳过，或检出新工件时，顺序可能变化。

参数 3：
通常指定为 1。
如果使用取多个的机械手，在用机械手连续拾取能抓住的工件量时，指定下一次拾取第几个工件。
例如，用能取 4 个的机械手抓住第 2 个工件时，指定为 3。

参数 4：
通常指定为 1。
如果使用取多个的机械手，在用机械手连续拾取能抓住的工件量时，指定能用机械手抓住的工件的个数。例如，对于能取 4 个的机械手，指定为 4。

参数 5：
指定存储至上游边界的距离的数值寄存器序号。
符号较上游边界更靠上游时为正，更高下游时为负。

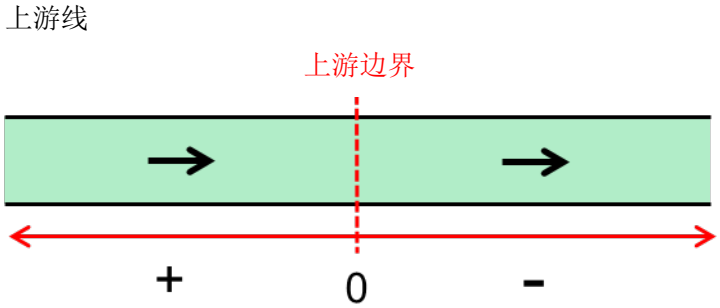


图 6.8.5.8 (a)

参数 6：
指定存储工件信息的视觉寄存器序号。

参数 7：
指定存储处理状态的数值寄存器序号。
存储的状态如下。
0： 队列中存在有指定条件相应的工件的情况。至上游边界的距离存储于指定数值寄存器中，工件信息存储于指定的视觉寄存器中。
1： 队列中不存在指定条件相应的工件的情况。对于存储距离的寄存器，以及存储工件信息的视觉寄存器，不能写入值。

参数 8（可省略）：
指定特定的模型 ID，想要了解工件的预估距离时，指定模型 ID。
可以省略，通常不指定。

如果这些参数无效，在运行程序时，状态窗口会显示“pkcsgetdist error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

Pkcsgetdist 错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：
传送带工作站 1 的 ID 存储于 R[11]，以能取 4 个的机械手只抓住 1 个工件的状态下(也就是说，下一次拾取第 2 个工件的情况)，将从第 3 个拾取的工件至上游边界的距离返回 R[15]，工件信息返回 VR[6]，处理的状态返回 R[14]。

```
CALL PKCSGETDIST(传送带工作站 ID=R[11:传送带工作站 1 ID],顺序=3,连续的标记=2,数量=4,距离 Reg=15,
补偿用 VR=6,状态 Reg=14)
```

6.8.5.9 PKCSGETPFRT.PC

返回在指定的传送带工作站刚调出 PKCSGETPFRT 之后，被处理工件的每分钟的处理数量(预计搬送率)。预计搬送率，以下面式子进行计算。

预计搬送率(个/min) = 预计搬送数量(个) x 传送带速度(mm/s) x 60(s/min)÷对应传送带工作站的距离(mm)

这里，“预计搬送数量”、“对应传送带工作站的距离”，分别如以下进行计算。

各传送带工作站中，规定有计算使用的范围，仅使用该范围内的工件计算预计搬送数量。该范围的下游侧，是该传送带工作站的放弃线。该范围的上游侧，是根据传送带工作站的以下 3 项之一。
1 最上游的传送带工作站的情况，传送带工作站在保持有 1 个以上工件的状态下，第一次调出 PKCSGETPFRT 时，传送带工作站所保持的工件的检出位置中，最上游侧的检出位置。
2 不是最上游的传送带工作站，［根据 Y 排序］无效的情况，是 1 个上游侧的传送带工作站的放弃线。

3 不是最上游的传送带工作站，［根据 Y 排序］有效的情况，是从 1 个上游侧的传送带工作站的放弃线，在［根据 Y 排序］中设定的仅［X 的范围］的下游侧位置。

预计搬送数量

计算所使用的范围所包括的工件中，基于传送带的负载平衡的设定、传送带工作站的［根据 Y 排序］的设定、托盘的层和初始状态的设定、以及现在的实绩，预计以该传送带工作站处理的放置数，称为预计搬送数量。无论机器人的搬运能力如何，根据基于负载平衡设定的此机器人应处理的比例及实际供给量而计算的值。实际上未必此处计算数量的工件机器人全部能够处理。

例如，负载平衡被设定为 50%的最上游的传送带工作站，在至今为止传送的 7 个工件中取 4 个，如果计算所使用的范围中，包括 3 个工件，则预计搬送数量为 1 个(＝(7＋3)÷2－4)。

传送带工作站对应距离

用来计算预想搬运数量的范围的上游侧和下游侧之间的距离，使其作为传送带工作站的对应距离。

注

以下情形，不可使用 PKCSGETPFRT.PC。

- **在传送带上，在［追踪类型］中设置［圆弧］时**
- **在传送带上使用［间隙检查］时**

6

参数 1：

指定传送带工作站的 ID。

可通过前述的 PKCSGETID 取得 ID。

参数 2：

指定存储预计搬送率的数值寄存器序号。

如果这些参数无效，在运行程序时，状态窗口会显示“pkcsgetpfrt error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

Pkcsgetpfrt 错误 17030：

指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：

传送带工作站 1 的 ID 存储于 R[11]时，将预计搬送率返回 R[12]。

```
CALL PKCSGETPFRT(传送带工作站 ID=R[11:传送带工作站 1 ID],工件流量 Reg=12)
```

6.8.5.10 PKCSGETNPRT.PC

返回指定的传送带工作站所保持的工件数量。

参数 1：

指定传送带工作站的 ID。

可通过前述的 PKCSGETID 取得 ID。

参数 2：

指定存储传送带工作站所保持的工件数量的数值寄存器序号。

使用例：

将传送带工作站 1 所保持的工件数量返回 R[13]。

```
CALL PKCSGETNPRT(传送带工作站 ID=R[11:传送带工作站 1 ID],工件数 Reg=13)
```

6.8.5.11 PKCSSETDSLIN.PC

变更指定的传送带工作站的放弃线。即时反映设定。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
指定放弃线的值。

使用例：
将传送带工作站 1 的放弃线变更为-200。

```
CALL PKCSSETDSLIN(传送带工作站 ID=R[11:传送带工作站 1 ID],放弃线=-200)
```

⚠ 注意

1 要正常反映设定内容，通常需要调出在主程序开头运行的 KAREL 程序 PKWCSTART，或 PKSNSTART 之前调出这些 KAREL 程序。

2 在开启设置画面的状态下，如果运行上述的 KAREL 程序，显示内容不会立即更新，而会在数秒后自动更新。

3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。

- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.5.12 PKCSGETENC.PC

取得在所指定的传送带工作站上使用中的编码器编号。

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
指定存储编码器编号的寄存器序号。

使用例：
在 R[7]中取得在传送带工作站 1 上使用中的编码器编号。

```
CALL PKCSGETENC(传送带工作站 ID=1,编码器编号 Reg=7)
```

6.8.5.13 PKCSGETTRID.PC

取得在所指定的传送带工作站上使用中的托盘 ID。

参数 1：
指定传送带工作站的 ID。
可通过前述的 PKCSGETID 取得 ID。

参数 2：
指定用来存储托盘 ID 之寄存器序号。

使用例：
在 R[7]中取得在传送带工作站 1 上使用中的托盘 ID。

```
CALL PKCSGETTRID(传送带工作站 ID=1,托盘 ID Reg=7)
```

6.8.5.14 PKCSCLRACK.PC

删除传送带工作站的未处理的 ACK。（强制）结束了机器人程序时，有时 PKCSGETQUE 正在被执行，且 PKCSACKQUE 的执行尚未完成。此时重启主程序，如果未执行的 PKCSACKQUE 被执行前执行 PKCSGETQUE，则会发生“GETQUE 之后未执行 ACKQUE”的报警。但是，如果执行此 PKCSCLRACK 之后再执行 PKCSGETQUE，就不会发生报警。但是，结束之前在 PKCSGETQUE 中取得的工件信息将会丢失。

参数 1：
指定传送带工作站的 ID。

使用例：
删除 ID 为 1 的传送带工作站的未处理的 ACK。

```
CALL PKCSCLRACK(传送带工作站 ID=1)
```

6.8.6 有关固定工作站的程序

固定工作站相关程序打头的 4 个字符是［PKFS］。［PK］后面的 2 个字符是［FS］。

6.8.6.1 PKFSGETID.PC

取得指定的固定工作站的 ID。

参数 1：
指定固定工作站名。

参数 2：
指定存储固定工作站的 ID 的数值寄存器序号。

如果这些参数无效，在运行程序时，状态窗口会显示以下信息，程序可能结束。各种原因和对策如下。

Cannot Find Fixed Station
不存在指定名称的固定工作站。

内部错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：
如果运行该 KAREL 程序，则固定工作站 1 的 ID 存储于 R[12]。

```
CALL PKFSGETID('FSTN1',固定工作站 ID R=12)
```

6.8.6.2 PKFSPUTQUE.PC

在指定的固定工作站放置新托盘。
用固定工作站的设置画面，指定放置的托盘的初始状态(“空”或“满”)。
处理放置的托盘内的全部单元后，准备其他托盘时，再次调出该 KAREL 程序。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

使用例：
固定工作站 1 放置新托盘。
ID 存储于 R[12]中。

```
CALL PKFSPUTQUE(固定工作站 ID=R[12:固定工作站 1 ID])
```

6.8.6.3 PKFSCLRQUE.PC

清除放置于指定的固定工作站的托盘的队列信息。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

使用例：
清除固定工作站 1 的工件信息。
ID 存储于 R[12]中。

```
CALL PKFSCLRQUE(固定工作站 ID=R[12:固定工作站 1 ID])
```

6.8.6.4 PKFSGETQUE.PC

从放置于指定的固定工作站的托盘中，取得未处理的单元信息。托盘的初始状态为“空”时，机器人将工件放置于取得的单元中。另一方面，托盘的初始状态为“满”时，机器人将放置于单元的工件拾取。
根据以下的优先顺序，选择未处理的单元信息。

- 根据 PKFSPUTQUE 放置的托盘内的单元。
- 托盘的初始状态为“空”时，在空的单元中，具有最小层编号的，也就是存在于最下层的单元。
  托盘的初始状态为“满”时，在放置有工件的单元中，具有最大层编号的，也就是存在于最上层的单元。
- 只在模型 ID 被指定的情况下，具有该模型 ID 的单元。
- 单元编号最小的单元。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

参数 2：
指定存储补偿量的视觉寄存器序号。

参数 3：
指定存储处理状态的数值寄存器序号。
0: 成功
1: 失败(队列中没有可处理的单元)

参数 4（可省略）：
指定模型 ID。

如果这些参数无效，在运行程序时，状态窗口会显示“P_FSGETQUE error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

P_FSGETQUE 错误 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：
从固定工作站 1 取得未处理的单元的信息。ID 存储于 R[12]中，VR[1]中是工件的信息，R[2]中是处理状态。

```
CALL PKFSGETQUE(固定工作站 ID=R[12:固定工作站 1 ID],补偿用 VR=1,状态 Reg=2)
```

⚠ 注意
相对各单元的位置的补偿量，是基于单元 1 和各单元的相对位置计算所得。因此，相对托盘的机器人位置的示教，必须在单元 1 的位置保持一致。

6.8.6.5 PKFSACKQUE.PC

以 PKFSGETQUE 处理的状态为 0，分配未处理单元的信息后，将对该单元进行的处理通知固定工作站。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

参数 2：
以下面的任意值，指定处理的结果。
1： 成功
在正常处理单元时，进行指定。通知后，该单元为已处理状态，不会再通过 PKFSGETQUE 进行取得。
2： 返回
在无法正常处理单元时，进行指定。通知后，该单元返回队列，可通过 PKFSGETQUE 再次取得。

使用例：
返回工件正常处理。ID 存储于 R[12]。

```
CALL PKFSACKQUE(固定工作站 ID=R[12:固定工作站 1 ID],成功)
```

6.8.6.6 PKFSPUTBUF.PC

本 KAREL 程序，仅在将托盘作为缓冲区使用时使用。
在指定的固定工作站放置新缓冲区。
用固定工作站的设置画面，指定放置的缓冲区的初始状态(“空”或“满”)。
将托盘作为缓冲区使用时，机器人可以再次拾取放置于单元的工件。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

参数 2：
指定用于区别缓冲区的识别编号。
作为识别编号，可指定 1 以上的值。如果指定相同的识别编号，多次调出该 KAREL 程序，则之前的缓冲区会被清除，新缓冲区则放置于固定工作站。
需要准备与之前的缓冲区不同的新缓冲区时，请指定与之前的缓冲区不同的识别编号。

使用例：
将缓冲区投入固定工作站 1。ID 存储于 R[12]中，识别编号为 1。

```
CALL PKFSPUTBUF(固定工作站 ID=R[12:固定工作站 1 ID],编号=1)
```

6.8.6.7 PKFSCLRBUF.PC

本 KAREL 程序，仅在将托盘作为缓冲区使用时使用。
清除放置于指定的固定工作站的缓冲区的队列信息。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

参数 2：
指定缓冲区的识别编号。
识别编号是以 PKFSPUTBUF 指定的。

使用例：
从固定工作站 1 清除识别编号 1 的缓冲区。ID 存储于 R[12]中。

```
CALL PKFSCLRBUF(固定工作站 ID=R[12:固定工作站 1 ID],编号=1)
```

6.8.6.8 PKFSGETBUF.PC

本 KAREL 程序，仅在将托盘作为缓冲区使用时使用。
从放置于指定的固定工作站的缓冲区中取得目标(空的单元信息或放置于单元的工件信息)。将托盘作为缓冲区使用时，由于机器人有将工件放置于空单元中的情况，以及拾取放置于单元中的工件的情况，因此需指定需处理的对象。
根据以下的优先顺序，选择目标。

- 被指定的识别编号的缓冲区内的单元。
- 目标为单元时，在空的单元中，是具有最小层编号的，也就是存在于最下层的单元。
  目标为工件时，在放置有工件的单元中，是具有最大层编号的，也就是存在于最上层的单元中放置的工件。
- 只在模型 ID 被指定的情况下，具有该模型 ID 的单元。即使目标是工件的情况，也通过单元的模型 ID 进行选择。
- 目标是单元的情况，是具有最小单元编号的单元。
  目标是工件，拾取顺序为先进先出的情况，是缓冲区内工件中，被最先放置的工件。拾取顺序为后进先出的情况，是缓冲区内工件中，被最后放置的工件。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

参数 2：
指定缓冲区的识别编号。
指定 1 以上的值。识别编号是以 PKFSPUTBUF 指定的。

参数 3：
指定目标。
1: 单元
2: 放置于单元的工件

参数 4：
指定存储补偿量的视觉寄存器序号。
目标是工件，且检出信息存储于缓冲区时，视觉寄存器的测量值、模型 ID、检出位置、编码器值中存储有工件的检出信息。

参数 5：
指定存储处理状态的数值寄存器编号。
0: 成功
1: 失败(队列中没有可处理的目标)

参数 6（可省略）：
指定模型 ID。
如果这些参数无效，在运行程序时，状态窗口会显示“P_FSGETBUF error(错误编号)”，程序可能结束。各种错误编号的原因和对策如下。

P_FSGETBUF 17030：
指定了不存在的数值寄存器序号。将数值寄存器赋予参数的情况，或进一步使用间接指定时，请确认对应的数值寄存器中，输入有存在的数值寄存器序号。

使用例：
从固定工作站 1 取得识别编号 1 的缓冲区的未处理的单元信息。ID 存储于 R[12]中，VR[1]中是工件的信息，R[2]中是处理状态。

```
CALL PKFSGETBUF(固定工作站 ID=R[12:固定工作站 1 ID],编号=1,单元=1,补偿用 VR=1,状态 Reg=2)
```

⚠ 注意
相对各单元的位置的补偿量，是基于单元 1 和各单元的相对位置计算所得。因此，相对托盘(缓冲区)的机器人位置的示教，必须在单元 1 的位置保持一致。

6.8.6.9 PKFSACKBUF.PC

本 KAREL 程序，仅在将托盘作为缓冲区使用时使用。
以 PKFSGETBUF 处理的状态为 0，分配目标的信息后，将对该目标进行的处理通知固定工作站。

参数 1：
指定固定工作站的 ID。
可通过前述的 PKFSGETID 取得 ID。

参数 2：
以下面的任意值，指定处理的结果。
1： 成功
在正常处理目标时，进行指定。通知后，该目标的状态发生如下变化。
目标是单元的情况，变化为工件被放置于单元的状态。也就是说，在通知后，通过 PKFSGETBUF 将工件指定为目标时，可取得放置该工件的单元。
目标是工件的情况，放置工件的单元变化为空的状态。也就是说，在通知后，通过 PKFSGETBUF 将单元指定为目标时，可取得该空的单元。
2： 返回
在无法正常处理目标时，进行指定。
通知后，该目标返回队列，可通过 PKFSGETBUF 再次取得。

参数 3：（可省略）
将工件的检出信息保存到缓冲区中时，指定检出信息被保存的视觉寄存器编号。这里存储于缓冲区的信息，可通过后面作为目标指定工件，调出 PKFSGETBUF 而再次取得。存储于缓冲区的信息，是测量值、模型 ID、检出位置、编码器值。

使用例：
返回目标正常处理。ID 存储于 R[12]。

```
CALL PKFSACKBUF(固定工作站 ID=R[12:固定工作站 1 ID],成功)
```

6.8.6.10 PKFSGETTRID.PC

取得在树状图视图的［固定工作站］设定的托盘的 ID。

参数 1：
指定固定工作站的 ID。

参数 2：
指定用来存储托盘 ID 之寄存器序号。

使用例：
在 ID 为 1 的固定工作站设定的托盘的 ID 在 R[7]中取得。

```
CALL PKFSGETTRID(固定工作站 ID=1,托盘 ID Reg=7)
```

6.8.6.11 PKFSGETUF.PC

取得在树状图视图的［固定工作站］设定的用户坐标系编号。

参数 1：
指定固定工作站的 ID。

参数 2：
指定保存用户坐标系编号的数值寄存器编号。

使用例：
在 ID 为 1 的固定工作站设定的用户坐标系编号在 R[7]中取得。

```
CALL PKFSGETTUF(固定工作站 ID=1,用户坐标系  Reg=7)
```

6.8.6.12 PKFSCLRACK.PC

删除固定工作站的未处理的 ACK。（强制）结束了机器人程序时，有时 PKFSGETQUE（或 PKFSGETBUF）正在被执行，且 PKFSACKQUE（或 PKFSACKBUF）的执行尚未完成。此时重启主程序，如果在未执行的 PKFSACKQUE（或 PKFSACKBUF）被执行之前执行 PKFSGETQUE（或 PKFSGETBUF），则发生 IRPK-051“GETQUE 之后未执行 ACKQUE”的报警。但是，如果执行此 PKFSCLRACK 之后执行 PKFSGETQUE（或 PKFSGETBUF），则不发生报警。但是，在结束之前在 PKFSGETQUE（或 PKFSGETBUF）中取得的工件信息将会丢失。

参数 1：
指定固定工作站的 ID。

使用例：
删除 ID 为 1 的固定工作站的未处理的 ACK。

```
CALL PKFSCLRACK (固定工作站 ID=1)
```

6.8.7 有关托盘的程序

托盘相关程序打头的 4 个字符是［PKTR］。［PK］后面的 2 个字符是［TR］。

6.8.7.1 PKTRSETCLPOS.PC

变更设定于托盘的类型。有多台机器人的情况，只要以其中 1 台机器人运行，则反映至全部机器人。

参数 1：
以字符串指定托盘的名称。

参数 2：
单元的序号，在 1 至 160 之间指定。
参数 3：
用序号指定变更的对象。变更对象如下所示。
1 : X
2 : Y
3 : Z
4 : R
5 : 模型 ID
6 : 层编号

⚠ 注意
如果层编号为 1 以上，则作为有效单元进行处理。因此，需要进行单元的追加时，请将层编号指定为 1 以上的值。但是，只能从使用中的最后的单元编号+1 的单元开始进行追加。另一方面，需要删除单元时，请将层编号指定为 0。但是，需从使用中的最后的单元起依次删除。如果要删除全部，也可以从最开始起删除。

参数 4：
指定变更对象的值。

使用例：
将“TRAY1”的“单元 1”的“X”的值变更为“33.3”。

```
CALL PKTRSETCLPOS('TRAY1',单元编号=1,X,数值=33.3)
```

对层编号 1 追加单元 2。

```
CALL PKTRSETCLPOS('TRAY1',单元编号=2,层编号,数值=1)
```

从层编号 1 删除单元 2。

```
CALL PKTRSETCLPOS('TRAY1',单元编号=2,层编号,数值=0)
```

⚠ 注意
1 为了正确反映设定内容，通常在主程序开头执行的 KAREL 程序 PKWCSTART 或 PKSNSTART 之前需要调用本 KAREL 程序。
2 在打开设定画面的状态下，即使执行本 KAREL 程序，显示内容也不会立即被更新，但数秒后会自动更新。
3 使用多台控制装置，调出变更设定的 KAREL 程序（PKCVSETLBD.PC、PKCVSETTRNAM.PC、PKCVENBLB.PC、PKCVENBSC.PC、PKSNENBSENS.PC、PKSNSETVPNAM.PC、PKSNSETVTOFS.PC、PKSNSETMTOFS.PC、PKCSSETDSLIN.PC、PKTRSETCLPOS.PC）时，要注意以下几点。
- **在调出 PKWCSTART 和 PKWCCHECK 之前调出变更所列举的设定的 KAREL 程序。**
- **为了等待设定变更被反映到全部控制装置中，在多台控制装置的所有控制装置中，在 PKWCSTART 和 PKWCCHECK 之前输入等待 5 秒左右的等待命令。调出 PKWCSTART 和 PKWCCHECK 时，若没有反映设定变更，则会发生 IRPK-079“需要同期数据”。即使等待 5 秒，仍然发生 IRPK-079 时，延长等待时间，一直到不再发生 IRPK-079 为止。**
- **调出多个变更所列举的设定的 KAREL 程序时，在一个控制装置上调出所有的 KAREL 程序。也可以变更其他控制装置的传感器和传送带工作站的设定。在多个控制装置上调出变更所列举的设定的 KAREL 程序时，在调用 PKWCSTART 和 PKWCCHECK 时，有的情况下会发生 IRPK-079“需要同期数据”。**

6.8.7.2 PKTRGETNUMCL.PC

取得托盘的指定层中所设定的单元数。

参数 1：
指定托盘 ID。
可通过前述 PKCSGETTRID 来取得在传送带上使用的托盘 ID。

参数 2：
指定层编号。
指定 0 时，取得托盘中所设定的全部单元数。
参数 3：
指定存储单元数的寄存器序号。

使用例：
在 R[7]中取得 ID 为 1 的托盘中所设定的所有单元数。

```
CALL PKTRGETNUMCL(托盘 ID=1,层编号=0,单元数 Reg=7)
```

6.9 机器人的程序

机器人程序根据系统创建程序。
本节介绍基本的程序例，并说明思考方式等。请参考介绍的程序例，配合客户负责的系统进行变更。

在“6.9.1 有关追踪程序”中对追踪程序进行说明。
在传送带上进行拾取动作、放置动作的程序，需要是追踪程序。追踪程序，需要进行与通常程序不同的设定。
6.9.2 之后对每个构成的样本程序进行说明。

而本节中，主要进行程序的逻辑生成。
实际示教机器人的位置的作业，在后面叙述的“6.11 机器人的位置示教”中进行。

> **注意**
> ***i*RPickTool 中，默认使停电处理无效。**

6.9.1 有关追踪程序

追踪传送带进行动作的机器人的程序，称为追踪程序。追踪程序与通常的程序不同，需要在程序的详细画面设定追踪参数设定表编号等。

设定方法如下。

1 请按下示教器的［SELECT］(列表)键。
显示程序一览画面。

2 从列表中选择追踪程序。
3 按下［>（下一页）］，再按下 F2［详细］。
4 按下 F3［下一步］。
则显示以下画面。

5 在［传送带追踪参数设定表编号］中，设定用传送带工作站指定的追踪参数设定表编号。
6 使［程序结束时继续追踪］有效。

注
使程序结束时继续追踪无效时，如果结束追踪程序，则机器人停止追踪动作。这样，即使立即进入下一个追踪程序，开始追踪动作，机器人的动作也无法顺畅连接。在视觉追踪系统中，有对机器人高速持续动作的要求，因此请使程序结束时继续追踪有效，令机器人的动作顺畅连接。

7 使［所选边界］保持为 0。如果为 0，使用以传送带工作站指定的追踪范围。

6.9.2 传送带—传送带的构成的样本程序

这里介绍从供给传送带拾取工件，将工件放置于输出传送带的机器人的程序例。参考这里介绍的程序，根据各系统进行变更。

一般来说，追踪的系统中，机器人的程序由以下 3 个程序构成。

- 主程序
- 在供给侧传送带进行拾取动作的程序
- 在输出侧传送带进行放置动作的程序

从主程序调出进行拾取动作的程序以及进行放置动作的程序。从传送带取，放置于工作台的情况，仅进行拾取动作的程序为追踪程序；从传送带取，放置于传送带的情况，进行拾取动作的程序和进行放置动作的程序都是追踪程序。
而这里介绍的程序例，使用以下的数值寄存器以及位置寄存器。请根据需要，适当变更寄存器序号。

表 6.9.2(a) 数值寄存器

R[1]	**周期停止标记。如果对该数值寄存器设定 1，则系统周期停止。**
R[2]	**PKCSGETQUE 的状态**
R[3]	**机器人的抓手区域*的计数。**
R[4]	**机器人的抓手区域的数量。请事先进行设定，如果是取 1 个的机械手，则设定为 1，如果是取 N 个的机械手，则设定为 N。**
R[11]	**存储供给侧传送带 CONV1 的传送带工作站 ID。**
R[12]	**存储输出侧传送带 CONV2 传送带工作站 ID。**

* 所谓抓手区域，表示拾取工件的部分。例如，有 4 个吸附垫的机械手，用各个衬垫分别拾取工件时，抓手区域的数量为 4。

表 6.9.2(b) 位置寄存器

PR[1～9]	第 N 个抓手区域用的拾取位置
PR[10]	用于生成拾取动作的接近点的工具偏移量
PR[11]	在输出侧传送带的放置位置
PR[12]	用于生成放置动作的接近点的工具偏移量

6.9.2.1 主程序

系统的主程序。

主程序的设定例

MAIN1.TP

```
  1:   R[1:周期停止]=0
  2:   CALL PKCSGETID('CONV1',工作站 ID Reg=11)
  3:   CALL PKCSGETID('CONV2',工作站 ID Reg=12)
  4:
  5:   UTOOL_NUM=0
  6:   UFRAME_NUM=0
  7: J P[1] 50% FINE
  8:
  9:   CALL PKWCSTART
 10:
 11:   LBL[100]
 12:   CALL PICK1
 13:   IF R[1:周期停止]=1,JMP LBL[900]
 14:   CALL DROP1
 15:   IF R[1:周期停止]=1,JMP LBL[900]
 16:   JMP LBL[100]
 17:
 18:   LBL[900]
 19:   CALL PKWCEND
[End]
```

在第 12 行调出 PICK1.TP，进行工件的拾取动作，在第 14 行调出 DROP1.TP，进行放置动作。主程序，主要是交替反复进行 PICK1.TP 和 DROP1.TP 的调出。取多个的机械手，以 PICK1.TP 进行全部的多个拾取动作，以 DROP1.TP 一起放置。

R[1]是周期停止标记。在拾取动作、放置动作的间隙，确认周期停止标记，如果标记竖起，则停止周期。

在第 2 行和第 3 行调出 PKCSGETID，在 R[11]、R[12]取得传送带工作站的 ID。这些分别可在 PICK1.TP、DROP1.TP 内使用。

在第 9 行调出 PKWCSTART，初始化传送带工作站，启动传感器任务。进入周期停止后，在第 19 行调出 PKWCEND，停止传感器任务。
使用 PKCSCLRQUE 以及 PKSNSTART，也可以分别进行传送带工作站的初始化以及传感器任务的启动。这种情况，将第 9 行更换为以下。

```
CALL PKWCCHECK
CALL PKCSCLRQUE(传送带工作站 ID=R[11:CSTN1])
CALL PKCSCLRQUE(传送带工作站 ID=R[12:CSTN2])
CALL PKSNSTART
```

首先，调出 PKWCCHECK，确认没有 *i*RPickTool 的无效设定。
然后，调出 PKCSCLRQUE，进行传送带工作站的初始化。
使用存储有以 PKCSGETID 取得的各传送带工作站的 ID 的数值寄存器。传送带工作站的初始化，在以 PKSNSTART 启动传感器前进行。

而在该程序例中，如果于第 13 行竖起周期停止标记，则无论此时处于持有工件 / 不持有工件状态，立即进入周期停止。如果在持有工件的状态，需要将持有的工件放置后进行周期停止时，请将第 9 行变更为以下语句。

```
13:    IF R[1:周期停止]=1 AND R[3:区域]=1,JMP LBL[900]
```

在后面叙述的进行拾取动作的程序 PICK1.TP 中，每次拾取工件，R[3]的值增加，因此如果是持有工件的状态，R[3]的值为 2 以上。利用这一特性，判别是处于持有工件的状态还是不持有的状态。

多台机器人的情况

多台机器人的情况，可以在全部机器人上直接使用 MAIN1.TP 以及 PICK1.TP、DROP1.TP。但是，需要变更追踪程序的追踪参数设定表编号。没有传感器(没有连接相机、光电管传感器)的机器人，如果执行 PKWCSTART 或 PKSNSTART，则发生 IRPK-031 的报警，但仅是报警信息，不会影响动作。

6.9.2.2 进行拾取动作的程序

在供给侧传送带上进行拾取动作的程序。

6

进行拾取动作的程序设定例子

PICK1.TP

```
 1:   UTOOL_NUM=0
 2:    UFRAME_NUM=0
 3:    R[3:区域]=1
 4:
 5:    LBL[100]
 6:    STOP_TRACKING
 7:    CALL PKCSGETQUE(传送带工作站 ID=R[11:CSTN1],连续的标记=R[3:区域],超时值(ms)=100,
       补偿用 VR=1,状态 Reg=2)
 8:    IF R[2:GETQ_状态]=0,JMP LBL[200]
 9:    IF R[1:周期停止]=1,JMP LBL[900]
10:    JMP LBL[100]
11:
12:    LBL[200]
13: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
14: L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[1]
15:    CALL VACUUM_ON
16: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
17:    CALL PKCSACKQUE(传送带工作站 ID=R[11:CSTN1],成功)
18:
19:    R[3:区域]=R[3:区域]+1
20:    IF R[3:区域]<=R[4:区域数],JMP LBL[100]
21:    LBL[900]
[End]
```

在程序的详细画面的［传送带追踪参数设定表编号］中，设定在传送带 CONV1 中选择该机器人的传送带工作站所指定的追踪参数设定表编号。
在第 1 行选择工具坐标系编号。工具坐标系编号指定设定成夹爪的工具坐标系编号。
例如，将夹爪的工具坐标系设定成 1 号。
在本例中，将夹爪的工具坐标系通用，在每个夹爪区域切换位置寄存器。其他方法也可以将位置寄存器通用，切换在各夹爪区域的设定的工具坐标系。

在第 2 行选择用户坐标系编号。用户坐标系编号必须选择 0。
如果在追踪程序中指定用户坐标系，机器人有可能会做出意外动作。（在追踪程序中，用户坐标系编号为 0 时，参照追踪坐标系，而不是机器人的世界坐标系。）

在第 7 行调出 PKCSGETQUE，至下一次拾取的工件进入追踪范围，最多等待 100ms。如果工件进入追踪范围，则 VR[1]中存储补偿数据，R[2]中存储 0。工件进入追踪范围后，跳至第 12 行，进行拾取动作，在第 17 行调出 PKCSACKQUE，通知传送带工作站，拾取结束。

> ⚠ 注意
> **使用取多个的机械手时，如 PICK1.TP 一个个依次处理工件。PKCSGETQUE 中，每次状态为 0 时，设定追踪动作的触发器。首先，对用机械手拾取的工件调出 PKCSGETQUE 和 PKCSACKQUE，一旦取得工件的信息，由于使用在最后的 PKCSGETQUE 中设定的追踪动作的触发器，可能无法在正确位置拾取最初拾取的工件。**

在 PKCSGETQUE 的第 3 个参数中，将待机时间指定为 100ms，因此如果调出 PKCSGETQUE 时取得的工件不在追踪范围内，则在待机时间 100ms 退出。这种情况，R[2]中是 0 以外的值，PICK1.TP 确认周期停止标记，再次调出 PKCSGETQUE，等待工件的分配。

> ⚠ 注意
> **运行 *i*RPickTool 的程序的机器人控制装置，并列进行 TP 程序的处理和传感器处理。因此，TP 程序的处理负担变重，延后传感器的处理。于是，可能发生虽然传送带上的工件进入了追踪范围，但是传感器未完成该工件的检出处理，未分配给机器人的情况。这种情况下，在循环中插入适当的待机，减轻 TP 程序的处理负担。例如，如果在 PKCSGETQUE 的第 3 个参数使超时为 0ms，则 PKCSGETQUE 被高速反复调出，因此 TP 程序很忙，传感器任务的处理时间可能变长。**

请根据使用的机械手，事先进行设定，如果是取 1 个的机械手，则 R[4]设定为 1，如果是取多个的机械手，则 R[4]设定为机械手的抓手区域的数量。

R[3]是下一个取得的抓手区域的序号，在第 3 行初始化为 1，每取 1 个，在第 19 行增加 1。于是，取多个的机械手的情况，取第 1 个时 R[3]＝1，取第 2 个时 R[3]＝2，依次递增，当超过 R[4]中指定的数字时，表示已拾取了全部抓手区域的工件。

在第 7 行的 PKCSGETQUE 的第 2 个参数中，指定存储有抓手区域序号的 R[3]。PKCSGETQUE 中，如果第 2 个参数为 1，则根据指定的负载平衡进行分配，如果是 2 以上，则与负载平衡无关，分配连续的工件。因此，取 N 个的机械手的情况下，取连续的 N 个后会跳过连续的 M 个（M: 取决于系统构成和设定)…，就会变成这样的动作。

> **注**
> **跳过数量 M，根据负载平衡决定。例如，取 4 个的机械手的情况，如果是 2 台机器人的等分配，第 1 台机器人取 4 个，跳过 4 个…，如果是 3 台机器人等分配，则第 1 台机器人取 4 个，跳过 8 个…。此外，在 2 台机器人的情况下，如果负载平衡设定为 1:2，则也是第 1 台机器人取 4 个，跳过 8 个…。**

> ⚠ 注意
> **从 R[3]的值增加 1 至调出 PKCSGETQUE 的瞬间，通过了“放弃线”的工件的信息，即使是连续的工件的信息，也移至下游的机器人。**

即使是取多个的机械手，在不连续取 N 个的情况下，以 PKCSGETQUE 的第 2 个参数中的常数，交付 1。此时，根据负载平衡的设定，决定每隔几个拾取。即使在这种情况下，如果所有的抓手区域已满，则跳过程序。

在第 13 行～16 行，进行拾取动作。机器人的示教点，不是程序的位置数据，而是在位置寄存器进行示教。由此，使程序运行中的示教点的微调整更为方便。此外，在存储有抓手区域序号的 R[3]间接指定位置寄存器。由此，机器人移动至选择的抓手区域相应的示教点。

在拾取位置指定“CNT0”(平顺 3)。由于在高速追踪中，如果指定“FINE”(定位)，可能来不及进行拾取动作。详细信息，请参照“6.12.4 示教点的平顺指定”。此外，使用直线距离指定指令、先执行指令，可进一步令追踪动作最佳化。详细信息，请参照“6.12.5 直线距离指定”、“6.12.8 先执行指令”。

> **注**
> **工具位于从手腕的机械接口偏移的位置时，追踪程序中，无论是取 1 个或多个的机械手，建议 TCP 只使用设定于手腕机械接口的 1 个。追踪动作中，在机器人的 TCP 位置进行目标位置是否在追踪范围内的确认。如果将 TCP 设定于从手腕的机械接口补偿的位置，则即使目标位置上的 TCP 位置在追踪范围内，手腕可能在追踪范围外，可能发生机器人无法到达的情况。**

拾取动作的接近点、逃避点，并不直接示教位置，而是使用事先设定于位置寄存器的工具偏移量，对于拾取位置相对地进行决定。由此，使拾取位置、接近点、逃避点的微调整更为容易。

在第 6 行调出追踪停止指令。这是停止追踪动作的指令。

本说明书介绍的样本程序，如“6.9.1 有关追踪程序”中的设定，程序结束后的追踪为有效。这是在进行连续的拾取/放置动作时，尽量使机器人高速动作的设定。
但是，如果发生暂时没有传送来拾取工件，则连续进行等待期间前的追踪动作，机器人可能离开动作范围。
停止追踪动作，避免机器人离开动作范围。即使在这种情况下，如果立即设定追踪的触发器，则能顺畅地开始下一个追踪动作。

注
生成取 1 个的机械手的专用程序时，可使其比 PICK1.TP 更简单些。请参照后面叙述的进行放置动作的程序。

6.9.2.3 进行放置动作的程序

在输出侧传送带上进行放置动作的程序。

进行放置动作的程序设定例子

DROP1.TP

```
  1:   UTOOL_NUM=0
  2:   UFRAME_NUM=0
  3:
  4:   LBL[100]
  5:   STOP_TRACKING
  6:   CALL PKCSGETQUE(传送带工作站 ID=R[12:CSTN2],连续的标记=1,超时值(ms)=100,补偿用 VR=2,
       状态 Reg=2)
  7:   IF R[2:GETQ_STAUTS]=0,JMP LBL[200]
  8:   IF R[1:周期停止]=1,JMP LBL[900]
  9:   JMP LBL[100]
 10:
 11:   LBL[200]
 12: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 13: L PR[11] 4000mm/sec CNT0 VOFFSET,VR[2]
 14:   CALL VACUUM_OFF
 15: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 16:   CALL PKCSACKQUE(传送带工作站 ID=R[12:CSTN2],成功)
 17:
 18:   LBL[900]
[End]
```

基本与进行拾取动作的程序相同。

在程序的详细画面的［传送带追踪参数设定表编号］中，设定在传送带 CONV2 中选择该机器人的传送带工作站所指定的追踪参数设定表编号。

主要差异在于，即使是取多个的机械手，由于是将拾取的工件一起放置，因此无论是取 1 个的机械手 / 取多个的机械手，都没有使用 R[3]、R[4]的抓手的管理。此外，同样道理，第 6 行的 PKCSGETQUE 的第 2 个系数为常数 1。

6.9.3 传送带—固定工作站的构成的样本程序

这里表示进行从传送带拾取，放置于固定工作站的作业的程序例。
程序内容，与从传送带向传送带搬送工件的情况类似。请一并参照“6.9.2 传送带—传送带的构成的样本程序”。

而这里介绍的程序例，使用以下的数值寄存器以及位置寄存器。请根据需要，适当变更寄存器序号。

表 6.9.3(a)数值寄存器

R[1]	**周期停止标记。如果对该数值寄存器设定 1，则系统周期停止。**
R[2]	**PKCSGETQUE 的状态**
R[5]	**PKFSGETQUE 的状态**

R[6]	**存储传送带 CONV1 的传送带工作站 ID。**
R[7]	**存储固定工作站 FSTN1 的固定工作站 ID。**

表 6.9.3(b)位置寄存器

PR[2]	**传送带上的拾取位置**
PR[3]	**固定工作站上的放置位置**
PR[11]	**用于生成拾取动作的接近点的工具偏移量**
PR[12]	**用于生成放置动作的接近点的工具偏移量**

6.9.3.1 主程序

系统的主程序。

主程序的设定例

MAIN1.TP

```
  1:  R[1:周期停止]=0
  2:  CALL PKCSGETID('CONV1',工作站 ID Reg=6)
  3:  CALL PKFSGETID('FSTN1',固定工作站 ID R=7)
  4:
  5:  ! 在固定工作站中准备托盘
  6:  CALL PKFSPUTQUE(固定工作站 ID=R[7:FSTN1_ID])
  7:
  8: J PR[1:Perch Pos] 100% FINE
  9:
 10:  CALL PKWCSTART
 11:
 12:  LBL[100]
 13:  CALL PICKCS
 14:  IF R[1:周期停止]=1,JMP LBL[900]
 15:
 16:  CALL DROPFS
 17:  IF R[1:周期停止]=1,JMP LBL[900]
 18:  JMP LBL[100]
 19:
 20:  LBL[900]
 21:  CALL PKWCEND
[End]
```

在第 13 行调出 PICKCS.TP，进行工件的拾取动作，在第 16 行调出 DROPFS.TP，进行放置动作。主程序，主要是交替反复进行 PICKCS.TP 和 DROPFS.TP 的调出。

R[1]是周期停止标记。在拾取动作、放置动作的间隙，确认周期停止标记，如果标记竖起，则停止周期。

在第 2 行和第 3 行调出 PKCSGETID，在 R[6]取得传送带工作站的 ID，在 R[7]取得固定工作站的 ID。这些分别可在 PICKCS.TP、DROPFS.TP 内使用。

在第 6 行固定工作站中，使用 PKFSPUTQUE 准备托盘。

在第 10 行调出 PKWCSTART，初始化传送带工作站，启动传感器任务。进入周期停止后，在第 21 行调出 PKWCEND，停止传感器任务。

多台机器人的情况

基本可以向其他机器人复制相同的程序进行使用。
但是，请注意以下几点。
MAIN1.TP 将固定工作站的名称变更为该机器人作业的对象。

PICKCS.TP、DROPFS.TP 可直接复制至其他机器人并使用。但是，需要根据作业的传送带工作站，变更追踪程序(PICKCS.TP)的追踪参数设定表编号。
没有传感器(没有连接相机、光电管传感器)的机器人，如果执行 PKWCSTART，则发生 IRPK-031 的报警，但仅是报警信息，不会影响动作。

6.9.3.2 进行拾取动作的程序

在传送带上进行拾取动作的程序。

进行拾取动作的程序设定例子

PICKCS.TP

```
  1:  UTOOL_NUM=0
  2:  UFRAME_NUM=0
  3:
  4:  LBL[100]
  5:  ! 从传送带工作站取得工件信息
  6:  CALL PKCSGETQUE(传送带工作站 ID=R[6:CSTN_ID],连续的标记=1,超时值(ms)=100,补偿用 VR=1,
      状态 Reg=2)
  7:  IF R[1:周期停止]=1,JMP LBL[900]
  8:  IF R[2:GETQ_STAUTS]>0,JMP LBL[100]
  9:
 10:L PR[2] 10000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[11]
 11:L PR[2] 10000mm/sec CNT0 VOFFSET,VR[1]
 12:  CALL VACUUM_ON
 13:L PR[2] 10000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[11]
 14:
 15:  ! 通知拾取了工件的事宜
 16:  CALL PKCSACKQUE(传送带工作站 ID=R[6:CSTN_ID],成功)
 17:
 18:  LBL[900]
[End]
```

在程序的详细画面的［传送带追踪参数设定表编号］中，设定在传送带 CONV1 中选择该机器人的传送带工作站所指定的追踪参数设定表编号。
程序的内容，与从传送带工作站的传送带工作站的作业的例子几乎相同。详细信息，请参照“6.9.2.2 进行拾取动作的程序”。但是，这里不使用抓手区域。

6.9.3.3 进行放置动作的程序

在固定工作站上进行放置动作的程序。

进行放置动作的程序设定例子

DROPFS.TP

```
  1:  UTOOL_NUM=0
  2:  UFRAME_NUM=0
  3:
  4:  LBL[100]
  5:  ! 等待托盘的准备结束
  6:  WAIT DI[21]=ON
  7:
  8:  ! 从托盘取得单元信息
  9:  CALL PKFSGETQUE(固定工作站 ID=R[7:FSTN1_ID],2,5)
 10:  IF R[1:周期停止]=1,JMP LBL[900]
 11:
 12:  ! 将工件放置于单元
 13:L PR[3] 10000mm/sec CNT100 VOFFSET,VR[2] Offset,PR[12]
 14:L PR[3] 10000mm/sec CNT0 VOFFSET,VR[2]
```

```
 15:   CALL VACUUM_OFF
 16:L PR[3] 10000mm/sec CNT100 VOFFSET,VR[2] Offset,PR[12]
 17:
 18:   ! 通知已将工件放置于单元的事宜
 19:   CALL PKFSACKQUE(固定工作站 ID=R[7:FSTN1_ID],成功)
 20:
 21:   ! 将空单元的信息返回托盘
 22:   CALL PKFSGETQUE(固定工作站 ID=R[7:FSTN1_ID],补偿用 VR=2,状态 Reg=5)
 23:   IF R[5:GETQ_STAUTS]>0,JMP LBL[316]
 24:
 25:   ! 通知托盘已满的事宜
 26:   CALL PKFSACKQUE(固定工作站 ID=R[7:FSTN1_ID],返回)
 27:
 28:   JMP LBL[900]
 29:
 30:   LBL[316]
 31:   ! 通知托盘已满的事宜
 32:   DO[21]=PULSE,1.0sec
 33:
 34:   ! 准备新托盘
 35:   CALL PKFSPUTQUE(固定工作站 ID=R[7:FSTN1_ID])
 36:
 37:   LBL[900]
[End]
```

在第 2 行指定用户坐标系序号。追踪程序中，需要使用户坐标系为 0，而固定工作站中可任意设定。

用 DI 信号确认第 6 行是否准备有托盘。

在第 9 行从托盘取得单元。通过对第 21 行以后的托盘的填充确认，如果没有空单元，则准备有新的托盘，并且在第 6 行等待托盘的准备结束，因此这里一定可以拾取工件。所以，这里不进行对 PKFSGETQUE 的状态的确认。

在第 13 行～16 行，进行放置动作。此外，使用直线距离指定指令、直线最高速度功能、先执行指令，可进一步令追踪动作最佳化。详细信息，请参照“6.12.5 直线距离指定”、“6.12.7 直线最高速度功能”、“6.12.8 先执行指令”。
在第 22、23 行使用 PKFSGETQUE，确认托盘内是否有空单元。如果没有空的，在第 31、32 行输出用于准备新托盘的信号(这里是 DO[21])，在第 35 行，使用 PKFSPUTQUE，对固定工作站准备新托盘。如果有空的，在第 26 行，将取得的单元返回队列，以 PKFSACKQUE 返还 ACK，使托盘的状态恢复原状。

6.9.4 传送带—缓冲区传送带的构成的样本程序

这里显示将托盘作为缓冲区使用的程序例。
程序内容，与从传送带向传送带搬送工件的情况类似。请一并参照“6.9.2 传送带—传送带的构成的样本程序”。

而这里介绍的程序例，使用以下的数值寄存器以及位置寄存器。请根据需要，适当变更寄存器序号。

表 6.9.4(a) 数值寄存器

R[1]	**周期停止标记。如果对该数值寄存器设定 1，则系统周期停止。**
R[2]	**PKCSGETQUE 的状态**
R[3]	**PKCSGETTIME 的状态**
R[5]	**PKFSGETQUE 的状态**
R[6]	**存储传送带 CONV1 的传送带工作站 ID。**
R[7]	**存储传送带 CONV2 的传送带工作站 ID。**
R[8]	**存储固定工作站 FSTN1 的固定工作站 ID。**
R[10]	**拾取状况。1：已拾取。2：已放置。**
R[13]	**工件的到达时间。**

表 6.9.4(b) 位置寄存器

PR[2]	**传送带上的拾取位置**
PR[3]	**在传送带的放置位置**
PR[5]	**固定工作站上的拾取位置**
PR[6]	**固定工作站上的放置位置**
PR[12]	**用于生成传送带上拾取动作的接近点的工具偏移量**
PR[13]	**用于生成传送带上放置动作的接近点的工具偏移量**
PR[15]	**用于生成固定工作站上拾取动作的接近点的工具偏移量**
PR[16]	**用于生成固定工作站上放置动作的接近点的工具偏移量**

6.9.4.1 主程序

系统的主程序。

主程序的设定例

MAIN1.TP

```
  1:  R[1:周期停止]=0
  2:
  3:  ! 取得各工作站的 ID
  4:  CALL PKCSGETID('CONV1',工作站 ID Reg=6)
  5:  CALL PKCSGETID('CONV2',工作站 ID Reg=7)
  6:  CALL PKFSGETID('FSTN1',固定工作站 ID R=8)
  7:
  8:  ! 在固定工作站中准备缓冲区
  9:  CALL PKFSPUTBUF(固定工作站 ID=R[8:BufferID],编号=1)
 10:
 11: J PR[1:Perch Pos] 100% FINE
 12:  CALL PKWCSTART
 13:
 14:  R[10:Pick]=0
 15:  LBL[100]
 16:  ! 从传送带工作站拾取工件
 17:  CALL PICKCS
 18:  IF R[1:周期停止]=1,JMP LBL[900]
 19:  IF R[10:Pick]=1,JMP LBL[200]
 20:
 21:  ! 从缓冲区拾取工件
 22:  CALL PICKBF
 23:  IF R[1:周期停止]=1,JMP LBL[900]
 24:  IF R[10:Pick]=0,JMP LBL[100]
 25:
 26:  LBL[200]
 27:  ! 在传送带工作站上放置工件
 28:  CALL DROPCS
 29:  IF R[1:周期停止]=1,JMP LBL[900]
 30:  IF R[10:Pick]=0,JMP LBL[100]
 31:
 32:  ! 在缓冲区内放置工件
 33:  CALL DROPBF
 34:  IF R[1:周期停止]=1,JMP LBL[900]
 35:  IF R[10:Pick]=1,JMP LBL[200]
 36:  JMP LBL[100]
 37:
 38:  LBL[900]
 39:  CALL PKWCEND
[End]
```

如果拾取工件时，工件还没有来，则从缓冲区取工件。
此外，如果在放置工件时未准备放置位置，则将工件放置于缓冲区。

根据 R［10:Pick］判断是否使用缓冲区。
机器人的机械手上有工件时，R［10:Pick］为 1，没有时为 0。
也就是说，从传送带拾取工件时，如果 R［10:Pick］=0，则从缓冲区拾取工件。
另一方面，将工件放置于传送带上时，如果 R［10:Pick］=0，则将工件放置于缓冲区内。

多台机器人的情况

基本可以向其他机器人复制相同的程序进行使用。

但是，请注意以下几点。
MAIN1.TP 将固定工作站的名称变更为该机器人作业的对象。
PICKCS.TP、DROPFS.TP 可直接复制至其他机器人并使用。
但是，需要根据作业的传送带工作站，变更追踪程序(PICKCS.TP)的追踪参数设定表编号。
没有传感器(没有连接相机、光电管传感器)的机器人，如果执行 PKWCSTART，则发生 IRPK-031 的报警，但仅是报警信息，不会影响动作。

6.9.4.2 进行拾取动作的程序

在传送带上进行拾取动作的程序。

进行拾取动作的程序设定例子

PICKCS.TP

```
  1:  UTOOL_NUM=0
  2:  UFRAME_NUM=0
  3:
  4:  STOP_TRACKING
  5:  ! 从传送带工作站取得工件信息
  6:  CALL PKCSGETQUE(传送带工作站 ID=R[6:CstnIDIn],连续的标记=1,超时值(ms)=0,补偿用 VR=1,状态
Reg=2)
  7:  IF R[1:周期停止]=1,JMP LBL[900]
  8:  IF R[2:CsGetQStat]=0,JMP LBL[300]
  9:  IF R[2:CsGetQStat]=1,JMP LBL[100]
 10:  IF R[2:CsGetQStat]=2,JMP LBL[900]
 11:  JMP LBL[900]
 12:
 13:  LBL[100]
 14:  ! 确认工件什么时候到达
 15:  CALL PKCSGETTIME(传送带工作站 ID=R[6:CstnIDIn],顺序=1,连续的标记=1,数量=1,时间 Reg=13,
      补偿用 VR=2,状态 Reg=3)
 16:  IF R[13:Time]>=1000,JMP LBL[900]
 17:
 18:  LBL[200]
 19:  ! 从传送带工作站取得工件信息
 20:  CALL PKCSGETQUE(传送带工作站 ID=R[6:CstnIDIn],连续的标记=1,超时值(ms)=100,补偿用 VR=1,状态
Reg=2)
 21:  IF R[1:周期停止]=1,JMP LBL[900]
 22:  IF R[2:CsGetQStat]=0,JMP LBL[300]
 23:  IF R[2:CsGetQStat]=1,JMP LBL[200]
 24:  IF R[2:CsGetQStat]=2,JMP LBL[900]
 25:  JMP LBL[900]
 26:
 27:  LBL[300]
 28:  ! 从传送带工作站拾取工件
 29:L PR[2] 10000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[12]
 30:L PR[2] 10000mm/sec CNT0 VOFFSET,VR[1]
 31:  CALL VACUUM_ON
 32:L PR[2] 10000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[12]
 33:
 34:  ! 通知拾取了工件的事宜
 35:  CALL PKCSACKQUE(传送带工作站 ID=R[6:CstnIDIn],成功)
 36:  R[10:Pick]=1
 37:
 38:  LBL[900]
[End]
```

在程序的详细画面的［传送带追踪参数设定表编号］中，设定在传送带 CONV1 中选择该机器人的传送带工作站所指定的追踪参数设定表编号。

与从传送带工作站的传送带工作站的作业例不同的点在于，如果工件还未到达领域内，则预测工件的到达时间，判断是否从缓冲区取工件。

第 6 行的 PKCSGETQUE 特地将待机时间设为 0。因为当拾取工件不在领域内时，立即至缓冲区确认。
因第 6 行的 PKCSGETQUE 而超时时，在第 15、16 行确认工件的到达时间。如果知道下一个应拾取的工件暂时不会来，则使 R［10:Pick］保持 0，结束该程序，至缓冲区确认。
如果确认了到达时间后，知道工件马上就来，则在第 20 行反复调出 PKCSGETQUE，等待工件。这里，为了减轻 CPU 负担，在待机时间中输入 100ms。

6.9.4.3 进行放置动作的程序

在传送带上进行放置动作的程序。

进行放置动作的程序设定例子

DROPCS.TP

```
 1:  UTOOL_NUM=0
 2:  UFRAME_NUM=0
 3:
 4:  STOP_TRACKING
 5:  ! 从托盘取得单元信息
 6:  CALL PKCSGETQUE(传送带工作站 ID=R[7:CstnIDOut],连续的标记=1,超时值(ms)=0,补偿用 VR=4,状态
Reg=2)
 7:  IF R[1:周期停止]=1,JMP LBL[900]
 8:  IF R[2:CsGetQStat]=0,JMP LBL[300]
 9:  IF R[2:CsGetQStat]=1,JMP LBL[100]
10:  IF R[2:CsGetQStat]=2,JMP LBL[900]
11:  JMP LBL[900]
12:
13:  LBL[100]
14:  ! 确认托盘的单元什么时候到达
15:  CALL PKCSGETTIME(传送带工作站 ID=R[7:CstnIDOut],顺序=1,连续的标记=1,数量=1,时间 Reg=13,
     补偿用 VR=4,状态 Reg=3)
16:  IF R[13:Time]>=1000,JMP LBL[900]
17:
18:  LBL[200]
19:  ! 从托盘取得单元信息
20:  CALL PKCSGETQUE(传送带工作站 ID=R[7:CstnIDOut],连续的标记=1,超时值(ms)=100,补偿用 VR=3,
     状态 Reg=2)
21:  IF R[1:周期停止]=1,JMP LBL[900]
22:  IF R[2:CsGetQStat]=0,JMP LBL[300]
23:  IF R[2:CsGetQStat]=1,JMP LBL[200]
24:  IF R[2:CsGetQStat]=2,JMP LBL[900]
25:  JMP LBL[900]
26:
27:  LBL[300]
28:  ! 将工件放置于单元
29:L PR[3] 10000mm/sec CNT100 VOFFSET,VR[4] Tool_Offset,PR[13]
30:L PR[3] 10000mm/sec CNT0 VOFFSET,VR[4]
31:  CALL VACUUM_OFF
32:L PR[3] 10000mm/sec CNT100 VOFFSET,VR[4] Tool_Offset,PR[13]
33:
34:  ! 通知已放置于单元的事宜
35:  CALL PKCSACKQUE(传送带工作站 ID=R[7:CstnIDOut],成功)
36:  R[10:Pick]=0
37:
38:  LBL[900]
```

```
[End]
```

基本与从传送带进行拾取动作的程序相同。

在程序的详细画面的［传送带追踪参数设定表编号］中，设定在传送带 CONV2 中选择该机器人的传送带工作站所指定的追踪参数设定表编号。

6.9.4.4 进行拾取动作的程序(缓冲区)

在固定工作站上进行拾取动作的程序。

进行拾取动作的程序（缓冲区）设定例子

PICKBUF.TP

```
  1:  UTOOL_NUM=0
  2:  UFRAME_NUM=0
  3:
  4:  ! 从缓冲区取得工件信息
  5:  CALL PKFSGETBUF(固定工作站 ID=R[8:BufferID],编号=1,单元=2,补偿用 VR=3,状态 Reg=5)
  6:  IF R[5:FsGetQStat]<=0,JMP LBL[100]
  7:  IF R[5:FsGetQStat]>0,JMP LBL[900]
  8:
  9:  LBL[100]
 10:  ! 从缓冲区拾取工件
 11:L PR[5] 10000mm/sec CNT100 VOFFSET,VR[3] Offset,PR[15]
 12:L PR[5] 10000mm/sec CNT0 VOFFSET,VR[3]
 13:  CALL VACUUM_ON
 14:L PR[5] 10000mm/sec CNT100 VOFFSET,VR[3] Offset,PR[15]
 15:
 16:  ! 通知已从缓冲区拾取工件的事宜
 17:  CALL PKFSACKBUF(固定工作站 ID=R[8:BufferID],成功)
 18:  R[10:Pick]=1
 19:
 20:  LBL[900]
[End]
```

在第 2 行指定用户坐标系序号。追踪程序中，需要使用用户坐标系为 0，而固定工作站中可任意设定。

在第 5 行从缓冲区取得工件。

在第 11 行～14 行，进行放置动作。此外，使用直线距离指定指令、直线最高速度功能、先执行指令，可进一步令追踪动作最佳化。详细信息，请参照“6.12.5 直线距离指定”、“6.12.7 直线最高速度功能”、“6.12.8 先执行指令”。

6.9.4.5 进行放置动作的程序(缓冲区)

在固定工作站上进行放置动作的程序。

进行放置动作的程序（缓冲区）设定例子

DROPBUF.TP

```
  1:  UTOOL_NUM=0
  2:  UFRAME_NUM=0
  3:
  4:  ! 从缓冲区取得单元信息
  5:  CALL PKFSGETBUF(固定工作站 ID=R[8:BufferID],编号=1,单元=1,补偿用 VR=2,状态 Reg=5)
  6:  IF R[5:FsGetQStat]<=0,JMP LBL[100]
  7:  IF R[5:FsGetQStat]>0,JMP LBL[900]
  8:
```

```
 9:   LBL[100]
10:   ！在缓冲区内放置工件
11:L PR[6] 10000mm/sec CNT100 VOFFSET,VR[2] Offset,PR[16]
12:L PR[6] 10000mm/sec CNT0 VOFFSET,VR[2]
13:   CALL VACUUM_OFF
14:L PR[6] 10000mm/sec CNT100 VOFFSET,VR[2] Offset,PR[16]
15:
16:   ！通知已将工件放置于单元的事宜
17:   CALL PKFSACKBUF(固定工作站 ID=R[8:BufferID],成功,2)
18:   R[10:Pick]=0
19:
20:   LBL[900]
[End]
```

基本与从固定工作站进行拾取动作的程序相同。

在第 5 行，从缓冲区取得空单元。从第 11 行至第 14 行执行放置动作。

6.10 基准位置设定

基准位置，是在机器人的程序示教拾取位置或放置位置时的工件的检出位置。
对于设定了基准位置的工件，对在相同传送带作业的全部机器人示教拾取位置或放置位置。
在传感器的设置画面的基准位置设定向导设定基准位置。
在不改变传送带上设定了基准位置的工件的位置的同时，进行各机器人的位置示教。
如果是固定工作站，不需要这里的作业。直接进行机器人的位置示教。

⚠ 注意
1 基准位置设定后，直至机器人的位置示教结束，即使传送带向前传送，也请注意不要改变相对传送带的工件的相对位置。如果在传送带上有多个传送带工作站，在全部传送带工作站上的机器人的位置示教结束之前，不得使工件动作。如果错误动作，请重新进行基准位置的设定。
2 如果有多台机器人，在停止传送带的状态下，于各机器人控制装置的编码器画面上，确认［当前计数值(计数)］是否为相同值。如果不同，请使全部机器人控制装置的电源关闭 / 开启。

注
1 追踪拾取位置和放置位置时，首先在进行拾取动作的程序中，进行基准位置设定和机器人的位置示教。之后，使传送带动作，令机器人以追踪动作拾取工件，进行放置动作程序的基准位置设定和机器人的位置示教。
2 如果在视觉程序中，将［使用基准数据］设定于［模型 ID 切换］，在对每个模型 ID 重复进行以下步骤。此时，通过模式匹配等检出工具分别设定模型 ID，添加基准数据，创建模型 ID 数量的基准数据后，进行基准位置设定。
3 需要在通过此传感器选择的控制装置中打开 iRPickTool 的设定画面，设定基准位置。

操作步骤因传感器的触发条件与触发的组合多少会有些不同。

以下组合的操作步骤请参阅“3.2.10 基准位置的设定、机器人的位置示教”。

- ［距离］+［通过视觉检出］的情况
  表示供给传送带中的步骤，除此以外的情况也相同。
- ［DI］/［RI］/［HDI］+［放入工件］的情况
  表示在输出传送带中使用托盘时的步骤，除此以外的情况也相同。

此外，在以下组合的情况下，与设定目标位置的同时设定基准位置，以避免实际检出工件。详情请参阅“6.5.6 目标位置设定”。

- ［距离］/［旗标］+［放入工件］的情况

6.10.1 ［DI］/［RI］+［通过视觉检出］

针对在传感器设定画面将触发条件设定成［DI］或［RI］，将动作设定成［通过视觉检出］时的基准位置设定步骤进行说明。
关于传感器的设定请参阅“6.5 传感器设定”。

1 使传送带停止。
2 选择传感器［SENS*］，按下 F3［基准位置］。

显示如下［基准位置向导　步骤 1/2］。

注
如果可以使用［位置寄存器编号］，通过使用该功能可以简化后述的机器人位置示教。详细请参阅“3.2.10.1 供给基准位置的设定、机器人的位置示教、在［位置寄存器编号］使用位置寄存器的编号时”。

3 将工件放在触发传感器上游。
放置在传送带宽度方向的中央附近。

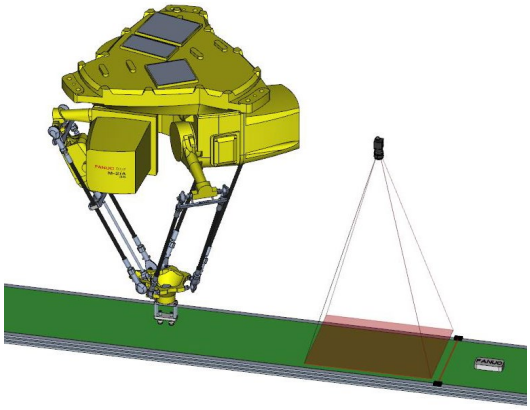
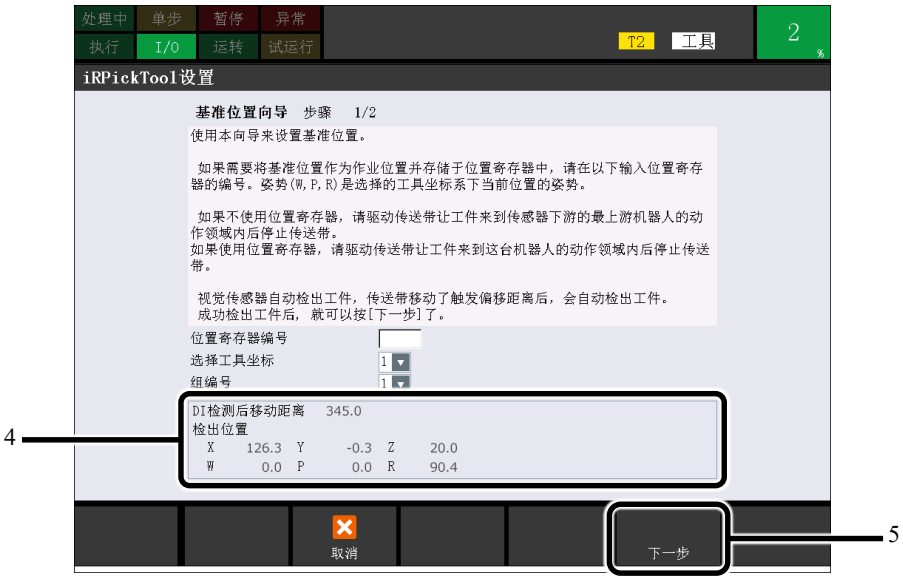


图 6.10.1(a) 将工件放在触发传感器上游

4 运行传送带，工件到达机器人动作范围内后关停。
如果检出成功，显示［检出位置］。
※检出不成功时
- 打开视觉程序，选择执行时监视“100%显示”。
- 打开执行时监视，确认图像。
- 如果检出时间偏离，请更改［触发偏移］值。

检出成功后，F5［下一步］有效。

5 按下 F5［下一步］。



6 按下 F5［结束］，

显示如下画面。
确认在［基准位置］中显示“向导成功”。

7 接下来示教机器人的位置。（请参阅“6.11 机器人的位置示教”。）

6.11 机器人的位置示教

在基准位置的设定后，进行机器人的位置示教。
以下组合的操作步骤请参阅“3.2.10 基准位置的设定、机器人的位置示教”。

- ［距离］+［通过视觉检出］的情况
  表示供给传送带中的步骤，除此以外的情况也相同。
- ［DI］/［RI］/［HDI］+［放入工件］的情况
  表示在输出传送带中使用托盘时的步骤，除此以外的情况也相同。

6.11.1 传送带工作站

1 在传感器的向导画面设定基准位置后，如果工件进入机器人的动作范围内，则停止传送带。

> ⚠ **注意**
> **停止传送带时，请注意传送带上工件不要偏离。**

2 从程序列表打开追踪程序。
3 确认选择了在追踪程序内使用的用户坐标系、工具坐标系。
4 以点动使机器人动作，示教拾取/放置位置。

> ⚠ **注意**
> **1 拾取/放置位置的示教，务必请在追踪程序内进行。由于是进行追踪坐标系的坐标值的示教。即使是使用位置寄存器的情况，也请不要在位置寄存器画面进行位置示教。在拾取/放置位置有机器人的状态下，将光标对准移动至追踪程序内的拾取/放置位置的动作指令的行编号，边按下 SHIFT 键，边按下 F5［TOUCHUP］(位置修正)，进行示教。此外，显示“从当前位置减去 VR 吗？”时，选择“否”。显示“VR［*］：未初始化。是否继续？”时，选择“是”。**
> **2 托盘时，针对单元 1 示教放置/拾取位置。根据各单元的 X、Y、Z、R 与基准单元的坐标值自动计算针对各单元的补偿量。通过使用视觉补偿命令，机器人根据存储在视觉寄存器的补偿量移向各单元。**
> **3 3 轴～5 轴机器人的情况，请示教示教位置的 W、P 为 0、180、-180 的其中之一。**
> **4 5 轴机器人的情况，由于构造上存在无法取得的姿势，示教为法兰盘面朝正下方(或正上方)的拾取/放置位置。详细信息请参照“操作说明书（基本操作篇）B-83284CM”中的“程序的构成”章、“动作指令”节、“动作类型”项中的“关于 5 轴机器人特有的动作”。**

> **注**
> **可使用工具补偿指令(或位置补偿指令)，以示教的拾取/放置位置为基准，相对生成接近点和逃避点的位置。这种情况，不需要接近点和逃避点的位置的示教。**

5 如果传送带上有多个传送带工作站，则在所有传送带工作站中反复进行上述步骤。

> **注**
> **各机器人的机械手 / 抓手为相同形状时，省略第 2 台以后的机器人位置示教，可直接复制第 1 台的机器人的示教位置使用。**

6.11.2 固定工作站

在托盘的设定画面按下 F3［基准单元］，将单元 1 的数值设定成基准单元，只对单元 1 示教放置/拾取位置。详情请参阅“6.3.3 基准单元的设定”。
使用视觉补偿指令，以该基准单元为基准，相对生成各单元的位置。

> **注**
> **针对单元 1 的放置/拾取位置可以对表示托盘坐标系的用户坐标系之外的任意用户坐标系进行示教。**

6.12 追踪动作的微调整

这里介绍使机器人的追踪动作进一步优化的方法。
假设基本动作的示教已大致完成。

边进行实际的追踪动作，边进行位置的调整。

如果搬运中有误差，应仔细观察在什么条件下、在什么方向上、有多少偏差，特定原因，并根据原因采取适当的措施。如果未能根据原因采取措施，则无法期待精度的改善。

在下面的节里，介绍各种情况下的对策。

6.12.1 最初的微调整

机器人的位置示教后，最先应进行的调整。多次传送工件，进行示教位置的微调整。该调整对各个机器人分别进行。
如果使用视觉，在下一节“6.12.2 追踪坐标系的设定误差的调整”的步骤中进行相同动作，因此请跳过本节，至下一节。

1 在追踪程序拾取位置之后，立即输入待机指令。
之所以输入待机指令，是因为可通过目视确认偏移。待机时间以 30 秒左右为宜。
2 以稼动时的速度使传送带动作。
3 与平常一样启动主程序，传送 1 个工件。

> **⚠ 注意**
> **运行追踪程序时，请以 AUTO 模式或 T2 模式运行。**
> **传送带动作时，即使以 T1 模式运行追踪程序，机器人也无法动作。这种情况，显示“LNTK-041 编码器在 T1 模式移动”的报警信息。**

4 机器人到达拾取工件的位置的正上方后，停止传送带。此时，请使程序保持运行状态。
5 传送带完全停止后，暂时停止程序。
6 在这种状态下，确认机器人的位置和工件位置的偏移量。
7 如果有偏移，以点动移动至正确位置。这里，将光标对准拾取位置的动作指令的行，进行位置修正。此时，会询问是否减去补偿量，请选择［是］。
8 从追踪程序中删除插入的待机指令。

6.12.2 追踪坐标系的设定误差的调整

使用相机检出工件时，如果相机认识的追踪坐标系和各机器人认识的追踪坐标系有偏移，则追踪动作会发生误差。该误差会以“如果工件不旋转，则正确补偿，如果工件旋转，则发生偏移”的形式发生。

该误差原因是，未正确设定追踪坐标系。本来，需从工具坐标系的设定、追踪坐标系的设定开始重新进行，但也可以通过以下步骤，对 *i*RVision 计算的补偿数据添加调整量，实现改善。该调整对各个机器人分别进行。

而这里介绍的程序例，将位置寄存器的 20 号作为存储调整量的地方使用。请根据需要适当变更。

1 在追踪程序的 PKCSGETQUE 后面追加以下的 1 行。

例：“6.9.2.2 进行拾取动作的程序”的 PICK1.TP 的情况

```
 5:  LBL[100]
 6:  STOP_TRACKING
 7:  CALL PKCSGETQUE(R[11:CSTN1], R[3:区域], 100, 1, 2)
 8:  IF R[2:GETQ_状态]=0,JMP LBL[200]
 9:  IF R[1:周期停止]=1,JMP LBL[900]
10:  JMP LBL[100]
11:
```

```
12:  LBL[200]
13:  CALL ADJ_OFS(1, 1, 20, 3)
```

2 根据存储调整的补偿数据的视觉寄存器的序号，变更追踪程序。

```
12:  LBL[200]
13:  CALL ADJ_OFS(1, 1, 20, 3)
14:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
15:L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[3]
16:  CALL VACUUM_ON
 17:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
```

3 为了使机器人在追踪程序内的拾取位置的正上方停止，请将动作指令和待机指令插入接近点下面一行。

```
12:  LBL[200]
13:  CALL ADJ_OFS(1, 1, 20, 3)
14:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
15:L PR[R[3:区域]] 4000mm/sec FINE VOFFSET,VR[3] Tool_Offset,PR[15]
16:  WAIT 10.00(sec)
17:L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[3]
18:  CALL VACUUM_ON
 19:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
```

之所以输入待机指令，是因为可通过目视确认偏移。待机时间以 10 秒左右为宜。此外，对 PR[15](用于生成拾取动作的接近点的工具偏移量，Z 以外是 0)的 Z 输入较小值。这是为了使机器人在拾取位置的正上方停止。

4 将 PR［20］设为 XYZWPR =(0, 0, 0, 0, 0, 0) 。
5 以稼动时的速度使传送带动作。
6 与平常一样启动主程序，传送 1 个工件。此时，工件的放置方向与基准位置设定时相同。

⚠ 注意
运行追踪程序时，请以 AUTO 模式或 T2 模式运行。
传送带动作时，即使以 T1 模式运行追踪程序，机器人也无法动作。这种情况，显示“LNTK-041 编码器在 T1 模式移动”的报警信息。

7 机器人到达接近点后，停止传送带。此时，请使程序保持运行状态。
8 传送带完全停止后，暂时停止程序。
9 确认工件位置和机器人位置的偏移。
10 如果有偏移，以点动移动至正确位置，进行示教修正。此时，会询问是否减去补偿量，请选择［是］。
11 再次启动主程序，传送 1 个工件。
这次，将工件设定于基准位置时，以 180 度相反方向放置。
12 机器人到达接近点后，停止传送带。此时，请使程序保持运行状态。
13 传送带完全停止后，暂时停止程序。
14 确认机器人的位置和工件位置的偏移量。
15 如有偏移，分别测定 X 方向(传送带的传送方向)和 Y 方向(与传送带传送方向正交的方向)的偏移，将 X 方向的调整量输入 PR[20]的 X，Y 方向的调整量输入 PR[20]的 Y。调整量设定为各偏移量的一半的值。从 ADJ_OFS 的行运行程序，确认是否移动至正确位置。如有偏移，再次返回步骤 4，进行作业。反复执行，直至机器人移动至正确位置。
16 确认移动至正确位置后，从追踪程序中删除插入的动作指令和待机指令。

有关 ADJ_OFS

ADJ_OFS 是在存储于视觉寄存器的补偿数据上，加上指定的调整量的 KAREL 程序。以下记述参数的意思。详细信息请参照“*i*RVision 操作说明书 （参考篇）”中的“其他选项功能”章、“视觉支持工具”节。

参数 1：存储补偿数据的寄存器种类。补偿数据存储于视觉寄存器中时，指定为 1
参数 2：补偿数据存储于视觉寄存器中时的视觉寄存器的序号
参数 3：存储调整量的位置寄存器的序号

参数 4：存储调整的补偿数据的视觉寄存器的序号

注
***i*RPickTool 中，即使没有安装视觉支持工具选项，也可以标准使用 ADJ_OFS。**

6.12.3 机器人的动态误差的调整

即使在使传送带停止的状态下，机器人正确来到工件的拾取位置上，如果在传送带动作的状态下，仍有发生偏移的可能。该误差是引起机器人动作的追踪延迟的误差。

注
1. 动态误差调整是调整在“6.12.1 最初的微调整”～“6.12.2 追踪坐标系的设定误差调整”中说明的误差，如果在停止传送带的状态下，则将其变成无误差状态后再进行。
2. 机器人在多个传送带上进行作业时，请在其中速度最大的传送带上进行调整。

按以下步骤调整该误差。

1 为了使机器人在追踪程序内的拾取位置的正上方停止，请将动作指令和待机指令插入接近点下面一行。

例：“6. 9.2.2 进行拾取动作的程序”的 PICK1.TP 的情况

```
12:   LBL[200]
13:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
14:L PR[R[3:区域]] 4000mm/sec FINE VOFFSET,VR[1] Tool_Offset,PR[15]
15:   WAIT 10.00(sec)
16:L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[1]
17:   CALL VACUUM_ON
```

之所以输入待机指令，是因为可通过目视确认偏移。待机时间以 10 秒左右为宜。此外，在 PR[15](用于生成拾取动作的接近点的工具偏移量。Z 以外为 0。)的 Z 中输入较小的值。这是为了使机器人在拾取位置的正上方停止。

2 以稼动时的速度使传送带动作。
3 与平常一样启动主程序，传送 1 个工件。此时，工件的放置方向与基准位置设定时相同。

⚠ 注意
运行追踪程序时，请以 AUTO 模式或 T2 模式运行。
传送带动作时，即使以 T1 模式运行追踪程序，机器人也无法动作。这种情况，显示“LNTK-041 编码器在 T1 模式移动”的报警信息。

4 在传送带动作中，且程序运行过程中，请确认机器人的位置正确追踪工件的拾取位置。
5 如果有偏移，在“6.2 机器人设定”中介绍的“动态追踪误差调整”中，输入以(5 量(mm)÷传送带速度(mm/s))×1000 计算的值(单位 ms)。传送带的速度，是机器人追踪工件时的速度。机器人偏移至工件在传送带上的上游侧，偏移量的符号为正，偏移至下游侧则为负。已经在［动态追踪误差调整］中设定了值的情况，则将上述计算所得的值与原来的值相加并输入。
6 从追踪程序中删除插入的动作指令和待机指令。

计算例

［动态追踪误差调整］中设定为 30 的状态下，传送带以速度 500 mm/s 动作时，工件的拾取位置在传送带上游侧偏移 3 mm 的情况，

［动态追踪误差调整］=+30(ms) + (3(mm) ÷ 500(mm/s)) × 1000 = +36 (ms)

6.12.4 示教点的平顺指定

通常，进行拾取动作的程序和放置动作的程序中，示教至接近点的移动、至拾取/放置位置的移动、至逃避点的移动等 3 个动作。根据这 3 个动作的 FINE(定位) / CNT(平顺)的设定情况，不仅机器人的动作路径变化，还影响循环时间。

接近点、逃避点

有关接近点和逃避点，由于不需要正确通过该点，因此尽量 CNT 指定较大的值，可加快循环时间。CNT 可指定在 0～100 的范围，因此 CNT100 最快。
但是，CNT 指定越大，机器人的动作路径就进行越大的内圈转动。机器人内圈转动，可能挂到传送带上其他的工件。
请观察动作路径和循环时间的关系，调整 CNT 的程度。

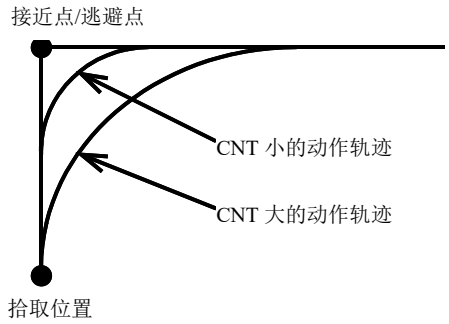


图 6.12.4(a)

拾取位置、放置位置

有关拾取/放置位置，由于需要正确到达该位置，因此通常并非 CNT，而是进行 FINE。
但是，如果是 FINE 指定，机器人的动作比 CNT 指定慢。为了尽量不影响路径，提高循环时间，请以 CNT 指定较小值。由于 CNT 可在 0～100 的范围内指定，请观察动作路径和动作时间的关系，在 0～2 的附近范围进行调整。

但是，在 CNT 指定的情况下，即使是较小值，机器人会有若干内圈转动，在未到达示教的拾取/放置位置之前，就开始向逃避点动作。例如，下图示教点 A 的情况，如果是 FINE，机器人下降至指定高度，如果是 CNT，则机器人不下降至指定高度，开始向逃避点上升。
于是，如示教点 B，将示教位置设定为比实际需要到达的位置稍低的位置。一下子下降较为危险，因此请边观察机器人的动作，边逐渐调整。

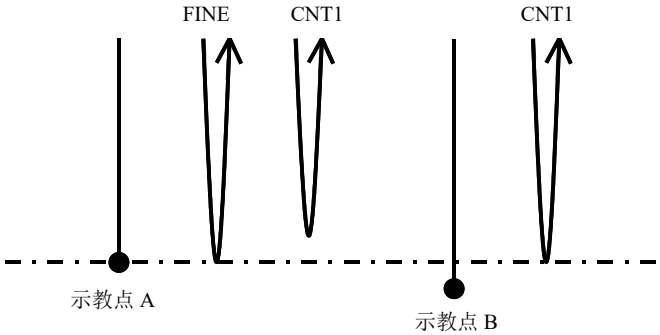


図 6.12.4(b)

6.12.5 直线距离指定

使接近点、逃避点 CNT 变大，且使拾取/放置位置并非 FINE 而是 CNT 的情况下，机器人不仅在高度方向未到达目标位置，在 XY 方向也可能未到达目标位置。该未到达的部分，看上去加在搬运误差上。

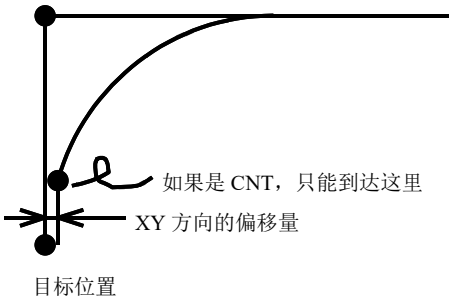


图 6.12.5(a)

这种情况，在拾取/放置位置的动作程序上，附加直线距离指定指令。
如果附加该指令，即使之前的点为 CNT，仅目标点之前指定的距离，机器人以直线动作。
但是，如果直线距离较长，则循环时间变长，因此请观察机器人的动作和循环时间的关系，进行调整。

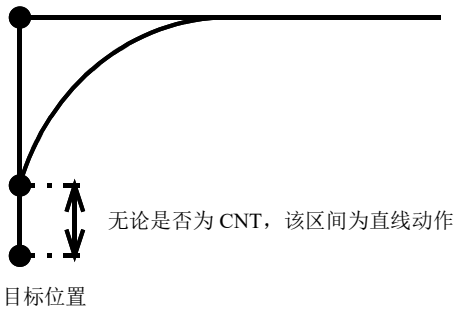


图 6.12.5(b)

以下是程序的修正例。是“6.9.2.2 进行拾取动作的程序”中介绍的程序的一部分。

```
13:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
14:L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[1] AP_LD10
15:   CALL VACUUM_ON
16:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10] RT_LD10
```

在第 14 行，附加直线距离指定指令。由此，机器人在目标点之前的 10 mm 进行直线动作。

6.12.6 法兰盘直线指令

法兰盘直线指令，是在 TP 程序的动作指令上附加的指令。当 TCP 和标牌中心有偏移时附加。

通常的直线动作，如下图 TCP 描画直线般进行控制。其结果，特别是如同取多个的机械手，当 TCP 和标牌中心有偏移时，标牌的中心曲折，机器人的动作距离变长，可能发生循环时间的不适当。

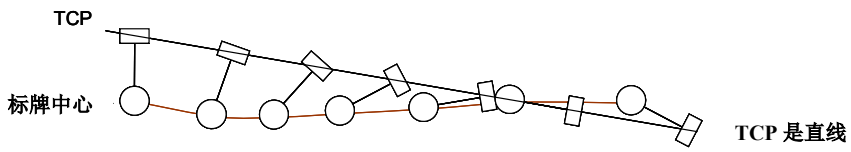


图 6.12.6(a)通常的直线动作

通过使用法兰盘直线指令，无法保证 TCP 的直线性，但能如下图所示，标牌的中心描画直线。其结果，机器人的动作距离为最小，从而使循环时间缩短。

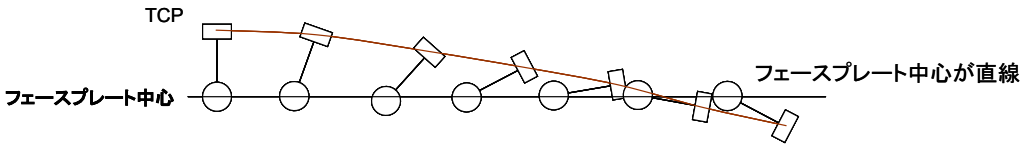


图 6.12.6(b)使用法兰盘直线指令时

特别是类似拳头机器人的手腕高速机器人，在使用 TCP 与标牌中心有偏移的机械手进行拾取时，在传送带上空反复直线动作的情况，其效果值得期待。
以下是程序的修正例。是“6.9.2.2 进行拾取动作的程序”中介绍的程序的一部分。

```
13:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10] FPLIN
14:L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[1] AP_LD10
15:   CALL VACUUM_ON
16:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10] RT_LD10
```

在第 13 行，附加法兰盘直线指令。在移动至接近位置时使用。

⚠ 注意
1 如果使用法兰盘直线指令，则 TCP 在直线时不动作。请在如同传送带的上空般，与周围无干涉的领域的动作中使用。
2 法兰盘直线指令的效果根据机器人、动作而不同。

6.12.7 直线最高速度功能

以直线动作指令指定的最高速度(例如 2000mm/sec)，是限制电机能力的值。实际上，机器人可以以比动作指令中记述的速度更高的速度动作。
在 *i*RPickTool 的适用例中，希望机器人能尽量高速地动作。通过使用该功能，机器人可以其具有的能力的最高速度直线动作。
但是本功能在追踪程序中自动无效。因此，只能在固定工作站上作业。

⚠ 注意
1 使用本功能的情况，机器人高速动作。执行动作指令时，请充分确认周围是否有危险。
2 追踪动作中，本功能自动无效。
3 先执行指令中，执行时间有差异。
4 可能无法保持直线距离指定功能中指定的距离。

以下是程序的修正例。是“6.9.3.2 进行拾取动作的程序”中介绍的程序的一部分。

```
10:L PR[2] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[11]
11:L PR[2] 10000mm/sec CNT0 VOFFSET,VR[1]
12:   CALL VACUUM_ON
 13:L PR[2] 10000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[11]
```

第 10 行的动作指令的速度中，选择［max_speed］(最大速度)。由此，在移动至接近位置中，本功能有效。本功能主要在移动至接近位置时使用。

6.12.8 先执行指令

如果高速稼动系统，使吸附 ON / OFF 的信号交换的等待时间，可能影响性能。这种情况，在拾取/放置位置的动作程序上，附加先执行指令。
如果附加该指令，可以在机器人的动作中输出信号，因此可以使信号交换的等待时间为零，缩短循环时间。

以下是程序的修正例。是“6.9.2.2 进行拾取动作的程序”中介绍的程序的一部分。

```
13:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
```

```
14:L PR[R[3:区域]] 4000mm/sec CNT0 VOFFSET,VR[1] TB .5sec,CALL VACUUM_ON
15:
16:L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
```

在第 14 行，附加先执行指令。这里，在目标点到达的 0.5 秒之前，运行子程序“VACUUM_ON”(吸附 ON)。在这里可以通过调整指定的秒数，在向着目标点之前的适当时间，使吸附 ON。

6.13 启动多台机器人时的注意事项

以多台机器人进行视觉追踪时，最后启动传感器执行的机器人控制装置的程序。为了以各机器人控制装置清除传送带工作站内的工件的信息，全部结束后，传感器任务开始工件的检出。
有多台执行传感器的机器人控制装置时，如“6.9.2.1 主程序”所述，在传感器启动之前插入待机语句，在全部的传送带工作站内的工件信息的清除结束后，启动传感器。

6.14 预整列设定

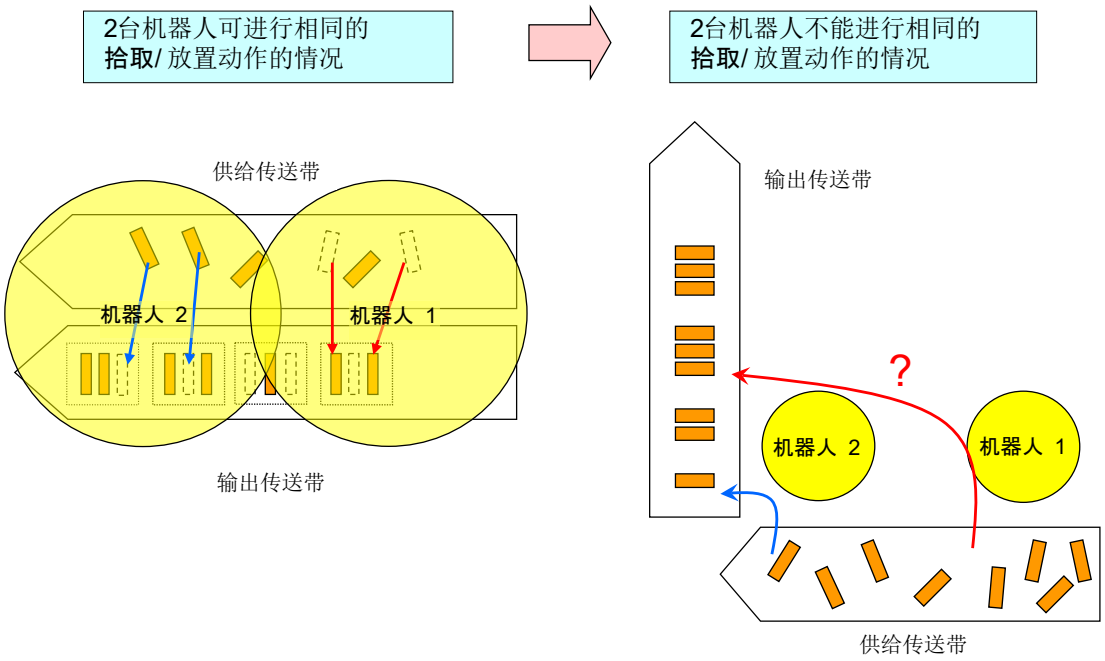
本节对相同传送带上的整列功能(预整列功能)的设定步骤进行说明。

北美专用设定时，需要北美专用选项 R856: *i*RPickTool / Pre-grouping，以使用预整列功能。

使用预整列功能的系统的基本构成，由配置于供给传送带的上游的相同供给传送带上整列工件的机器人，以及一同搬送配置于下游的整列工件的机器人组成。以下情况，有效发挥预整列功能。

例 1

如下图右侧所示，由于传送带的配置关系，不能将 2 台机器人设定为进行相同的拾取/放置动作。



例 2

即使将 2 台机器人设定为进行相同的拾取/放置动作，各台机器人的动作距离也变长，结果可能导致处理能力下降。

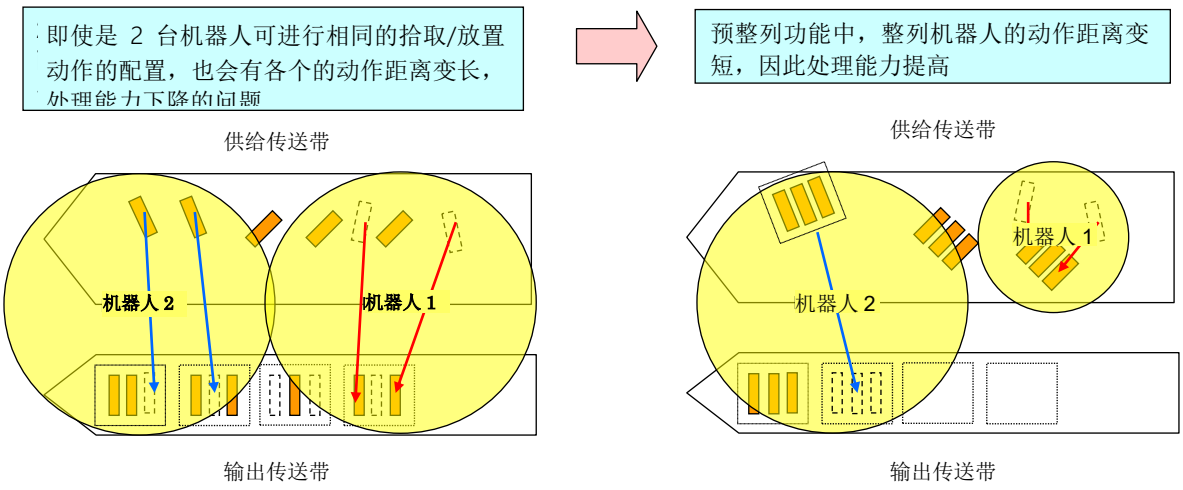


图 6.14(a)可有效预整列功能的构成例子

以用 2 台机器人在相同传送带上整列工件，将最下游的第 3 台机器人整列的工件一同交付至其他传送带的系统为例，对各步骤进行说明。

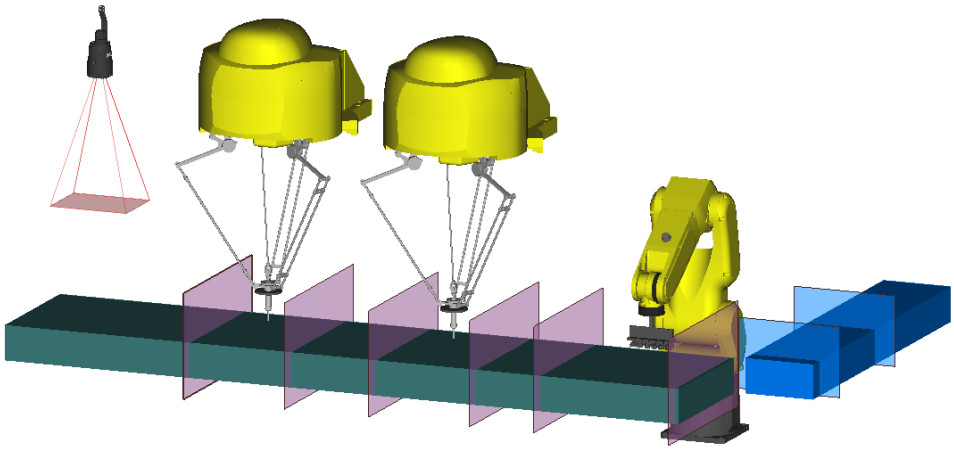


图 6.14(b)

这种情况下的系统构成和树状图构成，如图 6.14(c)所示。

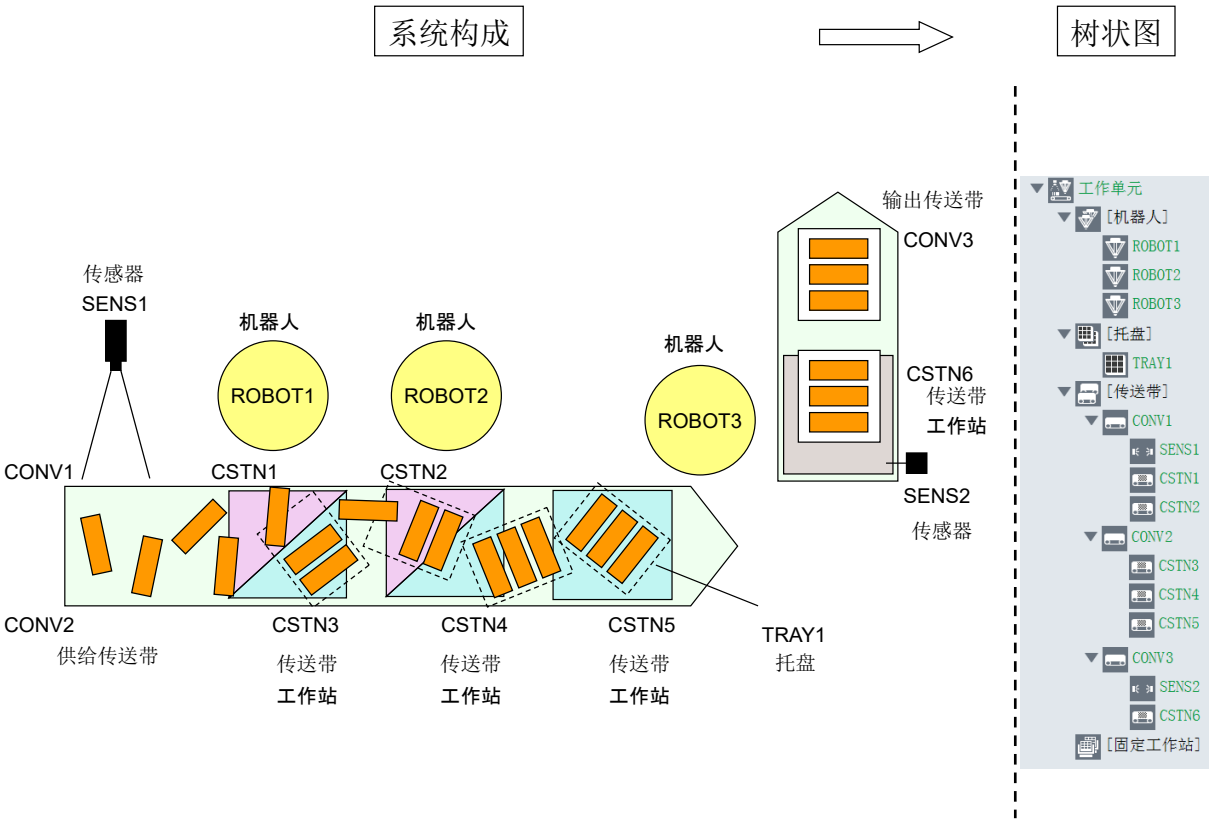


图 6.14(c)

在相同传送带上进行整列时，在相同的作业领域，设定进行拾取的传送带工作站，和放置拾取的工件的传送带工作站。
也就是说，对于相同的传送带，生成 2 个传送带对象。2 个传送带对象中，进行拾取的传送带，与通常的供给传送带一样进行设定。设定进行放置接着拾取的工件之作业的传送带。此传送带，与物理上进行拾取的传送带相同，包括作为其子对象的传送带工作站在内，需要进行后述的附加设定。

此外，图 6.14(c)的 CSTN5 是用于将 ROBOT3 整列的工件集中取出的传送带工作站。CSTN5 上虽然不进行放置已拾取工件的作业本身，但是不属于供给传送带 CONV1，而必须属于进行放置已拾取工件的作业之输出传送带 CONV2。

6.14.1 用语

对预整列功能特有的用语进行解说。

预整列传送带

在相同传送带上进行整列时，用于放置的传送带对象。图 6.14(c)中，CONV2 是预整列传送带。

输入传送带

在相同传送带上进行整列时，用于进行拾取的传送带对象。图 6.14(c)中，CONV1 是输入传送带。

虚拟托盘

如果在相同传送带上进行整列，实际并无托盘传送，而是看上去依照规定的整列图形，在虚拟的托盘内整列放置工件。指的就是这种虚拟的托盘。

该虚拟托盘，在后面叙述的托盘对象中定义。通过在预整列传送带进行设定，整列机器人可在执行时生成定义的托盘。根据传送来的工件的条件不同，虚拟托盘的位置和方向每次都不同。

基准工件

整列时不动的工件。

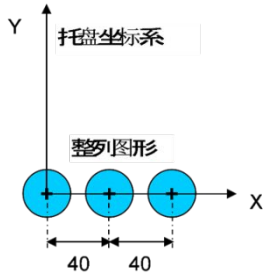
6.14.2 工作单元、机器人设定

没有预整列情况下的特记事项。请参照“6.1 工作单元设定”、“6.2 机器人设定”。

6.14.3 托盘设定

设定整列图形。

- 托盘坐标系的原点处理圆(无位相)的工件时，可在任意位置设定。
  例如设定单元 1 的 X、Y 为 0。
- 托盘坐标系的 X 方向上，处理圆(无位相)的工件时，与追踪坐标系的 X 方向相同，自动计算。但是，如果后面叙述的设定项目［搜索方向］有效时，从对应设定的多个方向中，根据情况选择托盘坐标系的 X 方向。
- 整列图形设定于层 1。
- 即使在层 1 的单元中设定模型 ID，也不能以预整列功能使用。使全部的模型 ID 为 1。



⚠ 注意

在层 1 内的各单元的坐标中，设定工件的中心位置。如果单元的坐标与工件的中心位置偏离，则无法正确计算为了与传送带上的工件整列而放置的工件间的干涉的检查，工件间可能发生冲撞。

注

如果单元和单元过于接近，判定调用 PKCSGENVRTRY 时基准工件与整列后的工件会碰撞，从而发生 IRPK-088“预整列的设定不正确。”的错误。

- 用于将整列的工件一同拾取的单元设定于层 2。
- 为了仅将完成整列图形的工件一同拾取，将层 2 的单元的模型 ID 设定为与层 1 的单元不同的模型 ID。
  例如，将模型 ID 设定为 2。一同拾取最下游工件的机器人，在 PKCSGETQUE 指定该模型 ID。

6.14.4 传送带设定

预整列中，必须设定两个传送带、输入传送带和预整列传送带。

1 两个传送带中，首先，设定输入传送带的对象。
请参照“6.4 传送带设定”。
预整列的情况，在用于一同拾取的最下游的传送带工作站中，所追加的预整列传送带的设置画面中设定追踪坐标系设定。输入传送带和预整列传送带，物理上属于相同的传送带，其追踪坐标系共同。因此，输入传送带不需要设定追踪坐标系。没有其他的特记事项。

2 设定预整列传送带的对象。
预整列传送带的对象不需要传感器。具有传感器对象的传送带对象，不能进行预整列的设定。首先，删除在生成传送带对象时自动生成的传感器对象。

3 追加需要数量的传送带工作站对象。

4 在［设置预整列］的［预整列］中选中［已使用］。

注
［追踪类型］中选择［圆弧］时，或选中［伺服传送带］的［已使用］时，不显示［设置预整列］的设定项目。

如果在［预整列］中选中［已使用］，会显示如下画面。

5　如果按下 F4［确定］，则显示预整列的设定项目。

注
1　预整列用中初始化的设定，是［负载平衡］的设定，以及子传送带工作站的［根据 Y 排序］。［负载平衡］为［无效］且初始化为［所有模型 ID 共通］。子传送带工作站的［根据 Y 排序］初始化为无效。
2　预整列传送带中，不能变更［追踪类型］、［伺服传送带］、［托盘］以及［负载平衡］的项目。
3　在相同传送带上以多台机器人进行整列时，为了使用负载平衡功能，在输入传送带的设置画面进行负载平衡的设定。

6　有关设置预整列以外的项目，建议“6.4 传送带设定”进行设定。

［设置预整列］

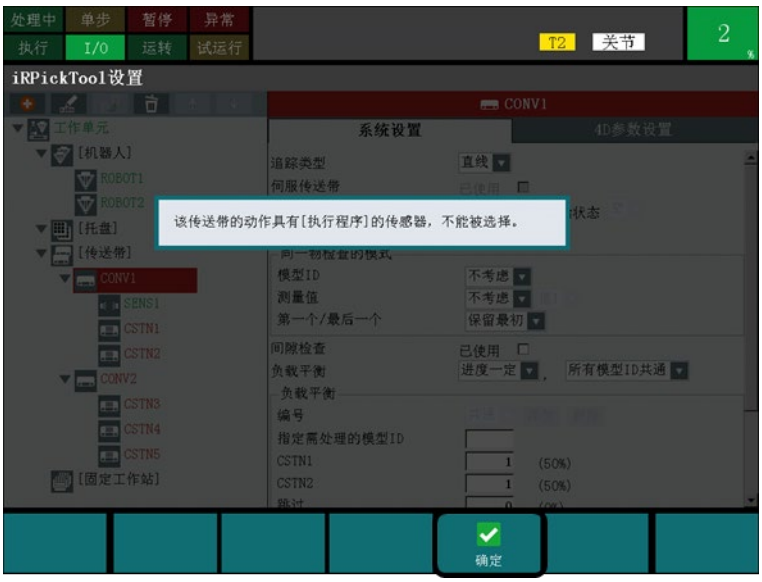
［输入传送带］

选择输入传送带的对象。

> **注**
>
> **［输入传送带］的选项中，仅显示满足以下条件的传送带对象。**
> - **在［追踪类型］中选择了［直线］。**
> - **没有选中［伺服传送带］的［已使用］。**
> - **［托盘］中选择了［不使用］。**
> - **没有选中［预整列］的［已使用］。**

以［输入传送带］选择的传送带中，［设置预整列］的项目如下图所示。

此外，［输入传送带］选择的传送带中，不能变更［追踪类型］、［伺服传送带］、［托盘］。如需要变更，会显示如下信息。如果按下 F4［确定］，则变更复原。



当选择输入传送带时，如果在预整列传送带的子工作站中，有与输入传送带的子工作站中选择的相同的机器人，则追踪参数设定只能在输入传送带的传送带工作站设置画面中进行。

> **⚠ 注意**
>
> **如果选择输入传送带，则预整列传送带的子工作站的［追踪参数设置］被覆盖改写。具体是，如果在子工作站中，有与输入传送带的子工作站中选择的机器人相同的情况，则覆盖改写为与在输入传送带设定的追踪参数设定表编号相同的值。**

如果在预整列传送带和输入传送带的各自的传送带工作站中，追加有相同的机器人，则在输入传送带的传送带工作站的设置画面中，选择追踪参数设定表编号和编码器序号。如果其中之一追加了机器人，则在该传送带工作站的设置画面中，选择追踪参数设定表编号和编码器序号。然后，在预整列传送带的设置画面，按下 F4［设定追踪坐标系］，开

始追踪坐标系的设定。请参照“3.2.4 追踪坐标系设定”。在完成设定第 1 台机器人的追踪坐标系后，就会开始设定第 2 台机器人的追踪坐标系。

[Y 的最大值]

可生成整列图形的 Y 方向的上限值。是追踪坐标系的值。

注
生成的整列图形，并不是视觉的模型原点，而是工件整体不要超出该上限值。

将在“5.4 工具坐标系的设定”设定的工具坐标系作为手动进给坐标系选择，点动 TCP，通过按下［记录］按钮，设定追踪坐标系的 Y 坐标值。例如，碰触相对传送带的进行方向的左侧皮带端，进行设定。

⚠ 注意
1 在该设定之前，需要先完成追踪坐标系的设定。
2 以打开着 *i*RPickTool 的设置画面的机器人控制装置的机器人进行碰触。

1 如果按下 F4［记录］按钮，则显示追踪坐标系中机器人的现在位置的 Y 值。

2 如果按下 F4［确定］，则设定该值。
如果按下 F5［取消］，则不设定。

注
如果不存在选择打开着 *i*RPickTool 的设置画面的控制装置的机器人的传送带工作站，当按下［记录］按钮时，发出 IRPK-065“未找到此传送带的工作站”的报警。

[Y 的最小值]

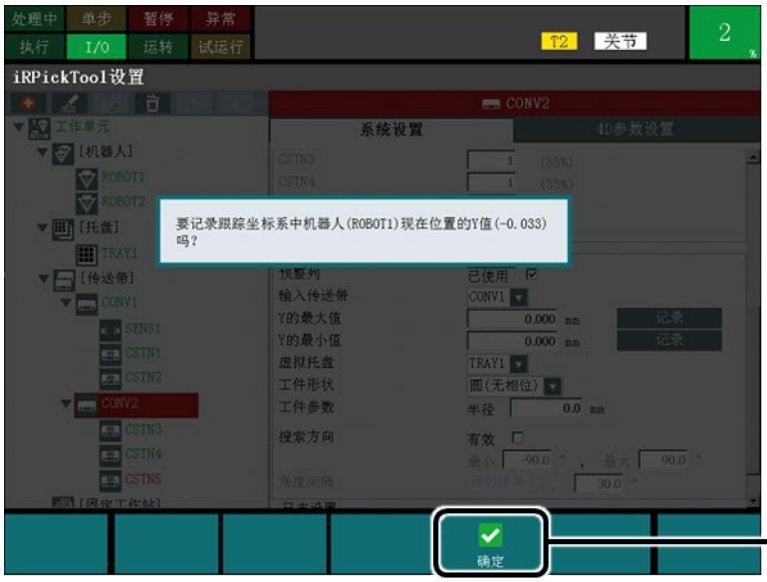
可生成整列图形的 Y 方向的下限值。是追踪坐标系的值。

注
生成的整列图形，并不是视觉的模型原点，而是工件整体不要超出该下限值。

将在“5.4 TCP 设定”设定的工具坐标系作为手动进给坐标系选择，点动 TCP，通过按下［记录］按钮，设定追踪坐标系的 Y 坐标值。例如，碰触相对传送带的进行方向的右侧皮带端，进行设定。

⚠ 注意
1 在该设定之前，需要先完成追踪坐标系的设定。
2 以打开着 *i*RPickTool 的设置画面的机器人控制装置的机器人进行碰触。

1 如果按下 F4［记录］按钮，则显示追踪坐标系中机器人的现在位置的 Y 值。



2 如果按下 F4［确定］，则设定该值。
如果按下 F5［取消］，则不设定。

> **注**
> **如果不存在选择打开着 *i*RPickTool 的设置画面的控制装置的机器人的传送带工作站，当按下［记录］按钮时，发出 IRPK-065“未找到此传送带的工作站”的报警。**

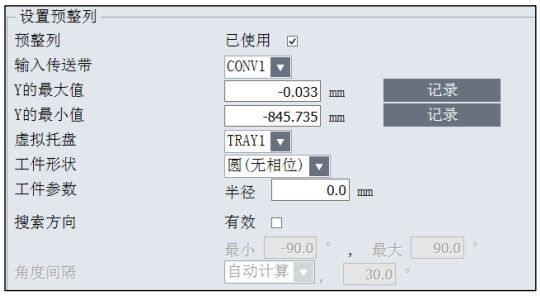
［虚拟托盘］

选择设定了整列图形的托盘。

［工件形状］

选择整列的工件的形状。可选择的工件形状为［圆(无相位)］和［长方形］。根据工件形状，切换至如下设置画面。

［工件形状］是［圆(无相位)］的情况



［工件参数］

在［半径］中设定圆的半径。

> **⚠ 注意**
> **半径设定得比实际小时，不会正确计算传送带上的工件和为进行排列而放置的工件之间的干涉检查，工件彼此之间有可能相互冲撞。将圆形以外的工件作为圆来设定时，设定完全将工件包括在内的圆的半径。**

注
对于被设定整列模式的托盘单元的间隔，如果半径设定的太大，判定调用 PKCSGENVRTRY 时基准工件与整列后的工件会碰撞，从而发生 IRPK-088“预整列的设定不正确。”的错误。

［搜索方向］

如果是没有相位的圆形工件，可在任意方向生成整列图形。

如果没有选中［有效］，则仅在一个方向生成图形，使托盘坐标系的 X 方向与追踪坐标系的 X 方向相同。

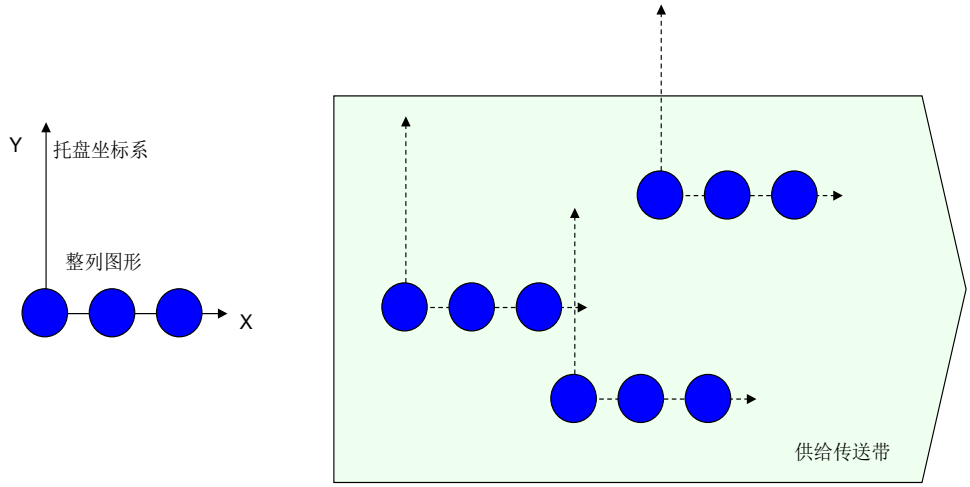


图 6.14.4 (a) 搜索方向没有变成［有效］时

如果选中［有效］，则在多个方向中，搜索托盘坐标系的 X 方向。以追踪坐标系的 X 方向为基准(0 度)，以［角度间隔］的角度间隔，在不超过［最小］、［最大］所设定的角度的范围内，搜索方向。

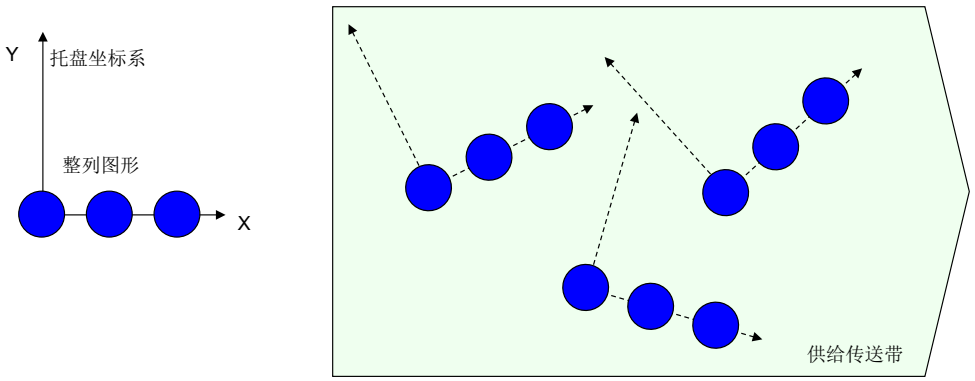


图 6.14.4 (b) 搜索方向变成［有效］时

［搜索角度间隔］

［搜索方向］有效时，可设定。可选择［自动计算］和［下一个值］。
如果选择［自动计算］，以自动计算的角度间隔，搜索整列图形。显示自动计算的角度间隔，但是不能变更。
如果选择［下一个值］，使用在文本框内输入的角度间隔，搜索整列图形。可设定的值的范围，从 1 度至 360 度。

［工件形状］是［长方形］的情况

［工件参数］

［X 方向长度］中设定托盘坐标系上的工件的 X 方向长度。［Y 方向长度］中设定托盘坐标系上的工件的 Y 方向长度。

> **⚠ 注意**
> **X 方向、Y 方向的长度设定得比实际的长方形的各边长度小时，不会正确计算传送带上的工件和为进行排列而放置的工件之间的干涉检查，工件彼此之间有可能相互冲撞。将长方形以外的工件作为长方形来设定时，设定完全将工件包括在内的长方形各边的长度。**

> **注**
> **对于设定整列模式的托盘单元的间隔，如果 X 方向、Y 方向的长度设定的太大，判定调用 PKCSGENVRTRY 时基准工件与整列后的工件会碰撞，从而发生 IRPK-088“预整列的设定不正确。”的错误。**

以上是在预整列传送带的设定。
最后，设定用于一同交付最下游机器人整列的工件的传送带的对象。
即使在传送带上传送的托盘中放置工件时，由于以 1 次的动作放置托盘内的全部工件，因此设定时，使［托盘］为［不使用］。
如果将多组一同拾取的工件放置于 1 个托盘内，则选择追加了放置工件组数量的单元的托盘。
其他的项目，请参照“6.4 传送带设定”。

6.14.5 传感器设定

进行在相同传送带上用于整列的输入传送带的设定，以及在一同拾取机器人的输出传送带上的传感器的设定。请参照“6.5 传感器设定”。
但是，在相同传送带上用于整列的输入传送带的传感器的情况，有特别的注意事项。

首先，如发生未检出，则判断为在该位置可放置用于整列的工件。

> **⚠ 注意**
> **如果发生视觉的未检出，则用于整列而放置的工件，可能与传送带上未检出的工件冲撞。**

注

在相同传送带上用于整列的输入传送带的子传感器，对触发条件选择［距离］，对触发选择［通过视觉检出］时，请根据“6.5.1 传感器详细设定”的“［距离］+［通过视觉检出］的情况”的［触发距离］项目中的注意事项，为防止未检出而尽量将［触发距离］设定得较短。

然后，如果用视觉或光电管传感器检出的位置，与实际的工件中心位置不一致，则无法正确计算传送带上的工件和搬送中用于整列而放置的工件间的干涉的检查，工件间可能发生冲撞。在参照“3.2.7.2 视觉程序”以及“6.5.3 传感器位置设定”的“6.5.3.1 不使用托盘的情况”时，请注意。

⚠ 注意

1 以相同传送带上用于整列的输入传送带的子传感器，对触发选择［通过视觉检出］时，使模型原点为工件中心。如果模型原点与工件的中心位置偏离，则无法正确计算为了与传送带上的工件整列而放置的工件间的干涉的检查，工件间可能发生冲撞。

2 以相同传送带上用于整列的输入传送带的子传感器，对触发选择［放入工件］时，通过传感器位置设定，使碰触位置为工件的中心位置，并非工件下游侧端(用光电管检知的位置)。如果碰触位置与工件的中心位置偏离，则无法正确计算为了与传送带上的工件整列而放置的工件间的干涉的检查，工件间可能发生冲撞。

此外，不使用视觉，而使用预整列传送带在［工件形状］中选择［长方形］时，需要注意设定的顺序。后述的传送带工作站的设定项目［虚拟托盘位置］的设定结束后，进行传感器位置或者目标位置的设定。

⚠ 注意

不使用视觉而使用预整列传送带在［工件形状］中选择［长方形］时，在后述的传送带工作站的设定项目［虚拟托盘位置］的设定结束后，请进行传感器位置或者目标位置的设定。否则，机器人不会在正确的位置拾取工件。

6.14.6 传送带工作站设定

如果在预整列传送带的工作站中，有与输入传送带的工作站中选择的相同的机器人，则无法进行追踪参数设定。在输入传送带的工作站的设置画面进行设定。如果在输入传送带的工作站中进行设定，则预整列传送带的工作站中，也自动反映设定。

［放弃线］

预整列传送带的工作站，指定判断是否放弃传送带上传送生成的整列图形上的放置位置的位置。预整列的情况，同时从输入传送带的工作站取得工件信息，以及从预整列传送带的工作站取得用于整列该工件的放置位置的信息，因此根据从对整列前的工件开始追踪动作，至该工件整列结束的时间的平均，以及传送带的移动速度，决定放弃线。详细信息请参照“6.6 传送带工作站设定”的［放弃线］项目。

［根据 Y 排序］

预整列传送带的工作站中，不能使用根据 Y 排序功能。母传送带中选择了［预整列］的［已使用］时，自动设定为无效，无法变更。

［停止/启动传送带］

预整列传送带的工作站中，不能使用停止/启动传送带功能。

［跳过此传送带工作站］

预整列传送带的工作站中，故意要对象机器人休息而使其有效时，输入传送带上对应的传送带工作站中，也将跳过此传送带工作站设定为有效。

［设置预整列］

预整列传送带中选择［输入传送带］时，选择与在输入传送带上的传感器之后的传送带工作站相同机器人的子传送带工作站中，追加显示预整列的设定项目。预整列传送带中没有传感器。该传送带工作站起到传感器的作用。使用后面叙述的 KAREL 程序，生成用于整列的放置位置的数据。
在母传送带上对［工件形状］选择［长方形］时的设定项目如下。

［虚拟托盘位置］

工件的检测位置和虚拟托盘的原点之相对位置。
［工件形状］为［圆(无相位)］时，将会自动进行计算，因而不显示此项目。
［工件形状］为［长方形］时，按照如下步骤予以设定。

> **⚠ 注意**
> **输入传送带上的传感器的触发为［放入工件］时，在进行此虚拟托盘位置的设定后，请进行传感器位置或者目标位置的设定。已经完成传感器位置或设目标位置的设定时，在进行虚拟托盘位置的设定后，请重新进行传感器位置或者目标位置的设定。**

1 在预整列传送带的设置画面上，在［设置预整列］的［预整列］中暂时去掉［已使用］的勾选。(如下步骤所需的操作。)
2 在输入传送带的设置画面上，选择设定了预整列的排列模式的托盘。
3 在输入传送带的子传感器的设置画面上，进行托盘位置或者传感器位置或目标位置的设定。
   - 输入传送带中的传感器的触发为［通过视觉检出］时，进行托盘位置的设定。请参照“6.5.5 托盘位置设定”。
   - 输入传送带中的传感器的触发为［放入工件］时，触发条件若是［DI］或者［HDI］，进行传感器位置的设定；触发条件若是［距离］，进行目标位置的设定。请参照“3.2.6 传感器位置设定”或“6.5.6 目标位置的设定”中的“6.5.6.2 使用托盘的情况”。

   这里，假设检测出的工件放置在托盘的单元 1 上，进行托盘坐标系的原点、X 方向点的碰触。

> ⚠ **注意**
> **在进行碰触时，务必假设已将被检测出的工件放在托盘的单元 1 上进行碰触。否则，不会正确计算传送带上的工件和为进行排列而放置的工件之间的干涉检查，工件彼此之间有可能相互冲撞。**

4 记录下已设定的［托盘位置］或者［传感器位置］或［目标位置］的 X、Y、R 的值。
5 在输入传送带的设置画面上，将［托盘］复原为［未使用］。
6 在预整列传送带的设置画面上，在［设置预整列］的［预整列］中再次选中［已使用］。显示“设置为预整列需要初始化设置，继续吗？”，选择［OK］。这样，［输入传送带］会成为未选择，所以要再次进行选择。
7 在传送带工作站的设置画面上，在［虚拟托盘位置］中输入 4 中记录下来的 X、Y、R 的值。

［生成线］

以追踪坐标系上的 X 值指定整列图形的生成开始位置。输入传送带中，传感器的触发是［通过视觉检出］时，则设定于相机视野的传送带移动方向的下游侧端更下游侧。输入传送带中，传感器的触发是［放入工件］时，则设定于仅距离传感器位置工件至少在下列值以上距离的下游侧位置。

- ［工件形状］若是［圆(无相位)］，该值是［工件参数］内指定的半径的值。
- ［工件形状］若是［长方形］，该值是长方形的对角线的长度的一半的值。如图 6.14.6(a)所示，使用在［工件参数］内指定的 X 方向的长度 x 与 Y 方向的长度 y，计算该值。

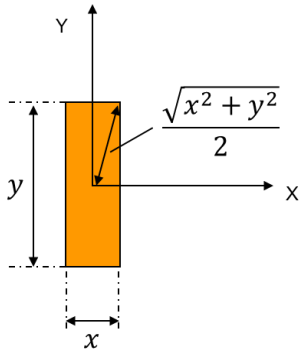


图 6.14.6(a)生成线的计算方法

⚠ 注意

1 在输入传送带使用视觉时，如果在视野中生成图形，可能在与生成的图形干涉的位置，检出新的工件。这种情况，用于整列而放置的工件，可能与新检出的工件冲撞。

2 即使在输入传送带上不使用视觉的情况下，若没有正确设定生成线，则在与所生成的模式相互干涉的位置会重新检测出工件，用于排列而放置的工件有可能与此新检测出的工件发生冲撞。

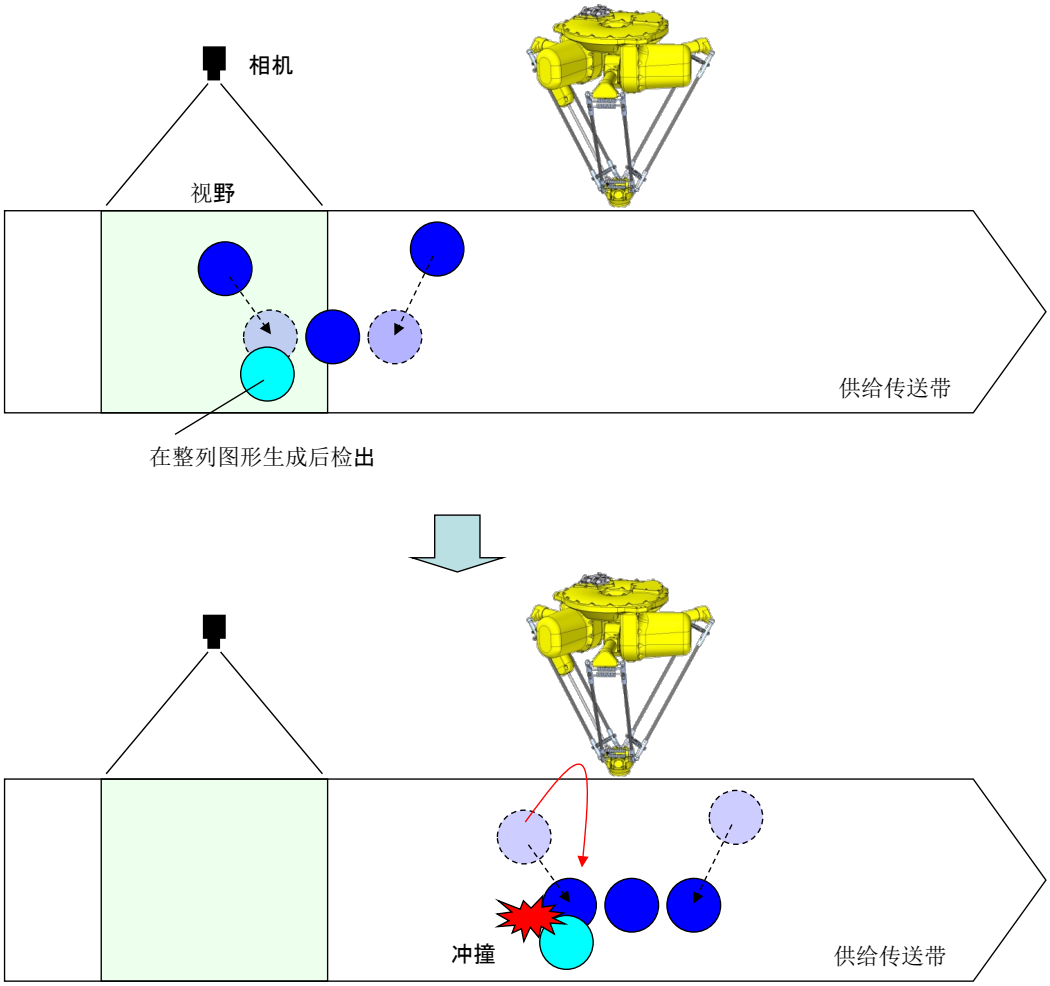


图 6.14.6(b)

[发信线]

在该传送带工作站以及相应的输入传送带工作站，指定将工件、用于整列的放置位置的数据发信至下游工作站的位置。

通常，在［通过视觉检出］的情况，如果模型原点超越放弃线，则工件数据送至下游的传送带工作站。此外，在［放入工件］的情况，如果以工件上的传感器位置的设定碰触的位置超越放弃线，则工件数据送至下游的传送带工作站。虽然工件数据送至下游的传送带工作站，但是工件的一部分还未超越放弃线。新生成的图形，与该工件的一部分发生干涉，用于整列而放置的工件可能和该工件发生冲撞。

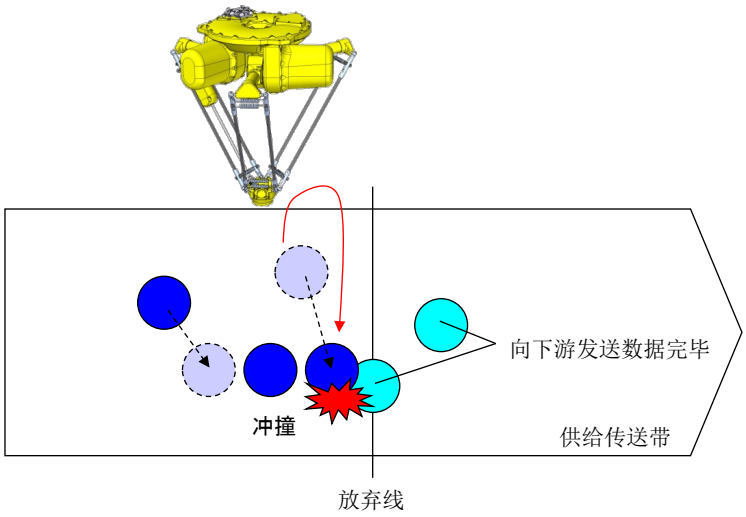


图 6.14.6(c)

为了在整列图形生成时进行正确的干涉检查，即使在超越了放弃线后，也请保持工件数据，直至超越发信线。

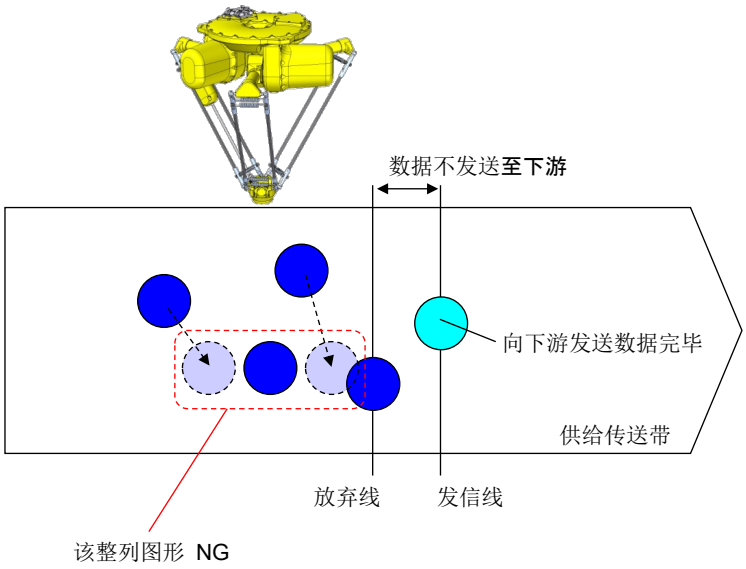


图 6.14.6(d)

在生成整列图形时的传送带上的位置，于整列图形整体位于比放弃线更上游侧的位置，生成整列图形。
［工件形状］是［圆(无相位)］的情况，设定为工件半径或更大值。［长方形］的情况，设定长方形的对角线的长度的一半或一半以上的值。请参阅图 6.14.6(a)。

> ⚠ **注意**
> **如果发信线未正确设定，则用于整列而放置的工件可能与传送带上其他的工件冲撞。**

［整列距离］

在传送带的移动方向，对于在一定距离内的工件生成整列图形。
如果构成整列图形的各工件过度远离传送带移动方向，即使拾取用于整列的工件，对应的放置位置也已经过了追踪范围，无法搬运。

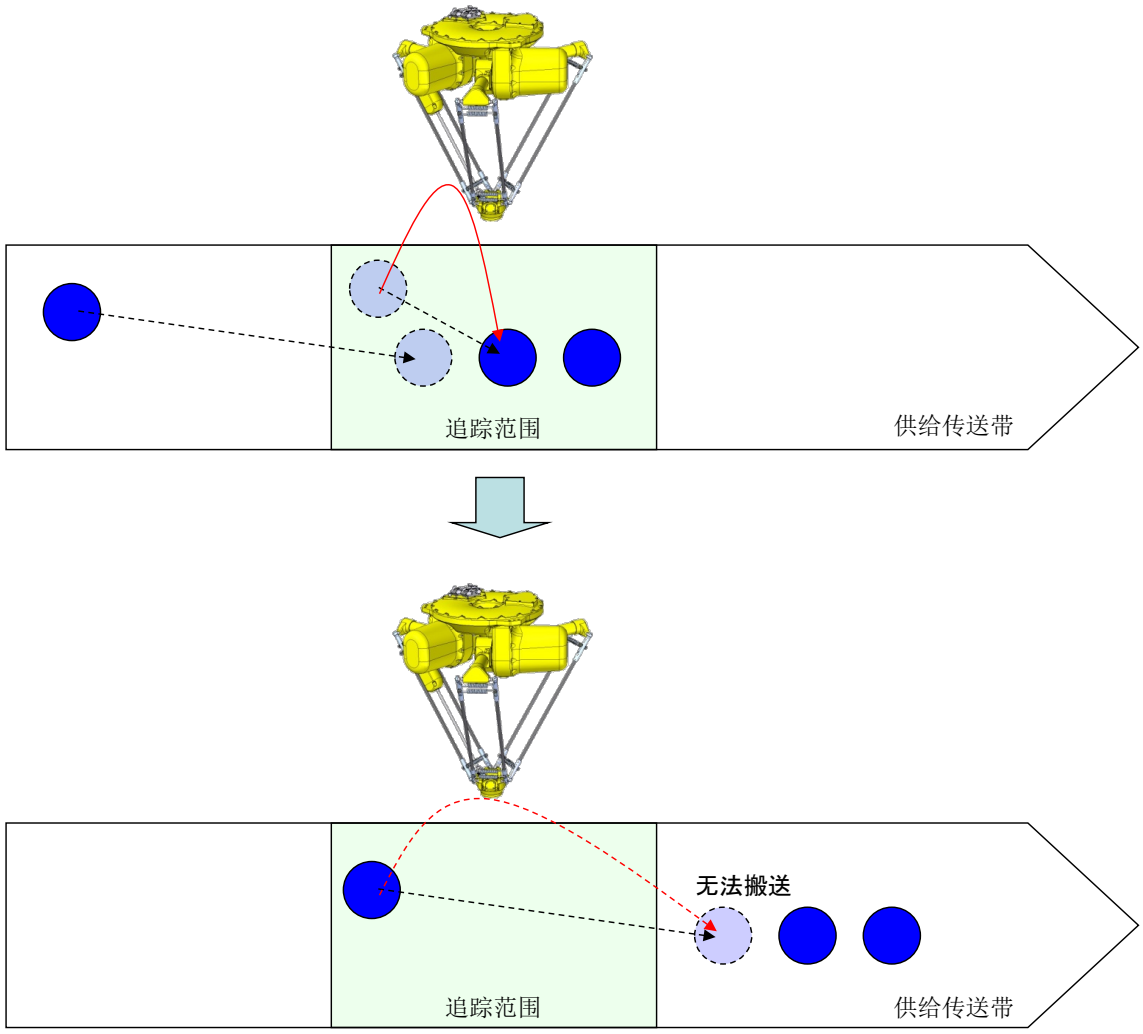


图 6.14.6(e)

如果选择［自动计算］，则使用输入传送带的工作站的上游边界与放弃线之间的距离。
如果选择［下一个值］，则使用在文本框内指定的距离。

［各项目的位置关系］

“生成线”、“发信线”、“上游边界”、“下游边界”、“放弃线”的位置关系，总结如下图。但是，有关“发信线”和“下游边界”，是根据本说明书设定的结果、“发信线”也可能在下游侧。

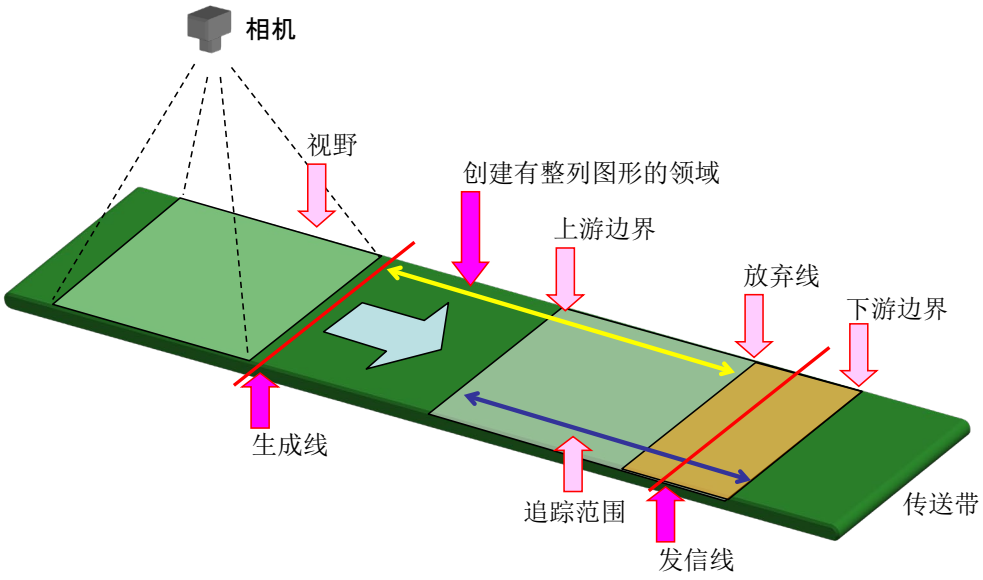


图 6.14.6(f)

6.14.7 固定工作站设定

使用固定工作站时，请参照“6.7 固定工作站设定”。

6.14.8 KAREL 程序

这里对预整列功能专用的 KAREL 程序的规格进行说明。全部有关传送带工作站的程序。

6.14.8.1 PKCSGENVRTRY.PC

在指定的传送带工作站，生成 1 个虚拟托盘。

参数 1：

指定传送带工作站的 ID。可通过 PKCSGETID 取得 ID。
预整列传送带的工作站中，对应的输入传送带的工作站，只能指定传感器之后配置的工作站。要指定预整列传送带上没有的工作站时，发出 IRPK-087“这是预整列的专用命令。”的错误。此外，预整列传送带的工作站中，如果对应的输入传送带的工作站指定了非传感器之后配置的工作站时，发出 IRPK-035“指定的传送带工作站无效。”的错误。

参数 2：

指定存储处理状态的数值寄存器序号。
存储的状态如下。

0： 生成 1 个虚拟托盘。
1： 无法生成以干涉为起因的虚拟托盘。
2： 工件间的距离比设定值或自动计算的值(上游边界和输入传送带上的放弃线之间的距离)大，因此不生成虚拟托盘。
3： 超越生成线的工件的数量少于生成虚拟托盘所需的工件，因此未生成虚拟托盘。

参数 3（可省略）：

指定特定的模型 ID，想要生成虚拟托盘时，指定模型 ID。
可以省略，通常不指定。

使用例：

在预整列传送带工作站生成虚拟托盘。预整列传送带工作站的 ID 存储于 R[12]，处理的状态存储于 R[4]。

```
CALL PKCSGENVRTRY(传送带工作站 ID=R[12:CStnIDOut],状态 Reg=4)
```

6.14.8.2 PKCSGETQPRGR.PC

从输入传送带的工作站取得 1 个工件的信息，从预整列传送带的工作站取得 1 个对应放置位置的单元的信息。取得的工件、单元信息存储于视觉寄存器。在输入传送带使负载平衡有效时，根据负载平衡的设定取得信息。

参数 1：

指定传送带工作站的 ID。

可通过 PKCSGETID 取得 ID。

只能指定预整列传送带工作站。要指定预整列传送带上没有的工作站时，发出 IRPK-087“这是预整列的专用命令。”的错误。

参数 2：

指定存储取得的工件信息的视觉寄存器序号。

参数 3：

指定存储取得的相应放置位置的单元信息的视觉寄存器序号。

指定与参数 2 相同视觉寄存器序号时，发出 IRPK-057“指定的参数值无效。”的错误。

参数 4：

指定存储处理状态的数值寄存器序号。

存储的状态如下。不会只取得工件的信息、单元的信息的其中一个信息。

0： 取得了工件的信息以及相应放置位置的单元信息。

1： 未取得工件的信息以及相应放置位置的单元信息。

使用例：

从输入传送带的工作站和预整列传送带的工作站，取得未处理的工件的信息，和放置位置的单元信息。预整列传送带工作站的 ID 存储于 R[12]，VR[1]中存储有工件的信息，VR[2]中存储有放置位置的单元信息，R[3]中存储有处理的状态。

```
CALL PKCSGETQPRGR(传送带工作站 ID=R[12:CStnIDOut],拾取补偿用 VR=1,放置补偿用 VR= 2,状态 Reg=3)
```

6.14.8.3 PKCSTRGPRGR.PC

如果以 PKCSGETQPRGR 取得的工件或放置位置的单元在追踪范围内，则将该编码器的值设定为追踪动作的触发器。仅待机指定的时间，直至取得工件或放置位置的单元传送至追踪范围。

参数 1：

指定传送带工作站的 ID。

可通过 PKCSGETID 取得 ID。

只能指定预整列传送带的传送带工作站或输入传送带的传送带工作站。指定其他的传送带工作站时，发出 IRPK-087“这是预整列的专用命令。”的错误。

参数 2：

以 ms 指定取得的工件或放置位置的单元被传送至追踪范围之前的待机时间。如果指定负数，则无限等待。如果指定为 0，则无需待机，进入下一个处理。

参数 3：

指定存储处理状态的数值寄存器序号。

存储的状态如下。

0： 追踪动作的触发器设定完成。

1： 取得的工件、或相应放置位置的单元未在时间内传送至追踪范围。

使用例：

在输入传送带的工作站设定追踪动作的触发器。

输入传送带工作站的 ID 存储于 R[11]，处理的状态存储于 R[3]。此外，如果工件未到达追踪范围，则等待 100ms。

```
CALL PKCSTRGPRGR(传送带工作站 ID=R[11:CStnIDIn],超时值(ms)=100,状态 Reg=3)
```

6.14.8.4 PKCSACKQPRGR.PC

以 PKCSGETQPRGR 处理的状态为 0，分配与工件相应放置位置的单元信息后，将对该工件和放置位置的单元进行的处理通知传送带工作站。

参数 1：

指定传送带工作站的 ID。

可通过 PKCSGETID 取得 ID。

只能指定预整列传送带的传送带工作站。要指定预整列传送带上没有的传送带工作站时，发出 IRPK-087“这是预整列的专用命令。”的错误。

参数 2：

以下面的任意值，指定处理的结果。

1： 成功

在正常将工件移动至相应放置位置时进行指定。

2： 返回

不立即处理工件，暂时返回传送带工作站的情况下进行指定。

3： 发信

不依照预定处理工件的情况下进行指定。

例如，使用机器人的 TP 程序评价被分配的工件的位置、模型 ID 等，故意阻止对该工件进行处理的情况，属于这种情况。这种情况，该工件的信息被转移至下一个传送带工作站。

4： 删除

工件的处理失败的情况下，进行指定。

这种情况，该工件的信息被删除。有关相同整列图形的，未处理的工件，在下一次调出 PKCSGENVRTRY 时再次利用。此外，通过使用 PKCSGETQCELL，将整列的工件一起拾取的机器人，也可拾取处理被中止，整列图形未完成的工件。

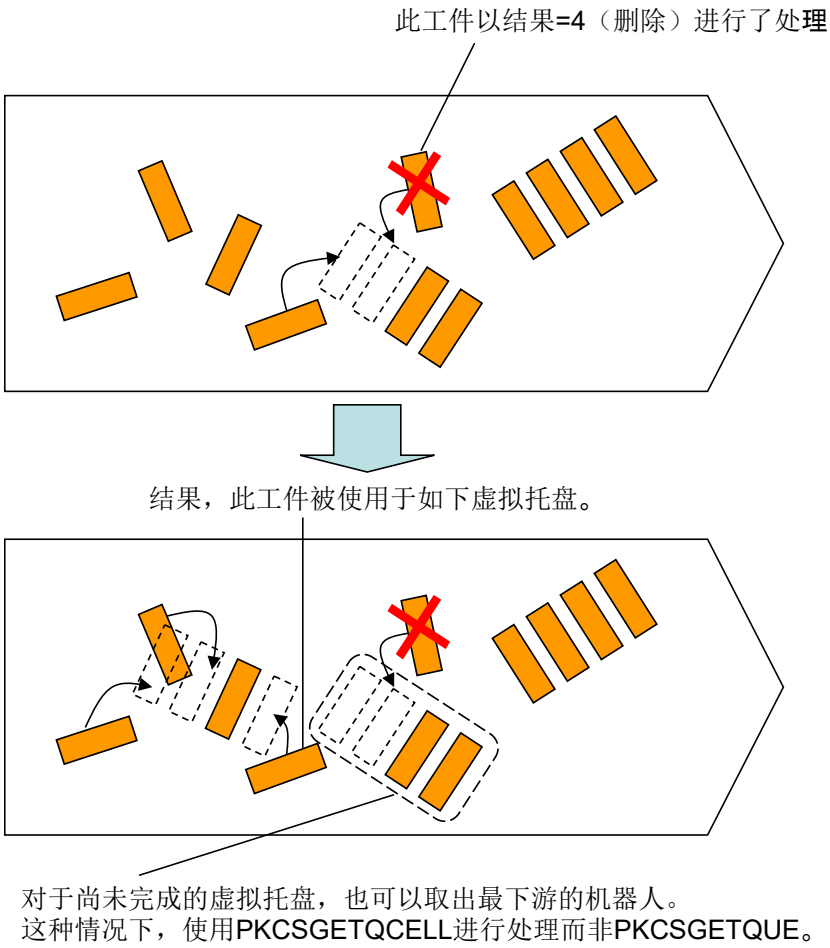


图 6.14.8.4(a)

使用例：

返回对应工件的放置位置正常处理。ID 存储于 R[12]。

```
CALL PKCSACKQPRGR(传送带工作站 ID=R[12:CStnIDOut],成功)
```

6.14.9 机器人的程序

介绍在相同传送带上进行整列的程序例，并说明思考方式等。请参考介绍的程序例，配合客户负责的系统进行变更。

6.14.9.1 传送带—传送带(相同，最上游)的构成的样本程序

介绍在传送带上紧跟传感器而配置的机器人的程序例。

与一般的追踪的系统不同，机器人的程序由以下 2 个程序构成。

- 主程序
- 在相同传送带上进行整列动作的程序

进行整列动作的程序，是从主程序调出的追踪程序。
而这里介绍的程序例，使用以下的数值寄存器以及位置寄存器。请根据需要，适当变更寄存器序号。

表 6..14.9.1(a) 数值寄存器

R[1]	**周期停止标记。如果对该数值寄存器设定 1，则系统周期停止。**
R[2]	**PKCSGETQPRGR 的状态**
R[3]	**PKCSTRGPRGR 的状态**

R[4]	**PKCSGENVRTRY 的状态**
R[11]	**存储输入传送带 CONV1 的传送带工作站 ID。**
R[12]	**存储预整列传送带 CONV2 的传送带工作站 ID。**

表 6..14.9.1(b) 位置寄存器

PR[1]	**传送带上的拾取位置(放置位置)**
PR[10]	**用于生成拾取动作的接近点的工具偏移量**
PR[12]	**用于生成放置动作的接近点的工具偏移量**

主程序

系统的主程序。

MAIN1.TP

```
  1:   R[1:周期停止]=0
  2:   CALL PKCSGETID('CONV1',工作站 ID Reg=11)
  3:   CALL PKCSGETID('CONV2',工作站 ID Reg=12)
  4:
  5:   UTOOL_NUM=1
  6:   UFRAME_NUM=0
  7:   J P[1:Perch Pos] 50% FINE
  8:
  9:   CALL PKWCSTART
 10:
 11:   LBL[100]
 12:   CALL PKCSGENVRTRY(传送带工作站 ID=R[12:CStnIDOut],状态 Reg=4)
 13:   IF R[4:生成状态]=0,JMP LBL[200]
 14:   IF R[4:生成状态]=1,JMP LBL[300]
 15:   IF R[4:生成状态]=2,JMP LBL[400]
 16:   WAIT      .02(sec)
 17:   IF R[1:周期停止]=1,JMP LBL[900]
 18:   JMP LBL[100]
 19:
 20:   LBL[200]
 21:   CALL PREGROUPING1
 22:   IF R[1:周期停止]=1,JMP LBL[900]
 23:   JMP LBL[100]
 24:
 25:   LBL[300]
 26:   CALL PKCSGETQUE(传送带工作站 ID=R[11:CStnIDIn],连续的标记=1,超时值(ms)=0,补偿用 VR=1,
       状态 Reg= 2)
 27:   IF R[2:GETQ 状态]<>0,JMP LBL[100]
 28:   CALL PKCSACKQUE(传送带工作站 ID=R[11: CStnIDIn],发信)
 29:   JMP LBL[100]
 30:
 31:   LBL[400]
 32:   CALL PICKCS
 33:   IF R[1:周期停止]=1,JMP LBL[900]
 34:   CALL DROPFS
 35:   IF R[1:周期停止]=1,JMP LBL[900]
 36:   JMP LBL[100]
 37:
 38:   LBL[900]
 39:   CALL PKWCEND
[End]
```

在第 12 行生成虚拟托盘。如果生成成功，跳至第 20 行，调出在传送带上进行整列的程序 PREGROUP1.TP。
PREGROUP1.TP 只继续可进行整列动作的部分，并结束。PREGROUP1.TP 结束后，确认周期停止标记，再次生成虚拟托盘。
生成的状态是 1(由于干涉，无法生成虚拟托盘)时，跳至第 25 行，故意跳过最下游的 1 个工件。这种情况，工件密集传送，机器人无法处理的可能性较高，因此故意跳过。
生成的状态是 2(工件间的距离较大)时，跳至第 31 行，从传送带上拾取 1 个工件，放置于固定工作站。这种情况，不常传送工件，因此机器人的动作路径变长，可以认为即使进行至传送带外的动作，跳过其他工件的可能性也较低。有关 PICKCS.TP、DROPFS.TP 的详细信息，请参照“6.9.3 传送带—固定工作站的构成的样本程序”。
生成的状态是 3(超越生成线的工件数量少于虚拟托盘所需的工件数量)时，稍许待机至需要数量的工件传送后，再反复调出生成虚拟托盘的指令。

进行整列动作的程序

在相同传送带上进行整列动作的程序。

PREGROUPING1.TP

```
  1:   UTOOL_NUM=1
  2:   UFRAME_NUM=0
  3:
  4:   LBL[100]
  5:   STOP_TRACKING
  6:   CALL PKCSGETQPRGR(传送带工作站 ID=R[12:CStnIDOut],拾取补偿用 VR=1,放置补偿用 VR=2,状态 Reg
=2)
  7:   IF R[2:GETQ 状态]=0,JMP LBL[200]
  8:   JMP LBL[800]
  9:
 10:   LBL[200]
 11:   CALL PKCSTRGPRGR(传送带工作站 ID=R[11:CStnIDIn],超时值(ms)=100,状态 Reg=3)
 12:   IF R[3:触发器状态]=0,JMP LBL[300]
 13:   IF R[1:周期停止]=1,JMP LBL[900]
 14:   JMP LBL[200]
 15:
 16:   LBL[300]
 17: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 18: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[1]
 19:   CALL VACUUM_ON
 20: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 21:
 22:   LBL[400]
 23:   STOP_TRACKING
 24:   CALL PKCSTRGPRGR(传送带工作站 ID=R[12:CStnIDOut],超时值(ms)=100,状态 Reg=3)
 25:   IF R[3:触发器状态]=0,JMP LBL[500]
 26:   JMP LBL[400]
 27:
 28:   LBL[500]
 29: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 30: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[2]
 31:   CALL VACUUM_OFF
 32: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
 33:   CALL PKCSACKQPRGR(传送带工作站 ID=R[12:CStnIDOut],成功)
 34:   JMP LBL[100]
 35:
 36:   LBL[800]
 37:   END
 38:
 39:   LBL[900]
 40:   CALL PKCSACKQPRGR(传送带工作站 ID=R[12:CStnIDOut],返回)
[End]
```

在第 6 行取得与工件数据对应的放置位置的数据。取得成功后，跳至第 10 行。如果未取得，跳至第 37 行，结束程序。

在第 11 行设定用于拾取动作的触发器。在 PKCSTRGPRGR 的第 2 个参数中，指定为 100ms，因此如果调出 PKCSTRGPRGR 时拾取的工件不在追踪范围内，则在待机时间 100ms 退出。这种情况，R[3]中是 1，PREGROUPING1.TP 确认周期停止标记，再次调出 PKCSTRGPRGR，等待工件进入追踪范围内。周期停止标记为 1 时，跳至第 39 行，调出 PKCSACKQPRGR，返回工件数据和放置位置的数据后，结束程序。

在第 17 行～20 行，进行拾取动作。

在第 24 行设定用于放置动作的触发器。在 PKCSTRGPRGR 的第 2 个参数中，指定为 100ms，因此如果调出 PKCSTRGPRGR 时，用于放置工件的位置不在追踪范围内，则最长待机 100ms，进入下一行。这种情况，R[3]是 1，再次调出 PKCSTRGPRGR，等待放置工件的位置进入追踪范围内。

在第 29 行～32 行，进行放置动作。使用与拾取动作相同的位置寄存器。

在第 33 行调出 PKCSACKQPRGR，通知传送带工作站，整列结束。然后跳至第 4 行，取得与下一个工件数据对应的放置位置的数据。

6.14.9.2 传送带—传送带(相同，下游)的构成的样本程序

介绍在传送带上非紧跟传感器而配置的机器人的程序例。不生成整列图形的点，与紧跟传感器配置的机器人不同。与“6.14.9.1 传送带—传送带(相同，最上游)的构成的样本程序”不同，仅介绍主程序。

主程序

系统的主程序。

MAIN1.TP

```
  1:   R[1:周期停止]=0
  2:   CALL PKCSGETID('CONV1',工作站 ID Reg=11)
  3:   CALL PKCSGETID('CONV2',工作站 ID Reg=12)
  4:
  5:   UTOOL_NUM=1
  6:   UFRAME_NUM=0
  7:   J P[1:Perch Pos] 50% FINE
  8:
  9:   CALL PKWCSTART
 10:
 11:   LBL[100]
 12:   CALL PREGROUPING1
 13:   IF R[1:周期停止]=1,JMP LBL[900]
 14:
 15:   CALL PICKCS
 16:   IF R[1:周期停止]=1,JMP LBL[900]
 17:   CALL DROPFS
 18:   IF R[1:周期停止]=1,JMP LBL[900]
 19:   JMP LBL[100]
 20:
 21:   LBL[900]
 22:   CALL PKWCEND
[End]
```

在第 12 行上调出传送带上进行整列的程序 PREGROUP1.TP。PREGROUP1.TP 只继续可进行整列动作的部分，并结束。PREGROUP1.TP 与“6.14.9.1 传送带—传送带(相同，最上游)的构成的样本程序”的“进行整列动作的程序”相同。

在没有可取得的数据的情况下结束整列动作时，在第 15 行调出最上游机器人不能整列，拾取跳过的工件的程序 PICKCS.TP。在第 17 行，调出将拾取的工件放置于固定工作站的程序 DROPFS.TP。有关 PICKCS.TP、DROPFS.TP 的详细信息，请参照“6.9.3 传送带—固定工作站的构成的样本程序”。
在最上游的机器人无法整列，处理 1 个跳过的工件后，跳至第 11 行，再次调出进行整列的程序 PREGROUP1.TP。

6.14.9.3 传送带—传送带的构成的样本程序

介绍一同拾取整列的工件，交付给其他传送带的最下游机器人的程序例。与“6.9.2 传送带—传送带的构成的样本程序”不同，仅介绍进行拾取动作的程序。主程序与进行放置动作的程序，请分别参照“6.9.2.1 主程序”和“6.9.2.3 进行放置动作的程序”。

进行拾取动作的程序

针对只汇总拾取整列图案已完成工件时的程序、以及未完成整列图案的工件也拾取时的程序进行说明。

只拾取整列图形已完成工件的程序例子

PICK1.TP

```
  1:    UTOOL_NUM=1
  2:    UFRAME_NUM=0
  3:
  4:    LBL[100]
  5:    STOP_TRACKING
  6:    CALL PKCSGETQUE(传送带工作站 ID=R[11:CStnIDIn],连续的标记=1,超时值(ms)=100,补偿用 VR=1,
        状态 Reg= 2,2)
  7:    IF R[2:GETQ 状态]=0,JMP LBL[200]
  8:    IF R[1:周期停止]=1,JMP LBL[900]
  9:    JMP LBL[100]
 10:
 11:    LBL[200]
 12: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 13: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[1]
 14:    CALL VACUUM_ON
 15: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 16:    CALL PKCSACKQUE(R[11:CStnIDIn], 1)
 17:
 18:    LBL[900]
[End]
```

对于通常的传送带—传送带的构成的样本程序，以第 6 行的 PKCSGETQUE 指定模型 ID 的点不同。在整列图形的托盘中，假设整列图形的层 1 中设定有模型 ID1，层 2 的单元中设定有模型 ID2。通过以 PKCSGETQUE 的参数 6 指定模型 ID2，仅层 1 的单元全部被处理的情况下，取得信息。

然后是拾取整列图形未完成，有缺齿的工件时的程序。

也拾取整列图形未完成工件的程序例子

PICK2.TP

```
  1:   UTOOL_NUM=1
  2:    UFRAME_NUM=0
  3:
  4:    LBL[100]
  5:    STOP_TRACKING
  6:    CALL PKCSGETQCELL(传送带工作站 ID =R[11:CStnIDIn],单元编号=4,超时值(ms)=100,补偿用 VR=1
        状态 Reg=2,空单元数 Reg=21)
  7:    IF R[2:GETQ 状态]=0,JMP LBL[200]
  8:    IF R[1:周期停止]=1,JMP LBL[900]
  9:    JMP LBL[100]
 10:
 11:    LBL[200]
```

```
 12: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 13: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[1]
 14:    CALL VACUUM_ON
 15: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 16:    CALL PKCSACKQUE(传送带工作站 ID =R[11:CStnIDIn],成功)
 17:
 18:    LBL[900]
[End]
```

与 PICK1.TP 不同，在第 6 行，不使用 PKCSGETQUE，而使用 PKCSGETQCELL 的点。整列 3 个工件的情况，为了一同拾取，以参数 2 指定层 2 中追加的单元 4。R[21]中返回整列图形的缺齿数量。

6.14.10 基准位置设定

在输入传送带以及一同交付整列的工件的传送带上，设定基准位置。请参照“6.10 基准位置设定”。

6

6.14.11 机器人的位置示教

在基准位置的设定后，进行机器人的位置示教。进行整列动作的程序，仅示教拾取位置。由于使用相同位置数据，因此不需要放置位置的示教。请参照“6.11 机器人的位置示教”。

6.14.12 追踪动作的微调整

请参照“6.12 追踪动作的微调整”。作为预整列时的特别记载事项，有关追踪坐标系的设定误差的调整方法，需要注意。

6.14.12.1 追踪坐标系的设定误差的调整

使用 ADJ_OFS 时，对“6.14.9.1 传送带－传送带(相同，最上游)的构成的样本程序”中介绍的进行整列动作的程序进行以下修正。
而这里介绍的程序例，将位置寄存器的 20 号作为存储调整量的地方使用。请根据需要适当变更。

PREGROUPING1.TP

```
  1:    UTOOL_NUM=1
  2:    UFRAME_NUM=0
  3:
  4:    LBL[100]
  5:    STOP_TRACKING
  6:    CALL PKCSGETQPRGR(传送带工作站 ID=R[12:CStnIDOut],拾取补偿用 VR=1,放置补偿用 VR=2,状态 Reg
=2)
  7:    IF R[2:GETQ 状态]=0,JMP LBL[200]
  8:    JMP LBL[800]
  9:
 10:    LBL[200]
 11:    CALL PKCSTRGPRGR(传送带工作站 ID=R[11:CStnIDIn],超时值(ms)=100,状态 Reg =3)
 12:    IF R[3:触发器状态]=0,JMP LBL[300]
 13:    IF R[1:周期停止]=1,JMP LBL[900]
 14:    JMP LBL[200]
 15:
 16:    LBL[300]
 17:    CALL ADJ_OFS(1, 1, 20, 3)
 18: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
 19: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[3]
 20:    CALL VACUUM_ON
 21: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
 22:
```

```
 23:    LBL[400]
 24:    STOP_TRACKING
 25:    CALL PKCSTRGPRGR(传送带工作站 ID =R[12:CStnIDOut],超时值(ms)=100,状态 Reg =3)
 26:    IF R[3:触发器状态]=0,JMP LBL[500]
 27:    JMP LBL[400]
 28:
 29:    LBL[500]
 30:    CALL ADJ_OFS(1, 2, 20, 4)
 31: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[4] Tool_Offset,PR[12]
 32: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[4]
 33:    CALL VACUUM_OFF
 34: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[4] Tool_Offset,PR[12]
 35:    CALL PKCSACKQPRGR(传送带工作站 ID=R[12:CStnIDOut],成功)
 36:    JMP LBL[100]
 37:
 38:    LBL[800]
 39:    END
 40:
 41:    LBL[900]
 42:    CALL PKCSACKQPRGR(传送带工作站 ID=R[12:CStnIDOut],返回)
[End]
```

- 在即将进行拾取动作、放置动作之前，调出 ADJ_OFS 调整补偿数据。
- 在拾取动作和放置动作使用存储于相同位置寄存器中的调整量。
- 设定调整量时，为了在拾取位置的正上方停止，请如下所示，将动作指令和待机指令插入接近点下面一行。

```
 17:    CALL ADJ_OFS(1, 1, 20, 3)
 18: L PR[1] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[10]
 19: L PR[1] 4000mm/sec FINE VOFFSET,VR[3] Tool_Offset,PR[15]
 20:    WAIT   10.00(sec)
 21: L PR[1] 4000mm/sec CNT0 VOFFSET,VR[3]
 22:    CALL VACUUM_ON
```

- 详细信息，请参照“6.12.2 追踪坐标系的设定误差的调整”。

7 Plug & Play 参考

本章针对您在使用 *i*RPickTool（基本功能+ Plug & Play）时参考的 Plug & Play 菜单项目详情及设定方法进行说明。
此外，还针对标准 TP 程序及 KAREL 程序等具体程序进行说明。

关于 *i*RPickTool（基本功能 + Plug & Play）的启动调试步骤，请参阅“4 *i*RPickTool（基本功能+Plug & Play）的启动调试步骤”。

*i*RPickTool 的画面菜单与本说明书相关项目如下图所示。
在以下项目中，“7.1 工作单元设定”针对 Plug & Play 各对象的最大数进行说明。“7.4 固定工作站设定”针对只可在 Plug & Play 设定的［跳过］设定进行说明。
“7.2 抓手和区域的设定”、“7.3 操作设定”为只可在 Plug & Play 中设定的项目。关于其他项目，请参阅“6 基本功能参考”中的各项目。

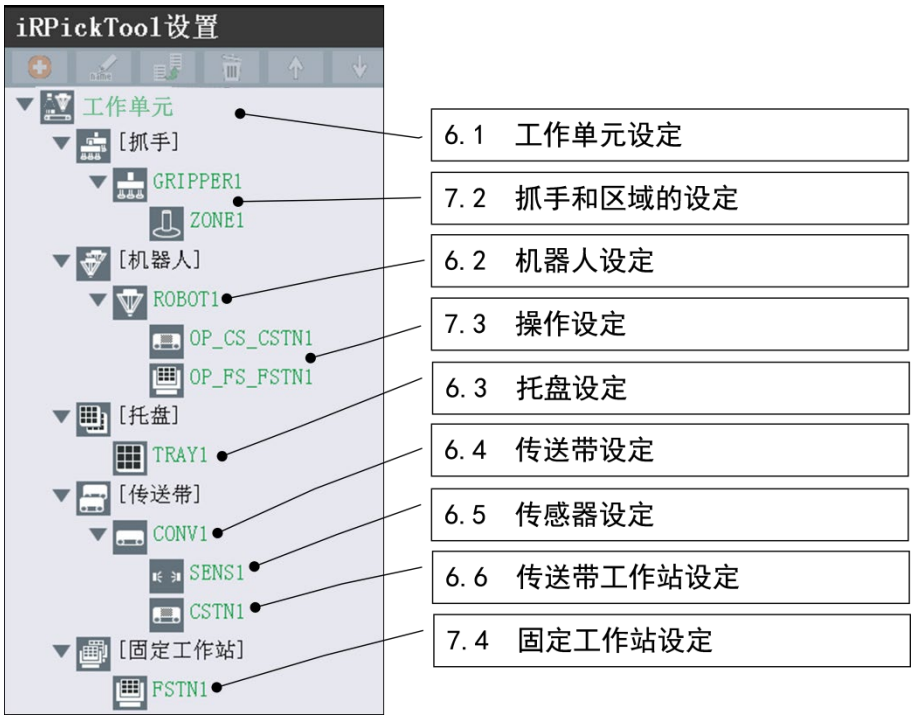


图 7(a) 与 *i*RPickTool 菜单相关的项目

除了图 7(a)的菜单项目之外，作为 Plug & Play 的参考，“7.5 机器人的程序”、“7.9 KAREL 程序”、“7.10 标准 TP 程序”对程序进行说明。“7.6 基准位置设定”、“7.7 机器人的位置示教”针对机器人位置的示教进行说明。

还有，“7.8 追踪动作的微调整”针对误差的调整，特别 Plug & Play 标准 TP 程序中的调整方法进行说明。

另外，“7.11 预整列设定”对预整列功能的专业化内容进行说明。
“7.12 Plug & Play 的缓冲区功能”针对在 Plug & Play 中使用的缓冲功能进行说明。

> ⚠ **注意**
> **在机器人间通信的设定中连接的所有机器人控制装置中，作为软件选项，请勿订购 *i*RPickTool Basic，而应订购 *i*RPickTool。**

基本操作

*i*RPickTool 按每个机器人或传送带等系统构成对象进行设定。
关于 *i*RPickTool 的基本操作，请参阅“6 基本功能参考”中的“基本操作”。

7.1 工作单元设定

关于工作单元的设定画面，请参阅“6.1 工作单元设定”。
另外，关于 Plug & Play 工作单元的设定步骤，请参阅“4 *i*RPickTool（基本功能+Plug & Play）的启动调试步骤”。

对象数量设定

工作单元可更改各对象的最大数量。
Plug & Play 可变更的最大数量如表 7.1(a)所示。

各对象的最大数的变更步骤，请参照“6.1 工作单元设定”的“对象数量设定”。

表 7.1(a)各对象可变更的最大数量

对象	最大数量
抓手	32
区域	100
机器人	32
托盘	16
传感器	32
传送带	32
传送带工作站	84
固定工作站	84

7.2 抓手和区域的设定

抓手的设定，是在 *i*RPickTool 的树状图的设定中最初进行的设定。因此，如下所示，抓手显示在树状图的最上方。

在卡爪中输入适合拾取/放置动作的待机时间。卡爪的子对象的区域设定中，输入工具坐标系、抓住输出信号、抓住确认输入信号等。
如果正确输入这些数值，就不需要重新编辑 Plug & Play 的标准 TP 程序。

在机器人拾取/放置工件时，参照抓手的设定。
抓手表示拾取工件的机械手整体，作为其子对象设定的区域，表示实际抓住工件的位置。下图是示意图。

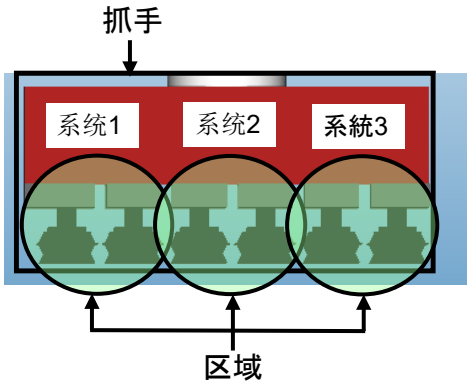


图 7.2(a) 抓手与区域示意图

1 个抓手中，可最多设定 20 个区域。
各区域中，包括工具坐标系、最多 4 个的区域工件有无 I/O、以及同样的最多 4 个的区域激活 I/O 的设定。以 1 个抓手拾取多个工件时，根据工件的数量设定区域的数量。
此外，抓手的数据可保存于方案中。由此，运行的各个方案中可使用各种区域工具坐标系、拾取/放置延迟时间等，因此无需对应各工件变更 TP 程序。
关于方案详情，请参阅“9.1 使用方案变更工作单元”。

7.2.1 抓手和区域的设定

如果添加抓手，就会自动添加区域。

区域的添加

按以下步骤在抓手中添加区域。

1 选择希望在树状图上添加的［抓手］。
2 按下树状图上的 新建按钮，或是选择［编辑树状图］中的［新建］。
则显示以下画面。

在［名称］中自动输入标准名称（例：ZONE2）。请根据需要进行更改。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。

3 按下 F4［确定］。
区域被追加至抓手中。

4 要在该抓手中再生成多个区域时，在各区域中反复上述步骤。

区域的删除

仅当 1 个抓手中有多个区域时，可以从抓手中删除区域。但是，可删除的区域，只能是该抓手中，显示在最下方的区域。抓手中必须有 1 个以上的区域，因此当只有一个区域时，不能删除。如果要删除所有区域，则删除抓手本身。

1 从树状图中选择删除［ZONE］（抓手中，显示于最下方的区域）。
2 按下树状图上的 按钮，或是选择［编辑树状图］中的［删除］。
3 于是会显示以下确认信息，按下 F4［确定］。

区域被从抓手中删除。

> **⚠ 注意**
> **如果要删除非显示于最下方的区域，则显示“在存在复数个领域的情况下，只能删除最后的领域”。**

抓手的删除

可进行抓手的删除。

1 在树状图中选择要删除的［抓手］。
2 按下树状图上的 按钮，或是选择［编辑树状图］中的［删除］。
3 于是会显示以下确认信息，按下 F4［确定］。

抓手及其下面的区域将被全部删除。

7.2.1.1 抓手的详细设定

进行抓手的详细设定。
从左侧的树状图中选择［抓手］。

［拾取延迟时间］

所谓拾取延迟时间，是指机器人到达拾取位置后的待机时间。例如，机械手是吸着型的情况，从衬垫接触工件到实际吸附需要花费时间时，在这里进行该时间的设定。以 ms 为单位输入。

［放置延迟时间］

所谓放置延迟时间，是指机器人到达放置位置后的待机时间。例如，机械手是抓住型的情况，从放置工件到实际卡盘放开工件需要花费时间时，在这里进行该时间的设定。以 ms 为单位输入。

［工件有无确认延迟时间］

所谓工件有无确认延迟时间，是指从发出区域激活 I/O，到执行工件有无检查的时间。调整工件有无检查传感器(机械手是吸附型时，是真空状态信号)的等待时间。在该延迟时间之前进行工件有无检查时，可能发生错误。

> **⚠ 注意**
> **如果设定延迟时间，请注意应用程序的周期时间会加长该延迟时间。此外，如果该延迟时间在追踪程序中被使用，也请注意在延迟时间结束之前，机器人在下游方向持续追踪工件。**

［区域数量］

抓手的区域数量为只读状态，在这里显示无法变更。但是，如果在抓手中追加或删除区域，则该数字自动更新。

7.2.1.2 区域的详细设定

进行区域的详细设定。
从左侧树状图中选择抓手下的［ZONE］。

［区域工具坐标系］

设定对应抓手的各区域的工具坐标系。
适当的工具坐标系值不清楚的情况下，在［MENU（菜单）］→［设置］→［坐标系］的工具坐标系画面上，使用机器人对工具坐标系实际进行示教。完成后，请返回此画面设定坐标系的值。

为了能在 *i*RPickTool 上活用“区域工具坐标系”，提供有如下 3 个程序。
PKGRGETZNCNT.PC: 向寄存器输出指定抓手的区域数。
PKGRGETZNUT.PC: 向位置寄存器输出指定抓手的指定区域的工具坐标系。
PK_GR_GET_UTOOLS1.TP: 使用上述 PKGRGETZNCNT.PC 和 PKGRGETZNUT.PC、以及 UTOOL［n］=PR［x］的指令，将指定抓手的各区域的工具坐标系设定在相应编号的工具坐标系中。(n 表示区域 ID。)

［区域激活 I/O］

行设定 1 个抓住时的输出信号。处理大工件等，在区域中使用多个输出信号时，可最多使用 4 行，输入区域的输出信号。输出值是［ON］、［OFF］或［脉冲］。指定［脉冲］为输出值时，以秒为单位指定脉冲输出时间。

在区域激活 I/O 的各行设定的条件，适用于“关闭抓手”或“拾取工件”动作。例如，设定为 RO［1］=［ON］时，RO［1］在拾取动作中为 ON，在放置动作中为 OFF。相反，设定为 RO［1］=［OFF］时，RO［1］在拾取动作中为 OFF，在放置动作中为 ON。

［区域工件有无 I/O］
行中设定 1 个确认是否抓住的输入信号。处理大工件等，在区域中使用多个输入时，可最多使用 4 行输入区域的输入信号。输入信号的值，是［ON］或［OFF］。

在区域工件有无 I/O 的各行指定的条件，适用于“关闭抓手”或“拾取工件”动作。例如，设定为 RI［1］=［ON］时，在拾取工件后，确认 RI［1］是否为 ON，在放置工件后，确认 RI［1］是否为 OFF。相反，设定为 RI［1］=［OFF］时，在拾取工件后，确认 RI［1］是否为 OFF，在放置工件后，确认 RI［1］是否为 ON。

7.3 操作的设定

操作是指机器人在各工作站执行的拾取/放置动作。
工作站包括［传送带工作站］和［固定工作站］，对于各工作站，在树状图上存在与机器人下面对应的操作。
操作包括在该工作站进行的作业类型（拾取/放置）、有多个工件时一次一次或是连续拾取工件、或是操作后是否执行抓住确认等设定项目。
正确设定这些项目，不用在 Plug & Play 的标准 TP 程序中编辑添加，就可以实施目标作业。

在传送带下生成传送带工作站时，或生成固定工作站集下的固定工作站后进行操作设定。
如果生成传送带工作站或固定工作站，则在以各个工作站设定的机器人下，自动生成操作。不能通过树状图上的 按钮，或 F2［编辑树状图］的［新建］生成操作。
同样，树状图上的 按钮，或 F2［编辑树状图］的［删除］，不能删除操作。
如果删除相应的传送带工作站或固定工作站，则也自动删除操作。

对应传送带工作站的操作中，以“OP_CS_xxxx”（OP_CS_是表示传送带工作站的操作的前缀，xxxx 是对应的传送带工作站的名称）的形式命名。
同样，对应固定工作站的操作中，以“OP_FS_xxxx”（OP_FS_是表示固定工作站的操作的前缀，xxxx 是对应的固定工作站的名称）形式命名。
无法更改操作名称，但是如果变更传送带工作站或固定工作站的名称，则操作的 xxxx 的部分也会被自动变更。

7.3.1 传送带工作站的操作

进行传送带工作站操作的设定。

传送带工作站的操作的生成

通过以下步骤，自动生成传送带工作站的操作。

以下画面为新建传送带工作站时的显示画面。
机器人下还不存在操作。

1 在设定传送带工作站的过程中选择［机器人］。
选择后，［OP_CS_CSTN1］立即被自动追加于选择的机器人下。

2 选择机器人下的操作［OP_CS_CSTN1］。
如下图所示，右侧显示所有的设定。

7.3.1.1 传送带工作站的操作的详细设定

进行传送带工作站操作的详细设定。
从树状图的左侧，选择机器人下的传送带工作站的操作。

［操作］

从下拉框选择机器人在传送带工作站执行的作业是［拾取］还是［放置］。

［多数拾取和放置］

只有［抓手］中的［区域］在 2 个以上时才能设定。
启用［多数拾取和放置］后，机器人只同时拾取/放置［区域］数量的工件。

> **⚠ 警告**
> **若在使用取多个的机械手一次性地将工件放置到托盘上，或者从托盘拾取时进行错误的设定，则有可能对系统造成重大影响，如机械手与托盘发生碰撞等。详情请参照“6.3.1 单元”的“取多个的机械手的托盘设定”。**

每次拾取 1 个工件后，移动至逃避位置（当待机位置有效，且不存在拾取工件时，则为待机位置），等待下一个工件。下一个工件到来后，则再次拾取工件，并返回逃避位置（当待机位置有效，且不存在拾取工件时，则为待机位置）。这一动作反复进行，直至拾取最后的工件。

如果不使其有效，则机器人个别拾取各个工件。

［接近时偏移量］

将工具坐标系上的值，输入以拾取/放置位置为基准的接近位置的偏移量。几乎所有的应用程序中，设定的偏移量只有 Z。

> **⚠ 注意**
> ***i*RPickTool 的标准 TP 程序中，接近位置偏移量不是位置补偿，而作为工具补偿使用。通常，工具坐标系的 Z 方向从标牌向着拾取/放置位置时，对接近位置偏移量的 Z 设定负的值。工具坐标的 Z 方向设定于与上述相反方向时，对接近位置偏移量的 Z 设定正的值。**

［待机位置］

启用待机位置时，如果机器人的拾取/放置的工件不存在，按之后说明的待机位置 X、Y 的设定，移动到计算出的待机位置，等待工件。标准的待机位置，是在采纳了拾取/放置位置和接近位置偏移量的 Z 值的高度，传送带工作站的追踪范围的上游侧边界。如果移动上游侧边界，待机位置也将随之自动移动。

如果使待机位置有效，则能改善应用程序的周期时间。如果不使用待机位置，当机器人的拾取/放置动作结束时，机器人在逃避位置待机。例如，进行［拾取］和［放置］的传送带工作站的情况，如果在拾取工件后，放置目标还没有来，则机器人在拾取工件的工作站上的逃避位置待机。在放置工件的工作站上，使待机位置有效时，机器人在放置工作站的上游侧边界待机。如果拾取工件的工作站和放置工作站之间距离较大时，通过使用该功能，可缩短周期时间，有望减少跳过的工件。

待机位置［X］

以毫米单位设定待机位置的偏移量。相应的传送带工作站的追踪坐标系 X 方向上的偏移量。如果输入正值，则沿着传送带的传送方向，待机位置向下游方向偏移。如果输入负值，待机位置向上游移动。

待机位置［Y］

以毫米单位输入待机位置的偏移量值。相应的传送带工作站的追踪坐标系 Y 方向上的偏移量。

［工件有无检查］

在相应的传送带工作站，设定机器人拾取/放置工件时的工件有无检查的有效/无效。

> ⚠ **注意**
> **即使将［工件有无检查］设定为有效，如果在相应区域的设定中，没有设定区域工件有无 I/O，则不进行工件有无检查。**

［基准位置 PR］

在该编号的位置寄存器中，记录基准位置。无法更改该值。Plug & Play 的基准位置设定的详细信息请参照“4.2.12 基准位置的设定、机器人的位置示教”。

［接近偏移量 PR］

*i*RPickTool 中运行标准 TP 程序时，这里显示的编号的位置寄存器中，存储接近位置偏移量。无法更改该值。接近位置偏移量被作为工具补偿来使用。

［工件数据 VR］

*i*RPickTool 中运行标准 TP 程序时，从队列取得工件信息，这里显示的编号的视觉寄存器中存储工件信息。无法更改该值。

7.3.2 固定工作站的操作

进行固定工作站操作的设定。

固定工作站的操作的新建

通过以下步骤，自动生成固定工作站的操作。

下图是新建固定工作站时的显示画面。
机器人下还不存在操作。

1 在设定固定工作站的过程中选择机器人。
选择后，［OP_FS_FSTN1］立即被自动追加于选择的机器人下。

2 选择机器人下的操作［OP_FS_FSTN1］。
如下图所示，右侧显示所有的设定。
此外，安装有 R857: *i*RPickTool /扩充功能包加载项时（北美专用 R827:*i*RPickTool / Production Control Add-on），显示缓冲区的设定项目。详情请参阅“7.12 Plug & Play 的缓冲区功能”。

7.3.2.1 固定工作站的操作的详细设定

进行固定工作站操作的详细设定。
从树状图的左侧，选择机器人下的固定工作站的操作。

[操作]

从下拉框选择机器人在固定工作站执行的作业是［拾取］还是［放置］。

[多数拾取和放置]

只有［抓手］中的［区域］在 2 个以上时才能设定。
启用［多数拾取和放置］后，机器人只同时拾取/放置［区域］数量的工件。
每次拾取 1 个工件后，移动至逃避位置，等待下一个工件。下一个工件到来后，再次拾取工件，并至逃避位置等待。
这一动作反复进行，直至拾取最后的工件。
如果不使其有效，则机器人个别拾取各个工件。

[接近时偏移量]

将工具坐标系上的值，输入以拾取/放置位置为基准的接近位置的偏移量。几乎所有的应用程序中，设定的偏移量只有 Z。

> **注意**
> ***i*RPickTool 的标准 TP 程序中，接近位置偏移量不是位置补偿，而作为工具补偿使用。通常，工具坐标系的 Z 方向从标牌向着拾取/放置位置时，对接近位置偏移量的 Z 设定负的值。工具坐标的 Z 方向设定于与上述相反方向时，对接近位置偏移量的 Z 设定正的值。**

[动作开始信号]

固定工作站中，可设定给予拾取/放置动作的开始许可的 DI。例如，在机器人动作前决定的位置配给托盘的系统中，托盘的配给结束，机器人可以动作的情况下，使这里设定的 DI 为 ON。使用该功能时，请设定合适的 DI 编号。

[动作完成信号]

固定工作站中，可设定通知机器人的拾取/放置动作完成的 DO。例如，如果机器人的动作结束，则机器人作业的托盘被排出，或在相同位置配给托盘的系统中，机器人的动作完成，需要排出托盘时，使该 DO 为 ON。

如果该 DO 为 ON，则成为传送机器人动作的工作站（此期间，前述的［动作开始信号］为 OFF）。重新准备机器人应作业的空的工作站的系统。

[动作开始信号模拟]
如果该模拟有效，且［动作开始信号］以及［动作完成信号］的设定中设定有有效的 DI/DO 编号，则 *i*RPickTool 依照后述的［待机时间］中设定的待机时间，自动进行传送工作站的模拟。

模拟中，*i*RPickTool 使［动作完成信号］中设定的 DO 为 ON，［动作开始信号］中设定的 DI 疑似性 OFF 后，经过［待机时间］中设定的时间后，再次 ON。即使实际没有输入［动作开始信号］，机器人也能在该固定工作站进行作业。

在实际生产前的试运行中，该功能在连续运行时能起作用。

[待机时间]
以 ms 单位指定［动作开始信号模拟］中的待机时间。
仅在使动作开始信号模拟有效时使用该值。

[工件有无检查]
在相应的固定工作站，设定机器人拾取/放置工件时的工件有无检查的有效/无效。

> **注意**
> **即使将［工件有无检查］设定为有效，如果在相应区域的设定中，没有设定区域工件有无 I/O，则不进行工件有无检查。**

[基准位置 PR]
在该编号的位置寄存器中，直接示教基准位置。无法更改该值。

[接近偏移量 PR]
*i*RPickTool 中运行标准 TP 程序时，这里显示的编号的位置寄存器中，存储接近位置偏移量。无法更改该值。接近位置偏移量被作为工具补偿来使用。

[工件数据 VR]
*i*RPickTool 中运行标准 TP 程序时，从队列取得工件信息，这里显示的编号的视觉寄存器中存储工件信息。无法更改该值。

7.4 固定工作站设定

进行固定工作站的设定。
本节针对只可在 Plug & Play 设定的［跳过］设定进行说明。关于其他设定，请参阅“6 基本功能参考”。

跳过

要在机器人上用该固定工作站操作时，请选中［有效］。机器人在多个固定工作站作业时，如发生因外部设备的故障等而无法在特定的固定工作站进行作业的情况，可通过使用该功能，跳过在该固定工作站的作业。

7.5 机器人的程序

Plug & Play 中备有标准 TP 程序，如果运行主程序 PK_MAIN1.TP，则根据“7.1 工作单元设定”～“7.4 固定工作站设定”的内容，进行需要的机器人的作业。
另外，对于实际示教机器人位置的作业，传送带工作站参照“4.2.12 基准位置的设定、机器人的位置示教”进行，固定工作站按“6.10 基准位置设定”与“6.11 机器人的位置示教”进行。

> ⚠ **注意**
> ***i*RPickTool 不支持停电处理。利用初始值禁用停止处理。接通电源后，务必从主程序 PK_MAIN1.TP 起始开始执行。**

有关标准 TP 程序下的机器人的动作，在“7.5.1 标准 TP 程序下的机器人的动作”进行说明。
此外，Plug & Play 的标准 TP 程序中，在“7.5.2 数值寄存器”以下所说明的规定的数值寄存器、位置寄存器、字符串寄存器、视觉寄存器、负载设定、用户报警、用户坐标系、工具坐标系为专用。请不要使用这些寄存器等。

> ⚠ **警告**
> **如果在 Plug & Play 中使用的数值寄存器、位置寄存器、字符串寄存器、视觉寄存器、负载设定、用户报警、用户坐标系、工具坐标系被用于其他用途，则机器人可能进行意想不到的动作，严重影响系统。**

7.5.1 标准 TP 程序下的机器人的动作

标准 TP 程序的构成前提，是机器人以一定的规则进行动作。
以下显示该前提。

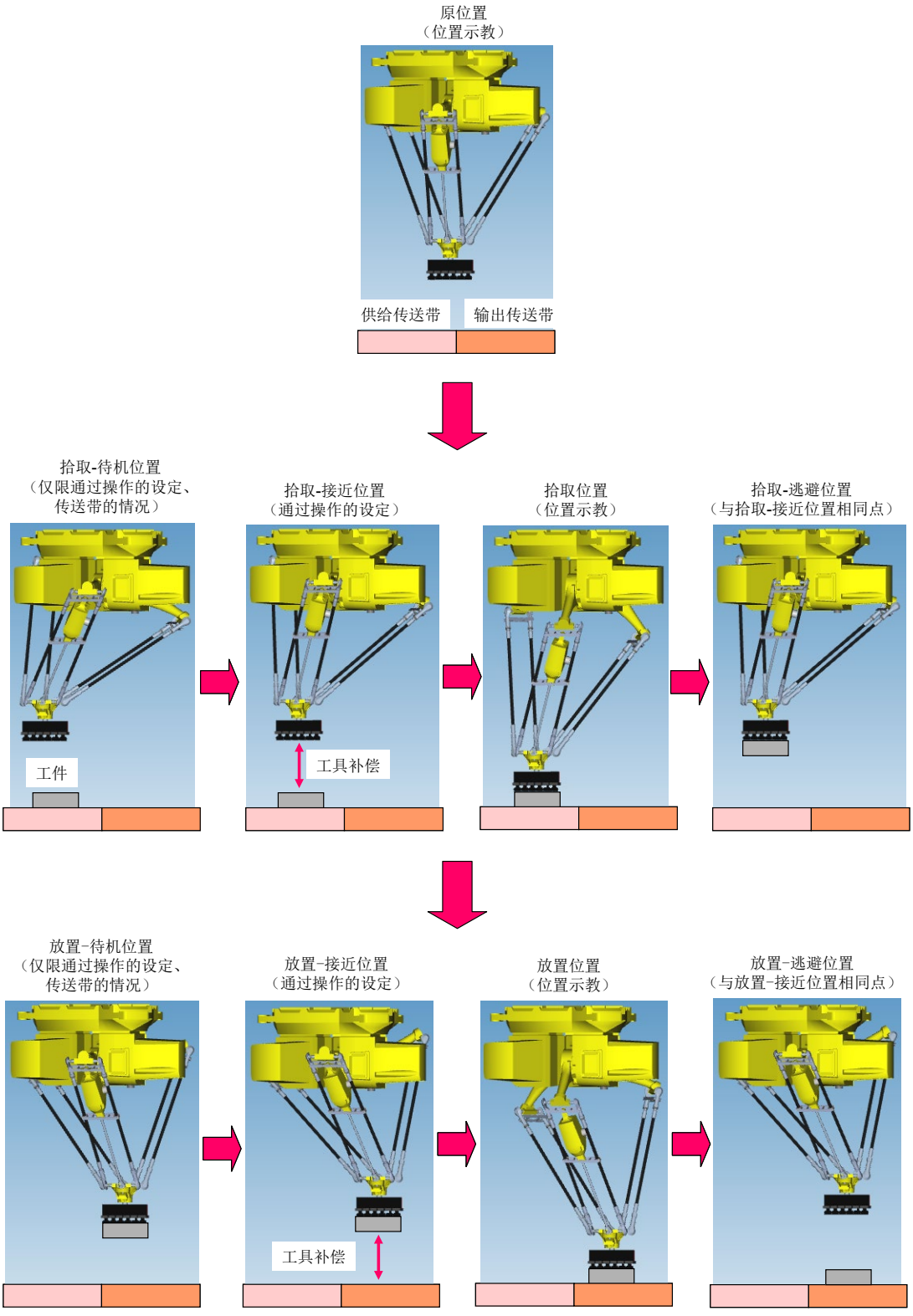


图 7.5.1(a) 机器人按一定规则动作

放置动作结束后，又返回拾取动作，反复进行拾取－放置的动作。仅在进行从传送带的拾取动作、放置动作时，才发生至各个拾取动作、放置动作的最左侧的待机位置的动作。固定工作站的情况下，不发生。

有关机器人的动作，有以下 3 点特别重要。
1　接近位置和逃避位置为相同位置。
2　用于示教接近位置而设定的值，是相对拾取位置、放置位置的工具补偿中的值。
3　一系列流程中应示教的点，仅限原位置、拾取位置、放置位置。

要使机器人进行与上述 3 点前提不同的动作时,请充分理解“6.8 KAREL 程序”、“7.9 KAREL 程序”、“7.10 标准 TP 程序”的内容后进行变更。

7.5.2 数值寄存器

在 Plug & Play 的标准 TP 程序中，使用特定的数值寄存器。现在，使用数值寄存器 95 号～230 号。
请不要将这些数值寄存器用于其他用途。由于不能使用这些数值寄存器，而需要更多数值寄存器时，请通过［控制启动］增加数值寄存器数量。

表 7.5.2(a) 数值寄存器

R［95］～［98］,［100］	**暂时使用的数值寄存器。这些在各种 TP 程序中被再次使用。**
R［99］	**使用中的方案的保存中使用的数值寄存器。**
R［97］～［100］	**所有组中共通使用的数值寄存器。 这些在组 1 以及组 2 程序中使用。**
R［101］～［108］	**组 1 的程序中共通使用的寄存器。**
R［109］～［125］	**组 1 的最初的拾取动作中使用的数值寄存器 （传送带工作站中和固定工作站中都使用）**
R［126］～［127］	**组 1 的缓冲区操作中使用的数值寄存器**
R［129］～［145］	**组 1 的最初的放置动作中使用的数值寄存器 （传送带工作站中和固定工作站中都使用）**
R［147］	**组 1 的缓冲区操作中使用的数值寄存器**
R［148］～［150］	**暂时使用的数值寄存器。这些在各种 TP 程序中被再次使用。**
R［151］～［158］	**组 2 的程序中共通使用的寄存器。**
R［159］～［175］	**组 2 的最初的拾取动作中使用的数值寄存器 （传送带工作站中和固定工作站中都使用）**
R［176］～［177］	**组 2 的缓冲区操作中使用的数值寄存器**
R［179］～［195］	**组 2 的最初的放置动作中使用的数值寄存器 （传送带工作站中和固定工作站中都使用）**
R［197］	**组 2 的缓冲区操作中使用的数值寄存器**
R［199］～［205］	**组 1 的缓冲区操作中使用的数值寄存器**
R［209］～［215］	**组 2 的缓冲区操作中使用的数值寄存器**
R［219］～［229］	**现在不使用**
R［230］	**组 1 的预整列功能中所使用的寄存器 （即使在不使用预整列功能的情况下，也用于判断是否使用预整列功能）**

7.5.3 位置寄存器

在 Plug & Play 的标准 TP 程序中，使用特定的位置寄存器。现在，使用位置寄存器 51 号～90 号。
请不要将这些位置寄存器用于其他用途。由于不能使用这些位置寄存器，而需要更多位置寄存器时，请通过［控制启动］增加位置寄存器数量。

表 7.5.3(a) 位置寄存器

PR［51］	**暂时使用的位置寄存器。**
PR［52］	**在全组的缓冲区中的动作的接近位置偏移量**
PR［53］	**在全组的缓冲区中的动作的基准位置**
PR［54］	**原位置**
PR［55］	**组 1 的机器人的待机位置**
PR［56］	**在传送带工作站的最初的拾取动作的接近位置偏移量**
PR［57］	**在传送带工作站的最初的拾取动作的基准位置**
PR［58］	**在传送带工作站的最初的放置动作的接近位置偏移量**
PR［59］	**在传送带工作站的最初的放置动作的基准位置**
PR［60］	**在固定工作站的最初的拾取动作的接近位置偏移量**
PR［61］	**在固定工作站的最初的拾取动作的基准位置**
PR［62］	**在固定工作站的最初的放置动作的接近位置偏移量**
PR［63］	**在固定工作站的最初的放置动作的基准位置**
PR［64］	**现在不使用**
PR［65］	**在传送带工作站的第 2 次拾取动作的接近位置偏移量**
PR［66］	**在传送带工作站的第 2 次拾取动作的基准位置的机器人待机位置**
PR［67］	**在传送带工作站的第 2 次放置动作的接近位置偏移量**
PR［68］	**在传送带工作站的第 2 次放置动作的基准位置**
PR［69］	**在固定工作站的，对于第 2 次拾取动作的接近位置偏移量**
PR［70］	**在固定工作站的对第 2 次拾取动作的基准位置**
PR［71］	**在固定工作站的，对于第 2 次放置动作的接近位置偏移量**
PR［72］	**在固定工作站的对第 2 次放置动作的基准位置**
PR［73］	**现在不使用**
PR［74］	**在传送带工作站的第 3 次拾取动作的接近位置偏移量**
PR［75］	**在传送带工作站的第 3 次拾取动作的基准位置的机器人待机位置**
PR［76］	**在传送带工作站的第 3 次放置动作的接近位置偏移量**
PR［77］	**在传送带工作站的第 3 次放置动作的基准位置**
PR［78］	**在固定工作站的，对于第 3 次拾取动作的接近位置偏移量**
PR［79］	**在固定工作站的对第 3 次拾取动作的基准位置**
PR［80］	**在固定工作站的，对于第 3 次放置动作的接近位置偏移量**
PR［81］	**在固定工作站的对第 3 次放置动作的基准位置**
PR［82］	**现在不使用**
PR［83］	**在传送带工作站的第 4 次拾取动作的接近位置偏移量**
PR［84］	**在传送带工作站的第 4 次拾取动作的基准位置的机器人待机位置**
PR［85］	**在传送带工作站的第 4 次放置动作的接近位置偏移量**
PR［86］	**在传送带工作站的第 4 次放置动作的基准位置**
PR［87］	**在固定工作站的，对于第 4 次拾取动作的接近位置偏移量**
PR［88］	**在固定工作站的对第 4 次拾取动作的基准位置**
PR［89］	**在固定工作站的，对于第 4 次放置动作的接近位置偏移量**
PR［90］	**在固定工作站的对第 4 次放置动作的基准位置**

7.5.4 字符串寄存器

在 Plug & Play 的标准 TP 程序中，使用以下的字符串寄存器。请不要将这些字符寄存器用于其他用途。

表 7.5.4(a) 字符串寄存器

SR［1］	**进行最初的拾取动作的工作站名（传送带工作站、固定工作站都使用）**
SR［2］	**进行最初的放置动作的工作站名（传送带工作站、固定工作站都使用）**
SR［3］	**暂时使用的字符串寄存器**
SR［4］	**SR［1］的工作站是传送带工作站时，该工作站所属的传送带名**
SR［5］	**SR［2］的工作站是传送带工作站时，该工作站所属的传送带名**

7.5.5 工件数据 VR

在 Plug & Play 的标准 TP 程序中，使用以下的视觉寄存器。请不要将这些视觉寄存器用于其他用途。

表 7.5.5(a) 工件数据 VR

VR［1］	**在组 1 的拾取动作中使用的工件数据**
VR［2］	**在组 1 的放置动作中使用的工件数据**
VR［3］	**在组 2 的拾取动作中使用的工件数据**
VR［4］	**在组 2 的放置动作中使用的工件数据**

无论是传送带工作站，还是固定工作站，都使用上述视觉寄存器。

7.5.6 负载设定

在 Plug & Play 的标准 TP 程序中，使用以下的负载设定。请分别设定不持有工件，和持有工件时的负载。此外，请不要将这些负载设定用于其他用途。

表 7.5.6(a) 负载设定

负载设定［1］	**不持有工件时的负载**
负载设定［2］	**持有工件时的负载**

7.5.7 用户报警

在 Plug & Play 的标准 TP 程序中，使用如下的用户报警［1］～［6］。请不要将这些用户报警用于其他用途。

报警 No.	报警信息
［1］:	［Invalid Robot Id ］
［2］:	［Invalid Gripper Id ］
［3］:	［No pick station ］
［4］:	［No drop station ］
［5］:	［Pick or Drop Qty > Zones ］
［6］:	［Cannot use PRGR and Buffers ］

7.5.8 用户坐标系

使用在树状图的［固定工作站］的设置画面中，设定的编号的用户坐标系。不会使用其他的用户坐标系。

7.5.9 工具坐标系

*i*RPickTool 所使用的工具坐标系的数量，根据机器人（属于动作组）使用的抓手的区域数不同而不同。
例如，如果抓手中只有 1 个区域，则只使用工具坐标系 1。
如果抓手中有 10 个区域，则 *i*RPickTool 的 TP 程序使用工具坐标系 1～10。

如果抓手中有 20 个区域（区域数量的最大值），则 *i*RPickTool 的 TP 程序使用工具坐标系 1～20。但是，在区域数超过 10 的情况下，需要进行追加设定，通过［控制启动］增加工具坐标系的数量。出厂时的设定为，工具坐标系的数量为 10。

7.6 基准位置设定

基准位置，是在机器人的程序示教拾取位置或放置位置时的工件的检出位置。
对于设定了基准位置的工件，对在相同传送带作业的全部机器人示教拾取位置或放置位置。

关于 Plug & Play 基准位置的设定，传送带工作站请参阅“4.2.12 基准位置的设定、机器人的位置示教”，固定工作站请参阅“6.10 基准位置设定”。

7.7 机器人的位置示教

在基准位置的设定后，进行机器人的位置示教。
关于机器人的位置示教，传送带工作站请参阅“4.2.12 基准位置的设定、机器人的位置示教”，固定工作站请参阅“6.11 机器人的位置示教”。

7.8 追踪动作的微调整

Plug & Play 中的机器人追踪动作微调方法与基本功能相同。
请参照“6.12 追踪动作的微调整”。

本节特别针对误差在 Plug & Play 标准 TP 程序中的调整方法进行说明。

7.8.1 追踪坐标系的设定误差的调整

使用相机检出工件时，如果相机认识的追踪坐标系和各机器人认识的追踪坐标系有偏移，则追踪动作会发生误差。该误差会以“如果工件不旋转，则正确补偿，如果工件旋转，则发生偏移”的形式发生。

该误差原因是，未正确设定追踪坐标系。本来，需从 TCP 的设定、追踪坐标系的设定开始重新进行，但也可以通过以下步骤，对 *i*RVision 计算的补偿数据添加调整量，实现改善。该调整对各个机器人分别进行。

而这里介绍的程序例，将位置寄存器的 20 号作为存储调整量的地方使用。请根据需要适当变更。

1　在即将移动至接近位置之前，追加以下的 1 行。

例：标准 TP 程序“7.10.2.12 PK_CV_PICK11.TP”的情况

```
56:   CALL ADJ_OFS(1,1,20,1)
57:   --
58:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
```

2　为了使机器人在追踪程序内的拾取位置的正上方停止，请将动作指令和待机指令插入接近点下面一行。

```
69:   LBL[88]
70:   --
71:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] Tool_Offset,PR[15]
72:   WAIT 10.00(sec)
73:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,
      CALL PKGRCLOSE(抓手 ID =R[104:Gripper Id G1],开始区域=R[107:Counter G1],
      结束区域=R[106:Gr EndZone G1])
74:   --
```

加入待机命令，是用于通过目视确认偏移。待机时间以 10 秒左右为宜。此外，对 PR［15］（用于生成拾取动作的接近点的工具偏移量，Z 以外是 0）的 Z 输入较小值。这是为了使机器人在拾取位置正上方停止。

3 使 PR［20］为(X, Y, Z, W, P, R) =(0, 0, 0, 0, 0, 0)。
4 以镓动时的速度使传送带动作。
5 与平常一样启动主程序（PK_MAIN1），传送 1 个工件。此时，工件的放置方向与基准位置设定时相同。

> ⚠ **注意**
> **运行追踪程序时，请以 AUTO 模式或 T2 模式运行。**
> **传送带动作时，即使以 T1 模式运行追踪程序，机器人也无法动作。这种情况，显示“LNTK-041 编码器在 T1 模式移动”的报警信息。**

6 机器人到达接近点后，停止传送带。此时，请使程序保持运行状态。
7 传送带完全停止后，暂时停止程序。
8 确认工件位置和机器人位置的偏移。
9 如果有偏移，以点动移动至正确位置，进行示教修正。此时，会询问是否减去补偿量，请选择［是］。
10 再次启动主程序，传送 1 个工件。
这次，将工件设定于基准位置时，以 180 度相反方向放置。
11 机器人到达接近点后，停止传送带。此时，请使程序保持运行状态。
12 传送带完全停止后，暂时停止程序。
13 确认机器人的位置和工件位置的偏移量。
14 如有偏移，分别测定 X 方向（传送带的传送方向）和 Y 方向(与传送带传送方向正交的方向)的偏移，将 X 方向的调整量输入 PR［20］的 X，Y 方向的调整量输入 PR［20］的 Y。调整量设定为各偏移量的一半的值。从 ADJ_OFS 的行运行程序，确认是否移动至正确位置。如有偏移，再次返回步骤 4，进行作业。反复执行，直至机器人移动至正确位置。
15 确认移动至正确位置后，从追踪程序中删除插入的动作指令和待机指令。

有关 ADJ_OFS

ADJ_OFS 是在存储于视觉寄存器的补偿数据上，加上指定的调整量的 KAREL 程序。以下记述参数的意思。详细信息，请参照“*i*RVision 操作说明书（参考篇）”。

参数 1：存储补偿数据的寄存器种类。补偿数据存储于视觉寄存器中时，指定为 1
参数 2：补偿数据存储于视觉寄存器中时的视觉寄存器的序号
参数 3：存储调整量的位置寄存器的序号
参数 4：存储调整的补偿数据的视觉寄存器的序号

> **注释**
> ***i*RPickTool 中，即使没有安装视觉支持工具选项，也可以标准使用 ADJ_OFS。**

7.8.2 机器人的动态误差的调整

即使在使传送带停止的状态下，机器人正确来到工件的拾取位置上，如果在传送带动作的状态下，仍有发生偏移的可能。该误差是引起机器人动作的追踪延迟的误差。

> **注释**
> **1 动态误差的调整，是进行 7.8.1 中介绍的误差调整，如果在停止传送带的状态下，则请在没有误差的状态下进行。**
> **2 机器人在多个传送带上进行作业时，请在其中速度最大的传送带上进行调整。**

该偏移通过以下步骤调整。

1 为了使机器人在追踪程序内的拾取位置的正上方停止，请将动作指令和待机指令插入接近点下面一行。

例：标准 TP 程序“7.10.2.12 PK_CV_PICK11.TP”的情况

```
70:  LBL[88]
```

```
71:   -- Move to pick pos
72:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] Tool_Offset,PR[15]
73:   WAIT 10.00(sec)
74:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,
      CALL PKGRCLOSE(抓手 ID =R[104:Gripper Id G1],开始区域=R[107:Counter G1],
      结束区域=R[106:Gr EndZone G1])
75:   -- Chk Skip Pick
```

加入待机命令，是用于通过目视确认偏移。待机时间以 10 秒左右为宜。此外，在 PR［15］(用于生成拾取动作的接近点的工具偏移量。Z 以外为 0。)的 Z 中输入较小的值。这是为了使机器人在拾取位置正上方停止。

2 以镓动时的速度使传送带动作。
3 与平常一样启动主程序（PK_MAIN1），传送 1 个工件。此时，工件的放置方向与基准位置设定时相同。

> ⚠ **注意**
> **运行追踪程序时，请以 AUTO 模式或 T2 模式运行。**
> **传送带动作时，即使以 T1 模式运行追踪程序，机器人也无法动作。这种情况，显示“LNTK-041 编码器在 T1 模式移动”的报警信息。**

4 在传送带动作中，且程序运行过程中，请确认机器人的位置正确追踪工件的拾取位置。
5 如果有偏移，在“6.12.3 机器人的动态误差的调整”中介绍的［动态追踪误差调整］中，输入以（偏移量（mm）÷传送带速度（mm/s））× 1000 计算的值（单位 ms）。传送带的速度，是机器人追踪工件时的速度。机器人偏移至工件在传送带上的上游侧，偏移量的符号为正，偏移至下游侧则为负。已经在［动态追踪误差调整］中设定了值的情况，则将上述计算所得的值与原来的值相加并输入。
6 从追踪程序中删除插入的动作指令和待机指令。

计算例

［动态追踪误差调整］中设定为 30 的状态下，传送带以速度 500 mm/s 动作时，工件的拾取位置在传送带上游侧偏移 3 mm 的情况，

［动态追踪误差调整］ =+30(ms) + (3(mm) ÷ 500(mm/s)) × 1000 = +36 (ms)

7.9 KAREL 程序

Plug & Play 中都使用标准程序，因此在通常情况下，无需在意在标准程序中使用的 KAREL 程序的存在，但是根据状况，发生需要变更标准程序的情况时，请充分理解以下的 KAREL 程序的规格，进行变更。

*i*RPickTool 中使用的 KAREL 程序，原则上有如下规定。

- 开头的 2 个字符以［PK］打头。
- 之后的 2 个字符表示相关对象。
- 第 5 个字符起，表示该程序的功能。

7.9.1 有关工作单元的程序

工作单元相关程序打头的 4 个字符是［PKWC］。打头的字符［PK］后面的 2 个字符是［WC］。

另外，工作单元相关程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

7.9.2 有关抓手的程序

抓手相关程序打头的 4 个字符是［PKGR］。［PK］后面的 2 个字符是［GR］。

7.9.2.1 PKGRCHKPP.PC

确认在关闭抓手后或拾取动作后，工件是否存在于抓手，或在打开抓手后或放置动作后，工件是否不存在于抓手。使用树状图的［抓手］以及［区域］的设定数据。

- 对应抓手的指定区域执行工件有无确认。
- 在树状图的［抓手］的设定画面中，“工件有无确认延迟时间”是 0 时，立即执行工件有无检查。工件有无确认延迟时间的值大于 0 时，在经过指定的延迟时间后，开始检查。从信号开启至实际进入真空状态，需要延迟时间的吸着机械手等部分抓手需要该时间。
- 由于在工件有无确认延迟时间中，机器人在下游方向追踪工件，因此在机器人的追踪动作的确认中，也可使用工件有无确认延迟时间。

参数 1：
指定抓手的 ID。

参数 2：
指定是“拾取”动作或“放置”动作。

注释
若已经显示参数的说明，则作为示教时的选项，显示
1 拾取
2 放置
内部设定为 0: 拾取/1: 放置。
有显示参数的说明时，“拾取”时指定 0，“放置”时指定 1。

参数 3：
指定返回状态的数值寄存器。态不是 0 时，表示不是工件有无检查中期待的结果，也就是说，在“拾取”动作中，抓手上没有工件。或者，在“放置”动作中，抓手上有工件。使用本 KAREL 程序时，请务必确认该状态。

参数 4：
指定进行工件有无检查的抓手的最初区域。

参数 5：
指定进行工件有无检查的抓手的最后区域。

使用例：
在 ID 为 1 的抓手的最初的 2 个区域的拾取动作结束后，确认是否存在工件。状态返回 R［7］。状态不是 0 时，表示有某个区域中没有工件。

```
CALL PKGRCHKPP(抓手 ID=1,拾取,状态 Reg=7,初始区域=1,结束区域=2)
```

7.9.2.2 PKGRCLOSE.PC

对于抓手指定的区域，执行关闭抓手或拾取动作。该程序使用树状图的［抓手］以及［区域］的设定数据，使区域激活 I/O 为 ON。

参数 1：
指定抓手的 ID。

参数 2：
指定抓手的最初区域。

参数 3：
指定抓手的最后区域。

使用例：
关闭 ID 为 1 的抓手的区域 1～2。

```
CALL PKGRCLOSE(抓手 ID=1,开始区域=1,结束区域=2)
```

7.9.2.3 PKGRGETDPDLY.PC

取得在树状图的［抓手］的设定画面中作为“放置延迟时间”设定的抓手的放置延迟时间。

参数 1：
指定抓手的 ID。

参数 2：
指定返回延迟时间的数值寄存器(单位 s)。

使用例：
R［7］中取得 ID 为 1 的抓手的放置延迟时间。

```
CALL PKGRGETDPDLY(抓手 ID=1,待机时间 Reg=7)
```


7.9.2.4 PKGRGETID.PC

取得在树状图的［机器人］的设定画面中设定的抓手的 ID。

参数 1：
指定机器人的 ID。

参数 2：
指定存储以参数 1 的机器人设定的抓手的 ID 的数值寄存器。

使用例：
R［7］中取得 ID 为 1 的机器人所使用的抓手的 ID。

```
CALL PKGRGETID(机器人 ID=1,抓手 ID Reg=7)
```

7.9.2.5 PKGRGETPKDLY.PC

取得在树状图的［抓手］的设定画面中作为“拾取延迟时间”设定的抓手的拾取延迟时间。

参数 1：
指定抓手的 ID。

参数 2：
指定返回延迟时间的数值寄存器(单位 s)。

使用例：
R［7］中取得 ID 为 1 的抓手的拾取延迟时间。

```
CALL PKGRGETPKDLY(抓手 ID=1,待机时间 Reg=7)
```

7.9.2.6 PKGRGETZNCNT.PC

取得抓手的区域数。

参数 1：
指定抓手的 ID。

参数 2：
指定返回区域数的数值寄存器。

使用例：
R［7］中取得 ID 为 1 的抓手的合计区域数。

```
CALL PKGRGETZNCNT(抓手 ID=1,区域数量 Reg=7)
```

7.9.2.7 PKGRGETZNUT.PC

在位置寄存器中取得抓手的区域的工具坐标系。

参数 1：
指定需要取得工具坐标系的机器人的 ID。

参数 2：
指定具有需要取得工具坐标系的区域的抓手的 ID。
参数 3：
指定需要取得工具坐标系的区域的 ID。

参数 4：
指定存储工具坐标系的位置寄存器序号。

使用例：
PR［51］中取得 ID 为 1 的机器人所具有的 ID 为 2 的抓手的 ID 为 3 的区域的工具坐标系。

```
CALL PKGRGETZNUT(机器人 ID=1, 抓手 ID=2,区域 ID=3,工具坐标系 PR=51)
```

7.9.2.8 PKGRINIT.PC

将抓手和所有的抓手的区域数据恢复出厂时的设定。仅在删除所有的抓手数据，从头开始重新设定时使用该程序。

无参数。

使用例：

```
CALL PKGRINIT
```

7.9.2.9 PKGROPEN.PC

对于抓手指定的区域，执行打开抓手或放置动作。该程序使用树状图的［抓手］以及［区域］的设定数据，使区域激活 I/O 为 ON。

参数 1：
指定抓手的 ID。

参数 2：
指定抓手的最初区域。

参数 3：
指定抓手的最后区域。

使用例：
打开 ID 为 1 的抓手的区域 1～2。

```
CALL PKGROPEN(抓手 ID=1,开始区域=1,结束区域=2)
```

7.9.3 有关机器人的程序

机器人相关程序打头的 4 个字符是［PKRB］。［PK］后面的 2 个字符是［RB］。

另外，以下程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

- PKRBGETID.PC
- PKRBGETSFLG.PC

7.9.3.1 PKRBGETCVCNT.PC

在树状图的设定中，指定的机器人取得进行指定动作（“拾取”或“放置”）的传送带的数量。

参数 1：
指定机器人的 ID。

参数 2：
指定是“拾取”动作或“放置”动作。

注释
若已经显示参数的说明，则作为示教时的选项，显示
1 拾取
2 放置
内部设定为 0: 拾取/1: 放置。
有显示参数的说明时，“拾取”时指定 0，“放置”时指定 1。

参数 3：
参数 1 的 ID 的机器人，指定存储进行参数 2 的动作的传送带数量的数值寄存器序号。

使用例：
R［7］中取得 ID 为 1 的机器人进行拾取动作的传送带的数量。

```
CALL PKRBGETCVCNT(机器人 ID=1,拾取,传送带计数 Reg=7)
```

7.9.3.2 PKRBGETFSCNT.PC

在树状图的设定中，指定的机器人取得进行指定动作（“拾取”或“放置”）的固定工作站的数量。

参数 1：
指定机器人的 ID。

参数 2：
指定是“拾取”动作或“放置”动作。

注释
若已经显示参数的说明，则作为示教时的选项，显示
1 拾取
2 放置
内部设定为 0: 拾取/1: 放置。
有显示参数的说明时，“拾取”时指定 0，“放置”时指定 1。

参数 3：
参数 1 的 ID 的机器人，指定存储进行参数 2 的动作的固定工作站数量的数值寄存器序号。

使用例：
R［7］中取得 ID 为 1 的机器人进行放置动作的固定工作站的数量。

CALL PKRBGETFSCNT(机器人 ID=1,放置,固定工作站计数 R=7)

7.9.3.3 PKRBGETPRGRT.PC

取得机器人按以下所示方式定义的预整列的类型。有关预整列功能的详情，请参照“7.11 预整列设定”。

0: 不使用预整列。
1: 在同一传送带上进行整列的最上游的机器人。生成虚拟托盘。
2: 在同一传送带上进行整列的非最上游的机器人。
3: 系对在同一传送带上已被整列的工件进行汇总搬运的机器人。

参数 1：
指定供给传送带的 ID。

参数 2：
指定供给传送带的传送带工作站的 ID。

参数 3：
指定输出传送带的 ID。

参数 4：
指定输出传送带的传送带工作站的 ID。

参数 5：
指定存储预整列类型值的寄存器序号。

使用例：
有关从 ID 为 1 的供给传送带上的 ID 为 2 的传送带工作站取出工件，向 ID 为 2 的输出传送带上的 ID 为 4 的传送带工作站放置工件的机器人，在 R［7］取得预整列的类型。

CALL PKRBGETPRGRT(拾取传送带 ID=1,拾取工作站 ID=2,放置传送带 ID=2,放置工作站 ID=4,
预整列类型 Reg=7)

7.9.4 有关机器人的程序

传送带相关程序打头的 4 个字符是［PKCV］。打头的字符［PK］后面的 2 个字符是［CV］。

另外，传送带相关程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

7.9.5 有关传感器的程序

传感器相关程序打头的 4 个字符是［PKSN］。［PK］后面的 2 个字符是［SN］。

另外，传感器相关程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

7.9.6 有关传送带工作站的程序

传送带工作站相关程序打头的 4 个字符是［PKCS］。［PK］后面的 2 个字符是［CS］。

另外，以下程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

- PKCSGETID.PC
- PKCSCLRQUE.PC
- PKCSGETQUE.PC
- PKCSGETQCELL.PC

- PKCSACKQUE.PC
- PKCSPUTQUE.PC
- PKCSGETTIME.PC
- PKCSGETDIST.PC
- PKCSGETPFRT.PC
- PKCSGETNPRT.PC
- PKCSSETDSLIN.PC
- PKCSGETENC.PC
- PKCSGETTRID.PC
- PKCSCALWPOS.PC
- PKCSCLRACK.PC

7.9.6.1 PKCSCALWPOS.PC

使用通过 PKCSGETQUE 取得的工件信息，在用于持有此工件的传送带工作站中，计算追踪范围上游侧边界的机器人的待机位置。在此程序中，除待机位置以外，也会输出是否移动到待机位置的判断结果。例如，工件已经在追踪范围内时，不需要移动到待机位置。此外，机器人已经在待机位置时，不需要移动到待机位置。

参数 1：
指定要计算待机位置的传送带工作站的 ID。ID 可使用 PKCSGETID 来取得。详细信息，请参照“6.8.5.1 PKCSGETID.PC”。

参数 2：
指定对传送带工作站的基准位置进行示教的位置寄存器编号。

参数 3：
使用 PKCSGETQUE，指定取得此传送带工作站的工件信息的视觉寄存器编号。

参数 4：
指定 PKCSGETQUE 的状态。

参数 5：
指定保存是否移动到待机位置的判断结果的数值寄存器编号。

参数 6：
指定存储计算的“待机位置”的位置寄存器编号。

参数 7：
指定距追踪范围上游边界的距离（单位 mm）。即使工件还未进入追踪范围，如果距离上游边界的距离在此距离范围内，则机器人不移动到待机位置。机器人从现在位置到达传送带工作站之前，工件已到达传送带工作站的追踪范围时，不需要移动到待机位置，是用于避免这种白移动到待机位置的设定值。此值根据传送带的速度进行调整。

使用例：
在 ID 为 1 的传送带工作站中，将基准位置示教为 PR[56]，根据 PKCSGETQUE 在 VR[1]中取得工件信息，在 VR[10]中取得 PKCSGETQUE 的状态。机器人从现在位置向待机位置移动期间，预计传送带将移动 150mm。在此条件下计算待机位置。是否移动到待机位置的判断结果返回 R[11]，计算的待机位置返回 PR[40]。

```
CALL PKCSCALWPOS(传送带工作站 ID=1,基准位置 PR=56,补偿用 VR=1,GetQ 状态=R[10],
移动判断用 Reg=11,待机位置 PR=40,无需移动范围=150)
```

7.9.6.2 PKCSGETAPPR.PC

取得通过树状图的［传送带工作站的操作］设定的接近位置偏移量。同时，指定的数值寄存器中，取得取得接近位置偏移量的位置寄存器序号。

参数 1：
指定传送带工作站的 ID。

参数 2：
指定存储存储接近位置偏移量的位置寄存器序号的数值寄存器序号。存储接近位置偏移量的位置寄存器序号，在树状图的［传送带工作站的操作］中指定。

使用例：
在位置寄存器中，取得以 ID 为 1 的传送带工作站相对应的传送带工作站的操作所设定的接近位置偏移量。同时，R［95］中，取得取得了接近位置偏移量的位置寄存器的序号。

```
CALL PKCSGETAPPR(传送带工作站 ID=1,接近位置 PR Reg=95)
```

7.9.6.3 PKCSGETOPPR.PC

取得通过树状图的［传送带工作站的操作］设定［基准位置 PR］的值，设定于数值寄存器。

参数 1：
指定传送带工作站的 ID。

参数 2：
指定存储［基准位置 PR］的值的数值寄存器序号。

使用例：
在 R［2］中，取得 ID 为 1 的传送带工作站相对应的传送带工作站的操作的［基准位置 PR］的值。

```
CALL PKCSGETOPPR(传送带工作站 ID=1,基准位置 PR Reg=2)
```

7.9.6.4 PKCSGETPMUL.PC

取得通过树状图的［传送带工作站的操作］设定的［多数拾取和放置］的设定(有效=1、无效=0)。

参数 1：
指定传送带工作站的 ID。

参数 2：
指定存储［多数拾取和放置］的有效/无效的数值寄存器序号。
1: 有效
0: 无效

使用例：
在 R［4］中，取得 ID 为 1 的传送带工作站相对应的传送带工作站的操作的［多数拾取和放置］的设定。

```
CALL PKCSGETPMUL(传送带工作站 ID=1,批量作业用 Reg=4)
```

7.9.6.5 PKCSGETPPCHK.PC

取得通过树状图的［传送带工作站的操作］设定的［工件有无检查］的设定（有效=1、无效=0）。

参数 1：
指定传送带工作站的 ID。

参数 2：
指定存储［工件有无检查］的有效/无效的数值寄存器序号。
1: 有效
0: 无效

使用例：
在 R［5］中，取得 ID 为 1 的传送带工作站相对应的传送带工作站的操作的［工件有无检查］的设定值。

CALL PKCSGETPPCHK(传送带工作站 ID=1,工件有无检查 Reg=5)

7.9.6.6 PKCSGETVR.PC

取得通过树状图的［传送带工作站的操作］设定的［工件数据 VR］的值，设定于数值寄存器。

参数 1：
指定传送带工作站的 ID。

参数 2：
指定存储［工件数据 VR］的值的数值寄存器序号。

使用例：
在 R［7］中，取得 ID 为 1 的传送带工作站相对应的传送带工作站的操作的［工件数据 VR］的值。

CALL PKCSGETVR(传送带工作站 ID=1,VR Reg=7)

7.9.6.7 PKCSGETWPOS.PC

取得通过树状图的［传送带工作站的操作］设定的［待机位置］的设定（有效=1、无效=0）。

参数 1：
指定传送带工作站的 ID。

参数 2：
指定存储［待机位置］有效/无效的数值寄存器序号。
1: 有效
0: 无效

使用例：
在 R［8］中，取得 ID 为 1 的传送带工作站相对应的传送带工作站的操作的［待机位置］的设定。

CALL PKCSGETWPOS(传送带工作站 ID=1,待机确认用 Reg=8)

7.9.7 有关固定工作站的程序

固定工作站相关程序打头的 4 个字符是［PKFS］。［PK］后面的 2 个字符是［FS］。

另外，以下程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

- PKFSGETID.PC
- PKFSPUTQUE.PC
- PKFSCLRQUE.PC
- PKFSGETQUE.PC
- PKFSACKQUE.PC
- PKFSPUTBUF.PC
- PKFSCLRBUF.PC
- PKFSGETBUF.PC
- PKFSACKBUF.PC
- PKFSGETTRID.PC
- PKFSGETUF.PC

7.9.7.1 PKFSGETAPPR.PC

取得通过树状图的［固定工作站的操作］设定的接近位置偏移量。同时，指定的数值寄存器中，取得取得接近位置偏移量的位置寄存器序号。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储存储接近位置偏移量的位置寄存器序号的数值寄存器序号。（存储接近位置偏移量的位置寄存器序号，在树状图的［固定工作站的操作］中指定。）

使用例：
在位置寄存器中，取得以 ID 为 1 的固定工作站相对应的固定工作站的操作所设定的接近位置偏移量。同时，R［95］中，取得取得了接近位置偏移量的位置寄存器的序号。

```
CALL PKFSGETAPPR(固定工作站 ID=1,接近位置 PR Reg=95)
```

7.9.7.2 PKFSGETIXDLY.PC

取得通过树状图的［固定工作站的操作］设定的［待机时间］。仅在固定工作站的作业开始信号模拟中，使用待机时间（单位 ms）。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［待机时间］的数值寄存器序号。

使用例：
在 R［6］中，取得以 ID 为 1 的固定工作站相对应的固定工作站的操作所设定的［待机时间］。

```
CALL PKFSGETIXDLY(固定工作站 ID=1,待机时间 Reg 6)
```

7.9.7.3 PKFSGETIXODO.PC

取得通过树状图的［固定工作站的操作］设定的［动作完成信号］的 DO 编号。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［动作完成信号］的 DO 编号的数值寄存器序号。

使用例：
在 R［7］中，取得以 ID 为 1 的固定工作站相对应的固定工作站的操作所设定的［动作完成信号］的 DO 编号。

```
CALL PKFSGETIXODO(固定工作站 ID=1,动作完成信号 Reg=7)
```

7.9.7.4 PKFSGETIXSIM.PC

取得通过树状图的［固定工作站的操作］设定的［动作开始信号模拟］的设定(有效=1、无效=0)。

参数 1：
指定固定工作站的 ID。
参数 2：
指定存储［动作开始信号模拟］有效/无效的寄存器序号。
1: 有效
0: 无效

使用例：
在 R［8］中，取得 ID 为 1 的固定工作站相对应的固定工作站的操作的［动作开始信号模拟］的设定值。

```
CALL PKFSGETIXSIM(固定工作站 ID=1,信号模拟 Reg=8)
```

7.9.7.5 PKFSGETOPPR.PC

取得通过树状图的［固定工作站的操作］设定的［基准位置 PR］的值，设定于数值寄存器。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［基准位置 PR］的值的数值寄存器序号。

使用例：
在 R［2］中，取得 ID 为 1 的固定工作站相对应的固定工作站的操作的［基准位置 PR］的值。

```
CALL PKFSGETOPPR(固定工作站 ID=1,基准位置 PR Reg=2)
```


7.9.7.6 PKFSGETPMUL.PC

取得通过树状图的［固定工作站的操作］设定的［多数拾取和放置］的设定(有效=1、无效=0)。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［多数拾取和放置］的有效/无效的数值寄存器序号。
1: 有效
0: 无效

使用例：
在 R［4］中，取得 ID 为 1 的固定工作站相对应的固定工作站的操作的［多数拾取和放置］的设定值。

```
CALL PKFSGETPMUL(固定工作站 ID=1,批量作业用 Reg=4)
```

7.9.7.7 PKFSGETPPCHK.PC

取得通过树状图的［固定工作站的操作］设定的［工件有无检查］的设定(有效=1、无效=0)。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［工件有无检查］的有效/无效的数值寄存器序号。
1: 有效
0: 无效

使用例：
在 R［5］中，取得 ID 为 1 的固定工作站相对应的固定工作站的操作的［工件有无检查］的设定值。

```
CALL PKFSGETPPCHK(固定工作站 ID=1,工件有无检查 Reg=5)
```

7.9.7.8 PKFSGETRDY.PC

取得固定工作站是否可进行作业（可= 0、不可= 1）。确认如果在固定工作站上启动机器人程序，*i*RPickTool 中固定工作站是否可进行动作。如果不能动作，也就是说，下一个工作站还没有到来的情况，则机器人不能在固定工作站进行拾取放置动作。动作可能的条件通常是，固定工作站的动作开始信号为 ON，动作完成信号为 OFF 时。但是，在固定

工作站的更换作业开始时，即使此时动作开始信号为 ON，也不能动作。之后，等待固定工作站的作业开始信号变为 OFF，再次变为 ON。（实际工作站上的动作是，动作结束的固定工作站被交付，等待新的固定工作站进入。）满足以上条件，且动作完成信号为 OFF 时，才能再次动作。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储固定工作站是否能动作的数值寄存器序号。

1: 可动作
0: 不可动作

使用例：
在 R［8］中取得 ID 为 1 的固定工作站是否可进行作业。

```
CALL PKFSGETRDY(固定工作站 ID=1,工作站就绪 Reg=8)
```

7.9.7.9 PKFSGETRDYDI.PC

取得通过树状图的［固定工作站的操作］设定的［动作开始信号］的 DI 编号。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［动作开始信号］的 DI 编号的数值寄存器序号。

使用例：
在 R［7］中，取得以 ID 为 1 的固定工作站相对应的固定工作站的操作所设定的［动作开始信号］的 DI 编号。

```
CALL PKFSGETRDYDI(固定工作站 ID=1,动作开始 DI Reg=7)
```

7.9.7.10 PKFSGETVR.PC

取得通过树状图的［固定工作站的操作］设定的［工件数据 VR］的值。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储［工件数据 VR］的值的数值寄存器序号。

使用例：
在 R［7］中，取得以 ID 为 1 的固定工作站相对应的固定工作站的操作所设定的［工件数据 VR］的值。

```
CALL PKFSGETVR(固定工作站 ID=1,VR Reg=7)
```

7.9.7.11 PKFSMONIDX.PC

固定工作站的更换开始后，立即使用的程序。该程序中，最初将固定工作站的状态设定为不可动作。然后，监视该固定工作站的动作开始信号变为 OFF（固定工作站的正在更换的状态），之后再变为 ON（更换结束，准备新的固定工作站的状态）。动作开始信号为 ON，且动作完成信号为 OFF 时，该固定工作站的状态设定为可动作。（如果机器人不在该状态下处于可动作状态，则无法在固定工作站内动作）。

参数 1：
指定固定工作站的 ID。

参数 2：
指定存储该程序的状态的数值寄存器序号。如果是 0 以外的情况，则表示对固定工作站的动作开始信号的监视失败。

使用例：
监视 ID 为 1 的固定工作站的动作开始信号。R［7］中取得该程序的状态。

```
CALL PKFSMONIDX(固定工作站 ID=1,状态 Reg=7)
```

7.9.7.12 PKFSSETRDY.PC

设定固定工作站是否可进行作业。一般在由于安全问题，需要使固定工作站的状态为不可动作时使用。请参照“7.9.7.8 PKFSGETRDY.PC”，决定内部使用怎样的处理方法，进行可动作的确认。

参数 1：
指定固定工作站的 ID。

参数 2：
设定固定工作站是否可进行作业。

注释
若已经显示参数的说明，则作为示教时的选项，显示
1 可动作
2 不可动作
内部设定为 1：可动作/0：不可动作。
不显示参数的说明时，“可动作”时请指定为 1，“不可动作”时请指定为 0。

使用例：
使 ID 为 1 的固定工作站为不可动作。

```
CALL PKFSSETRDY(固定工作站 ID=1,未就绪)
```

7.9.8 有关托盘的程序

托盘相关程序打头的 4 个字符是［PKTR］。［PK］后面的 2 个字符是［TR］。

另外，托盘相关程序与基本功能的 KAREL 程序相同。请参阅“6.8 KAREL 程序”。

7.10 标准 TP 程序

本节对以 Plug & Play 提供的标准 TP 程序进行说明。
这些标准 TP 程序，具有以下功能。有关机器人的实际动作，请参照“7.5.1 标准 TP 程序下的机器人的动作”。

- 以上支持所有的 FANUC 机器人。FANUC 机器人中，有 M-3*i*A/6S、M-2*i*A/3S 等的拳头机器人、LR Mate 系列等、3～6 轴的机器人。
- 支持以下的应用程序。
  - 机器人从传送带拾取、放置于传送带的应用程序
  - 机器人从传送带拾取、放置于固定工作站，或相反动作的应用程序
  - 机器人从固定工作站拾取、放置于固定工作站的应用程序
- 拾取/放置的对象，可选择工件或托盘的单元。
- 拾取/放置可以使 1 个，也可以同时多个(最多 20 个)。
- 此外，还具有在工件还没有到达追踪范围的情况下，自动移动至待机位置、进行抓手的工件有无检查等应用程序所必须的功能。

7.10.1 有关标准 TP 程序的名称

*i*RPickTool 的全部的 TP 程序，以“PK_”打头。

通过在示教器的［选择］画面按下 F1［类型］，选择“8 ‘PK_’”并进行过滤，能快速显示只有 *i*RPickTool 的标准 TP 程序。参照下图。

此外，如果选择“9 ‘PK’”，则还能同时显示 *i*RPickTool 的 KAREL 程序。

标准 TP 程序的名称，根据以下规则决定。请将其作为寻找程序时的标准。

- 名称 ＝PK_＋ 名称
- 名称 ＝PK_＋ 名称 ＋ 组
- 名称 ＝PK_＋ 对象名 ＋ 动作 ＋ 组编号 ＋ 作业顺序

以 2 个字符表示对象名。
- CV:传送带
- CS：传送带工作站
- FS：固定工作站
- GR：抓手

以下是动作例。
- GET_PICK_DATA11
- PICK11
- DROP11

动作的后面，以 1 位或 2 位数字结束。
2 位数字的情况，最初的数字表示组编号，第 2 个数字表示作业顺序。

例如，PICK11 的情况，表示由组 1 的机器人拾取工件，这是作为拾取工件作业的最初的作业。

此外，还有如下的更为简单规则的程序。
名称 ＝PK_＋ 名称 ＋ 组
例：
- PK_MAIN1：组 1 的机器人的主程序
- PK_INIT1：组 1 的机器人的初始化程序
- PK_PERCH1：组 1 的机器人的原位置移动程序

追加组时，需要自行追加 TP 程序，但只要依照相同规则对程序命名，就能有效地管理 TP 程序。

7.10.2 标准 TP 程序的详细

7.10.2.1 标准 TP 程序一览

以下是组 1 的标准 TP 程序一览。

No.	程序名	备注
2	PK_CV_DROP11	[DropG1Cv1]
3	PK_CV_GET_DROP_DATA11	[GetDpDataG1Cv1]
4	PK_CV_GET_PICK_DATA11	[GetPkDataG1Cv1]
5	PK_CV_PICK11	[PickG1Cv1]
6	PK_CV_WAITPOS1	[WaitG1Cv]
7	PK_CYCSTOP1	[CycleStopG1]
8	PK_DROP1	[DropSelectG1]
9	PK_FS_DROP11	[DropG1FS1]
10	PK_FS_GET_DROP_DATA11	[GetPkDataG1FS1]
11	PK_FS_GET_PICK_DATA11	[GetPkDataG1FS1]
12	PK_FS_INDEX	[IndexFS]
13	PK_FS_PICK11	[PickG1FS1]
14	PK_FS_READY	[ReadyCheckFS]
15	PK_GR_GET_ENDZONE1	[GripGetEZoneG1]
16	PK_GR_GET_UTOOLS1	[GripGetUToolsG1]
17	PK_INIT1	[InitG1]
18	PK_MAIN1	[MainG1]
19	PK_PERCH1	[PerchG1]
20	PK_PICK1	[PickSelectG1]

7.10.2.2 PK_MAIN1.TP: 组 1 的主程序

本程序是主程序。标准设定中，例如从 PLC 等，通过 UOP 启动程序，则该程序自动运行。

该程序以下面步骤动作。

1 进行寄存器等系统的初始化。取得通过树状图的设定，分配给寄存器。并且，进行工具坐标系和用户坐标系的设定。同时，根据运行的程序是追踪程序或非追踪程序，决定运行拾取/放置程序。
2 根据相机、DI、RI 或 HDI，开始检知工件的传感器任务。
3 使机器人移动至原位置。
4 使用预整列时，调出预整列的主程序。
5 不使用预整列时，以无限循环运行包括拾取/放置动作的程序。在拾取/放置动作分别结束时，确认是否有周期停止的要求。
6 如果有周期停止的要求，则使周期停止。
7 从拾取/放置程序的循环中（因周期停止等）退出后，或者从预整列的主程序退出后，结束传感器任务。

PK_MAIN1.TP

```
   1:  ! Main Prog for Grp:1
   2:
   3:  LBL[10]
   4:  --
   5:  CALL PK_INIT1
   6:  --
   7:  CALL PKSNSTART
   8:  --
   9:  CALL PK_PERCH1
  10:  ! Pre-grouping is done by
  11:  ! PK_PRGR_MAIN1.TP
  12:  !R[230] is set in PK_PRGR_INIT1.
  13:  !TP called in PK_INIT1.TP
  14:  IF R[230:PRGR Robot Type]<=0 OR R[230:PRGR Robot Type]>2,JMP LBL[100]
  15:  CALL PK_PRGR_MAIN1
  16:  JMP LBL[10000]
  17:
  18:  LBL[100:pk n pl]
  19:  --
  20:  CALL PK_PICK1
  21:  IF R[101:Cycle Stop G1]>0,JMP LBL[10000]
  22:  --
  23:  CALL PK_DROP1
  24:  LBL[200]
  25:  IF R[101:Cycle Stop G1]>0,JMP LBL[10000]
  26:  JMP LBL[100]
  27:
  28:  LBL[10000: End]
  29:  --
  30:  CALL PKSNSTOP
[End]
```

7.10.2.3 PK_INIT1.TP

本程序，是组 1 的主程序 PK_MAIN1.TP 的初始化程序。

该程序以下面步骤动作。

1 *i*RPickTool 的设定数据在所有的控制装置中被同步。
2 取得使用中的方案 ID。
3 取得组 1 的机器人的机器人 ID。
4 取得机器人的抓手 ID。
5 取得抓手的区域的设定数据，设定于工具坐标系。
6 恢复或切换方案时，使用 KAREL 程序 PKCOPYPR2POS. PC 向 PK_PERCH1.TP 的 P［1］反映位置数据。
7 判断进行拾取动作的工作站是传送带工作站还是固定工作站。
8 判断进行放置动作的工作站是传送带工作站还是固定工作站。
9 取得需要机器人和传送带工作站以及固定工作站相关的所有数据。
10 取得组 1 的缓冲区 ID。缓冲区 ID 比 0 大，也就是说，Plug & Play 的缓冲区功能有效时，执行 PK_BF_GET_DATA1.TP，将与缓冲区相关的系统变量取到寄存器或位置寄存器中。已获取的系统变量，将在从缓冲区获取工件，或者将工件放置在缓冲区中的程序中进行参照。Plug & Play 的缓冲区功能，包含在 R857: *i*RPickTool /扩充功能包加载项中（北美专用 R827: *i*RPickTool / Production Control Add-on）中。
11 取得预整列的类型。不使用预整列时，类型为 0。
12 对工件有无检查寄存器等特定寄存器进行初始化。

PK_INIT1.TP

```
  1:  ! PKINIT1
  2:  ! Check data is synchronized and
  3:  ! load balance data is valid
  4:  CALL PKWCCHECK
  5:
  6:  ! Group 1 inits
  7:  R[99:Active Recipe]=$PKWC.$ACT_RC_ID
  8:
  9:  ! Get robot from group
 10:  CALL PKRBGETID(组编号=1,机器人 ID Reg=103)
 11:
 12:  ! Get gripper for robot
 13:  ! PKGRGETID(rob_id, reg_gripId)
 14:  CALL PKGRGETID(机器人 ID=R[103:Robot Id G1],抓手 ID Reg=104)
 15:  ! Set $MNUTOOLs from Gripper Zone
 16:  CALL PK_GR_GET_UTOOLS1(机器人 ID=R[103:Robot Id G1],抓手 ID=R[104:Gripper Id G1])
 17:
 18:
 19:  --
 20:  --
 21:  IF R[105:Recipe loaded G1]<>1,JMP LBL[25]
 22:  --
 23:  UTOOL_NUM=0
 24:  CALL PKCOPYPR2POS(位置 PR=54,程序名称='PK_PERCH1',位置数据编号=1,组编号=1)
 25:  R[105:Recipe loaded G1]=0
 26:  LBL[25]
 27:
 28:  LBL[50: Pick Inits]
 29:  ! Get the count of pick conveyors
 30:  !   for this robot
 31:  CALL PKRBGETCVCNT(机器人 ID=R[103:Robot Id G1],拾取=0,传送带计数 Reg=95)
 32:  IF R[95:TemporaryUse1]=0,JMP LBL[100]
 33:  ! Get 1st pick conveyor's data
 34:  CALL PK_CV_GET_PICK_DATA11
 35:  JMP LBL[400]
 36:
 37:  LBL[100: GetPkFStn]
 38:  ! Get the count of pick FStns
 39:  !    for this robot
 40:  CALL PKRBGETFSCNT(机器人 ID=R[103:Robot Id G1],拾取=0,固定工作站计数 R=95)
 41:  IF R[95:TemporaryUse1]=0,JMP LBL[300]
 42:  ! Get pick FStn's data
 43:  CALL PK_FS_GET_PICK_DATA11
 44:  JMP LBL[400]
 45:
 46:  LBL[300: No Pick Stns]
 47:  UALM[3]
 48:  JMP LBL[10000]
 49:
 50:
 51:  LBL[400: Drop Inits]
 52:  ! Get the count of drop conveyors
 53:  !   for this robot
 54:  CALL PKRBGETCVCNT(机器人 ID=R[103:Robot Id G1],放置=1,传送带计数 Reg=95)
 55:  IF R[95:TemporaryUse1]=0,JMP LBL[500]
 56:  ! Get drop conveyor's data
 57:  CALL PK_CV_GET_DROP_DATA11
```

```
 58:   JMP LBL[1000]
 59:
 60:   LBL[500: GetDpFStn]
 61:   ! Get the count of drop FStns
 62:   !   for this robot
 63:   CALL PKRBGETFSCNT(机器人 ID=R[103:Robot Id G1],放置=1,固定工作站计数 R=95)
 64:   IF R[95:TemporaryUse1]=0,JMP LBL[600]
 65:   ! Get drop FStn's data
 66:   CALL PK_FS_GET_DROP_DATA11
 67:   JMP LBL[1000]
 68:
 69:   LBL[600: No Drop Stns]
 70:   UALM[4]
 71:   JMP LBL[10000]
 72:
 73:
 74:   LBL[1000: Buf Inits]
 75:   ! Buffer initializations
 76:   CALL PKRBGETBUFID(机器人 ID=R[103:Robot Id G1],缓冲区 ID Reg=199)
 77:   IF R[199:Buf Id G1]<=0,JMP LBL[2000]
 78:   CALL PK_BF_GET_DATA1
 79:
 80:   LBL[2000:Pre-group Inits]
 81:   ! Check for Pre-group option
 82:   !PKPRGRGETEN(Enb_Reg)
 83:   CALL PKPRGRGETEN(预整列有效 Reg=230)
 84:   IF R[230:PRGR Robot Type]=0,JMP LBL[3000]
 85:   !Check if pre-group is used
 86:   CALL PK_PRGR_INIT1(预整列类型 Reg=230)
 87:   IF R[230:PRGR Robot Type]=0,JMP LBL[3000]
 88:   IF R[199:Buf Id G1]<>0,JMP LBL[2025]
 89:   JMP LBL[3000]
 90:   LBL[2025]
 91:   ! Cannot use both PRGR and Buffer
 92:   UALM[6]
 93:
 94:   LBL[3000:User Init]
 95:   ! User initialization
 96:   R[117:Pk1x PPStat]=0
 97:   R[137:Dp1x PPStat]=0
 98:
 99:   LBL[10000:End]
[End]
```

7.10.2.4 PK_PERCH1.TP

本程序是使组 1 的机器人移动至原位置的程序。工具坐标系编号、用户坐标系编号均使用 0。P［1］为原位置，在第 17 行被复制到 PR［54］中。若保存方案，则 PR［54］的位置数据将被作为原位置来保存。

PK_PERCH1.TP

```
 1:  --
 2:
 3:  UTOOL_NUM=0
 4:  UFRAME_NUM=0
 5:  !*********************************
 6:  --
 7:  --
 8:  --
```

```
 9:   --
10:   --
11:   --
12:   --
13:   !*********************************
14:   PAYLOAD[1]
15:J P[1] 25% FINE
16:   --
17:   PR[54:Perch Pos]=P[1]
[End]
```

7.10.2.5 PK_PICK1.TP

本程序对组 1 的机器人的拾取动作是追踪还是非追踪进行判断，并调出适当的程序。

此外，本程序中，通常执行 PK_CV_PICK11.TP，从供给传送带拾取工件。另外，在使用了 Plug & Play 的缓冲区功能时，执行 PK_BUF_PICK1.TP，判断从传送带拾取工件，还是从缓冲区拾取工件。
Plug & Play 的缓冲区功能，包含在 R857: *i*RPickTool /扩充功能包加载项中（北美专用 R827: *i*RPickTool / Production Control Add-on）中。详情请参阅“7.12 Plug & Play 的缓冲区功能”。

使用预整列时，在本程序中，执行 PK_PRGR_CV_PICKVTRAY1，从供给传送带中汇总获取整列的工件。详情请参阅“7.11 预整列功能的设定”。

PK_PICK1.TP

```
 1:   !Pick program G1
 2:   !Choose trk or non-trk pick prog
 3:
 4:   R[107:Counter G1]=1
 5:   IF R[109:Pk1x CvId]=0,JMP LBL[1000]
 6:   IF (R[199:Buf Id G1]<>0),JMP LBL[50]
 7:   IF R[230:PRGR Robot Type]=3,JMP LBL[60]
 8:   ! Pick from conveyor
 9:   CALL PK_CV_PICK11
10:   JMP LBL[10000]
11:   ! Buffer handling
12:   LBL[50: Buf loop]
13:   CALL PK_BUF_PICK1
14:   JMP LBL[10000]
15:   LBL[60:Virtual Tray]
16:   !Pick virtual tray
17:   CALL PK_PRGR_CV_PICKVTRAY1
18:   JMP LBL[10000]
19:
20:   LBL[1000: FStn]
21:   ! Non-tracking pick from FStn
22:   CALL PK_FS_PICK11
23:   LBL[10000: End]
[End]
```

7.10.2.6 PK_DROP1.TP

本程序对组 1 的机器人的放置动作是追踪还是非追踪进行判断，并调出适当的程序。

此外，本程序中，通常执行 PK_CV_DROP11.TP 并向输出传送带放置工件。另一方面，在使用了 Plug & Play 的缓冲区功能时，执行 PK_BUF_DROP1.TP，判断向传送带放置工件，还是向缓冲区放置工件。
Plug & Play 的缓冲区功能，包含在 R857: *i*RPickTool /扩充功能包加载项中（北美专用 R827: *i*RPickTool / Production Control Add-on）中。详情请参阅“7.12 Plug & Play 的缓冲区功能”。

PK_DROP1.TP

```
   1:  !Drop program G1
   2:  !Choose trk or non-trk drop prog
   3:
   4:  R[107:Counter G1]=1
   5:  IF R[129:Dp1x CvId]=0,JMP LBL[1000]
   6:  IF (R[199:Buf Id G1]<>0),JMP LBL[50]
   7:  !Place on conveyor
   8:  CALL PK_CV_DROP11
   9:  JMP LBL[10000]
  10:  ! Buffer handling
  11:  LBL[50: Buffer drop]
  12:  CALL PK_BUF_DROP1
  13:  JMP LBL[10000]
  14:
  15:  LBL[1000: FixedPos]
  16:  ! Non-tracking drop at FStn
  17:  CALL PK_FS_DROP11
  18:  LBL[10000: End]
[End]
```

7.10.2.7 PK_GR_GET_UTOOLS1.TP

该程序以下面的步骤，对组 1 的机器人使用的抓手的各区域的工具坐标系进行设定。

1　使用 KAREL 程序 PKGRGETZNCNT，取得指定的抓手的［区域数量］。
2　使用 KAREL 程序 PKGRGETZNUT，取得在各区域设定的［区域工具坐标系］。
3　从工具坐标系的第 1 开始，依次根据［区域数量］设定取得的工具坐标系。

在调用此程序时，需要如下 2 个参数。
参数 1：机器人 ID
参数 2：抓手 ID

PK_GR_GET_UTOOLS1.TP

```
   1:  --
   2:  --
   3:  --
   4:  --
   5:  --
   6:
   7:  --
   8:
   9:  CALL PKGRGETZNCNT(抓手 ID=AR[2],区域数量 Reg=96)
  10:   FOR R[95:TemporaryUse1]=1 TO R[96:TemporaryUse2]
  11:
  12:   CALL PKGRGETZNUT(机器人 ID=AR[1],抓手 ID=AR[2],区域 ID=R[95:TemporaryUse1],工具坐标系
PR=51)
  13:
  14:   UTOOL[R[95]]=PR[51:TemporaryUse1]
  15:   ENDFOR
  16:   ! PickQty = PlaceQty = ZoneCnt
  17:   R[111:Pk1x Qty]=R[96:TemporaryUse2]
  18:   R[131:Dp1x Qty]=R[96:TemporaryUse2]
[End]
```

7.10.2.8 PK_CV_GET_PICK_DATA11.TP

本程序，通过以下的步骤，取得在传送带工作站的拾取动作相关的树状图的设定。

1 从$PKMAIN［1］.$PK_CV［1］取得最初的拾取动作的传送带 ID。
2 使用 KAREL 程序 PKCVGETNAME，取得最初的传送带的名称。
3 使用 KAREL 程序 PKCSGETID，取得传送带工作站的 ID。
4 使用 KAREL 程序 PKCSGETPMUL，取得在传送带工作站的操作中设定的［多数拾取和放置］的有效/无效的设定。
5 取得在传送带工作站的操作中设定的［接近时偏移量］，使用 KAREL 程序 PKCSGETAPPR，在相同的传送带工作站的操作设定的［接近偏移量 PR］中进行设定。
6 使用 KAREL 程序 PKRBGETSFLG，取得在机器人中设定的［跳过动作实施范围］的［标记编号］。
7 使用 KAREL 程序 PKCSGETTRID，取得在传送带上使用中的托盘的 ID。
8 用 KAREL 程序 PKCSCLRACK，清除未处理的 ACK，从而避免报警“GETQUE 之后未执行 ACKQUE”。
9 使用 KAREL 程序 PKCSCLRQUE，根据系统变量$PKCLRQ 的设定，清除传送带工作站的队列。

⚠ 注意
该程序，例如第 12 行：如同 CALL PKCSGETAPPR(传送带工作站 ID=R［110:Pk1x StnId］,接近位置 PR Reg=95) ，暂时被使用的数值寄存器——R［95］被多次使用。标准 TP 程序中，不使用这些被盖写的值，所以不成问题，但是在变更标准 TP 程序后使用时，请根据需要，采取退避至 R［95］以外的不会被盖写的寄存器等措施。

PK_CV_GET_PICK_DATA11.TP

```
  1:  --
  2:  --
  3:  --
  4:  --
  5:
  6:  --
  7:  R[109:Pk1x CvId]=$PKMAIN[1].$PK_CV[1]
  8:  --
  9:  CALL PKCVGETNAME(“传送带 ID”=R[109:Pk1x CvId],“传送带名称 SR”=4)
 10:  CALL PKCSGETID(“CONV1”=SR[4],“工作站 ID Reg”=110)
 11:  CALL PKCSGETPMUL(“传送带工作站 ID”=R[110:Pk1x StnI],“批量作业用 Reg”=112)
 12:  CALL PKCSGETAPPR(“传送带工作站 ID”=R[110:Pk1x StnI],“接近位置 PR Reg”=95)
 13:  CALL PKRBGETSFLG(“机器人 ID”=R[103:Robot Id G1],“跳过动作 Reg”=102)
 14:  !Get CStn's tray id
 15:  CALL PKCSGETTRID(“传送带工作站 ID”=R[110:Pk1x StnI],“托盘 ID Reg”=121)
 16:  --
 17:  CALL PKCSCLRACK(“传送带工作站 ID”=R[110:Pk1x StnI])
 18:  ! Clear the queues based on
 19:  ! $PKCLRQ for G1 CStns
 20:  R[95:TemporaryUse1]=$PKCLRQ
 21:  IF R[95:TemporaryUse1]=0,JMP LBL[5000]
 22:  CALL PKCSCLRQUE(“传送带工作站 ID”=R[110:Pk1x StnI])
 23:
 24:  LBL[5000: Eff Tools]
 25:  !Get position check flag
 26:  !PKSCSGETPOSCK(cs_id,
 27:  ! operation, reg_PosCheck)
 28:  CALL PKCSGETPOSCK(“传送带工作站 ID”=R[110:Pk1x StnI],“拾取”=0,“拾取可否检查 Reg”=115)
 29:  LBL[10000:End]
[End]
```

7.10.2.9 PK_CV_GET_DROP_DATA11.TP

本程序，通过以下的步骤，取得在传送带工作站的放置动作相关的树状图的设定。

1 从$PKMAIN［1］.$DP_CV［1］取得最初的拾取动作的传送带 ID。
2 使用 KAREL 程序 PKCVGETNAME，取得最初的传送带的名称。
3 使用 KAREL 程序 PKCSGETID，取得传送带工作站的 ID。
4 使用 KAREL 程序 PKCSGETPMUL，取得在传送带工作站的操作中设定的［多数拾取和放置］的有效/无效的设定。
5 取得在传送带工作站的操作中设定的［接近时偏移量］，使用 KAREL 程序 PKCSGETAPPR，在相同的传送带工作站的操作设定的［接近偏移量 PR］中进行设定。
6 使用 KAREL 程序 PKRBGETSFLG，取得在机器人中设定的［跳过动作实施范围］的［标记编号］。
7 使用 KAREL 程序 PKCSGETTRID，取得在传送带上使用中的托盘的 ID。
8 用 KAREL 程序 PKCSCLRACK，清除未处理的 ACK，从而避免报警“GETQUE 之后未执行 ACKQUE”。
9 使用 KAREL 程序 PKCSCLRQUE，根据系统变量$PKCLRQ 的设定，清除传送带工作站的队列。

⚠ 注意
该程序，例如第 12 行：如同 CALL PKCSGETAPPR(传送带工作站 ID=R［130:Dp1x StnId］,接近位置 PR Reg=95)，暂时被使用的数值寄存器——R［95］被多次使用。标准 TP 程序中，不使用这些被盖写的值，所以不成问题，但是在变更标准 TP 程序后使用时，请根据需要，采取退避至 R［95］以外的不会被盖写的寄存器等措施。

PK_CV_GET_DROP_DATA11.TP

```
   1:  --
   2:  --
   3:  --
   4:  --
   5:
   6:  --
   7:  R[129:Dp1x CvId]=$PKMAIN[1].$DP_CV[1]
   8:  --
   9:  CALL PKCVGETNAME(“传送带 ID”=R[129:Dp1x CvId],“传送带名称 SR”=5)
  10:  CALL PKCSGETID(“CONV1”=SR[5],“工作站 ID Reg”=130)
  11:  CALL PKCSGETPMUL(“传送带工作站 ID”=R[130:Dp1x StnI],“批量作业用 Reg”=132)
  12:  CALL PKCSGETAPPR(“传送带工作站 ID”=R[130:Dp1x StnI],“接近位置 PR Reg”=95)
  13:  CALL PKRBGETSFLG(“机器人 ID”=R[103:Robot Id G1],“跳过动作 Reg”=102)
  14:  !Get CStn's tray id
  15:  CALL PKCSGETTRID(“传送带工作站 ID”=R[130:Dp1x StnI],“托盘 ID Reg”=141)
  16:  --
  17:  CALL PKCSCLRACK(“传送带工作站 ID”=R[130:Dp1x StnI])
  18:  ! Clear the queues based on
  19:  ! $PKCLRQ for G1 CStns
  20:  R[95:TemporaryUse1]=$PKCLRQ
  21:  IF R[95:TemporaryUse1]=0,JMP LBL[5000]
  22:  CALL PKCSCLRQUE(“传送带工作站 ID”=R[130:Dp1x StnI])
  23:
  24:  LBL[5000: Eff Tools]
  25:  !Get position check flag
  26:  !PKSCSGETPOSCK(cs_id,
  27:  ! operation, reg_PosCheck)
  28:  CALL PKCSGETPOSCK(“传送带工作站 ID”=R[130:Dp1x StnI],“放置”=1,“拾取可否检查 Reg”=135)
  29:  LBL[10000:End]
[End]
```

7.10.2.10 PK_FS_GET_PICK_DATA11.TP

本程序，通过以下的步骤，取得在固定工作站的拾取动作相关的树状图的设定。

1 从$PKMAIN［1］.$PK_FS［1］ 取得最初的拾取动作的固定工作站 ID。
2 使用 KAREL 程序 PKFSGETPMUL，取得在固定工作站的操作中设定的［多数拾取和放置］的有效/无效的设定。
3 取得在固定工作站的操作中设定的［接近时偏移量］，使用 KAREL 程序 PKFSGETAPPR，在相同的固定工作站的操作设定的［接近偏移量 PR］中进行设定。
4 使用 KAREL 程序 PKFSGETRDYDI，取得在固定工作站的操作中设定的［动作开始信号］的 DI 编号。
5 使用 KAREL 程序 PKFSGETIXODO，取得在固定工作站的操作中设定的［动作完成信号］的 DO 编号。
6 使用 KAREL 程序 PKFSSETRDY，使固定工作站的状态初始化为不可动作。
7 使用 KAREL 程序 PKFSGETIXSIM，确认在固定工作站的操作中设定的［动作开始信号模拟］是否为有效。
8 使用 KAREL 程序 PKFSGETTRID，取得在固定工作站中设定的托盘的 ID。
9 用 KAREL 程序 PKFSCLRACK，清除未处理的 ACK，从而避免报警“GETQUE 之后未执行 ACKQUE”。
10 使用 KAREL 程序 PKFSCLRQUE 和 PKFSPUTQUE，根据系统变量$PKCLRQ 的设定对托盘进行初始化。

⚠ 注意
该程序，例如第 9 行：如同 CALL PKFSGETAPPR(固定工作站 ID=R［110:Pk1x StnId］,接近位置 PR Reg=95)，暂时被使用的数值寄存器——R［95］被多次使用。标准 TP 程序中，不使用这些被盖写的值，所以不成问题，但是在变更标准 TP 程序后使用时，请根据需要，采取退避至 R［95］以外的不会被盖写的寄存器等措施。

PK_FS_GET_PICK_DATA11.TP

```
   1:  --
   2:  --
   3:  --
   4:
   5:  --
   6:  R[109:Pk1x CvId]=0
   7:  R[110:Pk1x StnId]=$pkmain[1].$pk_fs[1]
   8:   CALL PKFSGETPMUL(固定工作站 ID=R[110:Pk1x StnId],批量作业用 Reg=112)
   9:  CALL PKFSGETAPPR(固定工作站 ID=R[110:Pk1x StnId],接近位置 PR Reg=95)
  10:  CALL PKFSGETRDYDI(固定工作站 ID=R[110:Pk1x StnId],动作开始 DI Reg=118)
  11:  CALL PKFSGETIXODO(固定工作站 ID=R[110:Pk1x StnId],动作完成信号 Reg=119)
  12:  CALL PKFSSETRDY(固定工作站 ID=R[110:Pk1x StnId],就绪=1)
  13:  CALL PKFSGETIXSIM(固定工作站 ID=R[110:Pk1x StnId],信号模拟 Reg=95)
  14:  CALL PKFSGETTRID(固定工作站 ID=R[110:Pk1x StnId],托盘 ID Reg=121)
  15:  CALL PKFSCLRACK(固定工作站 ID=R[110:Pk1x StnId])
  16:  IF R[121:Pk1x TrayId]=0,JMP LBL[10000]
  17:  R[95:TemporaryUse1]=$PKCLRQ
  18:  IF R[95:TemporaryUse1]=0,JMP LBL[1000]
  19:  CALL PKFSCLRQUE(固定工作站 ID=R[110:Pk1x StnId])
  20:  CALL PKFSPUTQUE(固定工作站 ID=R[110:Pk1x StnId])
  21:  --
  22:  R[122:Pk1x TrayCtr]=1
  23:   LBL[1000]
  24:
  25:   LBL[10000: End]
[End]
```

7.10.2.11 PK_FS_GET_DROP_DATA11.TP

本程序，通过以下的步骤，取得在固定工作站的放置动作相关的树状图的设定。

1 从$PKMAIN［1］.$DP_FS［1］ 取得最初的放置动作的固定工作站 ID。
2 使用 KAREL 程序 PKFSGETPMUL，取得在固定工作站的操作中设定的［多数拾取和放置］的有效/无效的设定。
3 取得在固定工作站的操作中设定的［接近时偏移量］，使用 KAREL 程序 PKFSGETAPPR，在相同的固定工作站的操作设定的［接近偏移量 PR］中进行设定。
4 使用 KAREL 程序 PKFSGETRDYDI，取得在固定工作站的操作中设定的［动作开始信号］的 DI 编号。
5 使用 KAREL 程序 PKFSGETIXODO，取得在固定工作站的操作中设定的［动作完成信号］的 DO 编号。
6 使用 KAREL 程序 PKFSSETRDY，使固定工作站的状态初始化为不可动作。
7 使用 KAREL 程序 PKFSGETIXSIM，确认在固定工作站的操作中设定的［动作开始信号模拟］是否为有效。
8 使用 KAREL 程序 PKFSGETTRID，取得在固定工作站中设定的托盘的 ID。
9 用 KAREL 程序 PKFSCLRACK，清除未处理的 ACK，从而避免报警“GETQUE 之后未执行 ACKQUE”。
10 使用 KAREL 程序 PKFSCLRQUE 和 PKFSPUTQUE，根据系统变量$PKCLRQ 的设定对托盘进行初始化。

> ⚠ **注意**
> **该程序，例如第 9 行：如同 CALL PKFSGETAPPR(固定工作站 ID=R［130:Dp1x StnId］,接近位置 PR Reg=95)，暂时被使用的数值寄存器——R［95］被多次使用。标准 TP 程序中，不使用这些被盖写的值，所以不成问题，但是在变更标准 TP 程序后使用时，请根据需要，采取退避至 R［95］以外的不会被盖写的寄存器等措施。**

PK_FS_GET_DROP_DATA11.TP

```
  1:  --
  2:  --
  3:  --
  4:
  5:  --
  6:  R[129:Dp1x CvId]=0
  7:  R[130:Dp1x StnId]=$pkmain[1].$dp_fs[1]
  8:  CALL PKFSGETPMUL(“固定工作站 ID”=R[130:Dp1x StnId],“批量作业用 Reg”=132)
  9:  CALL PKFSGETAPPR(“固定工作站 ID”=R[130:Dp1x StnId],“接近位置 PR Reg”=95)
 10:  CALL PKFSGETRDYDI(“固定工作站 ID”=R[130:Dp1x StnId],“动作开始 DI Reg”=138)
 11:  CALL PKFSGETIXODO(“固定工作站 ID”=R[130:Dp1x StnId],“动作完成信号 Reg”=139)
 12:  CALL PKFSSETRDY(“固定工作站 ID”=R[130:Dp1x StnId],“就绪”=1)
 13:  CALL PKFSGETIXSIM(“固定工作站 ID”=R[130:Dp1x StnId],“信号模拟 Reg”=95)
 14:  CALL PKFSGETTRID(“固定工作站 ID”=R[130:Dp1x StnId],“托盘 ID Reg”=141)
 15:  CALL PKFSCLRACK(“固定工作站 ID”=R[130:Dp1x StnId])
 16:  IF R[141:Dp1x TrayId]=0,JMP LBL[10000]
 17:  R[95:TemporaryUse1]=$PKCLRQ
 18:  IF R[95:TemporaryUse1]=0,JMP LBL[1000]
 19:  CALL PKFSCLRQUE(“固定工作站 ID”=R[130:Dp1x StnId])
 20:  CALL PKFSPUTQUE(“固定工作站 ID”=R[130:Dp1x StnId])
 21:  --
 22:  R[142:Dp1x TrayCtr]=1
 23:  LBL[1000]
 24:
 25:  LBL[10000: End]
[End]
```

7.10.2.12 PK_CV_PICK11.TP

本程序中，组 1 的机器人在传送带工作站上，以下面的步骤执行最初的拾取动作。

1 在开始拾取动作之前,将［跳过动作实施范围］的标记设定为 OFF。
2 使用 KAREL 程序 PKCSGETQUE，从队列中取得工件信息。
3 不管是否已从队列取得工件，在与此拾取动作对应的传送带工作站操作的设定中［待机位置］有效时，检查是否需要在传送带工作站的待机位置也即上游侧的边界等待机器人，若判断为需要进行等待，则使用 PKCSCALWPOS.PC 计算待机位置。
4 因工件不存在、或者工件尚未到达上游侧边界而机器人需要等待时，使用 PK_CV_WAITPOS1.TP 移动至待机位置。
5 未能利用 KAREL 程序 PKCSGETQUE 取得工件信息时，再次尝试取得工件信息。
6 通过 KAREL 程序 PKCSGETQUE 取得工件数据时，进入 7 以后步骤。
7 将所选的工具坐标系设定为抓手的最初区域的工具坐标系。
8 使得机器人移动至接近位置。
9 如果在移动过程中，［跳过动作实施范围］的标记为 ON（由于超出了下游侧的边界，机器人无法拾取工件），则使用 KAREL 程序 PKCSACKQUE，将工件数据以 ACK = 发信(3)返回队列，并返回程序开头，取得新的工件数据。
10 使得机器人移动至拾取位置。使用先执行指令，与动作并行执行“关闭抓手”动作。“关闭抓手”动作通过 KAREL 程序 PKGRCLOSE 执行。
11 如果在移动过程中，［跳过动作实施范围］的标记为 ON（由于超出了下游侧的边界，机器人无法拾取工件），则使用 KAREL 程序 PKCSACKQUE，将工件数据以 ACK = 发信(3)返回队列，并返回程序开头，取得新的工件数据。
12 移动至拾取位置后，使用 KAREL 程序 PKGRGETPKDLY 取得［拾取延迟时间］，在该时间内与传送带一起移动。
13 在与此拾取动作对应的传送带工作站操作的设定中［工件有无检查］有效时，使用 KAREL 程序 PKGRCHKPP 来检查是否已把持工件。
14 如果被判断为抓住失败，将工件数据以 ACK = 删除(4)返回队列，并返回程序开头，取得新的工件数据。
15 在判断为抓取成功时，在“抓手全部埋入的状态”也即负荷设定［2］中设定机器人的负荷设定。
16 使得机器人移动至逃避位置。
17 机器人个别拾取多个工件时，返回程序的开头并取得新的工件信息。只拾取一个工件时或者一次性拾取多个工件时，结束程序。

PK_CV_PICK11.TP

```
 1:  --
 2:  LBL[5]
 3:  F[R[102]]=(OFF)
 4:  UFRAME_NUM=0
 5:  UTOOL_NUM=1
 6:
 7:  LBL[10: Pick Loop]
 8:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
 9:  STOP_TRACKING
10:  --
11:  IF R[115:Pk1x UsePosChk]=0,JMP LBL[15]
12:  --
13:  --
14:  CALL PKCSGETQINRG(传送带工作站 ID=R[110:Pk1x StnI],连续取得的标记=R[107:Counter G],
     超时值(ms)=100,位置检查有效=R[115:Pk1x UsePos],基准位置 PR=57,接近时位置 PR=56,工具补偿=1,
     补偿用 VR=1,状态 Reg=123)
15:  JMP LBL[20]
16:
17:  LBL[15]
18:  --
19:  CALL PKCSGETQUE(传送带工作站 ID=R[110:Pk1x StnI],连续的标记=R[107:Counter G1],超时值(ms)=100,
     补偿用 VR=1,状态 Reg=123)
20:
21:  LBL[20]
```

```
  22:   //--
  23:
  24:
  25:   LBL[30]
  26:   --
  27:   CALL PKCSCALWPOS(传送带工作站 ID=R[110:Pk1x StnI],基准位置 PR=57,补偿用 VR=1,
        GetQ 状态=R[123:Pk1x GetQ stat],移动判断用 Reg=95,待机位置 PR=55,无需移动范围=150)
  28:   --
  29:   IF (R[95:TemporaryUse1]=0),CALL PK_CV_WAITPOS1(工具坐标编号=R[107:Counter G1],接近位置
PR=56,
        待机位置 PR=55,位置检查用 Reg=115)
  30:
  31:   LBL[40:ChkGetQStat]
  32:   IF ((R[199:Buf Id G1]>0) AND (R[123:Pk1x GetQ stat]>0)),JMP LBL[10000]
  33:   IF R[123:Pk1x GetQ stat]<>0,JMP LBL[10]
  34:
  35:
  36:   LBL[50]
  37:
  38:   LBL[60]
  39:   //--
  40:
  41:
  42:   LBL[70]
  43:   //--
  44:
  45:
  46:   LBL[80]
  47:
  48:   --
  49:   UTOOL_NUM=R[107:Counter G1]
  50:   IF R[115:Pk1x UsePosChk]=0,JMP LBL[85]
  51:   --
  52:   CALL PKCSCHKPOS(传送带工作站 ID=R[110:Pk1x StnI],补偿用 VR=1,基准位置 PR=57,接近位置 PR=56,
        工具补偿=1,状态 Reg=108)
  53:   IF R[108:PosNotReach G1]<>0,JMP LBL[1000]
  54:
  55:   LBL[85]
  56:   --
  57:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
  58:   --
  59:   IF (F[R[102]]),JMP LBL[1000]
  60:
  61:   --
  62:   CALL PK_GR_GET_ENDZONE1(拾取和放置计数=R[107:Counter G],批量作业=R[112:Pk1x PkMult],
        数量=R[111:Pk1x Qty],结束区域 Reg=106)
  63:   IF R[115:Pk1x UsePosChk]=0,JMP LBL[88]
  64:   --
  65:   CALL PKCSCHKPOS(传送带工作站 ID=R[110:Pk1x StnI],补偿用 VR=1,基准位置 PR=57,接近位置 PR=0,
        工具补偿=1,状态 Reg=108)
  66:   IF R[108:PosNotReach G1]<>0,JMP LBL[1000]
  67:
  68:   LBL[88]
  69:   --
  70:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,
       CALL PKGRCLOSE(抓手 ID=R[104:Gripper Id G1],开始区域=R[107:Counter G1],
       结束区域=R[106:Gr EndZone G1])
  71:   --
```

```
 72:   IF (F[R[102]]),JMP LBL[1000]
 73:   --
 74:   CALL PKGRGETPKDLY(抓手 ID=R[104:Gripper Id G1],待机时间  Reg=113)
 75:   WAIT R[113]
 76:   --
 77:   CALL PKCSGETPPCHK(传送带工作站 ID=R[110:Pk1x StnI],工件有无检查 Reg=116)
 78:   IF R[116:Pk1x UsePPChk]=0,JMP LBL[90]
 79:   --
 80:   --
 81:   --
 82:   CALL PKGRCHKPP(抓手 ID=R[104:Gripper Id G1],拾取=0,状态 Reg=117,初始区域=R[107:Counter G1],
       结束区域=R[106:Gr EndZone G1])
 83:   LBL[90]
 84:   PAYLOAD[2]
 85:   --
 86:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
 87:   IF R[117:Pk1x PPStat]<>0,JMP LBL[1500]
 88:   JMP LBL[2000]
 89:
 90:
 91:   LBL[1000: Missed part]
 92:   F[R[102]]=(OFF)
 93:   --
 94:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],发信)
 95:   IF R[199:Buf Id G1]>0,JMP LBL[10000]
 96:   JMP LBL[10]
 97:
 98:
 99:   LBL[1200: Return part]
100:   F[R[102]]=(OFF)
101:   --
102:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],返回)
103:   IF (R[199:Buf Id G1]>0),JMP LBL[10000]
104:   JMP LBL[10]
105:
106:
107:   LBL[1500: Dropped part]
108:   F[R[102]]=(OFF)
109:   --
110:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],删除)
111:   R[117:Pk1x PPStat]=0
112:   IF R[199:Buf Id G1]>0,JMP LBL[10000]
113:   JMP LBL[10]
114:
115:
116:   LBL[2000: Pick OK]
117:   --
118:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],成功)
119:   --
120:   IF R[112:Pk1x PkMult]=0,JMP LBL[2050]
121:   R[107:Counter G1]=R[107:Counter G1]+R[111:Pk1x Qty]
122:   JMP LBL[2100]
123:
124:   LBL[2050]
125:   R[107:Counter G1]=R[107:Counter G1]+1
126:   IF R[107:Counter G1]<=R[111:Pk1x Qty],JMP LBL[10]
127:
128:   LBL[2100]
```

```
  129:   R[205:Picked G1]=1
  130:   LBL[10000: End]
  131:   STOP_TRACKING
  132:   F[R[102]]=(OFF)
[End]
```

7.10.2.13 PK_CV_DROP11.TP

本程序中，组 1 的机器人在传送带工作站上，以下面的步骤执行最初的放置动作。

1 在开始放置动作之前，将［跳过动作实施范围］的标记设定为 OFF。
2 使用 KAREL 程序 PKCSGETQUE，从队列中取得工件信息。
3 不管是否已从队列取得工件，在与此拾取动作对应的传送带工作站操作的设定中［待机位置］有效时，检查是否需要在传送带工作站的待机位置也即上游侧的边界等待机器人，若判断为需要进行等待，则使用 PKCSCALWPOS.PC 计算待机位置。
4 因工件不存在、或者工件尚未到达上游侧边界而机器人需要等待时，使用 PK_CV_WAITPOS1.TP 移动至待机位置。
5 未能利用 KAREL 程序 PKCSGETQUE 取得工件信息时，再次尝试取得工件信息。
6 通过 KAREL 程序 PKCSGETQUE 取得工件数据时，进入 7 以后步骤。
7 将所选的工具坐标系设定为抓手的最初区域的工具坐标系。
8 使得机器人移动至接近位置。
9 如果在移动过程中，［跳过动作实施范围］的标记为 ON（由于超出了下游侧的边界，机器人无法拾取工件），则使用 KAREL 程序 PKCSACKQUE，将工件数据以 ACK = 发信(3)返回队列，并返回程序开头，取得新的工件数据。
10 使得机器人移动至放置位置。用先执行指令，与动作并行执行“打开抓手”动作。“打开抓手”动作通过 KAREL 程序 PKGROPEN 执行。
11 如果在移动过程中，［跳过动作实施范围］的标记为 ON（由于超出了下游侧的边界，机器人无法拾取工件），则使用 KAREL 程序 PKCSACKQUE，将工件数据以 ACK = 发信(3)返回队列，并返回程序开头，取得新的工件数据。
12 移动至放置位置后，使用 KAREL 程序 PKGRGETPKDLY，取得［放置延迟时间］，在该时间内与传送带一起移动。
13 通过该放置动作对应的传送带工作站的操作的设定，使［工件有无检查］有效时，使用 KAREL 程序 PKGRCHKPP，确认是否放开工件。
14 如果被判断为放开工件失败，将工件数据以 ACK = 4 返回队列，并返回程序开头，取得新的工件数据。
15 如果被判断为放开工件成功，将机器人的负载设定设定为“抓手为空的状态”，也就是负载设定［1］。
16 使得机器人移动至逃避位置。
17 机器人个别放置多个工件时，返回程序的开头，取得新的工件信息。只放置一个工件时或者一次性放置多个工件时，结束程序。

PK_CV_DROP11.TP

```
  1:  --
  2:  LBL[5]
  3:  F[R[102]]=(OFF)
  4:  UFRAME_NUM=0
  5:
  6:  LBL[10: Pick Loop]
  7:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
  8:  STOP_TRACKING
  9:  --
 10:  IF R[135:Dp1x UsePosChk]=0,JMP LBL[15]
 11:  --
 12:  --
 13:  CALL PKCSGETQINRG(传送带工作站 ID=R[130:Dp1x Stnl],连续取得的标记=R[107:Counter G],
      超时值(ms)=100,位置检查有效=R[135:Dp1x UsePos],基准位置 PR=59,接近时位置 PR=58,工具补偿=1,
      补偿用 VR=2,状态 Reg=143)
 14:  JMP LBL[20]
 15:
```

```
  16:   LBL[15]
  17:   --
  18:   CALL PKCSGETQUE(传送带工作站 ID=R[130:Dp1x StnI],连续的标记=R[107:Counter G1],超时值
(ms)=100,
        补偿用 VR=2,状态 Reg=143)
  19:
  20:   LBL[20]
  21:   //--
  22:
  23:
  24:   LBL[30]
  25:   --
  26:   CALL PKCSCALWPOS(传送带工作站 ID=R[130:Dp1x StnI],基准位置 PR=59,补偿用 VR=2,
        GetQ 状态=R[143:Dp1x GetQ stat],移动判断用 Reg=95,待机位置 PR=55,无需移动范围=150)
  27:   --
  28:   IF (R[95:TemporaryUse1]=0),CALL PK_CV_WAITPOS1(工具坐标编号=R[107:Counter G1],接近位置
PR=58,
        待机位置 PR=55,位置检查用 Reg=135)
  29:
  30:   LBL[40:ChkGetQStat]
  31:   IF ((R[199:Buf Id G1]>0) AND (R[143:Dp1x GetQ stat]>0)),JMP LBL[10000]
  32:   IF R[143:Dp1x GetQ stat]<>0,JMP LBL[10]
  33:
  34:
  35:   LBL[50]
  36:
  37:   LBL[60]
  38:   --
  39:
  40:
  41:   LBL[70]
  42:   --
  43:
  44:
  45:   LBL[80]
  46:
  47:   --
  48:   UTOOL_NUM=R[107:Counter G1]
  49:   IF R[135:Dp1x UsePosChk]=0,JMP LBL[85]
  50:   --
  51:   CALL PKCSCHKPOS(传送带工作站 ID=R[130:Dp1x StnI],补偿用 VR=2,基准位置 PR=59,接近位置 PR=58,
        工具补偿=1,状态 Reg=108)
  52:   IF R[108:PosNotReach G1]<>0,JMP LBL[1000]
  53:
  54:   LBL[85]
  55:   --
  56:L PR[59:Dp1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[58:Dp1 Cv Ap Ofst]
  57:   --
  58:   IF (F[R[102]]),JMP LBL[1000]
  59:
  60:   --
  61:   CALL PK_GR_GET_ENDZONE1(拾取和放置计数=R[107:Counter G],批量作业=R[132:Dp1x DpMult],
        数量=R[131:Dp1x Qty],结束区域 Reg=106)
  62:   IF R[135:Dp1x UsePosChk]=0,JMP LBL[88]
  63:   --
  64:   CALL PKCSCHKPOS(传送带工作站 ID=R[130:Dp1x StnI],补偿用 VR=2,基准位置 PR=59,接近位置 PR=0,
        工具补偿=1,状态 Reg=108)
  65:   IF R[108:PosNotReach G1]<>0,JMP LBL[1000]
```

```
 66:
 67:   LBL[88]
 68:   --
 69:L PR[59:Dp1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[2] TB R[145]sec,
       CALL PKGROPEN(抓手 ID=R[104:Gripper Id G1],开始区域=R[107:Counter G1],
       结束区域=R[106:Gr EndZone G1])
 70:   --
 71:   IF (F[R[102]]),JMP LBL[1000]
 72:   --
 73:   CALL PKGRGETDPDLY(抓手 ID=R[104:Gripper Id G1],待机时间 Reg=133)
 74:   WAIT R[133]
 75:   --
 76:   CALL PKCSGETPPCHK(传送带工作站 ID=R[130:Dp1x StnI],工件有无检查 Reg=136)
 77:   IF R[136:Dp1x UsePPChk]=0,JMP LBL[90]
 78:   --
 79:   --
 80:   --
 81:   CALL PKGRCHKPP(抓手 ID=R[104:Gripper Id G1],放置=1,状态 Reg=137,初始区域=R[107:Counter G1],
       结束区域=R[106:Gr EndZone G1])
 82:   LBL[90]
 83:   PAYLOAD[1]
 84:   --
 85:L PR[59:Dp1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[58:Dp1 Cv Ap Ofst]
 86:   IF R[137:Dp1x PPStat]<>0,JMP LBL[1500]
 87:   JMP LBL[2000]
 88:
 89:
 90:   LBL[1000: Missed part]
 91:   F[R[102]]=(OFF)
 92:   --
 93:   CALL PKCSACKQUE(传送带工作站 ID=R[130:Dp1x StnI],发信)
 94:   IF R[199:Buf Id G1]>0,JMP LBL[10000]
 95:   JMP LBL[10]
 96:
 97:   LBL[1200: Return part]
 98:   F[R[102]]=(OFF)
 99:   --
100:   CALL PKCSACKQUE(传送带工作站 ID=R[130:Dp1x StnI],返回)
101:   IF R[199:Buf Id G1]>0,JMP LBL[10000]
102:   JMP LBL[10]
103:
104:
105:   LBL[1500: Dropped part]
106:   F[R[102]]=(OFF)
107:   --
108:   CALL PKCSACKQUE(传送带工作站 ID=R[130:Dp1x StnI],删除)
109:   R[137:Dp1x PPStat]=0
110:   IF R[199:Buf Id G1]>0,JMP LBL[10000]
111:   JMP LBL[10]
112:
113:
114:   LBL[2000: Drop OK]
115:
116:   --
117:   CALL PKCSACKQUE(传送带工作站 ID=R[130:Dp1x StnI],成功)
118:   --
119:   IF R[132:Dp1x DpMult]=0,JMP LBL[2050]
120:   R[107:Counter G1]=R[107:Counter G1]+R[131:Dp1x Qty]
```

```
  121:   JMP LBL[2100]
  122:
  123:   LBL[2050]
  124:   R[107:Counter G1]=R[107:Counter G1]+1
  125:   IF R[107:Counter G1]<=R[131:Dp1x Qty],JMP LBL[10]
  126:
  127:   LBL[2100]
  128:   R[205:Picked G1]=0
  129:   LBL[10000: End]
  130:   STOP_TRACKING
  131:   F[R[102]]=(OFF)
[End]
```

7.10.2.14 PK_CV_WAITPOS1.TP

该程序是用于在没有工件时，组 1 的机器人能移动至传送带工作站的上游侧边界的待机位置，进行高效作业的程序。

调出该程序时，需要以下的 4 个参数。
参数 1：工具坐标系编号（通常对应此时使用的区域的编号）
参数 2：取得以传送带工作站的操作设定的［接近时偏移量］的位置寄存器序号
参数 3：存储有计算所得的待机位置的位置寄存器序号
参数 4：存储 0 的寄存器编号

该程序用于使用不同的参数，将机器人“放置”在传送带工件站或从传送带工件站的待机位置“拾取”。

PK_CV_WAITPOS1.TP

```
    1:   --
    2:   --
    3:   --
    4:   --
    5:   --
    6:   --
    7:   --
    8:
    9:   UTOOL_NUM=AR[1]
   10:   IF R[AR[4]]=0,JMP LBL[85]
   11:   PR[51:TemporaryUse1]=PR[AR[2]]
   12:   PR[51,3:TemporaryUse1]=PR[51,3:TemporaryUse1]*(-1)
   13:   --
   14:   CALL PKOPPOSCHK(位置 PR=AR[3],组编号=1,用户坐标编号=0,工具坐标编号=AR[1],状态 Reg=108,
         位置补偿 PR=51)
   15:   IF R[108:PosNotReach G1]<>0,JMP LBL[100]
   16:   LBL[85]
   17:   --
   18:L PR[AR[3]] max_speed CNT100 Tool_Offset,PR[AR[2]]
   19:   LBL[100]
 [End]
```

7.10.2.15 PK_FS_PICK11.TP

本程序中，组 1 的机器人在固定工作站上，以下面的步骤执行最初的拾取动作。

1 使用 KAREL 程序 PKFSGETUF，取得在树状图［固定工作站］的设定画面中设定的固定工作站的用户坐标系编号，设定于系统的用户坐标系编号。
2 用 TP 程序 PK_FS_READY，确认在固定工作站的拾取动作是否可动作。
定工作站不可动作的情况，反复执行 PK_FS_READY，直至可动作。“可动作”是指在固定工作站操作中设定的［动作开始信号］接通、［动作完成信号］断开、固定工作站处于非调换动作中这 3 个条件成立的状态。通过检查“可动作”，就可安全地在固定工作站上进行拾取动作。
4 使用 KAREL 程序 PKFSGETQUE，从队列取得工件数据。
无法从 PKFSGETQUE 取得工件数据的情况，表示这种情况是托盘内的所有工件已被拾取，托盘处于空状态。为了更换放有新工件的托盘，请使用 TP 程序 PK_FS_INDEX，执行固定工作站的更换作业。固定工作站的更换作业结束后，使用 KAREL 程序 PKFSPUTQUE，对固定工作站的托盘进行初始化。
6 PKFSGETQUE 取得工件数据时，进入 7 以后步骤。
7 将所选的工具坐标系设定为抓手的最初区域的工具坐标系。
8 使得机器人移动至接近位置。
9 机器人移动至接近位置后，使机器人移动至拾取位置。使用先执行指令，与动作并行执行“关闭抓手”动作。“关闭抓手”动作通过 KAREL 程序 PKGRCLOSE 执行。
10 机器人移动至拾取位置后，使用 KAREL 程序 PKGRGETPKDLY，取得［拾取延迟时间］，在该时间内待机。
11 通过该拾取动作对应的固定工作站的操作的设定，使［工件有无检查］有效时，使用 KAREL 程序 PKGRCHKPP，确认是否抓住工件。
12 如果被判断为抓住失败，将工件数据以 ACK = 返回(2)返回队列，并返回程序开头，取得新的工件数据。
13 在判断为抓取成功时，在“抓手全部埋入的状态”也即负荷设定［2］中设定机器人的负荷设定。
14 使得机器人移动至逃避位置。
15 机器人个别拾取多个工件时，返回程序的开头并取得新的工件信息。只拾取一个工件时或者一次性拾取多个工件时，结束程序。

PK_FS_PICK11.TP

```
 1:  --
 2:
 3:  R[107:Counter G1]=1
 4:
 5:  LBL[5]
 6:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
 7:
 8:  --
 9:  --
10:  CALL PKFSGETUF(固定工作站 ID=R[110:Pk1x StnId],用户坐标系  Reg=95)
11:  UFRAME_NUM=R[95:TemporaryUse1]
12:
13:  LBL[10]
14:  --
15:  --
16:  CALL PK_FS_READY(固定工作站 ID=R[110:Pk1x StnId],状态 Reg=95)
17:  IF R[95:TemporaryUse1]=0,JMP LBL[20]
18:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
19:  WAIT      .02(sec)
20:  JMP LBL[10]
21:
22:  LBL[20: Pick Loop]
23:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
24:
25:  --
26:  --
27:  CALL PKFSGETQUE(固定工作站 ID=R[110:Pk1x StnId],补偿用 VR=1,状态 Reg=123)
28:  IF R[123:Pk1x GetQ stat]=0,JMP LBL[200]
29:  !Index tray
```

```
30:   CALL PK_FS_INDEX(固定工作站 ID=R[110:Pk1x StnId],单元计数 Reg=122,状态 Reg=95)
31:   JMP LBL[5]
32:
33:   LBL[200]
34:
35:
36:   LBL[300]
37:   UTOOL_NUM=R[107:Counter G1]
38:   --
39:   CALL PK_GR_GET_ENDZONE1(拾取和放置计数=R[107:Counter G],批量作业=R[112:Pk1x PkMult],
      数量=R[111:Pk1x Qty],结束区域 Reg=106)
40:   --
41:L PR[61:Pk1 FS Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[60:Pk1 FS Ap Ofst]
42:   --
43:L PR[61:Pk1 FS Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,
      CALL PKGRCLOSE(抓手 ID=R[104:Gripper Id G1],开始区域=R[107:Counter G1],
      结束区域=R[106:Gr EndZone G1])
44:   --
45:   CALL PKGRGETPKDLY(抓手 ID=R[104:Gripper Id G1],待机时间  Reg=113)
46:   WAIT R[113]
47:   --
48:   CALL PKFSGETPPCHK(固定工作站 ID=R[110:Pk1x StnId],工件有无检查 Reg=116)
49:   IF R[116:Pk1x UsePPChk]=0,JMP LBL[400]
50:   --
51:   --
52:   --
53:   CALL PKGRCHKPP(抓手 ID=R[104:Gripper Id G1],拾取=0,状态 Reg=117,初始区域=R[107:Counter G1],
      结束区域=R[106:Gr EndZone G1])
54:   LBL[400]
55:   --
56:   PAYLOAD[2]
57:   --
58:L PR[61:Pk1 FS Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[60:Pk1 FS Ap Ofst]
59:   IF R[117:Pk1x PPStat]<>0,JMP LBL[1500]
60:   JMP LBL[2000]
61:
62:
63:   LBL[1500: PartGotdropped]
64:   --
65:   CALL PKFSACKQUE(固定工作站 ID=R[110:Pk1x StnId],返回)
66:   R[117:Pk1x PPStat]=0
67:   JMP LBL[20]
68:
69:
70:   LBL[2000: Pick ok]
71:   --
72:   CALL PKFSACKQUE(固定工作站 ID=R[110:Pk1x StnId],成功)
73:   --
74:   R[122:Pk1x TrayCtr]=R[122:Pk1x TrayCtr]+1
75:   --
76:   IF R[112:Pk1x PkMult]=0,JMP LBL[2050]
77:   R[107:Counter G1]=R[107:Counter G1]+R[111:Pk1x Qty]
78:   JMP LBL[3000]
79:
80:   LBL[2050]
81:   R[107:Counter G1]=R[107:Counter G1]+1
82:   IF (R[107:Counter G1]<=R[111:Pk1x Qty]),JMP LBL[20]
83:
```

```
 84:   LBL[3000: CheckIndex]
 85:   !Check if tray ready to be indexd
 86:   CALL PKFSGETQUE(固定工作站 ID=R[110:Pk1x StnId],补偿用 VR=1,状态 Reg=123)
 87:   IF R[123:Pk1x GetQ stat]=0,JMP LBL[3500]
 88:   !Index tray
 89:   CALL PK_FS_INDEX(固定工作站 ID=R[110:Pk1x StnId],单元计数 Reg=122,状态 Reg=95)
 90:   JMP LBL[10000]
 91:   LBL[3500]
 92:   --
 93:   CALL PKFSACKQUE(固定工作站 ID=R[110:Pk1x StnId],返回)
 94:
 95:
 96:   LBL[10000: End]
[End]
```

7.10.2.16 PK_FS_DROP11.TP

本程序中，组 1 的机器人在固定工作站上，以下面的步骤执行最初的放置动作。

1 使用 KAREL 程序 PKFSGETUF，取得在树状图［固定工作站］的设定画面中设定的固定工作站的用户坐标系编号，设定于系统的用户坐标系编号。
2 使用 TP 程序 PK_FS_READY，确认在固定工作站的放置动作是否可动作。
3 定工作站不可动作的情况，反复执行 PK_FS_READY，直至可动作。“可动作”是指在固定工作站操作中设定的［动作开始信号］接通、［动作完成信号］断开、固定工作站处于非调换动作中这 3 个条件成立的状态。通过检查“可动作”，就可安全地在固定工作站上进行放置动作。
4 使用 KAREL 程序 PKFSGETQUE，从队列取得工件数据。
5 无法从 PKFSGETQUE 取得工件数据的情况，表示这种情况是托盘内的所有单元都放有工件，托盘处于已满状态。为了更换新的空托盘，请使用 TP 程序 PK_FS_INDEX，执行固定工作站的更换作业。固定工作站的更换作业结束后，使用 KAREL 程序 PKFSPUTQUE，对固定工作站的托盘进行初始化。
6 PKFSGETQUE 取得工件数据时，进入 7 以后步骤。
7 将所选的工具坐标系设定为抓手的最初区域的工具坐标系。
8 使得机器人移动至接近位置。
9 机器人移动至接近位置后，使机器人移动至放置位置。用先执行指令，与动作并行执行“打开抓手”动作。“打开抓手”动作通过 KAREL 程序 PKGROPEN 执行。
10 移动至放置位置后，使用 KAREL 程序 PKGRGETPKDLY，取得［放置延迟时间］，在该时间内与传送带一起移动。
11 通过该放置动作对应的固定工作站的操作的设定，使［工件有无检查］有效时，使用 KAREL 程序 PKGRCHKPP，确认是否放开工件。
12 如果被判断为放开工件失败，将工件数据以 ACK = 返回(2)返回队列，并返回程序开头，取得新的工件数据。
13 如果被判断为放开工件成功，将机器人的负载设定设定为“抓手为空的状态”，也就是负载设定［1］。
14 使得机器人移动至逃避位置。
15 机器人个别放置多个工件时，返回程序的开头，取得新的工件信息。只放置一个工件时或者一次性放置多个工件时，结束程序。

PK_FS_DROP11.TP

```
  1:   --
  2:
  3:   R[107:Counter G1]=1
  4:
  5:   LBL[5]
  6:   IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
  7:
  8:   --
  9:   --
 10:   CALL PKFSGETUF(固定工作站 ID=R[130:Dp1x StnId],用户坐标系  Reg=95)
 11:   UFRAME_NUM=R[95:TemporaryUse1]
 12:
 13:   LBL[10]
```

```
14:  --
15:  --
16:  CALL PK_FS_READY(固定工作站 ID=R[130:Dp1x StnId],状态 Reg=95)
17:  IF R[95:TemporaryUse1]=0,JMP LBL[20]
18:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
19:  WAIT      .05(sec)
20:  JMP LBL[10]
21:
22:  LBL[20: Drop Loop]
23:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
24:
25:  --
26:  --
27:  CALL PKFSGETQUE(固定工作站 ID=R[130:Dp1x StnId],补偿用 VR=2,状态 Reg=143)
28:  IF R[143:Dp1x GetQ stat]=0,JMP LBL[200]
29:  !Index tray
30:  CALL PK_FS_INDEX(固定工作站 ID=R[130:Dp1x StnId],单元计数 Reg=142,状态 Reg=95)
31:  JMP LBL[5]
32:
33:  LBL[200]
34:
35:
36:  LBL[300]
37:  UTOOL_NUM=R[107:Counter G1]
38:  --
39:  CALL PK_GR_GET_ENDZONE1(拾取和放置计数=R[107:Counter G],批量作业=R[132:Dp1x DpMult],
       数量=R[131:Dp1x Qty],结束区域 Reg=106)
40:  --
41:L PR[63:Dp1 FS Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[62:Dp1 FS Ap Ofst]
42:  --
43:L PR[63:Dp1 FS Ref Pos] max_speed CNT0 VOFFSET,VR[2] TB R[145]sec,
       CALL PKGROPEN(抓手 ID=R[104:Gripper Id G1],开始区域=R[107:Counter G1],
       结束区域=R[106:Gr EndZone G1])
44:  --
45:  CALL PKGRGETDPDLY(抓手 ID=R[104:Gripper Id G1],待机时间 Reg=133)
46:  WAIT R[133]
47:  --
48:  CALL PKFSGETPPCHK(固定工作站 ID=R[130:Dp1x StnId],工件有无检查 Reg=136)
49:  IF R[136:Dp1x UsePPChk]=0,JMP LBL[400]
50:  --
51:  --
52:  --
53:  CALL PKGRCHKPP(抓手 ID=R[104:Gripper Id G1],放置=1,状态 Reg=137,初始区域=R[107:Counter G1],
       结束区域=R[106:Gr EndZone G1])
54:  LBL[400]
55:  --
56:  PAYLOAD[1]
57:  --
58:L PR[63:Dp1 FS Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[62:Dp1 FS Ap Ofst]
59:  IF R[137:Dp1x PPStat]<>0,JMP LBL[1500]
60:  JMP LBL[2000]
61:
62:
63:  LBL[1500: PartGotdropped]
64:  --
65:  CALL PKFSACKQUE(固定工作站 ID=R[130:Dp1x StnId],返回)
66:  R[137:Dp1x PPStat]=0
67:  JMP LBL[20]
```

```
 68:
 69:
 70:   LBL[2000: Pick ok]
 71:   --
 72:   CALL PKFSACKQUE(固定工作站 ID=R[130:Dp1x StnId],成功)
 73:   --
 74:   R[142:Dp1x TrayCtr]=R[142:Dp1x TrayCtr]+1
 75:   --
 76:   IF R[132:Dp1x DpMult]=0,JMP LBL[2050]
 77:   R[107:Counter G1]=R[107:Counter G1]+R[131:Dp1x Qty]
 78:   JMP LBL[3000]
 79:
 80:   LBL[2050]
 81:   R[107:Counter G1]=R[107:Counter G1]+1
 82:   IF (R[107:Counter G1]<=R[131:Dp1x Qty]),JMP LBL[20]
 83:
 84:   LBL[3000: CheckIndex]
 85:   !Check if tray ready to be indexd
 86:   CALL PKFSGETQUE(固定工作站 ID=R[130:Dp1x StnId],补偿用 VR=2,状态 Reg=143)
 87:   IF R[143:Dp1x GetQ stat]=0,JMP LBL[3500]
 88:   !Index tray
 89:   CALL PK_FS_INDEX(固定工作站 ID=R[130:Dp1x StnId],单元计数 Reg=142,状态 Reg=95)
 90:   JMP LBL[10000]
 91:   LBL[3500]
 92:   --
 93:   CALL PKFSACKQUE(固定工作站 ID=R[130:Dp1x StnId],返回)
 94:
 95:
 96:   LBL[10000: End]
[End]
```

7.10.2.17 PK_FS_INDEX.TP

本程序是执行固定工作站的更换的程序。

调出该程序时，需要以下的 3 个参数。
参数 1：固定工作站的 ID。
参数 2：表示固定工作站的托盘的单元编号的计数。
参数 3：存储状态的数值寄存器序号。存储于该数值寄存器的状态的值是 0 以外的值时，固定工作站的更换失败。

该程序以下面流程运行。
1　使在固定工作站的操作中设定的［动作完成信号］的 DO 为 ON。
2　使用 KAREL 程序 PKFSMONIDX，监视动作开始信号暂时 OFF 后，再次 ON。
3　为 ON 后，结束程序。如果在 ON 之前，PKFSMONIDX 发生错误，则将作为状态，表示固定工作站的更换失败的 1 存储于参数 3 的数值寄存器，并结束。

PK_FS_INDEX.TP

```
  1:   --
  2:   --
  3:   --
  4:   --
  5:   --
  6:
  7:   R[AR[3]]=0
  8:   R[96:TemporaryUse2]=0
  9:
 10:   --
 11:   CALL PKFSGETIXODO(固定工作站 ID=AR[1],动作完成信号 Reg=96)
```

```
 12:
 13:  --
 14:  IF R[96:TemporaryUse2]=0,JMP LBL[1500]
 15:
 16:
 17:  LBL[1000: Do_Idx]
 18:  --
 19:  DO[R[96]]=ON
 20:
 21:
 22:  --
 23:  --
 24:  --
 25:  --
 26:  CALL PKFSMONIDX(固定工作站 ID=AR[1],状态 Reg=AR[3])
 27:  LBL[1500]
 28:  --
 29:  CALL PKFSPUTQUE(固定工作站 ID=AR[1])
 30:  ! Reset tray counter
 31:  R[AR[2]]=1
 32:  JMP LBL[10000]
 33:
 34:  LBL[2000: Not ok]
 35:  R[AR[3]]=1
 36:
 37:  LBL[10000: End]
[End]
```


7.10.2.18 PK_FS_READY.TP

该程序是确认固定工作站是否能由机器人进行拾取/放置作业的程序。

在调用此程序时，需要如下 2 个参数。
参数 1：固定工作站的 ID。
参数 2：存储可作业状态的数值寄存器序号。储于该数值寄存器的状态的值是 0 以外的值时，固定工作站不可作业。
“不可作业”是指在固定工作站操作中设定的［动作开始信号］断开、［动作完成信号］接通、固定工作站处于调换动作中这 3 个条件中的至少一个以上成立的状态。

该程序以下面流程运行。
1 确认在固定工作站的操作中设定的［动作完成信号］的 DO 的状态。
2 如果 DO 为 ON，则将作为状态的“不可作业”存储于参数 2 的数值寄存器，并结束。
3 然后，确认是否在固定工作站的更换作业中。在更换作业中，特别是因某原因二未完成更换作业的情况，将作为状态的“不可作业”存储于参数 2 的数值寄存器，并结束。
4 然后，确认在固定工作站的操作中设定的［动作开始信号］的 DI 的状态。如果 DI 不是 ON，则将作为状态的“不可作业”存储于参数 2 的数值寄存器，并结束。
5 如果满足了全部的作业开始条件，则将表示“可作业”的 0 存储于参数 2 的数值寄存器，并结束。

PK_FS_READY.TP

```
  1:  --
  2:  --
  3:  --
  4:  --
  5:  --
  6:
  7:  R[96:TemporaryUse2]=0
  8:  R[AR[2]]=0
  9:
 10:  --
```

```
  11:   CALL PKFSGETIXODO(固定工作站 ID=AR[1],动作完成信号 Reg=96)
  12:
  13:   --
  14:   IF R[96:TemporaryUse2]<=0,JMP LBL[100]
  15:   IF DO[R[96]]=ON,JMP LBL[1000]
  16:
  17:   LBL[100]
  18:   --
  19:   --
  20:   --
  21:   CALL PKFSGETRDY(固定工作站 ID=AR[1],工作站就绪 Reg=97)
  22:   IF R[97:TemporaryUse3]<>1,JMP LBL[1000]
  23:
  24:   --
  25:   CALL PKFSGETRDYDI(固定工作站 ID=AR[1],动作开始 DI Reg=96)
  26:
  27:   --
  28:   IF R[96:TemporaryUse2]<=0,JMP LBL[2000]
  29:   IF DI[R[96]]=ON,JMP LBL[2000]
  30:   JMP LBL[1000]
  31:
  32:   LBL[1000: NotReady]
  33:   R[AR[2]]=1
  34:   JMP LBL[10000]
  35:
  36:   LBL[2000: Ready]
  37:   R[AR[2]]=0
  38:   JMP LBL[10000]
  39:
  40:   LBL[10000: End]
[End]
```

7.10.2.19 PK_GR_GET_ENDZONE1.TP

本程序根据最初的区域，［多数拾取和放置］的有效/无效的设定，计算组 1 的机器人使用的抓手的最后区域。在多个工件的拾取/放置动作中有效。

调出该程序时，需要以下的 4 个参数。
参数 1：最初的区域的编号。
参数 2：在传送带工作站的操作或固定工作站的操作中，设定的［多数拾取和放置］的有效/无效的设定。使用 KAREL 程序 PKCSGETPMUL 或 PKFSGETPMUL 进行取得。
参数 3：追加至抓手的区域数量。
参数 4：存储最后区域的编号的数值寄存器序号。

PK_GR_GET_ENDZONE1.TP

```
   1:   --
   2:   --
   3:   --
   4:   --
   5:   --
   6:   --
   7:   --
   8:   --
   9:
  10:   IF (AR[2]=0),JMP LBL[100]
  11:   R[AR[4]]=AR[1]+AR[3]-1
  12:   JMP LBL[1000]
  13:
```

```
  14:   LBL[100]
  15:   R[AR[4]]=AR[1]
  16:   LBL[1000]
  17:
[End]
```

7.10.2.20 PK_CYCSTOP1.TP

该程序记述有周期停止时的处理。标准 TP 程序中，作为假定形式，仅有暂时停止指令。请根据各个应用程序，插入必要的指令。

PK_CYCSTOP1.TP

```
  1:   --
  2:   --
  3:   PAUSE
 [End]
```

7.11 预整列设定

本节就 Plug & Play 中抓手和操作，对同一传送带上的整列功能（预整列功能）的设定步骤进行说明。
其他对象的预整列功能设定与基本功能的设定相同。请参照“6.14 预整列功能的设定”。

北美专用设定时，需要北美专用选项 R856: *i*RPickTool / Pre-grouping，以使用预整列功能。

7.11.1 抓手和区域的设定

有关在同一传送带上进行整列动作的机器人，没有特别记载事项。请参照“7.2 抓手和区域的设定”。

有关汇总拾取已被整列的工件的机器人，需要注意区域的设定。不是追加数量与工件数量对应的区域，而对于一次性汇总取出的工件设定 1 个区域。
譬如，图 6.14（c）所示的情况下，相对于 3 个工件设定 1 个区域。

7.11.2 操作的设定

有关在同一传送带上进行整列动作的机器人，没有特别记载事项。请参照“7.3 操作设定”。

有关汇总拾取已被整列的工件的机器人，需要注意传送带工作站操作、固定工作站操作的设定。

在拾取动作的传送带工作站操作，虽然一次性汇总拾取整列的工件，将［多数拾取和放置］置于无效。

在放置动作的传送带工作站操作、或者固定工作站操作中，［多数拾取和放置］的设定如下。
一次性放置全部工件时，将［多数拾取和放置］置于有效。
拾取动作执行 2 次以上再执行放置动作时，放置动作也以汇总取出的工件单位执行时，将［多数拾取和放置］置于无效。

7.11.3 追踪动作的微调整

请参照“6.12 追踪动作的微调整”。作为预整列时的特别记载事项，有关追踪坐标系的设定误差的调整方法，需要注意。

7.11.3.1 追踪坐标系的设定误差的调整（使用 ADJ_OFS 时）

使用 ADJ_OFS 时，按如下所示方式修正标准 TP 程序“7.11.4.3 PK_PRGR_CV_PICKDROP1.TP”。而这里介绍的程序例，将位置寄存器的 20 号作为存储调整量的地方使用。请根据需要适当变更。

```
 40:   LBL[80: PickPart]
 41:   CALL ADJ_OFS(1, 1, 20, 1)
 42:   ! Move to approach
 43:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
```

```
 64:   LBL[200: PlacePart]
 65:   CALL ADJ_OFS(1, 2, 20, 2)
 66:   ! Move to approach
 67:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[58:Dp1 Cv Ap Ofst]
```

- 在即将进行拾取/放置动作之前，调出 ADJ_OFS 调整补偿数据。
- 在拾取/放置动作中使用已被存储在相同位置寄存器中的调整量。
- 在设定调整量时，机器人会停止在追踪程序内的取出位置正上方，所以在紧靠接近点之后插入动作命令和待机命令。

```
 45:   ! Move to pick pos
 46:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] Tool_Offset,PR[15]
 47:   WAIT   10.00(sec)
 48:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,CALL PKGRCLOSE(抓手
ID=R[104:Gripper Id G1],开始区域=1,结束区域=1)
```

- 详细信息，请参照“7.8.1 追踪坐标系的设定误差的调整”。

7.11.4 标准 TP 程序

这里就 Plug & Play 中所提供的预整列的标准 TP 程序进行说明。

根据按以下所示方式定义的预整列的类型而动作。

1: 在同一传送带上进行整列的最上游的机器人。生成虚拟托盘。（机器人 1）
2: 在同一传送带上进行整列的非最上游的机器人。（机器人 2）
3: 系对在同一传送带上已被整列的工件进行汇总搬运的机器人。（机器人 3）

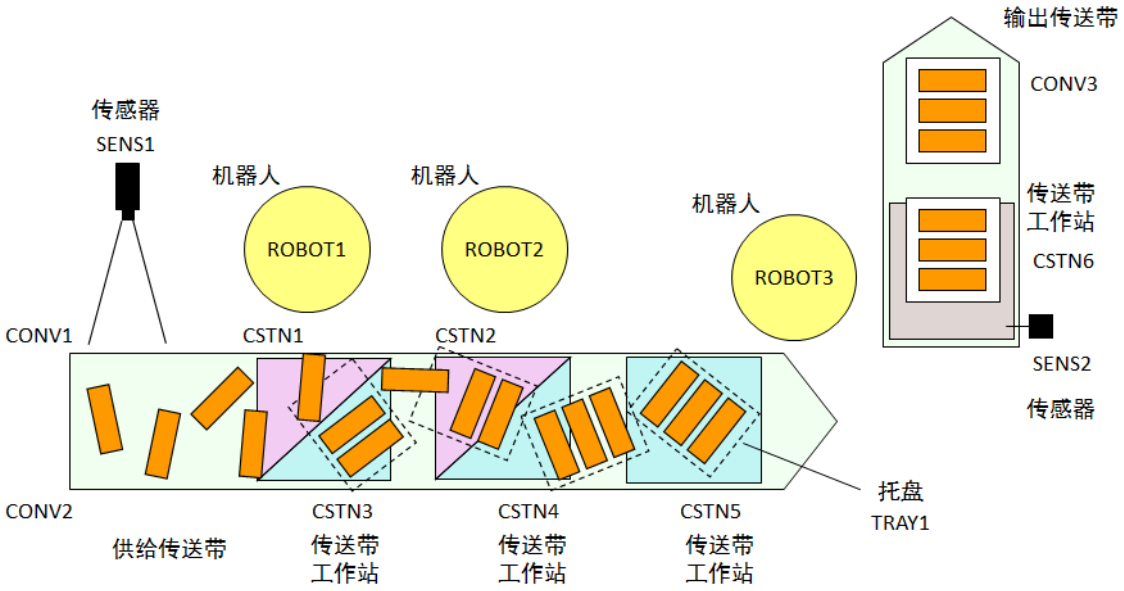


图 7.11.4(a) 系统构成

7.11.4.1 PK_PRGR_MAIN1.TP

本程序为组 1 的预整列的主程序。

此程序在预整列类型为 1 时，以无限循环的方式执行虚拟托盘的生成和 PK_PRGR_CV_PICKDROP1.TP。在生成虚拟托盘之前，检查是否有周期停止的要求。因干涉而未能生成虚拟托盘时，在利用 KAREL 程序 PKCSGETQUE 取得工件信息后，使用 KAREL 程序 PKCSACKQUE，将工件信息以 ACK = 发信(3)返回队列，在回到程序的开头后，尝试生成后续的虚拟托盘。因除此以外的理由而未能生成虚拟托盘时，在等待 0.02 秒后，再次尝试生成虚拟托盘。

此程序在预整列的类型为 2 时，以无限循环的方式执行 PK_PRGR_CV_PICKDROP1.TP。在 PK_PRGR_CV_PICKDROP1.TP 之前，检查是否有周期停止的要求。

PK_PRGR_MAIN1.TP

```
1:   ! Main Prog for Grp:1
  2:   ! for Pre-grouping
  3:
  4:   IF R[230:PRGR Robot Type]<=0 OR R[230:PRGR Robot Type]>2,JMP LBL[10000]
  5:
  6:   LBL[100:pk n pl]
  7:   IF R[101:Cycle Stop G1]>0,JMP LBL[10000]
  8:   ! Generate vtray only in the
  9:   ! sensor controller.
 10:   IF R[230:PRGR Robot Type]<>1,JMP LBL[200]
 11:   ! Generate Virtual tray
 12:   ! PKCSGENVRTRY(cstn_id, stat_reg)
 13:   CALL PKCSGENVRTRY(传送带工作站 ID=R[130:Dp1x StnI],状态 Reg=95)
 14:   IF R[95:TemporaryUse1]=0,JMP LBL[200]
 15:   IF R[95:TemporaryUse1]=1,JMP LBL[300]
 16:   IF R[95:TemporaryUse1]=2,JMP LBL[400]
 17:   WAIT       .02(sec)
 18:   JMP LBL[100]
 19:
 20:   LBL[200: Pre-group]
 21:   ! Do pre-grouping
 22:   CALL PK_PRGR_CV_PICKDROP1
 23:   IF R[230:PRGR Robot Type]<>1,JMP LBL[200]
 24:   JMP LBL[100]
 25:
 26:   LBL[300: Stat=1]
 27:   ! No virtual tray is generated.
 28:   ! Interference cannot be avoided
 29:
 30:   ! Strategy 1 (Default: Discard
 31:   ! the most downstream part and
 32:   ! re-generate tray again
 33:   CALL PKCSGETQUE(传送带工作站 ID=R[110:Pk1x StnI],连续的标记=1,超时值(ms)=0,补偿用 VR=1,状态
       Reg=123)
 34:   IF R[123:Pk1x GetQ stat]<>0,JMP LBL[375]
 35:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],发信=3)
 36:   JMP LBL[100]
 37:
 38:   ! Strategy 2: If it is feasible
 39:   !    to build the tray on a clear
 40:   !    area of the conveyor,
 41:   !    contact FANUC.
 42:   !CALL PK_PRGR_CV_SPECIAL_PICKDROP
 43:
 44:   JMP LBL[375]
 45:
 46:   ! Strategy 3: If a separate
 47:   ! Fixed station or indexing
 48:   ! conveyor is available, then
 49:   ! insert calls for the
 50:   ! pick and place programs
 51:   JMP LBL[375]
```

```
   52:
   53:   LBL[375]
   54:   WAIT       .02(sec)
   55:   JMP LBL[100]
   56:
   57:   LBL[400:Stat=2]
   58:   ! No virtual tray is generated.
   59:   ! Grouping cannot be achieved in
   60:   ! distance from the leading part
   61:   ! Insert program to handle this.
   62:   ! ex: Pick a part and place it
   63:   ! on a Fixed Stn or
   64:   ! Indexing conveyor
   65:   WAIT       .02(sec)
   66:   JMP LBL[100]
   67:
   68:   LBL[10000: End]
[End]
```

7.11.4.2 PK_PRGR_INIT1.TP

本程序为组 1 的预整列的初始化程序。

此程序取得预整列的类型。

PK_PRGR_INIT1.TP

```
    1:   ! PK_PRGR_INIT1.TP
    2:   !Initializations for pre-grouping
    3:
    4:   !AR[1]: Register num for PRGR tye
    5:
    6:   !Get PRGR type
    7:   CALL PKRBGETPRGRT(拾取传送带 ID=R[109:Pk1x CvId],拾取工作站 ID=R[110:Pk1x StnId],放置传送带
         ID=R[129:Dp1x CvId],放置工作站 ID=R[130:Dp1x StnId],预整列类型 Reg=AR[1])
    8:   LBL[10000:End]
[End]
```

7.11.4.3 PK_PRGR_CV_PICKDROP1.TP

此程序中，按以下步骤执行组 1 的机器人在同一传送带上的整列动作。

1 使用 KAREL 程序 PKCSGETQPRGR，从队列中取得工件信息。
2 不管是否已从队列取得工件，在与此拾取动作对应的传送带工作站操作的设定中［待机位置］有效时，检查是否需要在传送带工作站的待机位置也即上游侧的边界等待机器人，若判断为需要进行等待，则使用 PKCSCALWPOS.PC 计算待机位置。
3 因工件不存在、或者工件尚未到达上游侧边界而机器人需要等待时，使用 PK_CV_WAITPOS1.TP 移动至待机位置。
4 未能利用 KAREL 程序 PKCSGETQPRGR 取得工件信息时，结束程序。此时预整列的类型若不是 1，就会在等待 0.02 秒后结束程序。
5 能够利用 KAREL 程序 PKCSGETQPRGR 取得工件信息时，进入到 6 以后的步骤。
6 使用 KAREL 程序 PKCSTRGPRGR，设定拾取动作的追踪的触发。
7 未能利用 KAREL 程序 PKCSTRGPRGR 设定追踪的触发时，再次尝试触发的设定。
8 能够利用 KAREL 程序 PKCSTRGPRGR 设定追踪的触发时，进入到 9 以后的步骤。
9 使得机器人移动至接近位置。
10 使得机器人移动至拾取位置。使用先执行指令，与动作并行执行“关闭抓手”动作。“关闭抓手”动作通过 KAREL 程序 PKGRCLOSE 执行。
11 移动至拾取位置后，使用 KAREL 程序 PKGRGETPKDLY 取得［拾取延迟时间］，在该时间内与传送带一起移动。
12 在“抓手全部埋入的状态”也即负荷设定［2］中设定机器人的负荷设定。
13 使得机器人移动至逃避位置。
14 使用 KAREL 程序 PKCSTRGPRGR，设定放置动作的追踪的触发。

15 未能利用 KAREL 程序 PKCSTRGPRGR 设定追踪的触发时，再次尝试触发的设定。
16 能够利用 KAREL 程序 PKCSTRGPRGR 设定追踪的触发时，进入到 17 以后的步骤。
17 使得机器人移动至接近位置。
18 使得机器人移动至放置位置。用先执行指令，与动作并行执行“打开抓手”动作。“打开抓手”动作通过 KAREL 程序 PKGROPEN 执行。
19 移动至放置位置后，使用 KAREL 程序 PKGRGETPKDLY，取得［放置延迟时间］，在该时间内与传送带一起移动。
20 在“抓手为空的状态”也即负荷设定［1］中设定机器人的负荷设定。
21 使得机器人移动至逃避位置。
22 返回程序的开头，取得新的工件信息。

PK_PRGR_CV_PICKDROP1.TP

```
   1:  ! Pre-grouping Pick & Place G1
   2:  ! Note: This program does not
   3:  ! handle skip outbound flag
   4:  ! and gripper part presence check
   5:  ! If they are needed, modify this
   6:  ! program. For reference, use
   7:  ! PK_CV_PICK11.TP.
   8:  ! This program does not support
   9:  ! multiple-pick & wait position.
  10:
  11:  LBL[5]
  12:  UFRAME_NUM=0
  13:  UTOOL_NUM=1
  14:
  15:  !--------------------------------
  16:  LBL[10: Pick&Place Loop]
  17:  IF R[101:Cycle Stop G1]>0,JMP LBL[10000]
  18:  STOP_TRACKING
  19:  ! Get part from que
  20:  CALL PKCSGETQPRGR(传送带工作站 ID=R[130:Dp1x StnI],拾取补偿用 VR=1,放置补偿用 VR=2,状态
Reg=123) ;
  21:  !Convert GETQ stat
  22:  IF (R[123:Pk1x GetQ stat]=(1)),R[96:TemporaryUse2]=(2)
  23:  IF (R[123:Pk1x GetQ stat]=(0)),R[96:TemporaryUse2]=(1)
  24:  !Caclulate WAIT pos
  25:  CALL PKCSCALWPOS(传送带工作站 ID=R[110:Pk1x StnI],基准位置 PR=57,补偿用 VR=1,GetQ 状态
       =R[96:TemporaryUse2],移动判断用 Reg=95,待机位置 PR=55,无需移动范围=0)
  26:  ! Move to wait position
  27:  IF (R[95:TemporaryUse1]=0),CALL PK_CV_WAITPOS1(工具坐标编号=1,接近位置 PR=56,待机位置 PR=55,位
       置检查用 Reg=115)
  28:  IF R[123:Pk1x GetQ stat]=0,JMP LBL[20]
  29:  IF R[230:PRGR Robot Type]<>1,JMP LBL[9000]
  30:  JMP LBL[10000]
  31:
  32:  LBL[20:SetPickTrigger]
  33:  CALL PKCSTRGPRGR(“传送带工作站 ID”=R[110:Pk1x StnI],超时值(ms)=100,状态 Reg=95)
  34:  IF R[95:TemporaryUse1]=0,JMP LBL[80]
  35:  IF R[101:Cycle Stop G1]>0,CALL PKCSACKQPRGR(传送带工作站 ID=R[130:Dp1x StnI],返回=2)
  36:  IF R[101:Cycle Stop G1]>0,JMP LBL[10000]
  37:  JMP LBL[20]
  38:
  39:
  40:  LBL[80: PickPart]
  41:  ! Move to approach
  42:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
  43:
  44:  ! Move to pick pos
  45:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,CALL PKGRCLOSE(抓手
       ID=R[104:Gripper Id G1],开始区域=1,结束区域=1)
  46:  ! Pick delay
  47:  CALL PKGRGETPKDLY(抓手 ID=R[104:Gripper Id G1],待机时间  Reg=113)
  48:  WAIT R[113]
```

```
   49:   PAYLOAD[2] ;
   50:   ! Move to retreat
   51:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
   52:
   53:   !--------------------------------
   54:   ! Place the part in virtual tray
   55:   LBL[125: SetPlaceTrigger]
   56:   STOP_TRACKING
   57:   CALL PKCSTRGPRGR(传送带工作站 ID=R[130:Dp1x StnI],超时值(ms)=100,状态 Reg=95)
   58:   IF R[95:TemporaryUse1]=0,JMP LBL[200]
   59:   IF R[101:Cycle Stop G1]>0,CALL PKCSACKQPRGR(传送带工作站 ID=R[130:Dp1x StnI],删除=4)
   60:   IF R[101:Cycle Stop G1]>0,JMP LBL[10000]
   61:   JMP LBL[125]
   62:
   63:   LBL[200: PlacePart]
   64:   ! Move to approach
   65:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[58:Dp1 Cv Ap Ofst]
   66:
   67:   ! Move to place pos ;
   68:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[2] TB R[125]sec,CALL PKGROPEN(抓手
         ID=R[104:Gripper Id G1],开始区域=1,结束区域=1)
   69:   ! Place delay
   70:   CALL PKGRGETDPDLY(抓手 ID=R[104:Gripper Id G1],待机时间 Reg=133)
   71:   WAIT R[133]
   72:   PAYLOAD[1]
   73:   ! Move to retreat
   74:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[2] Tool_Offset,PR[58:Dp1 Cv Ap Ofst]
   75:   !--------------------------------
   76:
   77:   LBL[2000: Pick/Place OK]
   78:   CALL PKCSACKQPRGR(传送带工作站 ID=R[130:Dp1x StnI],成功=1)
   79:   JMP LBL[10]
   80:
   81:   LBL[9000]
   82:   WAIT      .02(sec)
   83:
   84:   LBL[10000: End]
   85:   STOP_TRACKING
[End]
```

7.11.4.4 PK_PRGR_CV_PICKVTRAY1.TP

此程序中，按以下步骤执行组 1 的机器人汇总拾取在同一传送带上被整列的工件之动作。

1 在开始拾取动作之前,将［跳过动作实施范围］的标记设定为 OFF。
2 使用 KAREL 程序 PKTRGETNUMCL，取得托盘的单元数。
3 将步骤 2 中取得的单元数作为层 2 的单元编号，使用 KAREL 程序 PKCSGETQCELL，从队列取得工件信息。
4 不管是否已从队列取得工件，在与此拾取动作对应的传送带工作站操作的设定中［待机位置］有效时，检查是否需要在传送带工作站的待机位置也即上游侧的边界等待机器人，若判断为需要进行等待，则使用 PKCSCALWPOS.PC 计算待机位置。
5 因工件不存在、或者工件尚未到达上游侧边界而机器人需要等待时，使用 PK_CV_WAITPOS1.TP 移动至待机位置。
6 未能利用 KAREL 程序 PKCSGETQCELL 取得工件信息时，再次尝试取得工件信息。
7 能够利用 KAREL 程序 PKCSGETQCELL 取得工件信息时，检查工件是否已被放置在层 1 的所有单元。工件尚未被放置在层 1 的全部单元中时，在 LBL［1000］以下的处理中，将工件信息以 ACK = 发信(3)返回队列，回到 2 并尝试取得新的工件信息。工件已被放置在层 1 的所有单元中时，进入到 8 以后的步骤。
8 将所选的工具坐标系设定为抓手的最初区域的工具坐标系。
9 使得机器人移动至接近位置。
10 移动过程中［跳过动作实施范围］的标记成为 ON（因超过下游侧的边界而机器人无法拾取工件）时，使用 KAREL 程序 PKCSACKQUE 将工件信息以 ACK = 发信(3)返回队列，回到 2 并取得新的工件信息。
11 使得机器人移动至拾取位置。使用先执行指令，与动作并行执行“关闭抓手”动作。“关闭抓手”动作通过 KAREL 程序 PKGRCLOSE 执行。

12 移动过程中［跳过动作实施范围］的标记成为 ON（因超过下游侧的边界而机器人无法拾取工件）时，使用 KAREL 程序 PKCSACKQUE 将工件信息以 ACK = 发信(3)返回队列，回到 2 并取得新的工件信息。
13 移动至拾取位置后，使用 KAREL 程序 PKGRGETPKDLY 取得［拾取延迟时间］，在该时间内与传送带一起移动。
14 在与此拾取动作对应的传送带工作站操作的设定中［工件有无检查］有效时，使用 KAREL 程序 PKGRCHKPP 来检查是否已把持工件。
15 在判断为抓取失败时，通过 ACK =删除(4)向队列发还工件信息，返回 2 并取得新的工件信息。
16 在判断为抓取成功时，在“抓手全部埋入的状态”也即负荷设定［2］中设定机器人的负荷设定。
17 使得机器人移动至逃避位置。
18 机器人多次拾取已被整列的工件时，返回 2 并取得新的工件信息。只拾取一次已被整列的工件时，结束程序。

PK_PRGR_CV_PICKVTRAY1.TP

```
  1:  ! Pre-grouping “Pick Virtual Tray”
  2:  ! for grp1 robot.
  3:  ! This program picks the entire
  4:  ! tray built by an upstream
  5:  ! pre-grouping robot in one shot.
  6:
  7:  ! It does not pick up unfilled
  8:  ! trays. Modification is needed
  9:  ! if it is necessary to handle
 10:  ! trays that are not full.
 11:
 12:  ! NOTE: It is necessary to define
 13:  ! one cell in the second layer of
 14:  ! the tray that has the same
 15:  ! coordinates as the cell 1 of
 16:  ! layer 1 but a different
 17:  ! model_id that is not used in
 18:  ! layer 1.
 19:
 20:  LBL[5]
 21:  F[R[102]]=(OFF)
 22:  UFRAME_NUM=0
 23:
 24:  LBL[10: Pick Loop]
 25:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
 26:  STOP_TRACKING
 27:  ! Get total cells in tray
 28:  CALL PKTRGETNUMCL(托盘 ID=R[121:Pk1x TrayId],层编号=0,单元数 Reg=122)
 29:  ! If R[122]=0, invalid tray
 30:  IF R[122:Pk1x TrayCtr]=0,CALL PKPOSTERR(错误代码=127038,中止=2)
 31:  ! Get tray cell from queue
 32:  CALL PKCSGETQCELL(传送带工作站 ID=R[110:Pk1x StnI],单元编号=R[122:Pk1x TrayCtr],超时值(ms)=100,
       补偿用 VR=1,状态 Reg=123,空单元数 Reg=96)
 33:  CALL PKCSCALWPOS(传送带工作站 ID=R[110:Pk1x StnI],基准位置 PR=57,补偿用 VR=1,GetQ 状态
       =R[123:Pk1x GetQ stat],移动判断用 Reg=95,待机位置 PR=55,无需移动范围=0)
 34:  IF (R[95:TemporaryUse1]=0),CALL PK_CV_WAITPOS1(工具坐标编号=R[107:Counter G1],接近位置 PR=56,待
       机位置 PR=55,位置检查用 Reg=115)
 35:  IF R[123:Pk1x GetQ stat]<>0,JMP LBL[10]
 36:  ! If unfilled cells exist,
 37:  ! skip them and don't pick.
 38:  IF R[96:TemporaryUse2]<>0,JMP LBL[1000]
 39:
 40:  LBL[50: Multipick]
 41:
 42:  LBL[80]
 43:
 44:  ! Set Utool
 45:  UTOOL_NUM=R[107:Counter G1]
 46:  ! Move to approach
 47:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
 48:  ! Flag on when Skip outbound true
 49:  IF (F[R[102]]),JMP LBL[1000]
```

```
   50:
   51:   ! Set Pick/drop registers
   52:   CALL PK_GR_GET_ENDZONE1(拾取和放置计数=R[107:Counter G],批量作业=R[112:Pk1x PkMult],数量
         =R[111:Pk1x Qty],结束区域 Reg=106)
   53:   ! Move to pick pos
   54:L PR[57:Pk1 Cv Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[125]sec,CALL PKGRCLOSE(抓手
         ID=R[104:Gripper Id G1],开始区域=R[107:Counter G1],结束区域=R[106:Gr EndZone G1])
   55:   ! Chk Skip Pick
   56:   IF (F[R[102]]),JMP LBL[1000]
   57:   ! Pick delay
   58:   CALL PKGRGETPKDLY(抓手 ID=R[104:Gripper Id G1],待机时间  Reg=113)
   59:   WAIT R[113]
   60:   ! Check part pres
   61:   CALL PKCSGETPPCHK(传送带工作站 ID=R[110:Pk1x StnI],工件有无检查 Reg=116)
   62:   IF R[116:Pk1x UsePPChk]=0,JMP LBL[90]
   63:   ! PKGRCHKPP grip_id, operation,
   64:   ! stat_reg, StartZone,
   65:   !EndZone (opt)
   66:   CALL PKGRCHKPP(抓手 ID=R[104:Gripper Id G1],拾取=0,状态 Reg=117,初始区域=R[107:Counter G1],结束区
         域=R[106:Gr EndZone G1])
   67:   LBL[90]
   68:   PAYLOAD[2]
   69:   ! Move to retreat
   70:L PR[57:Pk1 Cv Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
   71:   IF R[117:Pk1x PPStat]<>0,JMP LBL[1500]
   72:   JMP LBL[2000]
   73:
   74:   LBL[1000: Missed part]
   75:   F[R[102]]=(OFF)
   76:   ! Return part in VR1 to Queue
   77:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],发信=3)
   78:   JMP LBL[10]
   79:
   80:   LBL[1500: Dropped part]
   81:   F[R[102]]=(OFF)
   82:   ! Return part in VR1 to Queue
   83:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],删除=4)
   84:   R[117:Pk1x PPStat]=0
   85:   ! Insert code for rejecting parts
   86:   ! a reject location.
   87:   JMP LBL[10]
   88:
   89:
   90:   LBL[2000: Pick OK]
   91:   ! PKACKQ grp, vr, ack
   92:   CALL PKCSACKQUE(传送带工作站 ID=R[110:Pk1x StnI],成功=1)
   93:   --
   94:   IF R[112:Pk1x PkMult]=0,JMP LBL[2050]
   95:   R[107:Counter G1]=R[107:Counter G1]+R[111:Pk1x Qty]
   96:   JMP LBL[2100]
   97:
   98:   LBL[2050]
   99:   R[107:Counter G1]=R[107:Counter G1]+1
  100:   IF R[107:Counter G1]<=R[111:Pk1x Qty],JMP LBL[10]
  101:
  102:   LBL[2100]
  103:   LBL[10000: End]
  104:   STOP_TRACKING
  105:   F[R[102]]=(OFF)
[End]
```

7.12 Plug & Play 的缓冲区功能

本节中就 Plug & Play 中使用的缓冲区功能进行介绍。本功能包含在 R857: *i*RPickTool /扩充功能包加载项中（北美专用 R827: *i*RPickTool / Production Control Add-on）中，备有与本功能相关的用户接口、标准 TP 程序、KAREL 程序。设定可通过固定工作站操作的菜单进行。逐个拾取/放置工件时，可直接使用标准 TP 程序，无须进行特殊的编程。

注释
本功能包含在 R857: 本功能无法在 *i*RPickTool 基本上使用。
但是，*i*RPickTool 基本上提供有 PKFSCLRBUF.PC、PKFSPUTBUF.PC、PKFSGETBUF.PC 之类的、用来使用缓冲区的基本 KAREL 程序，客户自身可构筑使用了缓冲区的应用。

缓冲区是使用了托盘的固定工作站。预先已决定可作为缓冲区放置工件的单元，只要有尚未埋在缓冲区中的单元，机器人就可以将工件放置到缓冲区。此外，如果缓冲区中有工件，机器人就可以从缓冲区拾取工件。但是，如同分度传送带那样，它与当单元填满时排出缓冲区，重新设置别的缓冲区、这样的使用方法不对应。下图的示例中，显示由 4 个单元构成的缓冲区。

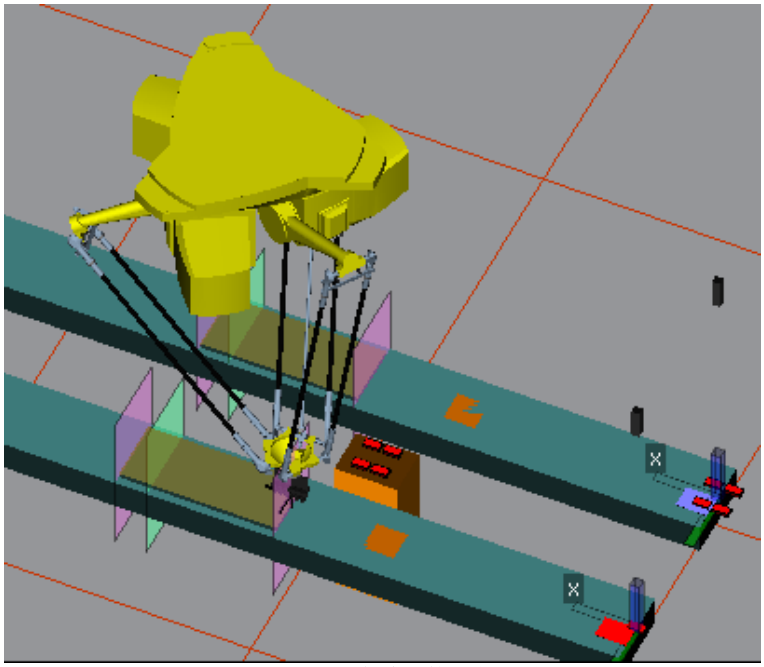


图 7.12(a) 由四个单元构成的缓冲区

本功能下，“拾取”－“放置”只支持“传送带工作站”－“传送带工作站”的情形。拾取”－“放置”在如下情况下不对应。

- “传送带工作站”－“固定工作站”
- “固定工作站”－“传送带工作站”
- “固定工作站”－“固定工作站”

本功能下，1 台机器人可使用的缓冲区只限于 1 个。
本功能下，如上所述，只与逐个拾取/放置工件的、“拾取 1 个”“放置 1 个”对应。

7.12.1 与缓冲区中的拾取/放置相关的机器人的动作

本功能下，在缓冲区进行拾取/放置的顺序，可以从“先进先出”和“后进先出”的 2 种中选择。标准设定下已选择“先进先出”。

从缓冲区拾取/放置工件时，机器人按照以下的规则动作。

- （拾取工件后）输出传送带的放置场所没有空间，且缓冲区中没有空间时，从供给传送带拾取工件，放置到缓冲区。

- 供给传送带上虽然有工件，但是尚未到达追踪区域，输出传送带的放置场所有空间，且工件在缓冲区时，动作根据工件到达追踪区域的上边位置为止的时间而不同。到达上边位置为止的时间比［拾取工件时使用缓冲区与否的判断时间］长时，从缓冲区拾取工件。到达上边位置为止的时间，受到供给传送带速度的影响。
- 虽然输出传送带的放置场所有空间，但是供给传送带上没有要拾取的工件，且缓冲区有工件时，从缓冲区拾取工件放置到输出传送带上。
- （拾取工件后）虽然输出传送带的放置场所有空间，但是尚未到达追踪范围，缓冲区有空间时，动作根据放置场所到达追踪范围的上边位置为止的时间而不同。到达上边位置为止的时间比［放置工件时使用缓冲区与否的判断时间］长时，将工件放置到缓冲区。到达上边位置为止的时间，受到输出传送带速度的影响。
- 供给传送带上没有要拾取的工件，且缓冲区上也没有工件时，在传送带工作站操作的设定中若已经勾选［待机位置］的［有效］，机器人就会在供给传送带的待机位置待机。
- （拾取工件后）输出传送带上没有放置场所，且缓冲区也没有空间时，在传送带工作站操作的设定中若已经勾选［待机位置］的［有效］，机器人就会在输出传送带的待机位置待机。

7.12.2 缓冲区的设定

要使用缓冲区，首先在树状图上新建固定工作站对象。
固定工作站的详情，请参照“6.7 固定工作站设定”。

在固定工作站的设定画面进行以下设定。

- 在［托盘］中选择适当的托盘，将［初始状态］设定为“空”。
- 勾选［缓冲区］的［有效］。
- 在［拾取顺序］中选择［先进先出］或［后进先出］。

［缓冲区］

作为缓冲区使用托盘时，选中［有效］。

［拾取顺序］

指定从缓冲区拾取工件的顺序。仅缓冲区有效时可以指定。

先进先出：
　拾取缓冲区内最先放入的工件。

后进先出：
　拾取缓冲区内最后放入的工件。

若创建固定工作站，则会在［机器人］中设定的机器人下，如下图所示那样，自动追加固定工作站操作（例：OP_FS_FSTN1）。

通过固定工作站操作的菜单，适当进行与缓冲区相关的设定。固定工作站操作的详情，请参考“7.3.2 固定工作站的操作”。

[拾取工件时使用缓冲区与否的判断时间]

供给传送带上虽然有工件，但是到达追踪范围为止，需要此值以上的时间时，缓冲区如有工件，机器人就会从缓冲区拾取工件。请在［拾取工件时使用缓冲区与否的判断时间］中，根据供给传送带的速度设定适当的值。

[放置工件时使用缓冲区与否的判断时间]

输出传送带上虽然有放置场所，但是到达追踪范围需要此值以上的时间时，缓冲区如有空间，机器人就会将工件放置到缓冲区。请在［放置工件时使用缓冲区与否的判断时间］中，根据输出传送带的速度设定适当的值。

注释
可以在这些值中输入负数。正数时表示到达追踪范围的“上边”位置为止的时间；负数是表示到达“上边”位置后的时间。

［拾取工件时使用缓冲区与否的判断时间/放置工件时使用缓冲区与否的判断时间］的设定，通过固定工作站操作的菜单也生产中也可随时变更。为了使得从缓冲区拾取工件的时机、放置到缓冲区上的时机适当，请调整这些时间。

7.12.3 缓冲区功能中使用的 KAREL 程序

本项中介绍与缓冲区功能相关的 KAREL 程序。

7.12.3.1 PKRBGETBUFID.PC

从机器人 ID 取得作为缓冲区使用的固定工作站的 ID。指定了不存在的机器人 ID 时，返还 0。此外，缓冲区不存在时，也返还 0。

参数 1：
指定机器人的 ID。

参数 2：
指定用来存储作为缓冲区使用的固定工作站 ID 的寄存器编号。

使用例：
在 R［88］取得 ID 为 1 的机器人作为缓冲区使用的固定工作站 ID。

```
CALL PKRBGETBUFID(机器人 ID=1,缓冲区 ID Reg=88)
```

7.12.3.2 PKFSGETBUFPT.PC

取得树状图的［固定工作站的操作］中设定的［拾取工件时使用缓冲区与否的判断时间］（单位 ms）。

参数 1：
指定作为缓冲区使用的固定工作站的 ID。

参数 2：
指定用来存储缓冲区使用判断时间的寄存器编号。

使用例：
在 R［95］取得固定工作站 ID 为 1 的缓冲区中使用的［拾取工件时使用缓冲区与否的判断时间］。

```
CALL PKFSGETBUFPT(缓冲区 ID=1,判断时间 Reg=95)
```

7.12.3.3 PKFSGETBUFDT.PC

取得树状图的［固定工作站的操作］中设定的［放置工件时使用缓冲区与否的判断时间］（单位 ms）。

参数 1：
指定作为缓冲区使用的固定工作站的 ID。

参数 2：
指定用来存储缓冲区使用判断时间的寄存器编号。

使用例：
在 R［95］取得固定工作站 ID 为 1 的缓冲区中使用的［放置工件时使用缓冲区与否的判断时间］。

```
CALL PKFSGETBUFDT(缓冲区 ID=1,判断时间 Reg=95)
```

7.12.4 缓冲区功能中使用的 TP 程序

本项中介绍与缓冲区功能相关的 TP 程序。要使用缓冲区功能，除了这里介绍的程序外，还需要以下的 TP 程序知识。

- PK_CV_PICK11.TP
- PK_CV_DROP11.TP
- PK_INIT1.TP
- PK_PICK1.TP
- PK_DROP1.TP

这些程序的详情，请参照“7.10 标准 TP 程序”。

7.12.4.1 PK_BF_GET_DATA1.TP

本程序在 PK_INIT1.TP 中使用。
本程序中，在寄存器和位置寄存器中取出缓冲区和相关的固定工作站的设定。同时进行以下处理。

- 使用 PKFSCLRACK.PC，擦除与缓冲区相关的全部未处理的 ACK。
- 使用 PKFSCLRBUF.PC，全部擦除缓冲区的单元的空信息。
- 使用 PKFSPUTBUF.PC，将缓冲区设定为初始状态（空）。

PK_BF_GET_DATA1.TP

```
1:  ! PK_BF_GET_DATA1
2:  ! Set Num Reg for G1 Buffer
3:
4:  CALL PKFSGETIXSIM(固定工作站 ID=R[199:Buf Id G1],信号模拟 Reg=95)
5:  CALL PKFSGETAPPR(固定工作站 ID=R[199:Buf Id G1],接近位置 PR Reg=95)
6:  CALL PKFSSETRDY(固定工作站 ID=R[199:Buf Id G1],就绪=1)
```

```
   7:  CALL PKFSCLRACK(固定工作站 ID=R[199:Buf Id G1])
   8:  CALL PKFSCLRBUF(固定工作站 ID=R[199:Buf Id G1],编号=1)
   9:  CALL PKFSPUTBUF(固定工作站 ID=R[199:Buf Id G1],编号=1)
  10:  --
  11:  R[204:Buf Ctr G1]=0
  12:  R[205:Picked G1]=0
  13:  R[126:Buf PP Stat G1]=0
  14:  LBL[1000]
  15:
  16:  LBL[10000: End]
  17:
[End]
```

7.12.4.2 PK_BF_PICK1.TP

从缓冲区拾取工件的程序。

- 计数器和工件数，使用与从传送带拾取工件的程序 PK_CV_PICK11.TP 相同的程序。
- 程序的构成与 PK_FS_PICK11.TP 类似。详情请参照“7.10.2.15 PK_FS_PICK11.TP”。

PK_BF_PICK1.TP

```
   1:  --
   2:
   3:  LBL[5]
   4:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
   5:
   6:  --
   7:  --
   8:  CALL PKFSGETUF(“固定工作站 ID”=R[199:Buf Id G1],“用户坐标系 Reg”=95)
   9:  UFRAME_NUM=R[95:TemporaryUse1]
  10:
  11:  LBL[10]
  12:  --
  13:  --
  14:  CALL PK_FS_READY(“固定工作站 ID”=R[199:Buf Id G1],“状态 Reg”=95)
  15:  IF R[95:TemporaryUse1]=0,JMP LBL[20]
  16:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
  17:  WAIT      .02(sec)
  18:  JMP LBL[10]
  19:
  20:  LBL[20]
  21:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
  22:
  23:  --
  24:  --
  25:  CALL PKFSGETBUF(“固定工作站 ID”=R[199:Buf Id G1],“编号”=1,“工件”=2,
       “补偿用 VR”=1,“状态 Reg”=123)
  26:  IF R[123:Pk1x GetQ stat]=0,JMP LBL[200]
  27:  JMP LBL[10000]
  28:
  29:  LBL[200]
  30:  CALL PK_GR_GET_ENDZONE1(“拾取和放置计数”=R[107:Counter G],
       “批量作业”=0,“数量”=1,“结束区域 Reg”=106)
  31:
  32:
  33:  LBL[300]
  34:  UTOOL_NUM=R[107:Counter G1]
  35:  --eg: Move to approach
```

```
  36:L PR[53:Bf Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[52:Bf Ap Ofst]
  37:   --eg: Move to pick pos
  38:L PR[53:Bf Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[127]sec,
      CALL PKGRCLOSE(“抓手 ID”=R[104:Gripper Id G1],“开始区域”=R[107:Counter G1],
      “结束区域”=R[106:Gr EndZone G1])
  39:   --
  40:   CALL PKGRGETPKDLY(“抓手 ID”=R[104:Gripper Id G1],“待机时间  Reg”=95)
  41:   WAIT R[95]
  42:   --
  43:   CALL PKFSGETPPCHK(“固定工作站 ID”=R[199:Buf Id G1],“工件有无检查 Reg”=95)
  44:   IF R[95:TemporaryUse1]=0,JMP LBL[400]
  45:   --
  46:   --
  47:   --
  48:   CALL PKGRCHKPP(“抓手 ID”=R[104:Gripper Id G1],
      “拾取”=0,“状态 Reg”=126,“初始区域”=R[107:Counter G1],“结束区域”=R[106:Gr EndZone G1])
  49:   LBL[400]
  50:   --
  51:   PAYLOAD[2]
  52:   --
  53:L PR[53:Bf Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[52:Bf Ap Ofst]
  54:   IF R[126:Buf PP Stat G1]<>0,JMP LBL[1500]
  55:   JMP LBL[2000]
  56:
  57:
  58:   LBL[1500: PartGotdropped]
  59:   --
  60:   CALL PKFSACKBUF(“固定工作站 ID”=R[199:Buf Id G1],“返回”=2)
  61:   R[126:Buf PP Stat G1]=0
  62:   JMP LBL[10000]
  63:
  64:
  65:   LBL[2000: Pick ok]
  66:   --
  67:   ! PKFSACKBUF(buf_id, ack, vr (opt
  68:   CALL PKFSACKBUF(“固定工作站 ID”=R[199:Buf Id G1],“成功”=1)
  69:   --eg: Update tray cell count ;
  70:   R[204:Buf Ctr G1]=R[204:Buf Ctr G1]-1
  71:   --
  72:   R[107:Counter G1]=R[107:Counter G1]+1
  73:   R[205:Picked G1]=1
  74:
  75:   LBL[3000: CheckIndex]
  76:   ! For indexing buffers only
  77:
  78:
  79:   LBL[10000: End]
[End]
```

7.12.4.3 PK_BF_DROP1.TP

将工件放置到缓冲区的程序。

- 计数器和工件数，使用与将工件放置到传送带上的程序 PK_CV_DROP11.TP 相同的程序。
- 程序的构成与 PK_FS_ DROP11.TP 类似。详情请参照“7.10.2.16 PK_FS_DROP11.TP”。

PK_BF_DROP1.TP

```
  1:  --
```

```
   2:
   3:  LBL[5]
   4:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
   5:
   6:  --
   7:  --
   8:  CALL PKFSGETUF(“固定工作站 ID”=R[199:Buf Id G1],“用户坐标系 Reg”=95)
   9:  UFRAME_NUM=R[95:TemporaryUse1]
  10:
  11:  LBL[10]
  12:  --
  13:  --
  14:  CALL PK_FS_READY(“固定工作站 ID”=R[199:Buf Id G1],“状态 Reg”=95)
  15:  IF R[95:TemporaryUse1]=0,JMP LBL[20]
  16:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
  17:  WAIT      .02(sec)
  18:  JMP LBL[10]
  19:
  20:  LBL[20]
  21:  IF R[101:Cycle Stop G1]=1,JMP LBL[10000]
  22:
  23:  --
  24:  --
  25:  CALL PKFSGETBUF(“固定工作站 ID”=R[199:Buf Id G1],“编号”=1,
          “单元”=1,“补偿用 VR”=1,“状态 Reg”=143)
  26:  IF R[143:Dp1x GetQ stat]=0,JMP LBL[200]
  27:  JMP LBL[10000]
  28:
  29:  LBL[200]
  30:  CALL PK_GR_GET_ENDZONE1(“拾取和放置计数”=R[107:Counter G],
          “批量作业”=0,“数量”=1,“结束区域 Reg”=106)
  31:
  32:
  33:  LBL[300]
  34:  UTOOL_NUM=R[107:Counter G1]
  35:  --
  36:L PR[53:Bf Ref Pos] max_speed CNT100 VOFFSET,
       VR[1] Tool_Offset,PR[52:Bf Ap Ofst]
  37:  --
  38:L PR[53:Bf Ref Pos] max_speed CNT0 VOFFSET,VR[1] TB R[147]sec,
       CALL PKGROPEN(“抓手 ID”=R[104:Gripper Id G1],“开始区域”=R[107:Counter G1],“结束区域”
=R[106:Gr EndZone G1])
  39:  --
  40:  CALL PKGRGETDPDLY(“抓手 ID”=R[104:Gripper Id G1],“待机时间 Reg”=95)
  41:  WAIT R[95]
  42:  --
  43:  CALL PKFSGETPPCHK(“固定工作站 ID”=R[199:Buf Id G1],“工件有无检查 Reg”=95)
  44:  IF R[95:TemporaryUse1]=0,JMP LBL[400]
  45:  --
  46:  --
  47:  --
  48:  CALL PKGRCHKPP(“抓手 ID”=R[104:Gripper Id G1],“放置”=1,“状态 Reg”=126,
          “初始区域”=R[107:Counter G1],“结束区域”=R[106:Gr EndZone G1]) ;
  49:  LBL[400]
  50:  --
  51:  PAYLOAD[1]
  52:  --
```

```
  53:L PR[53:Bf Ref Pos] max_speed CNT100 VOFFSET,VR[1] Tool_Offset,PR[52:Bf Ap Ofst]
  54:   IF R[126:Buf PP Stat G1]<>0,JMP LBL[1500]
  55:   JMP LBL[2000]
  56:
  57:
  58:   LBL[1500: PartGotdropped]
  59:   --eg: Return part in VR1 to Queue
  60:   CALL PKFSACKBUF(“固定工作站 ID”=R[199:Buf Id G1],“返回”=2)
  61:   R[126:Buf PP Stat G1]=0
  62:   JMP LBL[10000]
  63:
  64:
  65:   LBL[2000: Pick ok]
  66:   --
  67:   ! PKFSACKBUF(buf_id, ack, vr (opt
  68:   CALL PKFSACKBUF(“固定工作站 ID”=R[199:Buf Id G1],“成功”=1) ;
  69:   --eg: Update tray cell count
  70:   R[204:Buf Ctr G1]=R[204:Buf Ctr G1]+1
  71:   --
  72:   R[107:Counter G1]=R[107:Counter G1]+1
  73:   R[205:Picked G1]=0
  74:
  75:   LBL[3000: CheckIndex]
  76:   ! For indexing buffers only
  77:
  78:
  79:   LBL[10000: End]
[End]
```

7.12.4.4 PK_BF_OK2PICK1.TP

本程序在 PK_BUF_PICK1.TP 中使用，检查能否从缓冲区拾取工件。从缓冲区拾取工件，只能在缓冲区中有工件时进行。是否有工件，通过使用 PKFSGETBUF.PC 能否取得工件信息进行判断。使得取得的工件信息再返回缓冲区。

在调用此程序时，需要如下 2 个参数。
参数 1：作为缓冲区使用的固定工作站的 ID
参数 2：存储缓冲区状态的寄存器编号（缓冲区的状态　0：无工件、1：有工件）

PK_BF_OK2PICK1.TP

```
   1:  !PK_BF_OK2PICK1
   2:  ! Arguments
   3:  ! Arg 1: Buf Id
   4:  ! Arg 2: ok2Pick reg (0 or 1)
   5:
   6:  ! Get a part from buf
   7:  ! PKFSGETBUF(buf_id, index,
   8:  ! part_or_cell, vr, stat_reg,
   9:  ! model_id (opt))
  10:  CALL PKFSGETBUF(固定工作站 ID=AR[1],编号=1,工件,补偿用 VR=1,状态 Reg=123)
  11:  IF R[123:Pk1x GetQ stat]>0,JMP LBL[1000]
  12:  ! Ok
  13:  ! Return the part or cell
  14:  CALL PKFSACKBUF(固定工作站 ID=AR[1],返回)
  15:  R[AR[2]]=1
  16:  JMP LBL[10000]
  17:
  18:  LBL[1000: Not ok]
  19:  R[AR[2]]=0
  20:
  21:  LBL[10000: End]
[End]
```

7.12.4.5 PK_BF_OK2DROP1.TP

本程序在 PK_BUF_PICK1.TP 和 PK_BUF_DROP1.TP 中使用，检查能否将工件放置在缓冲区。将工件放置在缓冲区，只能在缓冲区内有空间时进行。是否有空间，通过使用 PKFSGETBUF.PC 能否取得空单元的信息来进行判断。使得取得的单元信息再返回缓冲区。

在调用此程序时，需要如下 2 个参数。
参数 1：作为缓冲区使用的固定工作站的 ID
参数 2：存储缓冲区状态的寄存器编号（缓冲区的状态　0：已满、1：有空间）

PK_BF_OK2DROP1.TP

```
   1:  !PK_BF_OK2DROP1
   2:  ! Arguments
   3:  ! Arg 1: Buf Id
   4:  ! Arg 2: ok2Pick reg (0 or 1)
   5:
   6:  ! Get an empty cell from buf
   7:  ! PKFSGETBUF(buf_id, index,
   8:  ! part_or_cell, vr, stat_reg,
   9:  ! model_id (opt))
  10:  CALL PKFSGETBUF(固定工作站 ID=AR[1],编号=1,单元,补偿用 VR=1,状态 Reg=143)
  11:  IF R[143:Dp1x GetQ stat]>0,JMP LBL[1000]
  12:  ! Ok
  13:  ! Return the part or cell
  14:  CALL PKFSACKBUF(固定工作站 ID=AR[1],返回)
  15:  R[AR[2]]=1
  16:  JMP LBL[10000]
  17:
  18:  LBL[1000: Not ok]
  19:  R[AR[2]]=0
  20:
  21:  LBL[10000: End]
[End]
```

7.12.4.6 PK_CV_OK2PICK1.TP

本程序在 PK_BUF_PICK1.TP 中使用，检查能否从传送带工作站拾取工件。从传送带工作站拾取工件，只能在工件在［拾取工件时使用缓冲区与否的判断时间］中设定的时间以内到达追踪范围时进行。否则，就需要检查能否从缓冲区拾取工件。工件到达追踪范围为止的时间，使用 PKCSGETTIME.PC 来取得。

在调用此程序时，需要如下 6 个参数。
参数 1：传送带工作站所属的传送带的 ID
参数 2：作为缓冲区使用的固定工作站的 ID
参数 3：顺序（参照“6.8.5.7 PKCSGETTIME.PC”的参数 2）
参数 4：连续拾取标记（参照“6.8.5.7 PKCSGETTIME.PC”的参数 3）
参数 5：存储［拾取工件时使用缓冲区与否的判断时间］的时间之寄存器编号
参数 6：存储传送带工作站状态的寄存器编号
（传送带工作站的状态　0：无工件、1：有工件）

PK_CV_OK2PICK1.TP

```
   1:  !PK_CV_OK2PICK1
   2:  ! Input
   3:  ! Arg 1: Cv Id
   4:  ! Arg 2: Buf id
   5:  ! Arg 3: Order
   6:  ! Arg 4: Pick Counter
   7:  ! Arg 5: Buf partTime Reg
   8:  ! Arg 6: ok2Pick reg (0 or 1)
   9:
  10:  ! Get the conveyor name
  11:  CALL PKCVGETNAME(传送带 ID=AR[1],传送带名称 SR=4)
  12:  ! Get conv id from name
  13:  CALL PKCSGETID(CONV1=SR[4],工作站 ID Reg=110)
  14:  ! Get pick time for buffer
  15:  ! PKFSGETBUFPT(buf_id, time_reg)
  16:  CALL PKFSGETBUFPT(缓冲区 ID=AR[2],判断时间 Reg=AR[5])
  17:
  18:  ! Check if the part is
  19:  ! sufficiently upstream
  20:  ! Get time from Up bound
  21:  ! PKCSGETTIME(cs_id, order,
  22:  !   pick_ctr, pick_qty,
  23:  ! time_reg, vr, stat_reg)
  24:  CALL PKCSGETTIME(传送带工作站 ID=R[110:Pk1x StnI],顺序=AR[3],连续的标记=AR[4],
       数量=R[111:Pk1x Qty],时间 Reg=95,补偿用 VR=1,状态 Reg=96)
  25:  IF R[96:TemporaryUse2]=3,JMP LBL[1000]
  26:  IF (R[95:TemporaryUse1]>R[AR[5]]),JMP LBL[1000]
  27:
  28:  LBL[500: Ok]
  29:  R[AR[6]]=1
  30:  JMP LBL[10000]
  31:
  32:  LBL[1000: Not ok]
  33:  R[AR[6]]=0
  34:
  35:  LBL[10000: End]
[End]
```

7.12.4.7 PK_CV_OK2DROP1.TP

本程序在 PK_BUF_PICK1.TP 和 PK_BUF_DROP1.TP 中使用，检查能否将工件放置到传送带工作站。将工件放置到传送带工作站，只能在托盘的单元等放置位置在［放置工件时使用缓冲区与否的判断时间］中设定的时间以内到达追踪范围时进行。否则，就需要检查能否将工件放置到缓冲区。放置位置到达追踪范围为止的时间，使用 PKCSGETTIME.PC 来取得。

在调用此程序时，需要如下 6 个参数。
参数 1：传送带工作站所属的传送带的 ID
参数 2：作为缓冲区使用的固定工作站的 ID
参数 3：顺序（参照“6.8.5.7 PKCSGETTIME.PC”的参数 2）
参数 4：连续拾取标记（参照“6.8.5.7 PKCSGETTIME.PC”的参数 3）
参数 5：存储［放置工件时使用缓冲区与否的判断时间］的时间之寄存器编号
参数 6：存储传送带工作站状态的寄存器编号
（传送带工作站的状态　0：无放置位置、1：有放置位置）

PK_CV_OK2DROP1.TP

```
  1:  !PK_CV_OK2DROP1
  2:  ! Input
  3:  ! Arg 1: Cv Id
  4:  ! Arg 2: Buf id
  5:  ! Arg 3: Order
  6:  ! Arg 4: Drop Counter
  7:  ! Arg 5: Buf partTime Reg
  8:  ! Arg 6: ok2Pick reg (0 or 1)
  9:
 10:  ! Get the conveyor name
 11:  CALL PKCVGETNAME(传送带 ID=AR[1],传送带名称 SR=5)
 12:  ! Get conv id from name
 13:  CALL PKCSGETID(CONV1=SR[5],工作站 ID Reg=130)
 14:  ! Get drop time for buffer
 15:  ! PKFSGETBUFDT(buf_id, time_reg)
 16:  CALL PKFSGETBUFDT(缓冲区 ID=AR[2],判断时间 Reg=AR[5])
 17:
 18:  ! Check if the part is
 19:  ! sufficiently upstream
 20:  ! Get time from Up bound
 21:  ! PKCSGETTIME(cs_id, order,
 22:  !   pick_ctr, pick_qty,
 23:  ! time_reg, vr, stat_reg)
 24:  CALL PKCSGETTIME(传送带工作站 ID=R[130:Dp1x StnI],顺序=AR[3],连续的标记=AR[4],
      数量=R[131:Dp1x Qty],时间 Reg=95,补偿用 VR=2,状态 Reg=96)
 25:  IF R[96:TemporaryUse2]=3,JMP LBL[1000]
 26:  IF (R[95:TemporaryUse1]>R[AR[5]]),JMP LBL[1000]
 27:
 28:  LBL[500: Ok]
 29:  R[AR[6]]=1
 30:  JMP LBL[10000]
 31:
 32:  LBL[1000: Not ok]
 33:  R[AR[6]]=0
 34:
 35:  LBL[10000: End]
[End]
```

7.12.4.8 PK_BUF_PICK1.TP

本程序在 PK_PICK1.TP 中使用，决定从缓冲区拾取工件（PK_BF_PICK1.TP），还是从供给传送带拾取工件（PK_CV_PICK11.TP）。

PK_BUF_PICK1.TP

```
   1:  ! PK_BUF_PICK1
   2:  ! Buffer Pick Grp1
   3:
   4:  LBL[50: Buf loop]
   5:  IF R[205:Picked G1]>0,JMP LBL[10000]
   6:  R[202:Buf Available G1]=0
   7:  R[203:CS Available G1]=0
   8:  R[97:TemporaryUse3]=0
   9:  LBL[100]
  10:  IF R[129:Dp1x CvId]>0,CALL PK_CV_OK2DROP1(“传送带 ID”=R[129:Dp1x CvId],
           “缓冲区 ID”=R[199:Buf Id G1],“顺序”=1,“放置编号”=1,“判断时间 Reg”=201,“放置 OK Reg”
=97)
  11:  CALL PK_BF_OK2PICK1(“缓冲区 ID”=R[199:Buf Id G1],“拾取 OK Reg”=202)
  12:  CALL PK_CV_OK2PICK1(“传送带 ID”=R[109:Pk1x CvId],“缓冲区 ID”=R[199:Buf Id G1],
           “顺序”=1,“拾取编号”=R[107:Counter G1],“判断时间 Reg”=200,“拾取 OK Reg”=203)
  13:  IF R[203:CS Available G1]>0,CALL PK_CV_PICK11
  14:  IF ((R[203:CS Available G1]<=0) AND (R[202:Buf Available G1]>0) AND (R[97:TemporaryUse3]>0)),
           CALL PK_BF_PICK1
  15:  IF R[205:Picked G1]<1,CALL PK_CV_PICK11
  16:  IF R[205:Picked G1]<1,JMP LBL[50]
  17:  LBL[10000]
[End]
```

7.12.4.9 PK_BUF_DROP1.TP

本程序在 PK_DROP1.TP 中使用，决定将工件放置到缓冲区（PK_BF_DROP1.TP），还是将工件放置到输出传送带（PK_CV_DROP11.TP）。

PK_BUF_DROP1.TP

```
   1:  ! PK_BUF_DROP1
   2:  ! Buffer Drop Grp1
   3:
   4:  LBL[50: Buf loop]
   5:  IF R[205:Picked G1]<1,JMP LBL[10000]
   6:  R[202:Buf Available G1]=0
   7:  R[203:CS Available G1]=0
   8:  LBL[100]
   9:  CALL PK_BF_OK2DROP1(“缓冲区 ID”=R[199:Buf Id G1],“放置 OK Reg”=202)
  10:  CALL PK_CV_OK2DROP1(“传送带 ID”=R[129:Dp1x CvId],“缓冲区 ID”=R[199:Buf Id G1],
           “顺序”=1,“放置编号”=R[107:Counter G1],“判断时间 Reg”=201,“放置 OK Reg”=203)
  11:  IF R[203:CS Available G1]>0,CALL PK_CV_DROP11
  12:  IF ((R[203:CS Available G1]<=0) AND (R[202:Buf Available G1]>0) AND (R[205:Picked G1]>0)),
           CALL PK_BF_DROP1
  13:  IF R[205:Picked G1]>0,CALL PK_CV_DROP11
  14:  IF R[205:Picked G1]>0,JMP LBL[50]
  15:  LBL[10000]
[End]
```

7.13 使用 *i*RPickTool 的跳过动作实施范围时的安全逃避动作

如果机器人在传送带上进行拾取部件的动作中/动作后（或者是放置部件的动作中/动作后）发生“LNTK-042 移出边界外”的错误，以后的逃避动作就会被跳过。
如图 7.13 所示，机器人将部件放置到深托盘或箱子时，如果跳过逃避动作，机器人有可能就会在移动到下一个示教点时，碰撞到现在托盘的侧壁。

注释
“LNTK-042 移出边界外”错误会在部件非常接近下游边界，内部追踪软件判断机器人在通过下游边界之前没有足够时间拾取部件时发生。因此，在初始状态下会跳过造成机器人超出下游边界的动作，机器人就会移动以拾取传送带上的下一个部件。

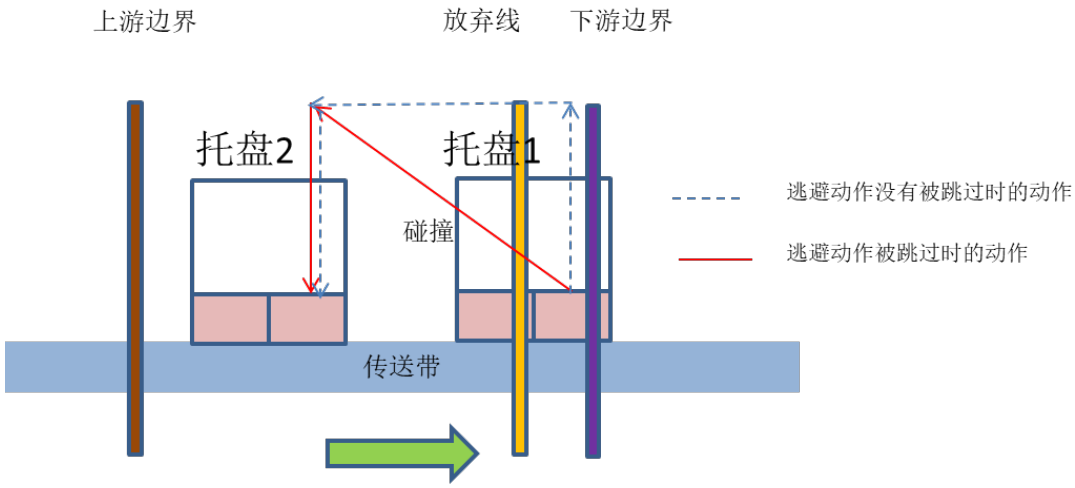


图 7.13 “LNTK-042 移出边界外”错误

*i*RPickTool（J945）中附带 PK_SAFE_RETREAT.TP 的 TP 程序。在此程序中，如果跳过逃避动作，就会使机器人相对于传送带垂直移动。逃避动作在图 7.13 中用虚线表示。利用从传送带取出工件的程序（PK_CV_PICK11.TP）或放置工件的程序（PK_CV_DROP11.TP）调用该程序，可避免该问题。

PK_SAFE_RETREAT.TP 中有 2 个参数。

参数 1：
工具坐标系编号。

参数 2：
用于使机器人安全逃避的工具补偿量。

如果在输入传送带上发生跳过动作实施范围错误，请修正 PK_CV_PICK11.TP 后再使用，如果在放置部件的动作中发生跳过动作实施范围错误，请修正 PK_CV_DROP11.TP 后使用。
PK_SAFE_RETREAT.TP 仅可以在 *i*RPickTool（J945）使用。

以下例子表示从传送带取出工件的程序 PK_CV_PICK11.TP 内 PK_SAFE_RETREAT.TP 程序的使用方法。如果发生“LNTK-042 移出边界外”错误，请注意标记 F[Reg[102］］会变成 ON。标记变成 ON 时，可以调用 PK_SAFE_RETREAT.TP。

```
71:   --
72:   IF (F[R[102]]),
   :  CALL PK_SAFE_RETREAT(
   :  工具坐标编号=R[107:Counter G1],
   :  工具补偿PR=56)
73:   --
74:   CALL PKGRGETPKDLY(
   :  抓手ID=R[104:Gripper Id G1],
   :  待机时间 Reg=113)
75:   WAIT R[113]
76:   --
```

```
85:   --
86:L   PR[57:Pk1 Cv Ref Pos] max_speed
   :   CNT100 VOFFSET,VR[1]
   :   Tool_Offset,PR[56:Pk1 Cv Ap Ofst]
   :
87:   IF (F[2]),CALL PK_SAFE_RETREAT(
   :   工具坐标编号=R[107:Counter G1],
   :   工具补偿PR=56)
88:   IF R[117:Pk1x PPStat]<>0,
   :   JMP LBL[1500]
89:   JMP LBL[2000]
```

7.14 多层托盘中的固定接近高度

如将部件放置在多层的托盘或箱子，如图 7.14(a)所示，如果机器人移到更高的层，则接近高度就发生变化。如果进行更高的移动，在背较高的托盘中，将有可能会降低效果。另外，还有可能在机器人的动作中发生极限错误，造成机器人停止。

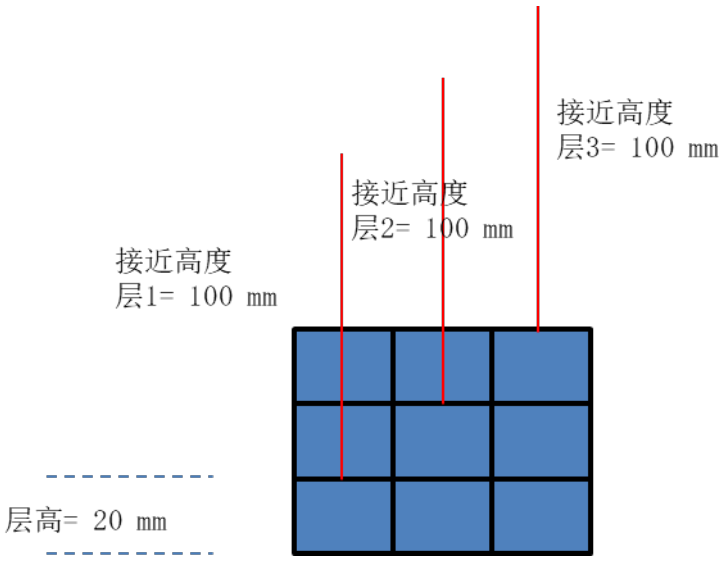


图 7.14(a) 机器人移动的接近高度按每个高层增加

使用以下 KAREL 程序，不论托盘为多少层，都可以将机器人移动到固定的接近高度。

- PKCSFIXAPRZ.PC
  传送带上的多层托盘专用
- PKFSFIXAPRZ.PC
  固定工作站上的多层托盘专用

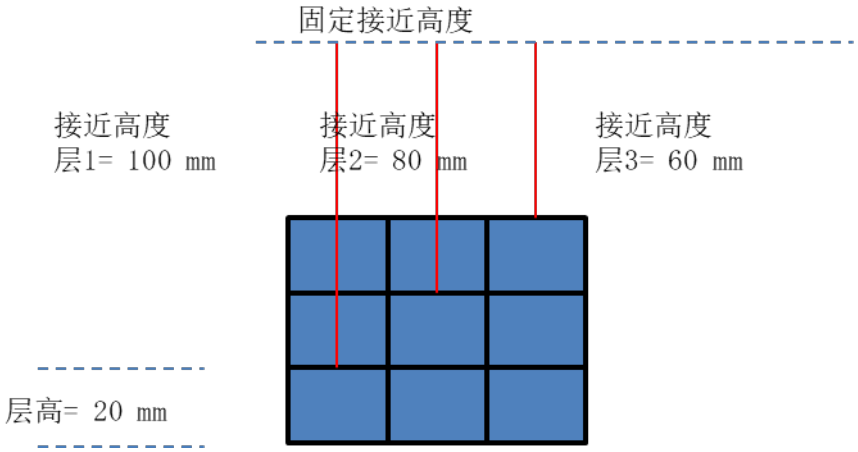


图 7.14(b) 固定接近高度

本功能仅可在 *i*RPickTool（J945）中使用。

7.14.1 PKCSFIXAPRZ.PC

PKCSFIXAPRZ.PC 可以从进行传送带拾取的程序 PK_CV_PICK11.TP 或进行放置动作的程序 PK_CV_DROP11.TP 调用。

参数 1：
整数。传送带工作站的 ID。

参数 2：
补偿的类型。在初始状态下设定为［工具补偿］，通常不需要更改。
向 PK_CV_PICK11.TP 或 PK_CV_DROP11.TP 的接近位置移动中，将补偿量应用于工具坐标系。

注释
向 PK_CV_PICK11.TP 或 PK_CV_DROP11.TP 的接近位置移动中，如希望将补偿量应用到跟踪坐标系，请将参数 2 更改成［位置补偿］。此外，需要确认和修正 TP 程序，以确保补偿量应用到跟踪坐标系后不会出现问题。

参数 3：
整数。PKCSGETQUE 的处理状态。

参数 4：
整数。0 之外时，表示该功能没有正常工作。

使用例：

PKCSFIXAPRZ(传送带工作站 ID=R[110:Pk1x StndId]，工具补偿, GetQ 状态=R[123:)Pk1x GetQ stat]，状态 Reg=1)

使用 PK_CV_PICK11.TP 程序的例子如下所示。调用 PKCSGETQUE 后（移动到接近位置前），需要调用 PKCSFIXAPRZ。

```
46:   LBL[80]
47:   CALL PKCSFIXAPRZ(
   :  传送带工作站ID=R[110:Pk1x StnId],
   :  位置补偿,
   :  GetQ状态=R[123:Pk1x GetQ stat],
   :  状态Reg=1)
48:
49:   UTOOL_NUM=R[107:Counter G1]
50:   IF R[115:Pk1x UsePosChk]=0,
   :  JMP LBL[85]
51:   --
```

注释
使用 PKCSFIXAPRZ.PC 时，在传送带工件站操作中［接近时偏移量］的［X］、［Y］、［W］、［P］、［R］中指定 0。

7.14.2 PKFSFIXAPRZ.PC

PKFSFIXAPRZ.PC 可以从进行拾取的程序 PK_FS_PICK11.TP 或进行放置动作的程序 PK_FS_DROP11.TP 调用。

参数 1：
整数。固定工作站的 ID。

参数 2：
补偿的类型。在初始状态下设定为［工具补偿］，通常不需要更改。
向 PK_CV_PICK11.TP 或 PK_CV_DROP11.TP 的接近位置移动中，将补偿量应用于工具坐标系。

注释
向 PK_CV_PICK11.TP 或 PK_CV_DROP11.TP 的接近位置移动中，如希望将补偿量应用到固定工件站的用户坐标系，请将参数 2 更改成［位置补偿］。此外，需要确认和修正 TP 程序，以确保补偿量应用到固定工作站的用户坐标系后不会出现问题。

参数 3：
整数。0 之外时，表示该功能没有正常工作。

使用例：

```
PKFSFIXAPRZ(固定工作站 ID=R[110:Pk1x StndId]，工具补偿，状态 Reg=1)
```

使用 PK_FS_PICK11.TP 程序的例子如下所示。移动到接近位置前，需要调用 PKFSFIXAPRZ。

```
 36:  LBL[300]
 37:  UTOOL_NUM=R[107:Counter G1]
 38:  --
 39:  CALL PK_GR_GET_ENDZONE1(
    :  拾取和放置计数=R[107:Counter G1],
    :  批量作业=R[112:Pk1x PkMult],
    :  数量=R[111:Pk1x Qty],
    :  结束区域Reg=106)
 40:  CALL PKFSFIXAPRZ(
    :  固定工作站ID=R[130:Dp1x StnId],
    :  工具补偿,状态Reg=1)
```

注释
使用 PKFSFIXAPRZ.PC 时，在固定工作站操作中［接近时偏移量］的［X］、［Y］、［W］、［P］、［R］中指定 0。

7.15 指定防止负载错误的 *i*RPickTool 机器人最高速度

*i*RPickTool（J945）的 TP 程序中，使用最高速度实现最高速度下的动作。关于使用最高速度的例子，说明如下。

```
86:L PR[57:Pk1 Cv Ref Pos］ max_speed CNT100 VOFFSET,VR[1］
    OFFSET,PR[56:Pk1 Cv Ap Ofst］      ;
```

如果机器人在初始状态下长时间地执行最大速度命令，*i*RPickTool 的高速应用程序将有中有会出现过热警报（需要机器人停止后重新启动）等负载循环相关问题。为防止循环错误，如下图所示，可在 *i*RPickTool 的［机器人］对象中的［机器人最高速度］设定合适的值。一经设定，该值可应用于使用最高速度命令的所有 TP 程序。
如果是拳头机器人，［机器人的最高速度］的初始值为 10,000 mm/s。
在 SCARA 机器人、LRMate 及其他串联型机器人中，［机器人的最高速度］的初始值为 2,000 mm/s。

［机器人的最高速度］的值也会被保存在方案中。

8 不同用途的设定例子

本章针对不同 *i*RPickTool 基本用途的系统设定例子举例说明。

8.1 根据传送带上的位置，由不同的机器人拾取

需要根据传送带上的工件位置，由其他机器人拾取的情况。
介绍的具体例是，准备有 2 台机器人，由上游侧的机器人拾取传送带右半部的工件，由下游侧的机器人拾取传送带左半部的工件的情况。传送带上的工件由相机检出。

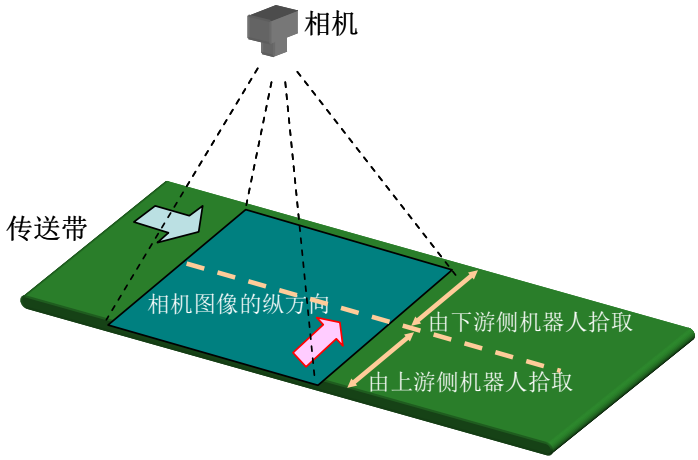


图 8.1(a)根据传送带上的工件位置更改取出机器人的设定例子

此外，还进行如下假设。

- 传送带的构成如下图 8.1(b)传送带构成示例图进行定义。
  与“2.4 系统构成例”中介绍的构成例 1 相同。

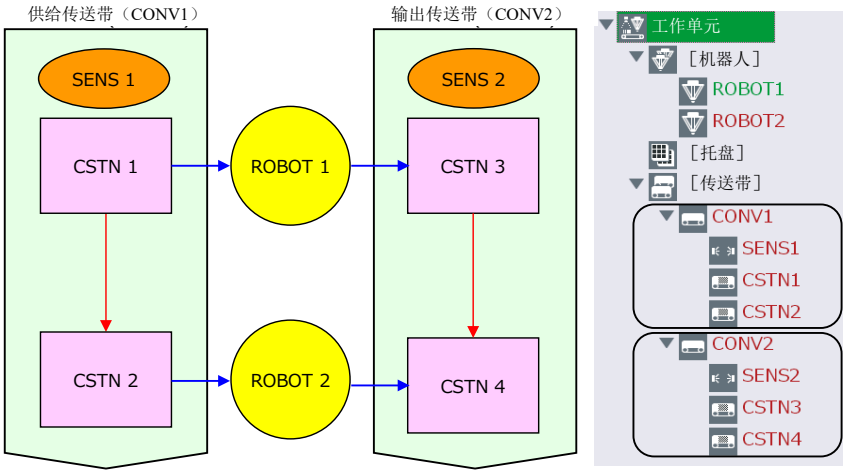
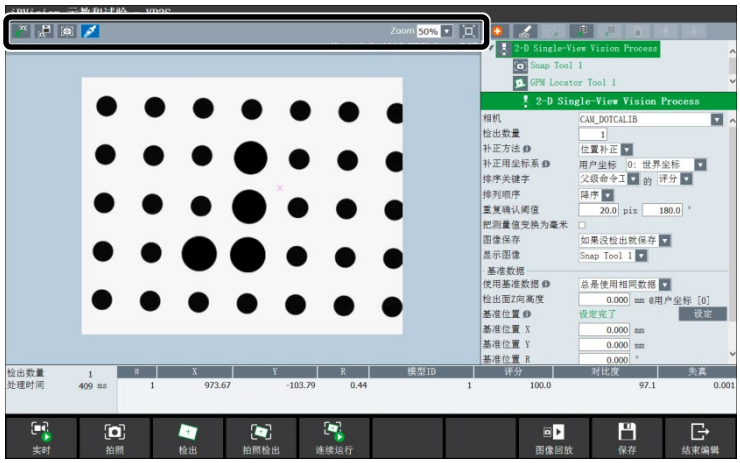


图 8.1(b)传送带构成示例

- 设置相机，使相机图像的横方向与传送带的行进方向平行。因此，传送带的宽方向则是相机图像上的纵方向。相机图像上的纵方向的坐标值在 *i*RVision 中以 Vt 表示。
- 令以图形匹配检出的工件的模型 ID 为 1。如果检测到的图像上 Vt 坐标值大于传送带中央的 Vt 坐标值，利用条件判断工具将模型 ID 设成 2（用“取样工具”确认传送带中央的 Vt 坐标值。）。

用 *i*RVision 确认像素的 Vt 坐标值

可以使用 *i*RVision 的取样工具确认在图像上选中的像素位置。



1 点击按钮。
取样工具启用，并在图像左上角显示 ╳ 。
2 点击图像上的任意像素。
╳标记移动，在右上角显示像素位置（Vt, Hz）。

条件判断工具的插入

在视觉程序中插入条件判断工具。

1 在编辑画面的树状图中，选择［GPM Locator Tool 1］，按下按钮。
则显示以下画面。

2　选择［条件判断］，按下 F4［下一个>］。

3　在［名称］中输入工具名（例：Conditional Execution Tool 1），按下 F4［确定］。

条件判断工具设定

如果 Vt 值大于传送带中央的 Vt 坐标值，（此处以 240 进行说明），设定时模型 ID 加上 1。

1 在树状图中选择［Conditional Execution Tool 1］。
则显示以下画面。

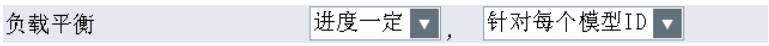
2 在［设定值］的［值 1］中选择［GPM Locator Tool 1］的［Vt］。
3 在［设定条件］中进行设定，使［条件 1］成为［值 1>240］。
4 在［设定处理］中选择［将随后的数据加到模型 ID］，值指定为 1。
5 按下 F5［结束编辑］。

6 按下［>（下一页）］，显示下一功能键，再按下 F5［保存］。

负载平衡设定

在传送带的负载平衡的设定中进行设定，分别由上游侧机器人 100%处理模型 ID1 的工件，下游侧机器人 100%处理模型 ID2 的工件。
打开传送带的设置画面，进行负载平衡的设定。这里只显示供给传送带。

1　打开传送带的设置画面。显示如下的对话框。
2　将［负载平衡］的左侧选框设定为［进度一定］或［保持个数比例］，使右侧的选框为［针对每个模型 ID］。



3　在［指定需处理的模型 ID］中指定 1 和 2。
4　负载平衡设定为由上游侧机器人 100％处理模型 ID1，下游侧机器人 100％处理模型 ID2。

8.2　根据工件的部分特征，由不同的机器人拾取

根据工件特征的不同进行种类的判别，根据种类由其他机器人拾取的情况。
这里作为具体例，介绍工件的一部分有孔，以及没有孔的工件 2 种工件传送时，根据有无孔切换分配目的地的例子。
下图是将螺母作为工件的情况。

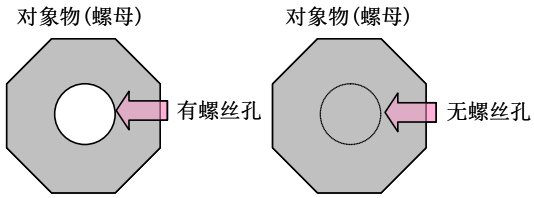


图 8.2(a)工件部分特征（部分有孔工件与无孔工件）示例

此外，还进行如下假设。

- 是否有孔，通过使用直方图工具（Histogram Tool）测量亮度的标准偏差进行判别。如果螺母表面均一，有孔时，亮度的标准偏差变大，无孔时，亮度的标准偏差变小。

	有孔	无孔
标准偏差	较大值	较小值

> **⚠ 注意**
> **根据背景等的不同，也会有无上述倾向的情况。这里介绍的仅是部分例子。**

> **注释**
> **孔的有无，也可以通过斑点工具，图形匹配工具等进行判别。这里作为一例，介绍使用直方图的方法。**

- 使图形匹配工具中示教的工件的模型 ID 为 1。
- 有孔的情况，使用条件判断工具，使模型 ID 为 2。
- 传送带定义如图 8.2(b)中的图所示。
  与“2.4 系统构成例”中介绍的构成例 1 相同。

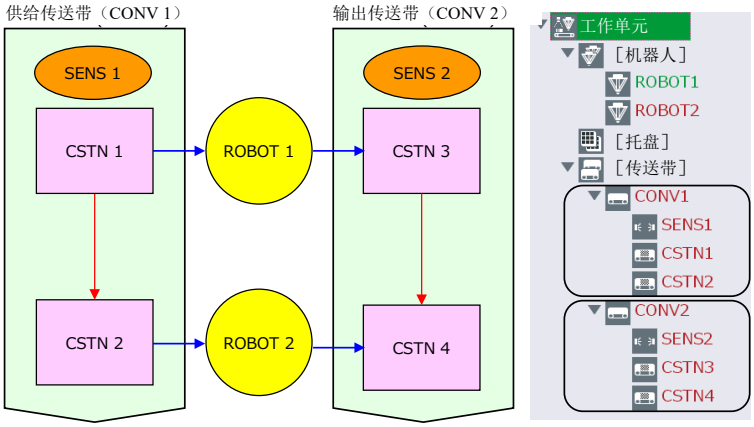


图 8.2(b)传送带构成示例

直方图工具的插入

在视觉程序中插入直方图工具。

1 在编辑画面的树状图中，选择［GPM Locator Tool 1］，按下按钮。
则显示以下画面。

2　选［Hitogram Tool］，按下 F4［下一个>］。

3　在［名称］中输入工具名（例：Histogram Tool 1），按下 F4［确定］。

8

在树状图上显示［Histogram Tool 1］。

4 选择树状图上的［Histogram Tool 1］。

5 按下［设定］按钮，示教［测量范围］。

6 按下 F5［结束编辑］。

7 按下［>（下一页）］，显示下一功能键，再按下 F5［保存］。

条件判断工具设定

设定条件判断工具，如果是有孔的工具，则使模型 ID 为 2。条件判断工具的插入方法，请参照“8.1 根据传送带上的位置，由不同的机器人拾取”。

假设有孔的情况的标准偏差为 50 左右，无孔的情况的标准偏差是 10 左右，则认为将中间的 30 左右作为阈值较为妥当。将条件判断工具设定为“如果直方图的标准偏差为 30 以上，则对模型 ID 加 1”。

1 在［设定值］中，对［值 1］选择［Histogram Tool 1］的［标准偏差］。

2 在［设定条件］中进行设定，使［条件 1］成为“如果值 1 比 30 大”。

3 在［设定处理］中选择［将随后的数据加到模型 ID］，值指定为 1。

负载平衡设定

在传送带的负载平衡的设定中进行设定，分别由上游侧机器人 100%处理模型 ID1 的工件，下游侧机器人 100%处理模型 ID2 的工件。

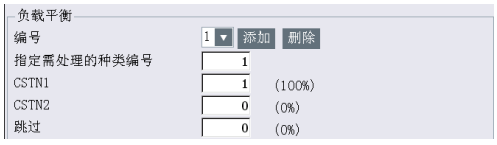
打开传送带的设置画面，进行负载平衡的设定。这里只显示供给传送带。

1　打开传送带的设置画面。显示如下的对话框。

2　将［负载平衡］的左侧选框设定为［进度一定］或［保持个数比例］，使右侧的选框为［针对每个模型 ID］。

3　在［指定需处理的模型 ID］中指定 1 和 2。

4　负载平衡设定为由上游侧机器人 100%处理模型 ID1，下游侧机器人 100%处理模型 ID2。



8

8.3 将工件多层堆起放入箱中

将工件多层堆起放入箱中的情形。
作为具体例，介绍在 1 段可放 4 个工件的箱子里，将工件 2 段堆起放入的情况。这里介绍在传送带上使用托盘的情况的例子，也可在固定工作站上使用。

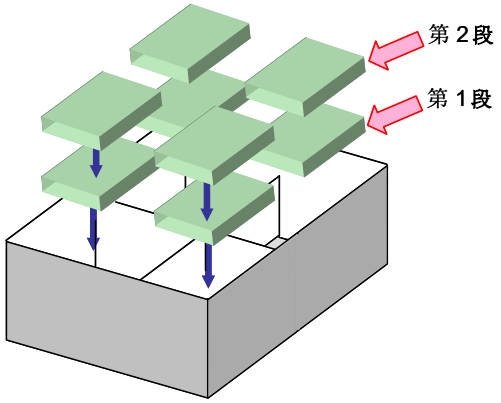


图 8.3(a)将每段可放 4 个工件的箱子堆起 2 段的例子

这种情况，在放置于下段之前，不能在上段放置工件。这种情况，设定为在使用托盘的“层”功能将下面的单元填满之前，上面的单元中不放置物品。

传送带如下图般定义。与“2.4 系统构成例”中介绍的构成例 1 相同。

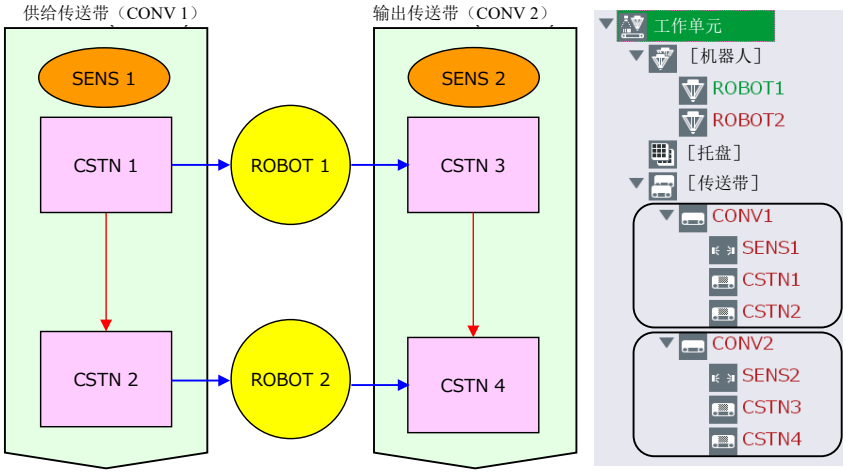


图 8.3(b)传送带构成示例

托盘设定

托盘的设定如下图所示。

层编号为 1 的单元 1～4 是第 1 段，层编号为 2 的单元的单元 5～8 是第 2 段。进行如此设定，将工件装入托盘时，在第 1 段的单元没有全部填满之前，第 2 段的单元 5～8 不会成为机器人的作业对象。
而同一段的单元，全部被追加至相同层。这样，相同段的单元，从下游侧的单元开始依次放置工件。

传送带设定

传送带的设定如下图所示。这里只显示输出传送带。不需要特别对供给传送带施加变更。

首先，选择［托盘］中设定的托盘，由于托盘内要装入工件，请在［初始状态］选择［空］。并且，通过使负载平衡无效，机器人在保持层的设定的同时，在单元中放置尽量多的工件。

注释

这里，使负载平衡无效，但也可以使其有效。这种情况，在保持层的设定的同时，分配单元的信息，从而维持负载平衡。

8.4 从箱子一端依次放置工件

在没有隔离的箱子中，从一端依次放置工件的情况。这里介绍在传送带上使用托盘的情况的例子，在固定工作站上的使用也同样。

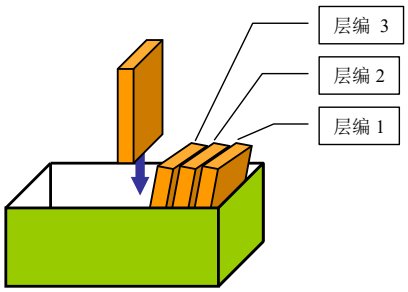


图 8.4(a) 从箱子一端依次放置工件的例子

这种情况，即使是只有 1 段的情况，也设定为：在使用托盘的“层”功能将一端的单元填满之前，旁边的单元中不放置工件。

托盘设定例

此时，托盘的各层设定如下例所示。

TRAY1

基准单元 未示教

层编号 1

单元	X(mm)	Y(mm)	Z(mm)	R(deg)	种类编号
1	80.0	20.0	-5.0	0.0	1

层编号 2

单元	X(mm)	Y(mm)	Z(mm)	R(deg)	种类编号
2	60.0	20.0	-5.0	0.0	1

层编号 3

单元	X(mm)	Y(mm)	Z(mm)	R(deg)	种类编号
3	40.0	20.0	-5.0	0.0	1

从单元 1 至单元 3，分别被追加至不同的层。单元 1 是最边缘端的单元。进行如此设定，通过在传送带上［初始状态］中选择［空］，一定以单元 1→单元 2→单元 3 的顺序，将工件放置于托盘中。

传送带的设定例

进行与“8.3 将工件多层堆起放入箱中”相同的设定。

具体请参照“8.3 将工件多层堆起放入箱中”。

8.5 在箱子中以各种方向放置

在箱子中放置工件时的方向，可能根据单元不同而不同。
例如，下图例中，单元 2 和单元 4 的方向为 90 度，与其他单元不同。

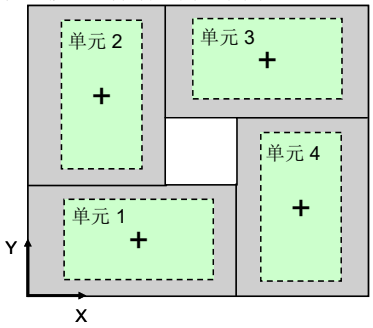


图 8.5(a) 根据单元不同，放在箱中的工件朝向不同的例子

这种情况，可以仅通过托盘的设定进行对应。

托盘设定

以下表示托盘的设定例。

使单元 1 的角度为 0 度，其他单元则以与单元 1 的相对角度表示。单元 2，单元 4 的角度是 90 度。

而机器人的位置示教与通常的托盘的情况相同，仅在单元 1 进行。由于在托盘的设定中，显示单元 2 和 4 有 90 度的不同，因此在放置单元 2 和 4 时，机器人自动改变 90 度方向。

8.6 根据工件种类不同，切换放置位置

在将拾取的工件装入箱子的系统中，有 2 种工件，根据种类不同，规定箱子中的放置位置的情况。这里介绍在传送带上使用托盘的情况的例子，在固定工作站上的使用也同样。

例如下图所示，有大工件和小工件 2 种，箱子里也同样有放置大工件的单元和放置小工件的单元的 2 种单元。这种情况下，根据拾取的工件种类，限定放置工件的单元。

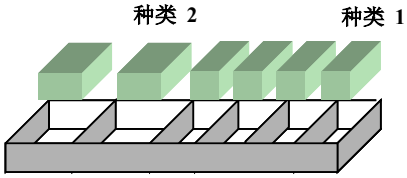


图 8.6(a)根据工件种类(大小)切换放置位置的例子

这种情况，托盘的设定中，对单元设定模型 ID，并在调用 PKCSGETQUE 时，指定需要分配的单元的模型 ID。

这里，在供给传送带中，对小工件的模型 ID 设定为 1，对大工件的模型 ID 设定为 2。

托盘设定

以下表示托盘的设定例。

单元 1～4 是小工件用的单元，单元 5～6 是大工件用的单元。
根据供给传送带上的工件的模型 ID，使单元的模型 ID 为 1 和 2。模型 ID1 的工件放置于模型 ID1 的单元中，模型 ID2 的工件放置于模型 ID2 的单元中。并不一定要使工件的模型 ID 和对应的单元的模型 ID 相同，但是这样能令操作更为方便。

所有的单元被追加至相同层。如果在不同的层中追加单元，则会发生当需要模型 ID1 的单元时，层设定中规定的模型 ID1 的单元无法分配的情况。

传送带设定

表示传送带的设定例。

供给传送带

输出传送带

供给传送带侧，在［负载平衡］的右侧选框中指定［所有模型 ID 共通］，由于负载平衡设定为 1:1，2 种工件混合传送，各机器人无论种类，分别处理 50%。

输出传送带侧，使负载平衡无效。这样，各机器人等待拾取的工件的模型 ID 相对应的单元的到来，尽量放置于更多的位置。

程序的变更例

这种情况，必须对进行机器人的放置动作的程序进行自定义。主程序以及进行拾取动作的程序，不需要进行变更。

对“6.9.2.3 进行放置动作的程序”中介绍的程序的进行以下变更。主要的变更有下划线标出。

DROP1.TP

```
 1:   UTOOL_NUM=0
 2:   UFRAME_NUM=0
 3:
 4:   LBL[100]
 5:   STOP_TRACKING
 6:   R[5:模型 ID]=VR[1].MODELID
 7:   CALL PKCSGETQUE(R[12:CSTN2], 1, 100, 2, 2, R[5:模型 ID])
 8:   IF R[2:GETQ_状态]=0,JMP LBL[200]
 9:   IF R[1:周期停止]=1,JMP LBL[900]
10:   JMP LBL[100]
11:
12:   LBL[200]
13: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
14: L PR[11] 4000mm/sec CNT3 VOFFSET,VR[2]
15:   CALL VACUUM_OFF
16: L PR[11] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[12]
17:   CALL PKCSACKQUE(传送带工作站 ID=R[12:CSTN2],成功)
18:
19:   LBL[900]
 [End]
```

在第 6 行，将从供给传送带拾取的工件的模型 ID 存储于 R［5］，在第 7 行的 PKCSGETQUE 中添加 1 个参数，将 R［5］作为模型 ID 进行指定。这样，R［5］中指定的模型 ID 的单元一定被分配。另外，若以 PKCSGETQUE 作为参数 6 指定模型 ID，则省略参数的说明。

⚠ 注意

如果没有拾取工件所对应的单元，则无法放置机器人抓取的工件。要正常稼动这种系统，必须使供给传送带上传送的工件 1，2 的比例，和输出传送带上放置传送的工件 1，2 的单元的比例一致。

8.7 根据种类，放置于其他的输出传送带

有多条输出传送带，根据从供给传送带拾取的工件的种类，进行放置于哪个输出传送带的切换的情况。

具体例如下图所示，认为有 1 条供给传送带，2 条输出传送带，各传送带上分别有 1 个传送带工作站的情况。

从供给传送带上传送的工件，有模型 ID1，2 共 2 种，模型 ID1 的工件放置于输出传送带 1，模型 ID2 的工件，放置于输出传送带 2。

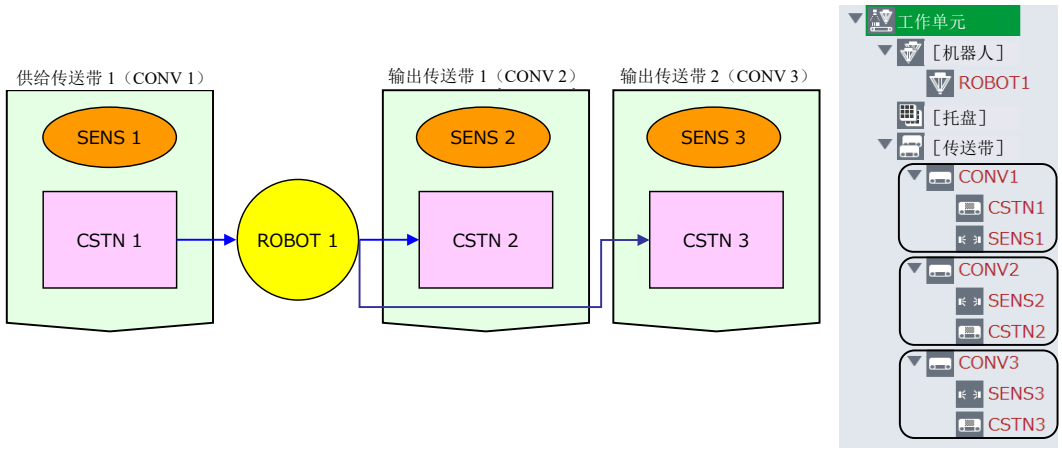


图 8.7(a) 根据工件种类(大小)放置于其他输出传送带的例子

传送带设定

设定为从供给传送带可拾取所有工件，与模型 ID 无关。通过使负载平衡无效，成为这种设定。2 根输出传送带的情况，同样也使负载平衡无效。

程序的变更例

这种情况，必须对机器人的主程序进行自定义。此外，还另外需要在第 2 输出传送带上进行放置动作的程序 DROP2.TP。进行拾取动作的程序，不需要特别进行变更。

对“6.9.2.1 主程序”中介绍的主程序 MAIN1.TP 进行以下变更。主要的变更有下划线标出。

MAIN1.TP

```
 1:   R[1:周期停止]=0
 2:   CALL PKCSGETID('CONV1',工作站 ID Reg=11)
 3:   CALL PKCSGETID('CONV2',工作站 ID Reg=12)
 4:   CALL PKCSGETID('CONV3',工作站 ID Reg=13)
 5: L P[1] 1000mm/sec FINE
 6:   CALL PKWCSTART
 7:
 8:   LBL[100]
 9:   CALL PICK1
10:   IF R[1:周期停止]=1,JMP LBL[900]
11:
12:   R[5:模型 ID]=VR[1].MODELID
13:   IF R[5:模型 ID]=2,JMP LBL[200]
14:   CALL DROP1
15:   JMP LBL[300]
16:   LBL[200]
17:   CALL DROP2
18:   LBL[300]
19:   IF R[1:周期停止]=1,JMP LBL[900]
20:   JMP LBL[100]
21:
```

```
 22:    LBL[900]
 23:    CALL PKWCEND
[End]
```

在第 12 行，将从供给传送带拾取的工件的模型 ID 存储于 R［5］。R［5］的值是 1 时，调出输出传送带 1 用的放置动作程序 DROP1.TP，是 2 时，调出输出传送带 2 用的放置动作程序 DROP2.TP。

表示在输出传送带 2 上进行放置动作的程序 DROP2.TP 的例。

DROP2.TP

```
  1:    UTOOL_NUM=0
  2:    UFRAME_NUM=0
  3:
  4:    LBL[100]
  5:    STOP_TRACKING
  6:    CALL PKCSGETQUE(传送带工作站 ID=R[13:CSTN3],连续的标记=1,超时值(ms)=100,补偿用 VR=3,
        状态 Reg=2)
  7:    IF R[2:GETQ_状态]=0,JMP LBL[200]
  8:    IF R[1:周期停止]=1,JMP LBL[900]
  9:    JMP LBL[100]
 10:
 11:    LBL[200]
 12: L PR[20] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[12]
 13: L PR[20] 4000mm/sec CNT3 VOFFSET,VR[3]
 14:    CALL VACUUM_OFF
 15: L PR[20] 4000mm/sec CNT100 VOFFSET,VR[3] Tool_Offset,PR[12]
 16:    CALL PKCSACKQUE(传送带工作站 ID=R[13:CSTN3],成功)
 17:
 18:    LBL[900]
[End]
```

基本与“6.9.2.3 进行放置动作的程序”中介绍的程序 DROP1.TP 相同，但是机器人的示教点的位置寄存器的编号不同。各传送带上，必须使用不同编号的位置寄存器。这里，将 PR［20］作为输出传送带 2 用的放置位置使用。

8.8 从 2 个以上的供给传送带上拾取

有多条供给传送带，从各条拾取工件，放置于输出传送带的情况。
即“2.4 系统构成例”中介绍的构成例 8 的情况。

具体例如下图所示，认为有 2 条供给传送带，1 条输出传送带，各传送带上分别有 1 个传送带工作站的情况。

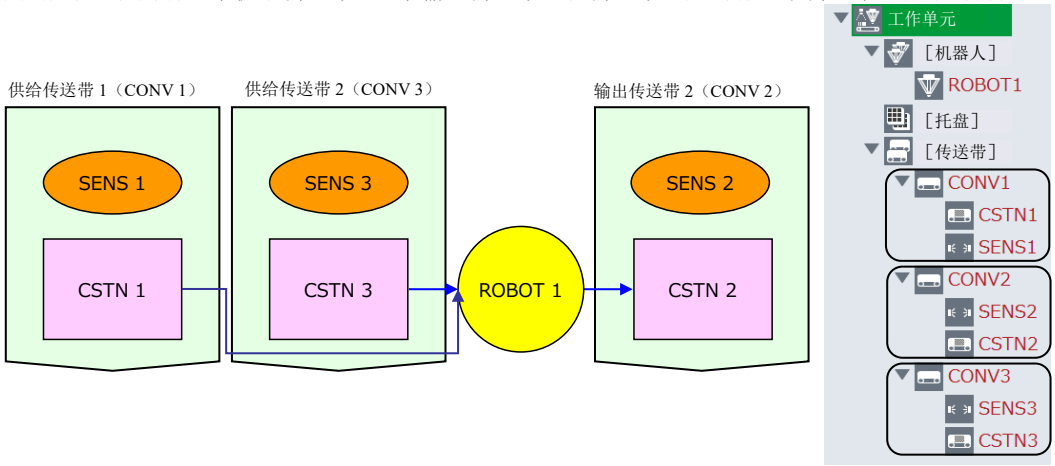


图 8.8(a) 从多个供给传送带取出工件的例子

机器人从各传送带交替拾取工件。如果工件的取得失败，则至其他传送带拾取工件。

主程序的变更例

这种情况，必须对机器人的主程序以及进行拾取动作的程序进行自定义。此外，还需要另外准备从第 2 根供给传送带进行拾取动作的程序。

以下是对“6.9.2.1 主程序”中介绍的程序 MAIN1.TP 和“6.9.2.2 进行拾取动作的程序”中介绍的程序 PICK1.TP 添加变更的例子。主要的变更有下划线标出。

MAIN1.TP

```
 1:   R[1:周期停止]=0
 2:   R[3:区域]=1
 3:   R[5:传送带选择]=1
 4:   CALL PKCSGETID('CONV1',工作站 ID Reg=11)
 5:   CALL PKCSGETID('CONV2',工作站 ID Reg=12)
 6:   CALL PKCSGETID('CONV3',工作站 ID Reg=13)
 7: L P[1] 1000mm/sec FINE
 8:   CALL PKWCSTART
 9:
10:   LBL[100]
11:   STOP_TRACKING
12:   IF R[5:传送带选择]=2,JMP LBL[200]
13:   CALL PICK1
14:   JMP LBL[300]
15:   LBL[200]
16:   CALL PICK2
17:   LBL[300]
18:   IF R[1:周期停止]=1,JMP LBL[900]
19:
20:   R[5:传送带选择]=3-R[5:传送带选择]
21:
22:   IF R[3:区域]<=R[4:区域数量] OR R[2:GETQ 状态]<>0,JMP LBL[100]
23:   R[3:区域]=1
24:
```

```
 25:    CALL DROP1
 26:    IF R[1:周期停止]=1,JMP LBL[900]
 27:    JMP LBL[100]
 28:
 29:    LBL[900]
 30:    CALL PKWCEND
[End]
```

传送带的切换使用 R［5］。R［5］是 1 时，从供给传送带 1 拾取，R［5］是 2 时，从供给传送带 2 拾取。

注释
如果两条传送带上都有待取的工件状态，优先从哪个传送带取工件，由应用程序决定，无法一概而论。这里编写了以不同于前一次取工件的传送带的传送带优先的程序。

PICK1.TP 是进行供给传送带 1 用的拾取动作的追踪程序，PICK2.TP 是进行供给传送带 2 用拾取动作的追踪程序。

PICK1.TP

```
  1:    UTOOL_NUM=1
  2:    UFRAME_NUM=0
  3:
  4:    CALL PKCSGETQUE(传送带工作站 ID=R[11:CSTN1],连续的标记=R[3:区域],超时值(ms)=100,补偿用
VR=1,
        状态 Reg=2)
  5:    IF R[2:GETQ_状态]<>0,JMP LBL[900]
  6:
  7: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
  8: L PR[R[3:区域]] 4000mm/sec CNT3 VOFFSET,VR[1]
  9:    CALL VACUUM_ON
 10: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 11:    CALL PKCSACKQUE(传送带工作站 ID=R[11:CSTN1],成功)
 12:    R[3:区域]=R[3:区域]+1
 13:
 14:    LBL[900]
[End]
```

没有对原来的 PICK1.TP 添加行，而是删除了若干行。与原来的 PICK1.TP 的不同之处在于，当第 4 行的 PKCSGETQUE 因超时而被跳过时，即此时应该被拾取的工件不在范围内，则立即结束 PICK1.TP。由此，接着调出 PICK2.TP，进行从传送带 2 的拾取。

PICK2.TP

```
  1:    UTOOL_NUM=1
  2:    UFRAME_NUM=0
  3:
  4:    CALL PKCSGETQUE(传送带工作站 ID=R[13:CSTN1],连续的标记=R[3:区域],超时值(ms)=100,补偿用
VR=3,
        状态 Reg=2)
  5:    IF R[2:GETQ_状态]<>0,JMP LBL[900]
  6:
  7:    R[6:位置 PR 编号]=R[3:区域]+20
  8: L PR[R[6:位置 PR 编号]] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[10]
  9: L PR[R[6:位置 PR 编号]] 4000mm/sec CNT3 VOFFSET,VR[2]
 10:    VACUUM_ON
 11: L PR[R[6:位置 PR 编号]] 4000mm/sec CNT100 VOFFSET,VR[2] Tool_Offset,PR[10]
 12:    CALL PKCSGETQUE(传送带工作站 ID=R[13:CSTN1],成功)
 13:    R[3:区域]=R[3:区域]+1
 14:
 15:    LBL[900]
[End]
```

基本与上面变更的 PICK1.TP 相同。与 PICK1.TP 的不同之处在于，由于机器人的示教点在供给传送带 1 和 2 不同，因此在位置寄存器的间接指定中，不直接使用 R［3］，而使用 R［6］。R［6］中是对 R［3］加 20 的值，这是假定作为供给传送带 2 的示教点而使用 PR［21~29］的情况。

8.9 拾取失败时发出通知

机器人拾取工件后，确认机器人实际是否拾取时，发现拾取失败的情况。
这种情况，工件的位置与用传感器检出时不同的可能性较高，因此用 PKCSACKQUE 指定“删除”，使以后的机器人不拾取该工件。
需要变更的只有“6.9.2.2 进行拾取动作的程序”中介绍的程序 PICK1.TP。
以下是示例。主要的变更有下划线标出。

PICK1.TP

```
  1:   UTOOL_NUM =0
  2:   UFRAME_NUM =0
  3:   R[3:区域]=1
  4:
  5:   LBL[100]
  6:   STOP_TRACKING
  7:   CALL PKCSGETQUE(传送带工作站 ID=R[11:CSTN1],连续的标记=R[3:区域],超时值(ms)=100,补偿用
VR=1,
       状态 Reg=2)
  8:   IF R[2:GETQ_状态]=0,JMP LBL[200]
  9:   IF R[1:周期停止]=1,JMP LBL[900]
 10:   JMP LBL[100]
 11:
 12:   LBL[200]
 13: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 14: L PR[R[3:区域]] 4000mm/sec CNT3 VOFFSET,VR[1]
 15:   CALL VACUUM_ON
 16: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
 17:   IF RI[R[3:区域]]=ON,JMP LBL[300]
 18:   CALL PKCSACKQUE(传送带工作站 ID=R[11:CSTN1],删除)
 19:   JMP LBL[100]
 20:
 21:   LBL[300]
 22:   CALL PKCSACKQUE(传送带工作站 ID=R[11:CSTN1],成功)
 23:   R[3:区域]=R[3:区域]+1
 24:   IF R[3:区域]<=R[4:区域数量],JMP LBL[100]
 25:   LBL[900]
[End]
```

在第 17 行进行工件的吸着确认。根据抓手区域的编号确认 RI，如果 RI 为 ON，则吸着成功，如果 RI 为 OFF，则吸着失败。如果吸着成功，则跳至 LBL［300］，正常结束。如果吸着失败，则在第 18 行以 PKCSACKQUE 报告“删除”（抓取失败），跳至 LBL［100］，去取下一个工件。

8.10 仅在单元都已满的情况下，取出托盘

由传送带上的最下游的机器人取出托盘本体时，用于确认所有单元是否都填满工件的方法。

作为具体例，认为托盘上有 4 个单元，仅当所有的单元都被工件填满时，最下游机器人取出托盘。最下游机器人仅进行取出托盘的作业。
此外，最下游机器人与其他机器人的机械手不同。

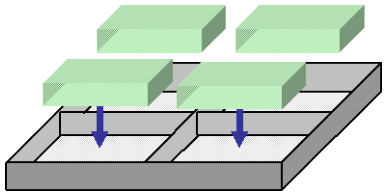


图 8.10(a) 只有所有单元都被填满时取出托盘的例子

这种情况，需要在托盘设定和负载平衡上下功夫。

这种情况，在托盘内进行 1 个份外的单元定义。最后的单元追加至比其他单元高 1 段的层，如果其他的单元不全部填满，则不分配至最后的单元。
此外，使最后的单元的模型 ID 与其他单元不同，仅分配至最下游的机器人。

托盘设定

托盘设定如下所示。

基准单元 X 20.0 Y 10.0 Z -5.0 R 0.0
层编号 1

单元	X(mm)	Y(mm)	Z(mm)	R(deg)	种类编号
1	20.0	10.0	-5.0	0.0	1
2	20.0	30.0	-5.0	0.0	1
3	50.0	30.0	-5.0	0.0	1
4	50.0	10.0	-5.0	0.0	1

层编号 2

单元	X(mm)	Y(mm)	Z(mm)	R(deg)	种类编号
5	20.0	10.0	-5.0	0.0	1

定义实际的托盘放置位置为 4 个，单元为 5 个。对单元 5 做以下设定。

- 追加至第 2 层。
- 使模型 ID 的值与其他单元不同。
- 使位置（X, Y, Z, R）与基准单元相同。

使单元的位置（X, Y, Z, R）与基准单元相同，是为了能够在进行其他单元的示教的同时，示教对于最后的单元的机器人的动作。

注释

通常，仅单元 1 进行机器人位置的示教。机器人对于各单元的位置可以将基准单元作为基准，根据托盘的设定自动计算。但是，由于对于单元 5 的作业（取出托盘本体）与对于其他单元的作业（将工件放置于单元）不同，因此需要能够另行示教机器人的位置。这种情况，为了方便，使单元 5 的 X, Y, Z, R 与基准单元为相同值。这样，由于单元 5 和基准单元为相同位置，以基准单元为基准计算所得的补偿量为 0，因此与基准单元相同，可自动示教相对单元 5 的机器人的位置。

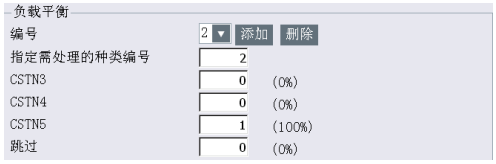
传送带设定

传送带的负载平衡设定如下所示。

该传送带上有 3 个传送带工作站，假设使用最初的 2 台机器人进行将工件放置于托盘内单元的作业，第 3 台机器人进行将托盘本体取出的作业。

最初的 2 个传送带工作站只需要分配模型 ID1 的单元，因此模型 ID1 的负载平衡如上图所示（第 1 台和第 2 台为各 50%，第 3 台为 0%）。

另一方面，最后的传送带工作站只需要分配模型 ID2 的单元，因此模型 ID2 的负载平衡如下图所示。



最后的传送带工作站中的模型 ID2 的负载平衡为 100%，但是托盘的［初始状态］为［空］，因此只要第 1 层的单元（单元 1～4）残留有 1 个，就不会分配第 2 层的单元（单元 5），所有最后的机器人只能接收单元 1～4 全部填满的托盘的信息。

基准位置设定和机器人的位置示教

在传感器任务的画面进行基准位置的设定后，当示教各机器人的位置时，仅最下游的机器人不是示教单元 1 的位置，而是示教托盘的取出位置。

> **⚠ 注意**
> **这里，以最下游的机器人仅取出托盘，在托盘的设定中，X, Y, Z, R 的基准单元和单元 5 一致为前提。**

程序的变更

程序不需要特别进行变更。
最下游的机器人也与其他机器人一样，通过 PKCSGETQUE 取得单元 5 的信息，进行托盘的取出。

8.11 系统重启时处理停止前已检出的工件

对因紧急停止等停止系统时，从主程序的开头重新开始时，系统停止前已检出未处理的传送带上的工件的处理方法进行说明。

为了删除已检出的工件信息，请通过调用 PKWCCHECK 和 PKSNSTART 来代替调用 PKWCSTART。不调用 PKCSCLRQUE。但是，系统停止前 PKCSGETQUE 被执行，且 PKCSACKQUE 的执行尚未结束时，如果从主程序的开头重新开始，在未执行的 PKCSACKQUE 被执行前，执行 PKCSGETQUE，会发生 IRPK-051“GETQUE 之后未执行 ACKQUE”的报警。为了避免发生此报警，执行 PKCSCLRACK。如果执行 PKCSCLRACK，在系统停止前通过 PKCSGETQUE 取得的工件信息将会丢失。通过 PKCSGETQUE 已取得工件信息时，举动与指定“4：删除”，PKCSACKQUE 被调用时相同。

主程序的变更例

对“6.9.2.1 主程序”中说明的程序 MAIN1.TP 追加更改的示例如下所示。主要的变更有下划线标出。

MAIN1.TP

```
  1:   R[1:CYCLE_STOP]=0
  2:   CALL PKCSGETID('CONV1', CStn ID Reg=11)
  3:   CALL PKCSGETID('CONV2', CStn ID Reg=12)
  4:
  5:   UTOOL_NUM =1
  6:   UFRAME_NUM =0
  7:   J P[1:Perch Pos] 50% FINE
  8:
  9:   CALL PKWCCHECK
 10:   CALL PKCSCLRACK(CStn ID=R[11:CSTN1])
 11:   CALL PKCSCLRACK(CStn ID=R[12:CSTN2])
 12:   CALL PKSNSTART
 13:
 14:   LBL[100]
 15:   CALL PICK1
 16:   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 17:   CALL DROP1
 18:   IF R[1:CYCLE_STOP]=1,JMP LBL[900]
 19:   JMP LBL[100]
 20:
 21:   LBL[900]
 22:   CALL PKWCEND
[End]
```

9 其他有用功能

本章介绍在使用 *i*RPickTool 的系统中有用的功能。通过灵活运用这些功能，实现构造更为稳定，适应性强的系统。

9.1 使用方案变更工作单元

本节对方案的设定进行说明。

> ⚠ **注意**
> **使用方案时，需要符合通过 ROS 以太网接口包（RIPE）连接的所有控制装置的软件版本。**

使用方案时，选定一个方案（称为活动方案。）执行与该内容相对应的工作单元。
方案的切换，也就是工作单元的更换，只能在非生产过程中，也就是说，机器人程序停止时进行。如果开始方案的切换，*i*RPickTool 更换为新使用中的方案所对应的工作单元的内容。

方案具有以下功能。
1 可根据 PLC 进行变更。
2 可导出至存储卡或 USB 设备。
3 可从存储卡或 USB 设备导入。

在工作单元中包括的控制装置中，所有的方案仅存储于 1 个控制装置中。存储这些所有方案的控制装置被称为“方案管理”。

可以从示教器时保存方案。
在示教器时的［文件］画面选择［全部］保存，将方案通用设定备份成 IRPICKTOOL_CFG.ZIP 文件，各方案备份成 IRPK_RECIPEx.ZIP（x 表示方案编号）文件。
IRPICKTOOL_CFG.ZIP 文件、IRPK_RECIPEx.ZIP 文件也可读入到机器人控制装置。读入时展开的内容，放置于 \IRPICKTOOL\控制装置内部的目录。

9.1.1 方案设定

方案设定，在工作单元的设定（系统设置）页中进行。工作单元的设定详情，请参阅“6.1 工作单元设定”或者是“7.1 工作单元设定”。
［工作单元］的设定页中，F3 上有名为［方案］的功能键。该页也可以进行“方案管理”的设定。
所谓方案管理，是指存储方案的控制装置。如果工作单元中有多个控制装置，标准情况下，方案通过 RIPE 存储于机器人间通信的网络内主控制装置中。有关机器人间通信的详细信息，请参照“5.1.4 机器人间的通信设定”。

1 从［工作单元］的设定画面按下 F3［方案］。

2 显示方案的设定画面。
如果已有方案，会在［方案列表］中显示方案名称。

［操作方案 DI］
如果从外部的 PLC 使这里设定的 DI 为 ON，则进行方案的操作。详细信息，请参照“9.1.3 利用 PLC 的方案操作”。

［方案错误 DO］
由于某原因导致方案操作失败时，会使这里设定的 DO 为 ON。

［方案名称］
可以直接编辑这里显示的名称。关于编辑方法请参阅“9.1.2 方案的详细设定”中的“方案名称的变更”。

9.1.2 方案的详细设定

进行方案的详细设定。

方案的生成

生成方案。
如果生成方案，所有的对象（工作单元、抓手、托盘、机器人、传送带、传感器、传送带工作站、固定工作站）中，现在设定的 iRPickTool 的数据，会被作为 XML 文件保存于名为 FR:\IRPICKTOOL\IRPK_RECIPEx\（x 是在［方案列表］中显示的方案 ID）的控制装置内部目录下。该 XML 文件中也包括追踪坐标系、固定工作站的用户坐标系。此外，使用 Plug & Play 时，原位置、基准位置、接近位置等需要通过 *i*RPickTool 示教机器人的位置。

在方案的设定画面按下 F1［方案操作］，选择［新建］。

以名称 RecipeX（X 是被分配的方案 ID）生成的新方案。
正在新建方案时，画面中显示“正在新建方案”。
如果正常创建方案，则显示确认信息。新建的方案中，包括有以树状图设定的现在的所有的 *i*RPickTool 数据。
如下图在［方案列表］中显示创建的方案。

方案名称的变更

更改方案的名称。

1 在［方案列表］处选择要更改名称的方案。
2 方案的当前名称显示在下方的［编辑方案］处，直接编辑。

3 选择含当前名称的文本框，更改名称。
4 按下示教器上的［ENTER］键，更改方案名称，并在［方案列表］中显示新的方案名称。
更改名称后，务必参考后述的“方案的保存”，进行方案的保存。
方案名称可使用最多 20 个半角字符。
类似“iRPickTool Recipe 2”的方案名称，可插入空白。包含“&”、“'”、“"”、“/”或“\”的名称不能被指定。

方案的删除

删除方案。

如果删除方案，实际是从 FR:\IRPICKTOOL\IRPK_RECIPEx 目录（x 是在［方案列表］中显示的方案 ID］删除 XML 文件内的方案相关的所有数据。其方案名称也从［方案列表］中删除。

1 在［方案列表］处选择要删除的方案。
高亮显示选中的方案。
2 按下 F1［方案操作］，选择［删除］。
则显示确认信息。
3 按下 F4［确定］。

4 在删除方案过程中，画面上可能显示“正在删除方案”。
如果正常删除方案，显示以下消息。
5 按下 F4［确定］。

方案的保存

保存方案。

如果保存方案，则现在的 *i*RPickTool 的数据保存于使用中的方案。使用中的方案，始终在［方案列表］中显示，［状态］显示为［有效］。
在创建方案后，或最后保存方案后，对 *i*RPickTool 的数据实施某种变更的情况下进行方案的保存。

> **⚠注意**
> **非使用中的方案，也可能在［方案列表］中被加亮。但是，在保存方案时，无论是否加亮显示，一定保存于使用中的方案。**
> **此外，*i*RPickTool 的数据的全套被完全保存。**

1 按下 F1［方案操作］，选择［保存］。
则显示确认信息。
2 按下 F4［确定］。

3 方案保存过程中，画面可能显示“正在保存方案”。
如果正常保存方案，显示以下消息。

4 按下 F4［确定］。

方案的恢复

恢复方案。

如果恢复方案，则恢复最后保存的使用中的方案的数据，成为现在的方案。例如：在最后保存方案后，对 *i*RPickTool 的数据实施了某种变更，但废弃该变更，返回最后保存的版本（也就是说，需要取消至此的变更）的情况下，进行该恢复。

> **注意**
> **非使用中的方案，也可能在［方案列表］中被加亮。但是，在恢复方案的情况下，无论是否加亮显示，一定从使用中的方案恢复数据。此外，*i*RPickTool 的数据的全套被完全恢复。**

1 按下 F1［方案操作］，选择［恢复］。
则显示确认信息。
2 按下 F4［确定］。

3 方案恢复过程中，画面上可能显示［正在恢复方案］。
如果正常恢复方案，显示以下消息。

4　按下 F4［确定］。

方案的导出

导出方案。

如果导出方案，则存储于控制装置的 FR:IRPICKTOOL 的方案，将被复制至存储卡或 USB 存储器等外部存储设备中。导出目的地的装置，可从［文件］画面进行变更。按下 F5［工具］，选择［1 切换设备］，选择准备的装置［4 存储卡（MC:）］、［7 USB 盘（UD1:）］、［1 TP 上的 USB（UT1:）］等。装置的准备结束后，执行以下步骤。

1　按下 F1［方案操作］，选择［导出］。
　则显示确认信息。
2　按下 F4［确定］。

在方案导出中，画面上可能显示“正在导出方案”。
如果正常导出方案，显示以下消息。

3 按下 F4［确定］。

方案的导入

导入方案。

如果导入方案，则存储于存储卡或 USB 存储器等外部存储设备中的方案，将被复制至控制装置的 FR:IRPICKTOOL 下。可从［文件］设置画面变更导入源的装置。按下 F5［工具］，选择［1 切换设备］，选择存储有方案的装置［4 存储卡（MC:）］、［7 USB 盘（UD1:）］、［1 TP 上的 USB（UT1:）］等。装置的准备结束后，执行以下步骤。

1 按下 F1［方案操作］，选择［导入］。
则显示确认信息。
2 按下 F4［确定］。

在方案导入中，画面上可能显示“正在导入方案”。
如果正常导入方案，显示以下消息。

3 按下 F4［确定］。

方案的切换

从现在的方案，切换至不同的方案。

在需要从现在的产品生产，切换至使用以前保存的不同数据生产不同产品时，进行方案的切换。例如，需要从 6 英寸的披萨生产切换至 18 英寸披萨生产的情况。

1 在［方案列表］中，点击需要切换使用的其他方案。该方案的行加亮。
2 按下 F2［方案变更］。
则显示以下确认信息。
3 按下 F4［确定］。

在方案切换过程中，画面上可能显示［正在切换方案］。
如果正常切换方案，显示以下消息。

4　按下 F4［确定］。

方案操作的规则设定

依照若干规则进行方案操作。
规则的设定，在按下方案主画面的 F3［设置］后所显示的画面上进行。
现在，只能进行有关方案导入操作（恢复、导入、切换）的设定。

要切换方案，或读入保存的方案时，基本使用切换对象的方案，或保存的方案的所有设定。

但是，有些设定可能会有需要直接使用现在使用的设定的情况。例如，在现在的工作单元，再次设定用于改善系统精度的追踪坐标系、基准位置。在很多情况下，为了维持高精度，在切换或导入方案后，应使用现在的设定，而不使用切换目的地的方案或以前保存的方案的设定。这种情况，可进行该规则的设定，对于特定项目，无需改变现在的设定，可直接使用。

例如，并非使用保存于现在要导入的方案的追踪坐标系，而是要使用现在的设定的情况，则取消对“传送带工作站: 追踪坐标系，编码器倍率”的勾选。同样，对于各项目中，恢复、导入、切换的各方案操作，需要使用现在的设定时，取消勾选。各项目的规则设定结束后，按下 F5［关闭］，关闭设置画面。

注释
1 无法对该画面中未记载的设定项目进行规则设定。
2 有关方案的保存，保存所有的设定。无法进行不保存特定项目的设定。
3 有关该画面中记载的项目，如果任何方案中都必须使用相同设定，在事先取消对象项目勾选的情况下，为了始终使用最新设定，结果在任何方案中，都可以使用相同的设定。
4 标准情况下，该画面显示的所有项目，不导入方案的值，而使用现在的设定值。北美专用设定时，在标准中勾选该画面显示的所有项目。

9.1.3 利用 PLC 的方案操作

可以从 PLC 操作方案。该操作按以下步骤进行。
PLC 只能利用方案管理控制装置进行操作方案。

1 在方案管理机器人控制装置的［方案］的设置画面，设定［操作方案 DI］的 DI 编号。
2 从 PLC 使通过 1 设的 DI 为 ON。
3 如果使 DI 为 ON，则通过机器人控制装置检出，运行用于与 PLC 进行信号交换的程序。备有名为 PK_PLC_RECIPE_OPERATION.TP 的标准程序。
4 使用 KAREL 程序可操作方案。KAREL 程序的参数在标准程序中通过组输入信号提供。
5 如果在方案的切换中发生错误，则以［方案错误 DO］设定的 DO 变为 ON。

如果方案变更已执行，则不允许进行通过 PLC 的方案变更。例如，在［方案］的设置画面执行某方案操作过程中，从 PLC 使［操作方案 DI］的 DI 为 ON 时，不进行方案的变更，因此［方案错误 DO］的 DO 变为 ON。

注释
通过 PLC 的方案的操作，有时需要 5 秒以上。

9.1.3.1 PLC 方案操作所需的设定

进行 PLC 方案操作所需的设定。

PLC 方案操作所需的设定

1 在方案管理机器人控制装置的［工作单元］的设置画面，按下 F3［方案］，显示［方案］的设置画面。

2　设定上方［方案设置］框内的［操作方案 DI］的 DI。

3　如果按下［ENTER］键，则显示以下信息。
4　按下 F4［确定］。

5　需要确认 PLC 是否出错时，设定［方案错误 DO］的 DO。
6　再次接通机器人控制装置的电源。

与 PLC 的信号交换程序 PK_PLC_RECIPE_OPERATION.TP 的变更

下面示出作为标准 TP 程序提供的 PK_PLC_RECIPE_OPERATION.TP。请根据需要，变更此程序。基本流程如下。

1　PLC 通过分组输入（GI）等指定操作与参数。
2　根据指定的操作调用 KAREL 程序。
3　确认 KAREL 程序的状态。

本程序是与 PLC 通信，取得操作和参数，进行方案操作的程序。如果将在树状图中的［方案］设定画面所设定的［方案操作 DI］DI 设成 ON，就会自动执行。

仅在非生产中的情况，也就是所有机器人程序结束的情况下可运行。

本程序在与 PLC 进行机械手摆动后，通过分组 I/O 获取操作和参数，但通信手段可根据应用程序变更。

程序的详细信息如下。

1 DO［32］是为了确认机器人是否可对 PLC 接收数据而输出的 DO。
2 DI［32］是在 PLC 处于对机器人可接收数据状态时输出的 DI。
3 GI［1］用于 PLC 向机器人发送操作与参数。
4 GO［1］用于机器人通过接收确认向 PLC 重新发送接收的操作与参数。
5 根据从 PLC 发送的操作选择 KAREL 程序。通过操作，将从 PLC 发送的参数指定成 KAREL 程序参数，调用 KAREL 程序。
6 KAREL 程序的状态表示方案操作是否成功。在［方案］的设定画面中，在“方案错误 DO”中设定 DO 编号，状态为 0 之外时，将打开“方案错误 DO”的 DO。

PK_PLC_RECIPE_OPERATION.TP

```
 1:  !---------------------------------
 2:  ! Communication program between a
 3:  ! PLC and iRPickTool to perform
 4:  ! Recipe operations.
 5:  ! This program runs if the Recipe
 6:  ! Change DI setup in the Recipe
 7:  ! menu is turned on.
 8:  ! Users can modify the PLC
 9:  ! handshake - you can change
10:  ! the DI/DO, GI/GO and registers
11:
12:  !---------------------------------
13:
14:  !Recipe Operation Function Codes
15:  ! CREATE       = 1
16:  ! DELETE       = 3
17:  ! SAVE_ACTIV_RECIPE = 4
18:  ! RESTORE_ACTIVE_RECIPE = 5
19:  ! EXPORT       = 6
20:  ! IMPORT       = 7
21:  ! SWITCH       = 8
22:  !
23:  ! Numeric registers R[95]-R[98]
24:  ! are used in this program.
25:  ! Users can re-assign the
26:  ! registers to other numbers
27:  ! to suit their conveninence.
28:  !
29:  !R[95] is used for storing status
30:
31:
32:  ! Turn off DO[33] (Success)
33:  DO[33:RecipeSuccess]=OFF
34:  ! Turn off DO[34] (Error)
35:  DO[34:RecipeFault]=OFF
36:
37:  !Tell PLC robot ready to receive
38:  DO[32:RecipeRobotReady]=ON
39:  ! Wait for PLC ready to send
40:  WAIT DI[32:RecipePLCReady]=ON TIMEOUT,LBL[4000]
41:  !Read function code
42:  R[97:TemporaryUse3]=GI[1:RecipeReadData]
43:  ! Echo Recipe Operation to PLC
44:  GO[1:RecipeEchoData]=R[97:TemporaryUse3]
```

```
   45:   ! Signal PLC that data received.
   46:   DO[32:RecipeRobotReady]=OFF
   47:   ! Wait for PLC acknowledgement.
   48:   WAIT DI[32:RecipePLCReady]=OFF TIMEOUT,LBL[4000]
   49:
   50:   SELECT R[97:TemporaryUse3]=1,JMP LBL[100]
   51:           =3,JMP LBL[300]
   52:           =4,JMP LBL[400]
   53:           =5,JMP LBL[500]
   54:           =6,JMP LBL[600]
   55:           =7,JMP LBL[700]
   56:           =8,JMP LBL[800]
   57:           ELSE,JMP LBL[4000]
   58:
   59:   LBL[100:Create]
   60:   !Tell PLC robot ready to receive
   61:   DO[32:RecipeRobotReady]=ON
   62:   ! Wait for PLC ready to send
   63:   WAIT DI[32:RecipePLCReady]=ON TIMEOUT,LBL[4000]
   64:   ! Read recipe_id register
   65:   R[98:TemporaryUse4]=GI[1:RecipeReadData]
   66:   ! Echo Recipe Id reg to PLC
   67:   GO[1:RecipeEchoData]=R[98:TemporaryUse4]
   68:   ! Signal PLC that data received.
   69:   DO[32:RecipeRobotReady]=OFF
   70:   ! Wait for PLC acknowledgement.
   71:   WAIT DI[32:RecipePLCReady]=OFF TIMEOUT,LBL[4000]
   72:   !Tell PLC robot ready to receive
   73:   DO[32:RecipeRobotReady]=ON
   74:   ! Wait for PLC ready to send
   75:   WAIT DI[32:RecipePLCReady]=ON TIMEOUT,LBL[4000]
   76:   ! Read string reg id for storing
   77:   ! recipe name
   78:   R[96:TemporaryUse2]=GI[1:RecipeReadData]
   79:   ! Echo string reg Id to PLC
   80:   GO[1:RecipeEchoData]=R[96:TemporaryUse2]
   81:   ! Signal PLC that data received.
   82:   DO[32:RecipeRobotReady]=OFF
   83:   ! Wait for PLC acknowledgement.
   84:   WAIT DI[32:RecipePLCReady]=OFF TIMEOUT,LBL[4000]
   85:   CALL PKRCCREATE(“方案 ID Reg”=R[98:TemporaryUse],“方案名称 SR”=R[96:Temporary],“状态
Reg”=95)
   86:   JMP LBL[5000]
   87:
   88:   LBL[300:Delete]
   89:   !Tell PLC robot ready to receive
   90:   DO[32:RecipeRobotReady]=ON
   91:   ! Wait for PLC ready to send
   92:   WAIT DI[32:RecipePLCReady]=ON TIMEOUT,LBL[4000]
   93:   ! Read recipe id to delete
   94:   R[98:TemporaryUse4]=GI[1:RecipeReadData]
   95:   ! Echo Recipe Id to PLC
   96:   GO[1:RecipeEchoData]=R[98:TemporaryUse4]
   97:   ! Signal PLC that data received.
   98:   DO[32:RecipeRobotReady]=OFF
   99:   ! Wait for PLC acknowledgement.
  100:   WAIT DI[32:RecipePLCReady]=OFF TIMEOUT,LBL[4000]
  101:   CALL PKRCDELETE(“方案 ID”=R[98:TemporaryUse4],“状态 Reg”=95)
```

```
  102:   JMP LBL[5000]
  103:
  104:   LBL[400:Save]
  105:   CALL PKRCSAVE(“状态 Reg”=95)
  106:   JMP LBL[5000]
  107:
  108:   LBL[500:Restore]
  109:   CALL PKRCRESTORE(“状态 Reg”=95)
  110:   JMP LBL[5000]
  111:
  112:   LBL[600:Export]
  113:   CALL PKRCEXPORT(“状态 Reg”=95)
  114:   JMP LBL[5000]
  115:
  116:   LBL[700:Import]
  117:   CALL PKRCIMPORT(“状态 Reg”=95)
  118:   JMP LBL[5000]
  119:
  120:   LBL[800:Switch]
  121:   !Tell PLC robot ready to receive
  122:   DO[32:RecipeRobotReady]=ON
  123:   ! Wait for PLC ready to send
  124:   WAIT DI[32:RecipePLCReady]=ON TIMEOUT,LBL[4000]
  125:   ! Read recipe id to switch to
  126:   R[98:TemporaryUse4]=GI[1:RecipeReadData]
  127:   ! Echo Recipe Id to PLC
  128:   GO[1:RecipeEchoData]=R[98:TemporaryUse4]
  129:   ! Signal PLC that data received.
  130:   DO[32:RecipeRobotReady]=OFF
  131:   ! Wait for PLC acknowledgement.
  132:   WAIT DI[32:RecipePLCReady]=OFF TIMEOUT,LBL[4000]
  133:   CALL PKRCACTIVE(“方案 ID”=R[98:TemporaryUse4],“状态 Reg”=95)
  134:   JMP LBL[5000]
  135:
  136:   LBL[4000:PLC Recipe Error]
  137:   ! 127294 = PLC Rec Op failed
  138:   R[95:TemporaryUse1]=127294
  139:   JMP LBL[5000]
  140:
  141:   LBL[5000:Check status]
  142:   IF (R[95:TemporaryUse1]<>0) THEN
  143:   ! Turn Error DO on
  144:   DO[34:RecipeFault]=ON
  145:   ! Post error and abort
  146:   CALL PKPOSTERR(“错误代码”=R[95:TemporaryUse1],中止)
  147:   ELSE
  148:   ! Turn success DO on
  149:   DO[33:RecipeSuccess]=ON
  150:   ENDIF
  151:   JMP LBL[10000]
  152:
  153:   LBL[10000:End]
/END
```

9.1.4 KAREL 程序

> ⚠ **注意**
> **使用此处介绍的 KAREL 程序进行方案操作时，最初调用 PKWCEND 来停止传感器。此外，方案操作过程中强制结束时，下次再进行方案操作时可能会发生 IRPK-083“另一个方案正在执行中。”。为了防止方案操作过程中强制结束，建议在使用了此处介绍的 KAREL 程序的 TP 程序的详细信息设定中，将忽略暂停设为 ON。**

9.1.4.1 PKRCCREATE.PC

本 KAREL 程序，仅在生成方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 创建方案。
- 设定创建的方案 ID。
- 设定创建的方案名称。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定存储方案 ID 的数值寄存器。

参数 2:
指定存储方案名称的字符串寄存器。
参数 3:
指定返回状态的数值寄存器。

使用例：
生成方案。将方案 ID 返回 R［2］，将方案名称返回 SR［3］，将状态返回 R［1］。

```
CALL PKRCCREATE (方案 ID Reg=2,方案名称 SR=3,状态 Reg=1)
```

9.1.4.2 PKRCDELETE.PC

本 KAREL 程序，仅在删除方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 确认指定的方案是否有效。
- 如果有效，删除指定的方案。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定要删除的方案 ID。

参数 2:
指定返回状态的数值寄存器。

使用例：
删除 ID 为 3 的方案。将状态返回 R［1］。

```
PKRCDELETE (方案 ID=3,状态 Reg=1)
```

9.1.4.3 PKRCSAVE.PC

本 KAREL 程序，仅在保存方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 保存方案。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定返回状态的数值寄存器。

使用例：
保存方案。将状态返回 R［1］。

```
PKRCSAVE (状态 Reg=1)
```

9.1.4.4 PKRCRESTORE.PC

本 KAREL 程序，仅在恢复方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 恢复方案。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定返回状态的数值寄存器。

使用例：
恢复方案。将状态返回 R［1］。

```
PKRCRESTORE (状态 Reg=1)
```

9.1.4.5 PKRCEXPORT.PC

本 KAREL 程序，仅在导出方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 导出方案。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定返回状态的数值寄存器。

使用例：
导出方案。将状态返回 R［1］。

```
PKRCEXPORT (状态 Reg=1)
```

9.1.4.6 PKRCIMPORT.PC

本 KAREL 程序，仅在导入方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 导入方案。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定返回状态的数值寄存器。

使用例：
导入方案。将状态返回 R［1］。

```
PKRCIMPORT(状态 Reg=1)
```

9.1.4.7 PKRCACTIVE.PC

本 KAREL 程序，仅在切换方案时使用。只能在树状图上设定的所有控制装置的程序结束时运行。

该程序如下动作。
- 如果［方案错误 DO］的 DO 为 ON，设成 OFF。
- 确认指定的方案是否有效。
- 如果有效，则导入指定的方案。
- 设定状态。
- 如果状态不是 0，将［方案错误 DO］的 DO 设成 ON。

参数 1:
指定要切换的方案的 ID。
参数 2:
指定返回状态的数值寄存器。

使用例：
切换至 ID 为 3 的方案。将状态返回 R［1］。

```
CALL PKRCACTIVE (方案 ID=3,状态 Reg=1)
```

9.2 确认生产状况

可确认 *i*RPickTool 的各对象的生产状况。

1 按下示教器的［MENU］键。
2 选择［下页］。
3 选择［状态］。
4 按下 F1［类型］键，从选择项中选择［*i*RPickTool］。

基本操作
这里对各个对象的操作中的共通部分进行说明。

连续更新的停止与重启
需要特意停止连续更新时，在连续更新时按下 F2［停止更新］。

或是按下 F2［连续更新］，重启连续更新。

生产状况复位

将生产时间、统计值复位，此时如果需要进行再次统计，则按下 F3［重置］。

更新树状图

在登录生产状况的状态下，在 *i*RPickTool 的设置画面进行追加传送带工作站等树状图的操作时，生产状况画面中不会自动反映其设定内容。这种情况，按下［＞（下一页）］，按下 F1［更新树状图］。这样，新追加的传送带工作站等会在树状图上显示。
而暂时从生产状况退出，即使再次登录，树状图也已被更新。

9.2.1 确认传送带的状态

可确认在传送带整体的工件的处理状况。

从树状图选择目标传送带。
不使用托盘时，显示以下画面。

使用托盘时，显示以下画面。

[生产时间]

表示开始生产后的时间。

[传送带速度（mm/s）]

表示传送带的速度。圆弧传送带时的单位是 deg/s。

[托盘]

在该传送带使用的托盘的名称。仅在使用托盘时显示。

[显示单位]

选择是以单元单位表示总搬运数等，还是以托盘单位表示。仅在使用托盘时显示。

[搬运数]

在该传送带整体上处理的工件的总数。使用托盘时，根据显示单位显示在该传送带整体处理的单元或托盘的总数。

[平均搬运速度（个/分）]

以生产时间除以在该传送带整体处理的工件总数，所得的平均搬运数。使用托盘时，根据显示单位，显示以生产时间除以在该传送带整体处理的单元或托盘的总数所得的平均搬运数。

[失败数]

任何传送带工作站都没有处理的工件总数。不包括视觉检出失败的工件。

[重置]

如果按下 F3［重置］键，则各数据被复位。复位后，再次开始记录。

9.2.2 确认传感器的状态

可确认传感器的状态。

从树状图选择目标传感器。

[状态]
表示该传感器的运行状态。如果该传感器正在运行，则以绿字显示［运行中］。如果停止，则以红字显示［暂停中］。

[视觉程序]
使用该传感器的视觉程序名。仅在使用视觉时显示。

[重叠次数]
使用视觉的情况，在前一个视觉的执行结束之前，后面进入的触发信号的次数。如果不是 0，表示系统上未能来得及执行视觉，必须降低触发信号进入的频率。
具体请将“3.2.5 传感器的设定”，“4.2.6 传感器的设定”（或“6.5.1 传感器详细设定”的“6.5.1.2 ［触发距离］＋［动作时间］的情况”）中设定的［触发距离］设定得比下面的［动作时间］更长。

[动作时间]
该传感器的检出时间，以及将检出的工件投入最上游传送带工作站所需的时间。如果该值比触发间隔大，则［重叠次数］增加。

[触发间隔]
触发所需的时间间隔。

[检出个数]
使用视觉时，在 1 次图像处理中检出的工件的个数（在相机视野中看到的工件个数）。

[重置]
如果按下 F3［重置］键，则各数据被复位。复位后，再次开始记录。

9.2.3 确认传送带工作站的状态

可确认在各传送带工作站的工件的处理状况。

从树状图选择目标传送带工作站。
不使用托盘时，显示以下生产状况的画面。

使用托盘时，显示以下生产状况的画面。

此外，如果按下 F4［队列］，则显示如下传送带工作站内的从下游的 10 个工件的状态。
以下的画面是不使用托盘的情况。X, Y, R 表示在各自的追踪坐标系上的位置和角度。

使用托盘时，显示以下传送带工作站内的各单元的状态。

使用圆弧传送带时，如下所示，不使用 X 和 Y，而使用角度和半径（＝圆弧传送带的中心和工件的距离）表示工件的现在位置。

如果按下 F4［生产状况］，则再次显示生产状况的画面。

生产状况

列表显示各模型 ID 的工件的处理状况。

［托盘］
在传送带上使用的托盘的名称。仅在使用托盘时显示。

［显示单位］
选择是以单元单位表示总搬运数等，还是以托盘单位表示。仅在使用托盘时显示。

［模型 ID］
在该传送带工作站上处理的工件的模型 ID。

［供给数］
该传送带工作站上传送的工件的总数。

［搬运数］
在该传送带工作站上处理的工件的总数。使用托盘时，根据显示单位，显示在该传送带工作站上处理的单元或托盘的总数。

［平均搬运速度（个/分）］
以生产时间除以在该传送带工作站上处理的工件总数，所得的平均搬运数。使用托盘时，根据显示单位，显示以生产时间除以在该传送带工作站上处理的单元或托盘的总数所得的平均搬运数。

队列

列表显示传送带工作站内的各工件的状态。从最下游的工件依次显示，最多显示 10 个工件。

［编号］
识别各工件、单元的编号。

［层］
仅在使用托盘时显示的各单元的层编号。

［模型 ID］
各工件、单元的模型 ID。使用托盘时，是以托盘设定的各单元的模型 ID。不使用托盘，使用视觉时，使视觉检出的模型 ID。

［X］，［Y］，［Z］，［R］
　各工件、单元的现在位置。显示从追踪坐标系看的位置。仅在使用托盘时显示 Z。
　圆弧传送带的情况，不以 X 和 Y，而以角度和半径表示。

9.2.4 确认固定工作站的状态

可确认在各固定工作站的单元、托盘的处理状况。

从树状图选择目标固定工作站。

［生产时间］
　表示生产时间。

［托盘］
　放置于该固定工作站的托盘的名称。

［显示单位］
　选择是以单元单位表示总搬运数等，还是以托盘单位表示。

［搬运数］
　在该固定工作站上处理的单元或托盘的总数。

［平均搬运速度（个/分）］
　以生产时间除以在该固定工作站上处理的单元或托盘的总数，所得的平均搬运数。

［空单元 / 所有单元］
　相对在现在作业的托盘中的所有单元数的空单元数。

［重置］
　如果按下 F3［重置］键，则各数据被复位。复位后，再次开始记录。

9.3 限制最终轴的轴可动范围

机器人伴随姿势变化进行直线动作时，使轴的旋转角度为最小地进行动作。这称为距离最短动作。如同在视觉追踪，拾取侧、放置侧的最终轴旋转角每次变化的系统中，在反复距离最短动作期间，可能最终轴的角度超出轴可动范围，发生行程极限报警。

本功能是在由于距离最短动作而发生行程极限时，特地绕远来回避最终轴的行程极限的功能。

本功能仅在以下条件成立时有效。
- 追踪类型为［直线］。
- 正在追踪拾取动作。
- 为直线动作。
- 有手腕姿势。

本功能默认为有效。设定以下系统变量。
- $LNCFG_GRP［组编号］.$LMT_CHK_ENB
  本功能有效无效的标记。无效时，设定为 FALSE。
- $LNCFG_GRP［组编号］.$LMT_CHK_UL
  从最终轴上限角度的偏离（单位：度）。根据需要进行设定。
- $LNCFG_GRP［组编号］.$LMT_CHK_LL
  从最终轴下限角度的偏离（单位：度）。根据需要进行设定。

任何的系统变量，变更都立即适用。

例如，在 6 轴机器人中，J6 轴的行程极限是 –270～270 度时，如果设定$LMT_CHK_UL = 90 度、$LMT_CHK_LL = 90 度，则 J6 轴的动作范围限制在 –180～180 度的范围。

9.4 超出领域时，跳过追踪动作

如果在传送带工作站的设定中，正确设定［放弃线］，则在工件超出［放弃线］时，机器人判断不追赶该工件。然后，该工件的信息被转移至下一个传送带工作站。但是，如果发生机器人的倍率低、传送带比预计快、机器人在远离的位置等多重条件，则可能发生追不上［放弃线］上游的工件。本功能是在发生这种情况时，中途中止追赶工件的功能。

通常情况，追踪程序由向以下 3 个位置的动作构成。

P1: 接近点
P2: 拾取位置
P3: 逃避点

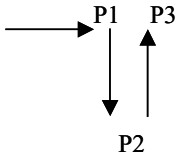


图 9.4(a) 追踪程序的位置

本功能是在运行向上述各位置移动的动作程序时，对目标位置是否在领域内进行判断。
如果判断为在领域外，则跳过向该位置移动的追踪动作程序，移至下一个指令行。

要使用本功能，需在机器人的设置画面设定［跳过动作实施范围］的各项目。

［有效］
切换本功能的有效无效。

［跳过动作实施时间］
动作开始，判断是否在领域内时，加上这里设定的值。单位为 ms。

在去拾取上述动作模式的工件时，当机器人到达 P1 时，工件还在领域内，机器人从 P1 向 P2 移动期间，或从 P2 向 P3 移动期间，工件至领域外的情况下，才开始放弃该工件，不向 P1 移动，而移动至下一个工件的方式更为有效。
这种情况，只要设定 P1→P2→P3 的移动所需时间，则不需进行不必要的向 P1 的移动。

例如，通过时间指令等，测量从 P1 经过 P2，到达 P3 所需的时间。如果该时间为 100ms，则将 100ms 输入［跳过动作实施时间］。

这样，在向 P1 的动作开始时，对仅领先 100ms 的位置，也就是 P3 位置是否在领域内进行判断。如果判断为在领域外，则跳过 P1、P2、P3 所有的动作程序。

［标记编号］
如果发生跳过，则这里指定的编号的标记变为“ON”。但是，系统软件只是使标记为“ON”。如果标记为“ON”，则需要在机器人程序中为“OFF”。

注释
通过取多个的机械手拾取工件后，在等待下一个工件期间，继续追踪动作，如果至领域外，则本功能并不有效。这是因为没有考虑等待下一工件间的追踪动作的继续。
这种情况，如“6.9.2.2 进行拾取动作的程序”所说明的，在调出 PKCSGETQUE 之前，执行“停止追踪”指令，停止追踪动作。

程序示例

使用该功能的情况，需要对进行拾取动作或放置动作的追踪程序进行变更。以下是变更后的程序例。主要的变更有下划线标出。标记编号为 1，从 P1 经过 P2 到达 P3 的时间为 100ms。

［标记编号］=1
［跳过动作实施时间］100（ms）

PICK1.TP

```
   1:   UTOOL_NUM=1
   2:   UFRAME_NUM=0
   3:   R[3:区域]=1
   4:
   5:   LBL[100]
   6:   STOP_TRACKING
   7:   F[1]=(OFF)
   8:   CALL PKCSGETQUE(传送带工作站 ID=R[11:CSTN1],连续的标记=R[3:区域],超时值(ms)=100,
        补偿用 VR=1,状态 Reg=2)
   9:   IF R[2:GETQ_状态]=0,JMP LBL[200]
  10:    IF R[1:周期停止]=1,JMP LBL[900]
  11:    JMP LBL[100]
  12:
  13:    LBL[200]
  14: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
  15:    IF (F[1]=ON),JMP LBL[300]
  16: L PR[R[3:区域]] 4000mm/sec CNT3 VOFFSET,VR[1]
  17:    IF (F[1]=ON),JMP LBL[300]
  18:    CALL VACUUM_ON
  19: L PR[R[3:区域]] 4000mm/sec CNT100 VOFFSET,VR[1] Tool_Offset,PR[10]
  20:
  21:    CALL PKCSACKQUE(传送带工作站 ID=R[11:CSTN1],成功)
  22:    R[3:区域]=R[3:区域]+1
  23:    IF R[3:区域]<=R[4:区域数],JMP LBL[100]
  24:    JMP LBL[900]
  25:
  26:    LBL[300]
  27:    CALL PKCSACKQUE(传送带工作站 ID=R[11:CSTN1],发信)
  28:    JMP LBL[100]
  29:
  30:    LBL[900]
[End]
```

当判断为在领域外的情况，通过 PKCSACKQUE 指定“发信”，将未能拾取该工件事宜通知下游侧机器人，由下游侧的机器人拾取该工件。

标记在程序中明确恢复“OFF”。

9.5 确认机器人的扭矩

如果电机过热，机器人就会因警报而停止。追踪动作中，看各轴上施加的扭矩值，确认电机是否过热。
扭矩监视画面的显示方法如下。

1 在示教器上按下［MENU］键。
2 选择［下页］。
3 选择［状态］。
4 按下 F1［类型］键，从选择项中选择［轴］。
5 按下［＞（下一页）］键后，按下 F2［监控］键。则显示以下画面。

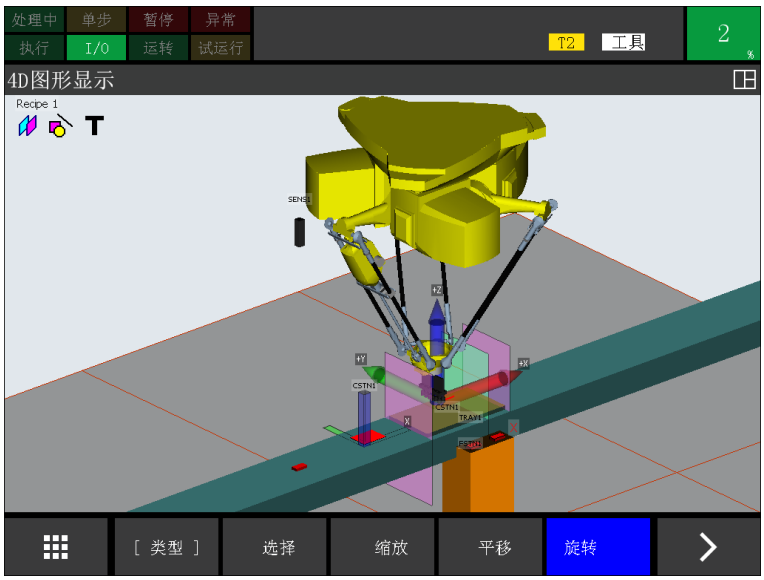
在系统连续正常运转的状态下，此处所显示的各轴［平均］值慢慢上升时，电机就有可能发生过热。
这种情况下，使追踪的动作程序速度变慢。

9.6 *i*RPickTool 的 4D 图形显示

本节，对 *i*RPickTool 的 4D 图形功能选项的设定进行说明。

当指定 4D 图形功能选项时，可通过 iPendant 使工作单元内的所有对象视觉化。例如，可以显示机器人、抓手、追踪坐标系、追踪范围的边界、放弃线、托盘及其单元、固定工作站的位置等。只通过数值，就能视觉性确认不易想象的设定，因此有助于确认是否正确进行了设定。

下图是以 4D 图形表示由 1 台机器人构成的工作单元的设定例。



4D 图形设定在树状图上进行。
如果指定了 4D 图形功能选项，则在工作单元、抓手、机器人、托盘、传送带、传送带工作站、传感器、固定工作站等对象的右侧窗格显示［4D 参数设置］标签。该标签中，备有对象的尺寸、颜色、透明度、显示、以及位置等各对象可变更的设定。
下图是传送带的［4D 参数设置］标签的显示。

9.6.1 4D 工作单元的显示

对象的［4D 参数设置］结束后，通过示教器的［MENU］键，移动至［4D 图形］－［4D 图形显示］。可显示现在的工作单元的情况，确认各对象的设定。

9.6.2 4D 工作单元的生成

空的工作单元

指定 4D 图形功能选项，未在树状图上进行 *i*RPickTool 的设定时，［4D 图形显示］中显示空的工作单元。

空的工作单元，至少包括 1 台机器人。工作单元中有多台机器人，通过 RIPE 进行机器人间通信的设定时，显示网络内的所有机器人。在树状图的传送带设定中，使用向导的追踪坐标系的设定结束时，则正确显示机器人的间隔，如果未结束，将发生所有的机器人相互重叠显示等不正确的显示。

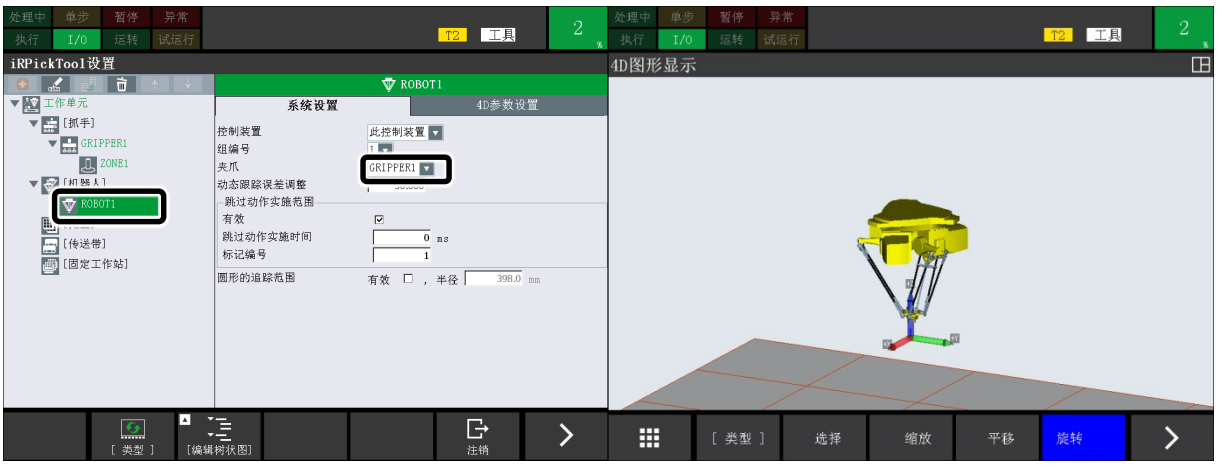
空的工作单元中，不显示抓手、传送带、固定工作站等的设定。
请参照下图的空工作单元。

4D 工作单元的新建

在树状图中生成工作单元时，也自动生成［4D 图形显示］的 4D 工作单元。

4D 抓手的新建

在树状图中生成抓手，如下图所示，通过［机器人］的设定分配该抓手，则［4D 图形显示］中也自动生成抓手。可确认手臂前端的抓手，以及显示［机器人］名称的机器人上的文本显示。



4D 抓手的变更

在树状图上，选择需要改变 4D 设定的抓手（GRIPPER1 等），按下［4D 参数设置］标签，则显示所有可变更的设定。在该设置画面，可变更抓手的尺寸、颜色、显示的有效/无效、以及以机器人的标牌为基准的位置，但是不能变更抓手显示所使用的形状。需要变更时，请参照“9.6.7 用户定义对象形状的使用”。

［长度］

显示的抓手的长度（单位 mm）。

［宽度］

显示的抓手的宽度（单位 mm）。

［高度］

显示的抓手的高度（单位 mm）。

［颜色］

显示的抓手的颜色。从下拉框进行选择。

［可见］

使用复选框，使抓手的显示为有效或无效。

［位置］

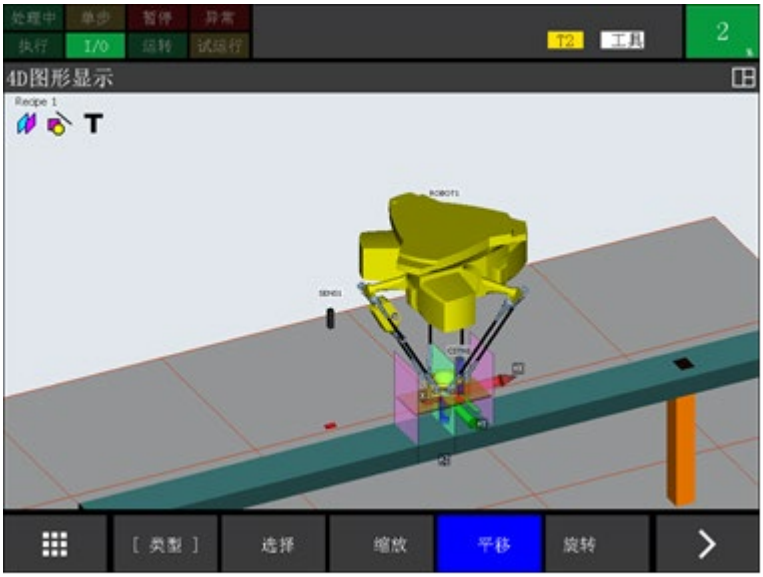
从机器人的标牌的抓手的原点位置。初始值全部为 0。

4D 传送带的新建

如果在树状图上新建传送带，则如下图所示，在［4D 图形显示］画面自动生成长 4000 mm 、宽 250 mm、高 100 mm 的传送带。显示的传送带的原点，在皮带的上方中央。该传送带的原点，与该传送带内最初的传送带工作站的追踪坐标系一致。如果没有示教追踪坐标系，则其值与机器人的世界坐标系的原点相同，都是 0。因此，在第一次新建最初的传送带时，通过机器人的世界坐标系的原点显示传送带。然后使用向导示教追踪坐标系时，传送带移动至正确位置。

此外，如果在树状图上新建传送带，将同时生成传感器和传送带工作站。传感器以从传送带内最初的传送带工作站的追踪坐标系的初始值的高度为 900 mm，长 25 mm、宽 25 mm、高 100 mm 的黑长方体形式显示。但是，在传送带工作站的设定中，有效的追踪参数设定表的设定结束之前，传送带工作站不在［4D 图形显示］的画面上显示。设定追踪参数设定表后，传送带工作站上，传送带上以褐色显示追踪范围，同时在传送带上还显示 3 个边界线（追踪范围的上游边界、下游边界、以及放弃线）。上游侧或下游侧的边界为紫色，放弃线以淡绿色显示。

如果在［传送带］的设置画面执行向导，正确设定追踪坐标系，或在［传感器］的设置画面执行向导，正确设定传感器位置（需通过手动设定高度），则如下图所示，传送带、传感器、以及传送带工作站在工作单元内的所有正确位置上显示。



9

4D 传送带的变更

在树状图上，选择需要改变 4D 设定的传送带（CONV1 等），按下［4D 参数设置］标签，则显示所有可变更的设定。在该设置画面，可变更传送带的尺寸、颜色、显示、以及位置，但是不能变更传送带显示所使用的 CAD 文件。变更 CAD 文件时，请参照本章的“9.6.7 用户定义对象形状的使用”章节。

［传送带长度］

传送带皮带的长度（单位 mm）。

［传送带宽度］

传送带皮带的宽度（单位 mm）。

［支架高度］

传送带支架的长度（单位 mm）。

［传送带颜色］

传送带皮带的颜色。从下拉框进行选择。

［可见］

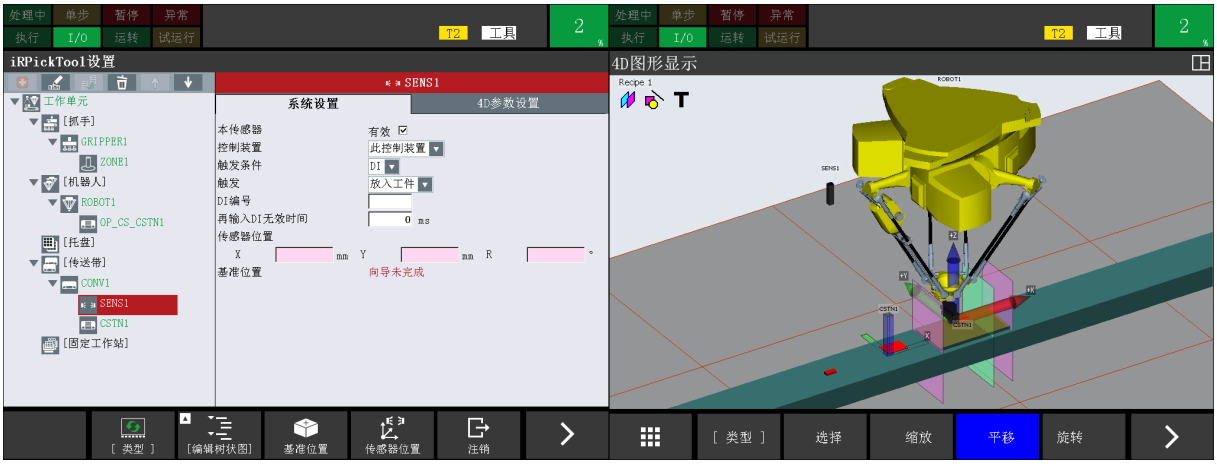
使用复选框，使传送带的显示为有效或无效。

［位置］

从传送带内最初的传送带工作站的追踪坐标系的，所显示的传送带的原点位置。传送带的原点，在皮带的上方中央。

4D 传感器的新建

如果在树状图上新建传感器，则如下图所示，在［4D 图形显示］画面上作为长 25 mm、宽 25 mm、高 100 mm 的长方体自动显示。传感器的原点，是长方体的中心。传感器的原点，作为初始值被设定于距离传送带内最初的传送带工作站的追踪坐标系 900mm 的上方。在［传感器］的设置画面中，如果通过向导设定传感器位置，则传感器的位置 X,Y 在正确位置显示，但如果未能在［4D 图形显示］的画面中正确变更高度，则显示的高度始终为初始值 900 mm。



4D 传感器的变更

在树状图上，选择需要改变 4D 设定的传送带（SENS1 等），按下［4D 参数设置］标签，则显示所有可变更的设定。在该设置画面，可变更传感器的尺寸、颜色、显示、以及位置，但是不能变更传感器显示所使用的形状。需要变更时，请参照“9.6.7 用户定义对象形状的使用”。

［长度］

显示的传感器的长度（单位 mm）。

［宽度］

显示的传感器的宽度（单位 mm）。

［高度］

显示的传感器的高度（单位 mm）。

［颜色］

显示的传感器的颜色。从下拉框进行选择。

［不透明度］

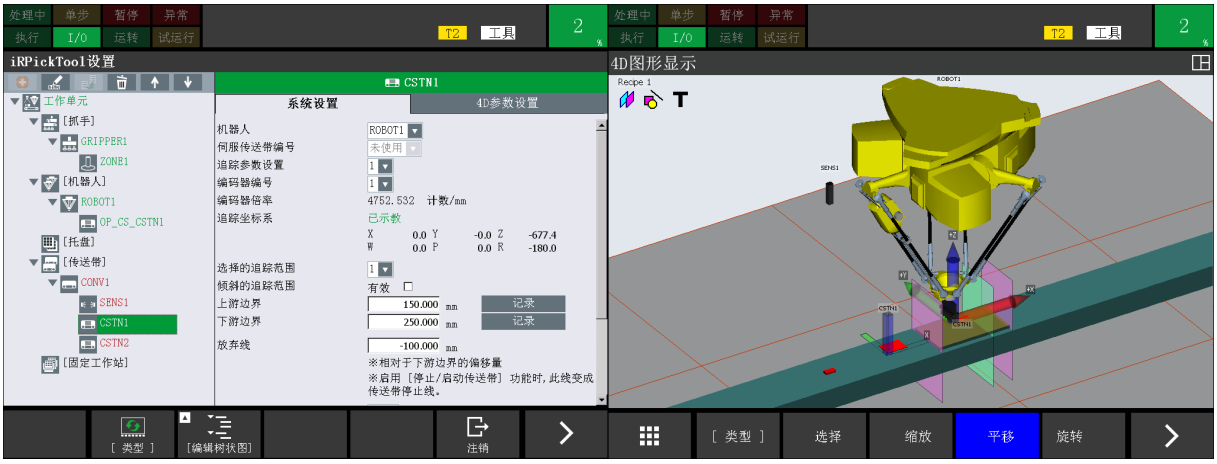
显示的传感器的不透明度。输入 0～100 的值。0 是完全透明，100 是完全不透明。

［可见］
使用复选框，使传感器的长方体的显示为有效或无效。

［位置］
从传送带内最初的传送带工作站的追踪坐标系的传感器原点位置。

4D 传送带工作站的新建

在树状图上新建传送带工作站后，如下图所示那样在［4D 图形显示］的画面上，以在传送带上追踪坐标系向着 3 个方向伸展的细小的长方体，用紫色分别自动显示追踪范围的上游侧的边界、下游侧的边界。放弃线以淡绿色（春绿色）显示。追踪范围以棕褐色显示。



4D 传送带工作站的变更

在树状图上选择希望变更 4D 设定的传送带工作站（CSTN1 等），按下［4D 参数设置］标签，显示可变更的所有设定。可以在此设置画面上，变更表示追踪范围、追踪坐标系、追踪范围的边界和放弃线的形状的大小、颜色、透明度、显示/非显示。此外，也可以切换传送带工作站整体的显示/非显示。

［基准高度］
表示追踪范围的长方体的高度（单位 mm）。

［基准颜色］
追踪范围的显示颜色。从下拉框进行选择。

［基准不透明度］
表示追踪范围的长方体的不透明度。输入 0～100 的值。0 是完全透明，100 是完全不透明。

［基准可见］
使用复选框，使得追踪范围的显示有效或者无效。

［追踪边界颜色］
追踪范围的边界的显示颜色。从下拉框进行选择。

［追踪边界不透明度］
追踪范围的边界的不透明度。输入 0～100 的值。0 是完全透明，100 是完全不透明。

［追踪边界可见］
使用复选框，使得追踪区域的边界的显示有效或者无效。

［视觉追踪边界颜色］
放弃线的显示颜色。从下拉框进行选择。

［视觉追踪边界不透明度］
表示放弃线的侧壁的不透明度。输入 0～100 的值。0 是完全透明，100 是完全不透明。

［视觉追踪边界可见］
使用复选框，使得放弃线的显示有效或者无效。

［追踪边界可见］
使用复选框，使得追踪坐标系的显示有效或者无效。

［传送带工作站可见］
使用复选框，使得传送带工作站整体的显示有效或者无效。
包括追踪坐标系、追踪范围、追踪范围的边界和放弃线。

4D 托盘的新建

在树状图上新建托盘，如果如下图所示，在固定工作站上设定该托盘，则在固定工作站上自动显示托盘的单元。同样也可以在传送带上设定托盘。单元在托盘坐标系的［托盘］的设置画面中设定的位置上显示。在固定工作站上使用时，托盘坐标系与通过固定工作站的设定选择的用户坐标系相同。在传送带工作站上使用时，托盘坐标系被作为从传送带最初的传送带工作站的追踪坐标系的偏移值计算。

而仅通过［托盘］的集下的定义，不会显示托盘。只会在设定了固定工作站或传送带的情况下显示。

4D 托盘的变更

在树状图上，选择需要改变 4D 设定的托盘（TRAY1 等），按下［4D 参数设置］标签，则显示所有可变更的设定。在该设置画面上，可设定工件的编号、托盘坐标系的显示的有效/无效。

［工件］

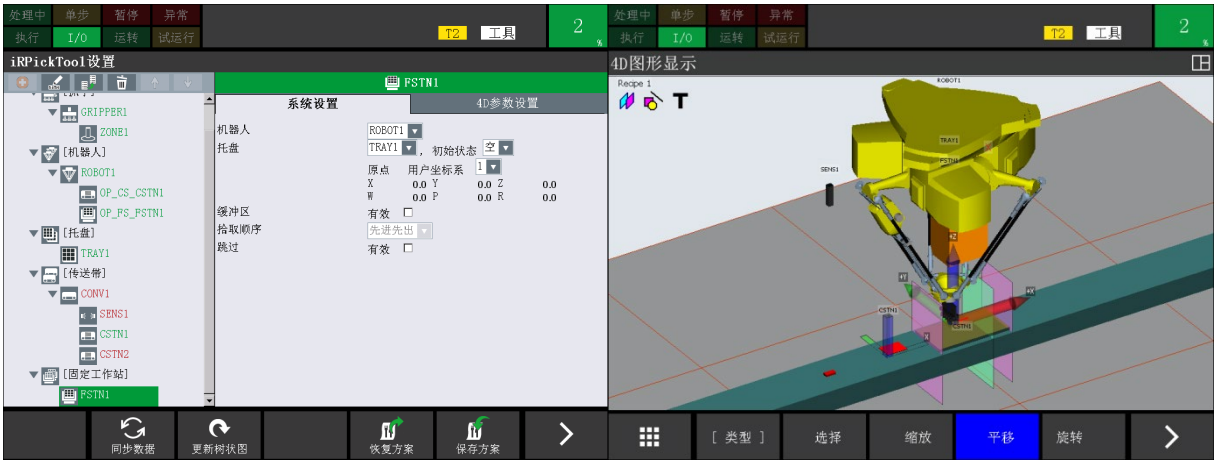
可利用下拉框来选择在托盘的各单元位置显示的工件的编号。可通过系统变量$pkpt4d，设定尺寸和颜色。

［坐标可见］

使用复选框，使托盘坐标系的显示为有效或无效。仅在托盘坐标系显示时，［4D 图形显示］的画面上显示托盘的名称。

4D 固定工作站的新建

如果在树状图上新建固定工作站，则如下图所示，在［4D 图形显示］画面上作为长 200 mm、宽 150 mm、高 500 mm 的长方体自动生成。显示的固定工作站的原点，与长方体中心上，树状图的［固定工作站］的设置画面中设定的编号的用户坐标系的原点一致。如果没有示教用户坐标系编号，则与机器人的世界坐标系的原点一致。因此，新建时，固定工作站在机器人的世界坐标系的原点上显示。如果后来设定用户坐标系编号，则固定工作站根据该用户坐标系显示。



4D 固定工作站的变更

在树状图上，选择需要改变 4D 设定的固定工作站（FSTN1 等），按下［4D 参数设置］标签，则显示所有可变更的设定。在该设置画面，可变更固定工作站的尺寸、颜色、显示、以及位置，但是不能变更固定工作站显示所使用的形状。需要变更时，请参照“9.6.7 用户定义对象形状的使用”。

［长度］

显示的固定工作站的长度（单位 mm）。

［宽度］

显示的固定工作站的宽度（单位 mm）。

［高度］

显示的固定工作站的高度（单位 mm）。

［颜色］

显示的固定工作站的颜色。从下拉框选择颜色。

［不透明度］

显示的固定工作站的不透明度。输入 0～100 的值。0 是完全透明，100 是完全不透明。

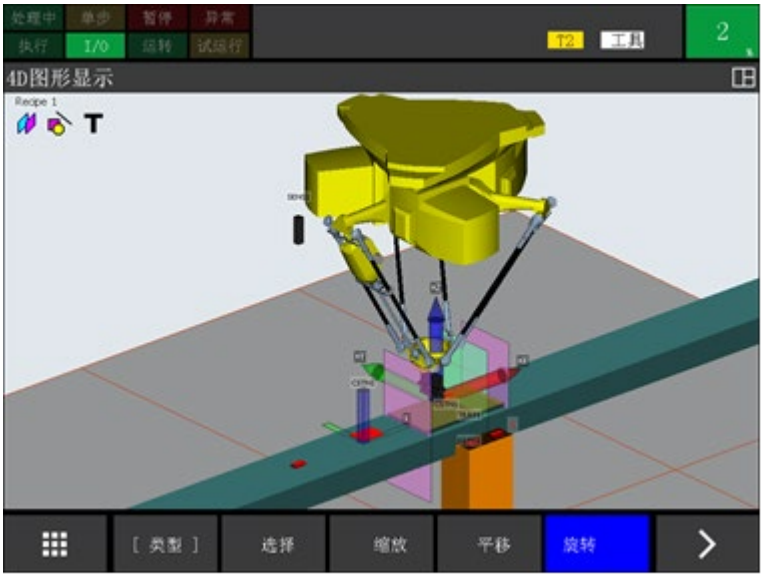
［可见］

使用复选框，使固定工作站的显示为有效或无效。

［位置可见］
使向托盘坐标系的相对固定工作站的机器人接近位置，以及拾取/放置位置的表示为有效或无效。仅在 Plug & Play 的情况下正确表示。

［位置］
从指定的用户坐标系的固定工作站原点位置。

如果正确设定用户坐标系，则［4D 图形显示］画面如下图显示。



9.6.3 图标的使用方法

［4D 图形显示］画面左上角，有如下图的 3 个图标。

示教器对应触摸屏的情况，可通过点击上述图标，简单地进行 4D 图形显示相关的操作。要使［4D 图形显示］画面上的触摸有效，必须最开始按下 F2［选择］。通过按一次各图标，对各个图标相应的以下设定的有效/无效进行切换。

边界的显示/非显示图标
是切换所有的传送带工作站的边界的显示/非显示的图标。边界包括追踪范围的上游侧边界、下游侧边界、以及放弃线。

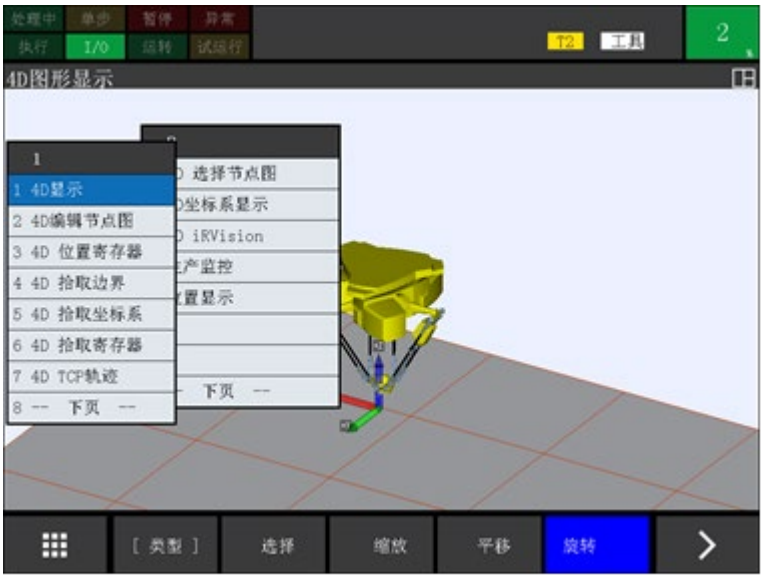
*i*RPickTool 的显示/非显示图标

是切换所有的 *i*RPickTool 的对象的显示/非显示的图标。需要确认“视觉点动”、“4D 编辑节点图”、“4D 位置寄存器”、“4D *i*RVision 显示”等，以及其他通用 4D 图形功能的显示时使用。

文本的显示/非显示图标
是切换文本的显示/非显示图标。*i*RPickTool 作为文本，显示机器人、传送带工作站、固定工作站、传感器、托盘坐标系、以及追踪坐标系的名称。切换上述各项的显示/非显示。

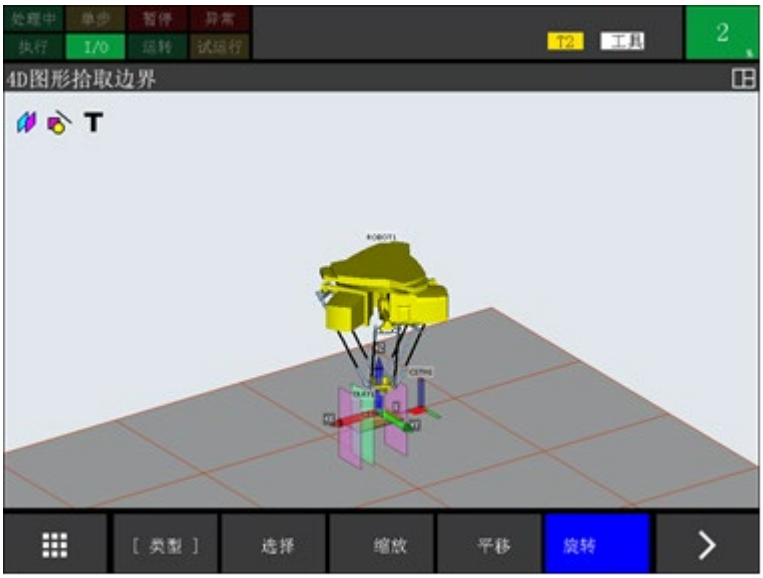
9.6.4 特定信息的选择显示

在［4D 图形显示］画面，如果按下 F1［类型］、将显示［4D 拾取边界］、［4D 拾取坐标系］、［4D 拾取寄存器］3 个 *i*RPickTool 相关的选择项，如果选择这些选择项，将仅显示各个的特定信息。要确认这些设定时发挥作用。



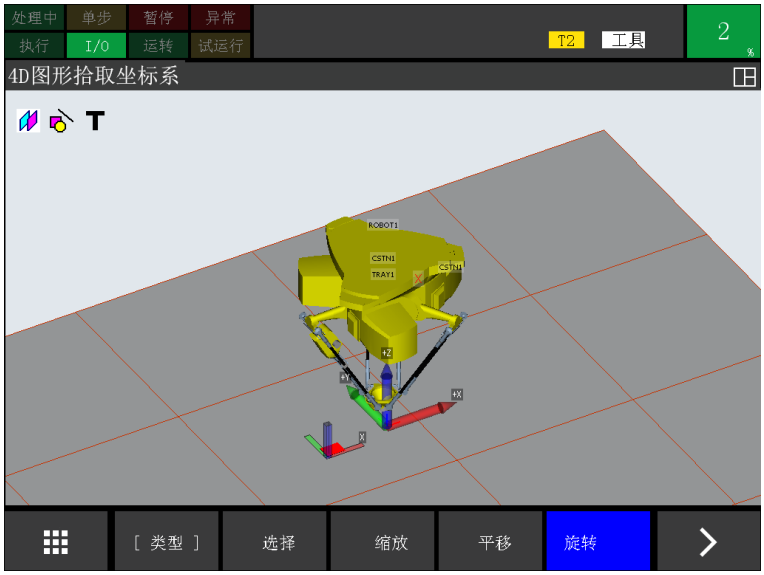
9.6.4.1 4D 拾取边界

如果选择［4D 拾取边界］，则只有所有的传送带工作站的追踪范围的边界，以及所有的传送带上的追踪坐标系，与机器人一起显示。这有助于确认以机器人为基准的追踪范围的边界，以及追踪坐标系的位置。



9.6.4.2 4D 拾取坐标系

如果选择［4D 拾取坐标系］，则只有所有的传送带上的追踪坐标系，以及固定工作站坐标系，与机器人一起显示。有助于确认是否示教所有的坐标系，或是否在正确位置进行示教。



9.6.5 4D 工作单元的删除

通过从 *i*RPickTool 的树状图删除所有对象，可以删除整个 4D 工作单元。
此外，传感器或传送带工作站等各个对象可以在树状图删除，但不可从 4D 图形显示删除机器人。

在需要暂时使 *i*RPickTool 的对象为非显示时，可使用上述 *i*RPickTool 的显示/非显示 图标 。

9.6.6 使用方案的工作单元的变更

导入方案后，工作单元中显示根据保存于该方案的设定数据的对象。所有的对象和其位置，都根据导入方案的数据。

［4D 图形显示］画面中，画面左上方的图标上，始终显示使用中的方案的名称。请参照画面。下图显示的文本“Recipe 1”，是使用中的方案的名称。

9.6.7 用户定义对象形状的使用

在［4D 图形显示］画面显示 *i*RPickTool 的各对象时，使用根据 *i*RPickTool 中准备的标准 CAD 文件的形状，一般情况下不能变更这些形状。

但是，需要使用单独的 CAD 文件，形成更接近实际系统的形状的 4D 图形的情况下，可使用名为“ROBOGUIDE 4D 编辑工具”的 FANUC 产品进行编辑。“ROBOGUIDE 4D 编辑工具”中，可生成工作单元，并将其下载至控制装置。

要显示使用单独的 CAD 文件的形状，有关抓手、传感器、传送带、固定工作站等对象，必须使标准形状不在 *i*RPickTool 中显示。因此，按下树状图的［工作单元］的设置画面的［4D 参数设置］标签，使［显示物理模型］的设定无效。如果设为无效，在 *i*RPickTool 中，仅追踪范围的边界等无法用户定义的对象显示标准形状。有关设定，请参照以下画面。

9.6.8 制约事项

符合以下任何一个条件时，请勿显示 4D 图形。有的情况下将无法正确显示。

- 1 个工作单元上使用 5 台以上机器人的情况。
- 使用了 2 个以上供给传送带或者输出传送带的情况。
- 使用了圆弧追踪的情况。

9.7 伺服传送带追踪拾取功能

9.7.1 概要

所谓伺服传送带追踪拾取功能，是在作为机器人的附加轴驱动传送带的同时，进行追踪拾取的功能。对于传送带，可更正确地追踪机器人。
作为间歇运转传送带使用时，如果将间歇运转伺服传送带追踪功能的选项用作连续运转传送带，将订购连续运转伺服传送带追踪功能选项。
要使用本功能，需要伺服传送带追踪拾取的选项。此外，为控制附加轴，需要独立附加轴的选项。
伺服传送带追踪拾取的选项中，包括多动作和连续回转功能。这些功能，用于使作为伺服传送带使用的附加轴在一定方向连续回转。

9.7.2 设定

请依照以下步骤，启动伺服传送带的系统。

9.7.2.1 附加轴设定
9.7.2.2 伺服传送带设定
9.7.2.3 伺服传送带用 TP 程序的生成
9.7.2.4 追踪参数设定表的设定
9.7.2.5 主程序的创建例

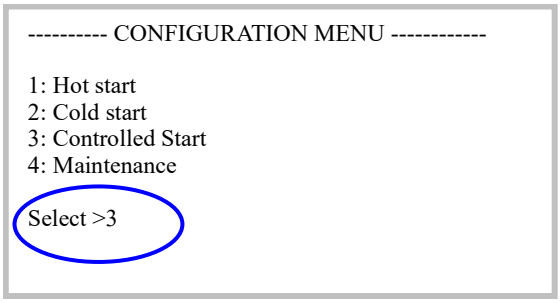
9.7.2.1 附加轴设定

附加轴的设定，在控制启动菜单的机器人设定画面中进行。

操作 控制启动

步骤

1 同时按住［PREV］（前一步）键和［＞（下一页）］键，使电源开启。
请按住键一段时间。
2 显示以下画面后，选择［Controlled Start］（控制启动）。请按下“3”，按下［ENTER］键。稍等片刻后，显示控制启动菜单。



操作 机器人设定画面

步骤

1 按下［MENU］键，选择［维护］。请将光标对准［维护］，按下［ENTER］键。显示机器人的设定画面。

2 请将光标对准［Independent Axes］的行，按下 F4［手动］。

3 显示独立附加轴的设定菜单。要追加轴时，选择［Add Axis］。请按下“2”，按下［ENTER］键。显示监视的设定画面。如果轴的设定结束，则返回该画面。要结束设定，同样请选择［4. EXIT］。

```
*** Group 2 Total Axes Installed =  0
  1. Display/Modify Nobot Axis 1~4
  2. Add Axis
  3. Delete Axis
  4. EXIT
 Select Item? 2
```

注释
请参照伺服电机选定结果，依照以下步骤进行各项目的设定。

4 请设定作为附加轴使用的伺服电机的种类。请选择［Standartd Method］，依次选择 MOTOR SIZE 和 MOTOR TYPE。

```
-- MOTOR SELECTION

  1: Standard Method
  2: Enhanced Method
  3: Direct Entry Method
Select ==> 1
```

```
      MOTOR SIZE （Beta standard, Beta is）

80. biS0.2     84. biS1     88. biS12
81. biS0.3     85. biS2     89. biS22
82. biS0.4     86. biS3
83. biS0.5     87. biS6
  0. Next page
Select ==> 85
```

```
      MOTOR TYPE

1. /2000      11. /4000
2. /3000      12. /5000
              13. /6000
Select ==> 11
```

例如： 选择 β *i*S2/4000 时，选择 85 后，选择 11。

5 请设定放大器的最大电流。例如，最大电流为 20A 的情况，请选择 10。

```
CURRENT LIMIT FOR AMPLIFIER

2.   4A        10.  20A
5.  40A        12. 160A
7.   80A

Select==> 10
```

6　轴的类型，请设定［Rotary Axis］（旋转轴类型）。

```
-- INDEPENDENT AXES TYPE  -
1. Linear Axis
2. Rotary Axis

Select? 2
```

7　请设定传送带的齿轮比。

```
-- GEAR RATIO --
Enter Gear Ratio?
```

（a）　“间歇运转传送带”设定时

请设定移动传送带 1 个间隔所需的电机的旋转数。通过这种设定，360°移动附加轴，则传送带移动 1 个间隔。

间隔　　使传送带移动 1 个容器时，传送带的移动距离

容器　　以一定的距离分隔传送带的间隔

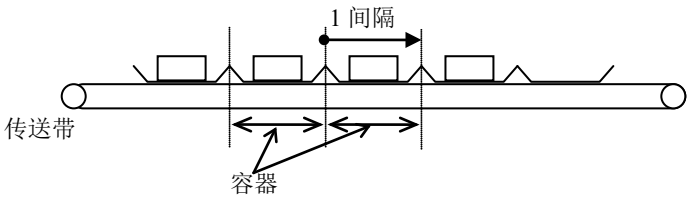


图 9.7.2.1 (a) 传送带上的容器和间隔

电机“M”旋转时，设传送带为“N”旋转，使用下面的式子求齿轮比。请输入求得的值。

$$\text{齿轮比} = \frac{M}{N \times \text{传送带上的容器数}}$$

- 从伺服传送带的设定画面上设定的值（电机侧齿轮数等），通过下面式子可计算齿轮比。请参照 9.7.2.2。

$$\text{齿轮比} = \frac{\text{转轴输入侧齿轮数} \times \text{传送带皮带孔数}}{\text{电机侧齿轮数} \times \text{转轴输出侧齿轮数} \times \text{伺服传送带容器数}}$$

（b） “连续运转传送带”设定时

请设定使传送带移动任意距离所需的电机的旋转数。通过这种设定，360°移动附加轴，则传送带只移动该距离。
电机“M”旋转时，设传送带为“N”旋转。从传送带 1 周的皮带的齿轮数，以及某距离的传送带皮带齿轮数，通过下面式子求齿轮比。

$$\text{齿轮比} = \frac{\text{M x某距离的皮带的齿轮数}}{\text{N x传送带 1 周的皮带齿轮数}}$$

- 从伺服传送带的设定画面上设定的值（电机侧齿轮数等），通过下面式子可计算齿轮比。请参照 9.7.2.2。

$$\text{齿轮比} = \frac{\text{转轴输入侧齿轮数 x 传送带齿轮}}{\text{电机侧齿轮数 x 转轴输出侧齿轮}}$$

8 请作为附加轴的最大速度设定标准值。请选择［2 No Change］（不变更）。

```
--MAX JOINT SPEED SETTING --
  Suggested Speed = 800.000(deg/s)
  (Calculated with Max Motor Speed)

Enter (1:Change, 2:No Change)? 2
```

9 请定义电机的旋转方向和传送带的移动方向。
电机正旋转时，传送带向通常令其动作的方向动作的情况，请设定“1”。
电机正旋转时，传送带向通常令其动作的方向的相反方向动作的情况，请设定“2”。

```
-- MOTOR DIRECTION
INDEPENDENT AXES 1 Motion Sign = TRUE
Enter (1:TRUE, 2:FALSE)?
```

从法兰盘正面看电机，逆时针方向是电机的正旋转方向。

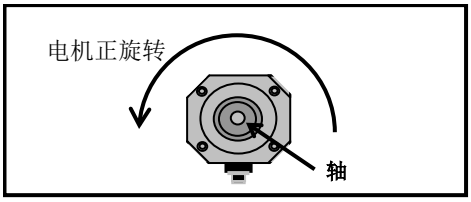


图 9.7.2.1 (b) 从法兰盘正面看电机的正旋转方向

10 设定附加轴的动作范围的上限和下限。
请设定上限 180（deg），下限-180（deg）。

```
-- UPPER LIMIT --
Enter Upper Limit ( deg)? 180
```

```
--LOWER LIMIIT--
Enter Lower Limit ( deg)? -180
```

11 请设定零点标定位置。
请在附加轴的动作范围内，制定能进行零点标定的机械位置。

通常，请设定为 0。

```
--MASTER POSITION －
Enter Master Position (deg)? 0
```

12 请设定加减速时间常数。加减速时间常数，是速度启动所需时间的设定。
以 2 段进行加减速。分别设定第 1 段和第 2 段的时间常数。

首先，设定第 1 段的时间常数。如果可以直接是标准值，则输入“2”，需变更时间常数时，输入“1”。

```
-- ACC/DEC TIME--
Default Value of acc_time1 = 384(ms)
Enter (1:Change, 2:No Change)?
```

变更值时，请输入第 1 段的时间常数。

```
Enter Accel Time 1 (ms)?
```

然后，设定第 2 段的时间常数。请设定为第 1 段时间常数的一半的值。如果可以直接是标准值，则输入“2”，需变更时间常数时，输入“1”。

```
Default value of acc_time2 = 192 (ms)
Enter (1:Change, 2:No Change)?
```

变更值时，请在以下画面设定第 2 段的时间常数。

```
Enter Accel Time 2 (ms)?
```

13 请设定最小时间常数。
即使通过加速度（ACC）指令较快计算加减速时间常数，也不得使实际的加减速时间常数比该值小。这称为最小时间常数。

请设定通过步骤 12 设定的第 1 段和第 2 段的时间常数合计值。

```
-- MIN_ACCEL TIME --
Default Value of min_acctime = 384(ms)
Enter (1:Change, 2:No Change)?
```

变更值时，请在以上画面输入“1”，在以下画面设定最小时间常数。

```
Enter Minimum Accel Time (ms)?
```

14 设定惯性比。惯性比表示相对电机旋转的负载的大小。
（需要根据伺服传送带的规格，事先求得惯性比。）

```
-- LOAD RATIO --
Load Ratio = (Load Inertia + Motor Inertia) / Motor Inertia

Enter Load Ratio -> (0:None 1->5:Valid)
```

15 设定放大器编号。

```
--SELECT AMP NUMBER--
Enter amplifier number (1-56)?
```

16 请设定在附加轴中使用的放大器的种类。

```
-- SELECT AMP TYPE --
1.      A06B-6107 series 6 axes amplifier
2.      A06B-6117 series Alpha i amp. or
A06B-6130 series Beta i amp.

Select?
```

17 定制动器编号。

```
--BRAKE SETTING --
Enter Brake Number (0~16)?
```

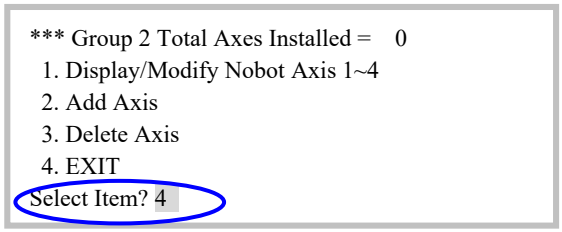
18 请进行制动器控制的设定。
所谓制动器控制，是当轴不动作一定时间时，自动进行制动的功能。未进行程序运行或点动时，只要程序中因待机指令等使轴不动作时，制动器控制也动作。

```
--SERVO TIMEOUT --
Servo off is Disable
Enter (1:Enable, 2:Disable)?
```

请设定在制动器控制有效时，轴停止几秒后进行制动。
可设定为 0 秒至 30 秒直接的值。

```
-- SERVO TIMEOUT VALUE --
Enter Servo Off Time? (0.0~30.0)
```

19 上述项目设定结束后，显示独立附加轴的设定菜单。
请选择［EXIT］，结束设定。



20 结束后，请按下［FCTN］（辅助）键，选择［冷开机］（Coldstart），进行冷启动。

9.7.2.2 伺服传送带设定

请以下面步骤进行伺服传送带的设定。这里进行的设定，自动反映于编码器的设定和连续回转功能的设定。

注释
可在伺服传送带的设定画面中，变更使用的标记。

操作 伺服传送带设定

步骤

1 按下［MENU］键，显示菜单。
2 选择［设置］。
3 按下 F1［类型］，显示菜单。
4 选择［伺服传送带］。显示伺服传送带的设定画面。

5 请在［1 编码器编号］中设定作为伺服传送带设定的编码器的编码器编号。
6 ［2 伺服传送带类型］中，根据订购的伺服传送带追踪功能选项进行以下设定。无法更改该项目。

间歇运转：每隔一定距离反复停止、移动的传送带。
订购了间歇运转伺服传送带追踪功能选项时被选择。

连续运转：以一定速度连续动作的传送带。
订购了间歇运转伺服传送带追踪功能选项时被选择。

7 请在［6 追踪参数设定表］中，设定使用伺服传送带的追踪参数设定表的编号。

8 将光标对准［2 伺服传送带类型］的行，按下 F2［详细］或［ENTER］键，则显示详细设定画面。显示的详细设定画面，根据传送带的类型不同而不同。

伺服传送带类型选择“间歇运转”的情况

9 显示以下的详细设定画面。由于不显示 F1［类型］，因此请在移动至其他画面时，按下［PREV］键。

10 请在［1 机器人组］中设定驱动传送带的附加轴的组编号和轴的编号。

11 请分别以整数设定［2 电机齿轮齿数］、［3 输入侧齿轮齿数］、［4 输出侧齿轮齿数］、［5 传送带齿数］。请在［6 传送带上的容器数］中设定传送带上的容器数。它们的关系如下图所示。通过输入这些值，设定伺服传送带的齿轮比。

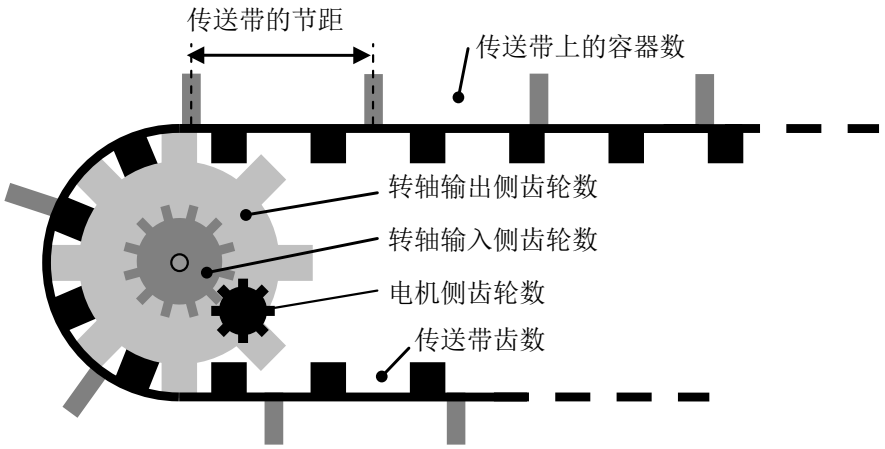


图 9.7.2.2 (a) 电机、转轴和传送带的关系

※ 根据电机和传送带的旋转数的关系进行设定时，请进行如下设定。电机“M”旋转，传送带“N”旋转时，请将［2 电机齿轮齿数］设定为“N”，将［3 输入侧齿轮齿数］设定为“M”，在［4 输出侧齿轮齿数］和［5 传送带齿数］中设定“1”。但是，请使 M 和 N 为整数。

12 请在［7 传送带的节距 （mm）］中设定 1 间隔的伺服传送带的移动距离。

13 请在［8 传送带驱动用 DI］中，设定用于驱动伺服传送带而使用的 DI 的编号。

14 请在［9 触发后延迟时间（ms）］中，设定从用于驱动伺服传送带的 DI 触发后，至实际驱动传送带的延迟时间。

15 在［10 传送带开始 DO］中，要运行伺服传送带用的 TP 程序时，需输出 DO 的情况，请输入输出所使用的 DO 的编号。同时也请参考后面叙述的伺服传送带用的 TP 程序 INDXG*.TP。

16 请设定［11 传送带标识 1］伺服传送带中使用的标记的编号。（请确认指定的标记未被用于其他目的。）

17 请设定［12 传送带标识 2］伺服传送带中使用的标记的编号。（请确认指定的标记未被用于其他目的。）

18 上述设定后，请按下 F2［执行］，进行电源的 OFF/ON。

注释
通过上述设定，还进行驱动传送带的轴的连续回转功能的设定，以及指定编号的编码器的设定。

伺服传送带类型选择“连续运转”的情况

9 显示以下的详细设定画面。由于不显示 F1［类型］，因此请在移动至其他画面时，按下［PREV］键。

10 请在［1 机器人组］中设定驱动传送带的附加轴的组编号和轴的编号。

11 请分别以整数设定［2 电机侧齿轮齿数］、［3 输入侧齿轮齿数］、［4 输出侧齿轮齿数］、［5 传送带齿数］。请在［6 传送带长度（mm）］中设定对应传送带侧齿轮数的长度。它们的关系如下图所示。通过输入这些值，设定伺服传送带的齿轮比。

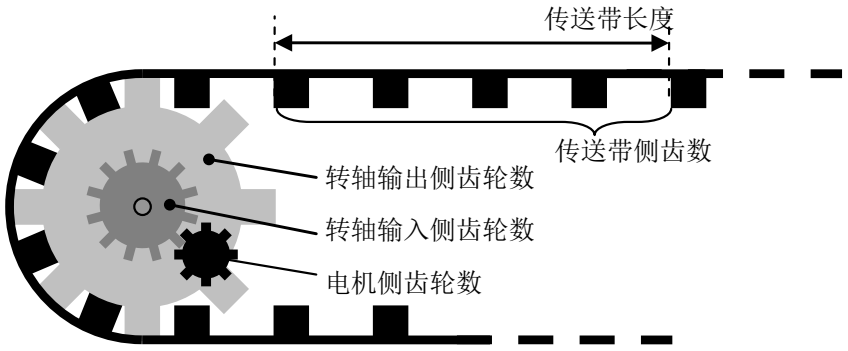


图 9.7.2.2 (b) 电机、转轴和传送带的关系

※ 根据电机和传送带的旋转数的关系进行设定时，请进行如下设定。电机“M”旋转，传送带“N”旋转时，请将［2 电机侧齿轮齿数］设定为“N”，将［3 输入侧齿轮齿数］设定为“M”。请在［4 输出侧齿轮齿数］中设定传送带 1 周的传送带的齿轮数，在［5 传送带侧齿数］中设定对应传送带长度的传送带的齿轮数。请使输入的值整数。

12 请在［7 传送带驱动用 DI］中，设定用于驱动伺服传送带而使用的 DI 的编号。

13 请设定［8 传送带标识］伺服传送带中使用的标记的编号。（请确认指定的标记未被用于其他目的。）

14 上述设定后，请按下 F2［执行］，电源的 OFF/ON。

注释
通过上述设定，还进行驱动传送带的轴的连续回转功能的设定，以及指定编号的编码器的设定。

9.7.2.3 伺服传送带用 TP 程序的生成

由于作为附加轴设定，因此需事先准备用于使伺服传送带动作的 TP 程序。通过以下步骤，可生成使伺服传送带动作的 TP 程序。

操作 伺服传送带用 TP 程序的生成

步骤

注释
在生成伺服传送带驱动用的 TP 程序之前，请在［动作性能/负载］画面设定追踪中使用的负载，并切换为该负载。

1 在伺服传送带的设定画面，请将光标对准［程序创建/速度设定］，按下 F2［详细］或［ENTER］键。

则显示以下画面。由于不显示 F1［类型］，因此请在移动至其他画面时，按下［PREV］键。

2 请在［1 伺服传送带速度 （part/min）］中，以整数设定伺服传送带的速度。
请设定 1 分钟内移动的间隔数。

3 请在［2 伺服传送带停止时间（ms）］中，以整数设定伺服传送带的停止时间。
请设定“-1”或 0 以上的值。计算的伺服传送带的速度类型，有以下不同。
作为间隔进给的传送带使用时，请对伺服传送带的停止时间设定 0 以上的值。传送带以一定速度动作时，请将伺服传送带的停止时间设定为“-1”。

1 间隔的传送带的移动时间，根据伺服传送带速度和停止时间计算。
- 伺服传送带停止时间≧0 的情况
  1 间隔的传送带的移动时间［ms］=（60*1000/伺服传送带速度）－伺服传送带停止时间

- 伺服传送带停止时间=-1 的情况
  1 间隔的传送带的移动时间［ms］=（2*60*1000/伺服传送带速度）

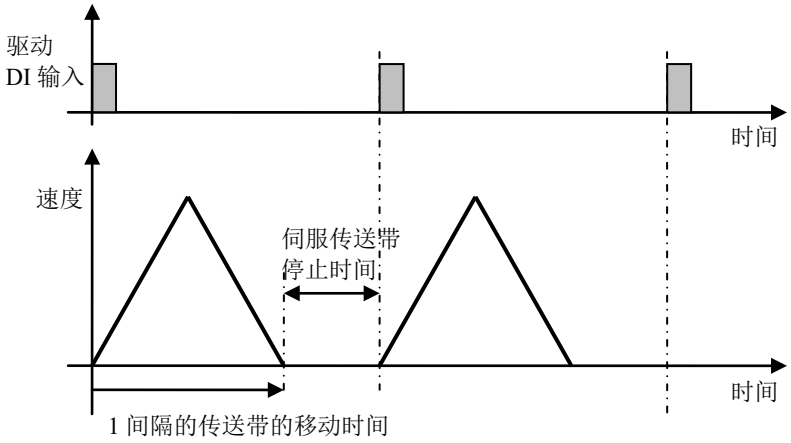


图 9.7.2.3 (a) 速度类型（伺服传送带停止时间≧0 的情况）

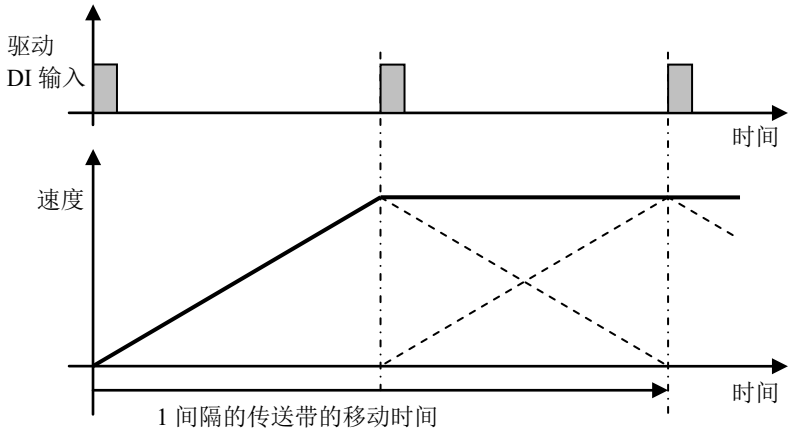


图 9.7.2.3 (b) 速度类型（伺服传送带停止时间=-1 的情况）

4 请在［3 伺服传送带寄存器开始编号］中，设定在 TP 程序内使用的数值寄存器的开始编号。TP 程序内，使用 2 个数值寄存器。
例如设定为“60”，则使用 R［60］和 R［61］。R［60］，用于设定 DO 编号，DO 编号是用于确认传送带移动至初始位置的编号。R［61］用于对传送带的间隔数进行计数。请参照后面叙述的样本程序。

5 上述设定后，请将光标对准［4 生成伺服传送带程序］，按下 F2［创建］。在正常处理的情况，显示“完成”，生成 INDXG*.TP（*是伺服传送带的组编号。如果伺服传送带是第 2 组，则为 INDXG2.TP）。

注意事项

- INDXG*.TP 是系统的默认程序，因此请不要直接变更。请务必变更为其他名称后使用。
- 根据设定不同，会发生显示“工件速度超过允许范围的最大值”的信息的情况。这表示，从伺服传送带的设定计算所得的加速度，比传送带、机器人的容许加速度大，这种情况，不生成 TP 程序。请变更伺服传送带速度、伺服传送带停止时间。
- 如果已经存在有 INDXG*.TP， 需要变更程序的情况，请按住［SHIFT］，并按下 F3［替换］。在正常处理的情况，显示“完成”，通过现在的设定生成的 TP 程序，将覆盖已存在的 INDXG*.TP。
  - 仅按下 F3［替换］的情况，会显示“按住 Shift+替换键,替换程序”的信息。不会覆盖 TP 程序。请按住［SHIFT］，并按下 F3［替换］。
  - 在选择 INDXG*.TP 的状态下，不能更新 INDXG*.TP。显示“处理失败”信息。
  - 根据设定不同，会发生显示“工件速度超过允许范围的最大值”的信息的情况。不会覆盖 TP 程序。请变更伺服传送带速度、伺服传送带停止时间。

样本程序

以下是自动生成的 TP 程序 INDXG*.TP 的例。

例:INDXG*.TP

```
1:  J P[1] 50% CNT0;                                  至初始位置的动作
2:     IF R[60:G2 Ready DO]=0, JMP LBL[3] ;
3:     DO[R[60]]=ON;                          使 R[60]中指定的 DO 为 ON
4:     LBL[3] ;
5:     R[61:G2 cur slot ID]=0;                        以“0”复位 R[61]
6:     LBL[1] ;
7:     $INDEXER[1].$INDEX_MV=1 ;                     用于进行驱动用 DI 的输入等待的设定
8:  J P[2] 180msec CNT100 INC ACC66;                  1 间隔的动作
9:     $INDEXER[1].$INDEX_MV=0 ;                     用于进行驱动用 DI 的输入等待的设定
10:    R[61:G2 cur slot ID]=R[61:G2 cur slot ID]+1;   传送带动作次数的计数
11:    IF R[61:G2 cur slot ID]=32, JMP LBL[2] ;       传送带 1 周后，至 LBL[2]
12:    JMP LBL[1] ;
13:    LBL[2] ;
14: J P[1] 100% CNT100      ;                         用于解除误差的动作
15:    R[61:G2 cur slot ID]=0      ;                  以“0”复位 R[61]
16:    JMP LBL[1] ;
```

```
/POS
P[1:“” ]{
GP2:
UF : 0, UT : 1,
J1=        0.000 deg
};
P[2:“” ]{
GP2:
UF : 0, UT : 1,
J1=     360.000 deg
};
```

- 第 1 行的动作指令，是用于使传送带移动至初始位置的语句。
- 第 2 至第 4 行，输出用于确认传送带移动至初始位置的 DO。
- 第 8 行的动作指令，是用于使传送带移动 1 间隔的语句。作为附加轴，动作 360 度。
- 第 7 行的$INDEXER［*］.$INDEX_MV=1 和第 9 行的$INDEXER［*］.$INDEX_MV=0 的设定，是用于监视作为伺服用 DI 设定的 DI 的输入的设定（*是伺服传送带的编号）。DI 从 OFF 至 ON 期间，不执行第 8 行的动作。
- 通过第 11 行的“IF”，进行传送带是否运转 1 周的确认。该例中，以 32 间隔为传送带运转 1 周。
- 第 14 行的动作指令，是用于消除移动时的误差的语句，传送带每运转 1 周就执行 1 次。
- R［60］、R［61］以伺服传送带数值寄存器开始编号进行设定。R［60］，用于设定 DO 编号，DO 编号是用于确认传送带移动至初始位置的编号。R［61］用于对传送带的间隔数进行计数。

操作 “连续运转”伺服传送带用 TP 程序的生成

步骤

1 在伺服传送带的设定画面，请将光标对准［程序创建/速度设定］，按下 F2［详细］或［ENTER］键。

则显示以下画面。由于不显示 F1［类型］，因此请在移动至其他画面时，按下［PREV］键。

2 请在［1 传送带速度（mm/sec）］中，以正的整数设定伺服传送带的速度。

注释
这里设定的速度，是通过 TP 程序使传送带动作时的速度。变更速度时，请不要直接变更 TP 程序，而改变该值。

3 上述设定后，请将光标对准［2 创建传送带程序］，按下 F2［创建］。在正常处理的情况，显示“完成”，生成 CNVYG*.TP（“*”是伺服传送带的组编号。如果伺服传送带是第 2 组，则为 CNVYG2.TP）。

注意事项

- CNVYG*.TP 是系统的默认程序，因此请不要直接变更。请务必变更为其他名称后使用。
- 如果已存在有 CNVYG*.TP，则不显示 F2［创建］，而显示 F3［替换］。如果按住［SHIFT］键，同时按下 F3［替换］，则再次生成程序。在正常处理的情况，显示“完成”将覆盖已存在的 CNVYG*.TP。
  - 仅按下 F3［替换］的情况，会显示“按住 Shift+替换键,替换程序”的信息。不会覆盖 TP 程序。请按住［SHIFT］，并按下 F3［替换］。
  - 在选择 CNVYG*.TP 的状态下，不能更新 CNVYG*.TP。
- 变更传送带的速度的情况，请变更［1 传送带速度 （mm/sec）］的值。无需为了变更传送带的速度，编辑 TP 程序。［1 传送带速度 （mm/sec）］的值即使 TP 程序运行中，也会反映到传送带的速度。如果指定的值较大，则显示“传送带速度超过允许范围”的信息，速度不会被变更。

样本程序

以下是自动生成的 TP 程序 CNVYG*.TP 的例。

例:CNVYG*.TP

```
1:    LBL[1] ;
2:    $INDEXER[1].$INDEX_MV=1 ;                用于进行驱动用 DI 的输入等待的设定
3:  J P[2] 48msec CNT100 INC ACC100;             传送带的动作
4:    $INDEXER[1].$INDEX_MV=0 ;                用于进行驱动用 DI 的输入等待的设定
5:    JMP LBL[1] ;
```

- 第 2 行的$INDEXER［*］.$INDEX_MV=1 和第 4 行的$INDEXER［*］.$INDEX_MV=0 的设定，是为了作为连续运转的传送带而使用所必要的设定（*是伺服传送带的编号）。并且还是用于监视作为伺服用 DI 而设定的 DI 的输入的设定。DI 是 ON 时，持续执行第 3 行的动作。如果使 DI 为 OFF，则传送带停止。

9.7.2.4 追踪参数设定表的设定

追踪参数设定表的设定方法，与通常的追踪设定方法相同。请进行在伺服传送带的设定画面中指定的追踪参数设定表的设定。请参照“3.2.8 传送带工作站的设定”或“4.2.9 传送带工作站的设定”进行设定。

9.7.2.5 主程序的创建例

TP 程序的示教方法，与原来的追踪拾取功能几乎相同。需要以多任务功能启动用于使传送带动作的 TP 程序的部分，与原来的追踪拾取功能不同。

以下的主程序的例，使传送带动作的 TP 程序为 CONVEYOR.TP。使传送带动作的 TP 程序 （INDXG*.TP、CNVYG*.TP） ，可通过“9.7.2.3 伺服传送带用 TP 程序的生成”的步骤生成。

主程序的例

```
1:  J P[1] 100% FINE;
2:    LINE [1] =ON ;
3:    RUN CONVEYOR ;                  运行用于驱动传送带的程序
4:    LBL[1] ;
5:    WAIT DI[10]=ON;
6:    LINECOUNT [1] R[1] ;
7:    SETTRIG LNSCH [1] R[1] ;
8:    CALL TRAK      ;                调出追踪用程序
9:    CALL NORM      ;                调出通常动作的程序
10:   JMP LBL[1] ;
```

- 第 1 行，使机器人移动至初始位置。
- 第 3 行，以“RUN”（执行）指令运行用于使传送带动作的程序（上述例中为 CONVEYOR.TP）。
- 第 5 行，进行传送带上传送工件的检出等待。DI［10］为检出工件的输入信号。
- 第 6、7 行，在追踪动作前，进行触发值的取得和设定。
- 第 8 行，“调出”（CALL）追踪用的程序。
- 第 9 行，“调出”（CALL）通常动作的程序。

9.7.3 伺服传送带用功能

9.7.3.1 伺服传送带用的动态误差的微调整参数

在伺服传送带追踪功能中，使用以下参数代替在“6.2 机器人设定”中所说明的［动态追踪误差调整］，调节追踪动作的动态误差。

关于调节方法，请参阅“Tracking OPERATOR'S MANUAL B-83474EN”中的“FINE TUNING HIGH SPEED ACCURACY”。可按伺服传送带设定。

$SLTK_GRP［伺服传送带的组编号］.$SRVO_DELAY

9.7.3.2 等待伺服传送带停止的功能

本功能无效的情况（标准为无效），无论伺服传送带的动作如何，机器人开始或结束追踪动作。因此，例如伺服传送带在加速过程中，如果机器人开始追踪动作，则为了赶上传送带，机器人的加速度加大，可能发生动作困难的情况。如果使本功能有效，则在伺服传送带停止状态下，机器人开始或结束追踪动作。由此，追踪动作开始、结束时的机器人的加速度受到控制。

注释
在本功能有效的情况下，如果没有伺服传送带停止的期间，则无法开始或结束追踪。

用于切换本功能的有效/无效的标记是$INDX_TRAK［参数设定表的编号］的 bit1、bit2。要使功能有效，请通过以下步骤，使这些值为 TRUE。

- $INDX_TRAK［参数设定表的编号］的 bit1 为 TRUE 的情况，在传送带停止状态下，开始追踪动作。
- $INDX_TRAK［参数设定表的编号］的 bit2 为 TRUE 的情况，在传送带停止状态下，结束追踪动作。

注释
要变更本功能的设定，请重新生成伺服传送带驱动用的 TP 程序。

操作 等待伺服传送带停止的功能设定

步骤

1 $INDX_TRAK［参数设定表的编号］的 bit2 设定

1.1 请确认设定。请将$INDX_TRAK［参数设定表的编号］的值除以“2”。
（“2”是 2 的 1 次方值。）
- 计算所得的值的整数部分是奇数时，bit1 为 TRUE。
- 计算所得的值的整数部分是偶数时，bit1 为 FALSE。

1.2 请根据现在的设定，变更$INDX_TRAK［参数设定表的编号］的值。
- bit1 为 TRUE，需要变更为 FALSE 时，请从$INDX_TRAK［参数设定表的编号］减去“2”。
- bit1 为 FALSE，需要变更为 TRUE 时，请对$INDX_TRAK［参数设定表的编号］加上“2”。

2 $INDX_TRAK［参数设定表的编号］的 bit2 设定

2.1 请确认设定。请将$INDX_TRAK［参数设定表的编号］的值除以“4”。
（“4”是 2 的 2 次方值。）
- 计算所得的值的整数部分是奇数时，bit1 为 TRUE。
- 计算所得的值的整数部分是偶数时，bit1 为 FALSE。

2.2 请根据现在的设定，变更$INDX_TRAK［参数设定表的编号］的值。
- bit2 为 TRUE，需要变更为 FALSE 时，请从$INDX_TRAK［参数设定表的编号］减去“4”。
- bit2 为 FALSE，需要变更为 TRUE 时，请对$INDX_TRAK［参数设定表的编号］加上“4”。

3 设定后，请重新开启控制装置的电源。

9.7.3.3 伺服传送带专用 KAREL 程序

可以使用下面的 KAREL 程序。

注释
也可以使用 *i*RPickTool 的 KAREL 程序。

通过使用这些，可不使用传感器，在每动作一定距离后取得触发值，并保存多个触发值。并且，还可以使用放弃线、停止/启动传送带、分配开始线的功能。

- SLTKINIT
- SLTKREST
- SLTKPSHQ
- SLTKPOPQ
- SLTKDELQ
- SLTKRSTQ

- SLTKGTPP

下面对各 KAREL 程序进行说明。然后，对放弃线、停止/启动传送带、分配开始线的功能进行说明。

注释
有关伺服传送带专用 KAREL 程序的功能与 *i*RPickTool 的功能不可以同时使用。使用 *i*RPickTool 的功能时，不是使用伺服传送带专用 KAREL 程序，而是使用 *i*RPickTool 的 KAREL 程序。

SLTKINIT

用于对触发值的信息的初始化。通常情况下，只需在系统启动时，执行 1 次该 KAREL 程序。

参数 1: 指定使用伺服传送带的参数设定表的编号。
参数 2: 指定记忆传送带的初始位置的位置寄存器的编号。
参数 3: 指定位置寄存器的编号。

现在的传送带的位置和初始位置（由参数 2 指定的位置寄存器的位置）的差，在参数 3 指定的位置寄存器中设定。设定的值，为 1 间隔内（0 至 360 度的范围）的值。动作开始时，可用于对准传送带的位置。

SLTKREST

伺服传送带中，1 间隔的距离已定，因此可每隔一定距离，作为触发值取得编码器的值。该程序用于设定成为触发基准的值。作为触发的基准值，设定 SLTKREST 运行时的编码器的值。传送带在初始位置动作后，需要执行 1 次。

参数 1: 指定使用伺服传送带的参数设定表的编号。

以下的例，是使用 SLTKINIT 在 PR［2］中设定现在的传送带的位置和初始位置的差。通过各轴动作，驱动传送带轴，使传送带移动至初始位置。通过 SLTKREST 设定触发的基准值。

```
1:   CALL SLTKINIT(1,1,2) ;          初始化和位置寄存器设定
2:  J PR[2] 50% CNT0 INC;            至初始位置的动作
3:   CALL SLTKREST(1) ;              触发的基准值设定
```

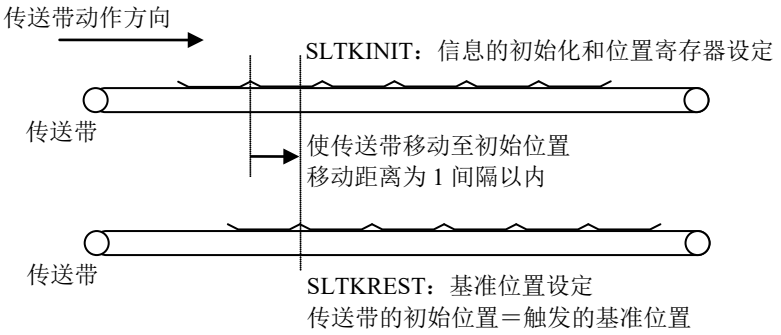


图 9.7.3.3(a) SLTKINIT, SLTKREST

SLTKPSHQ

使传送带只移动一定的间隔数后，用于保存触发值。

参数 1: 指定使用伺服传送带的参数设定表的编号。
参数 2: 指定传送带的间隔数。

可保存的触发值的数量，可以在下一个系统变量中变更。变更后，需要进行电源的 OFF/ON。
$SLTKSCH［n］.$QUE_SIZE: 标准值 20、最大值：100
n 是使用伺服传送带的参数设定表的编号

注释
如果保存的触发值的个数成为$SLTKSCH［n］.$QUE_SIZE 中指定的值，则删除保存的值中最老的触发值。

如果执行 SLTKPSHQ，从通过 SLTKREST 设定的触发的基准值和间隔数计算触发值并保存。第一次执行 SLTKPSHQ 时，触发的基准值作为触发值被保存。通常，传送带动作指定间隔数后执行。
例如，1 次间隔进给，保存 1 次触发值的情况，对第 2 个参数设定“1”。2 次间隔进给保存 1 次触发值的情况，对第 2 个参数设定“2”。下图是保存触发值的同时，进行 4 次间隔进给的情况。

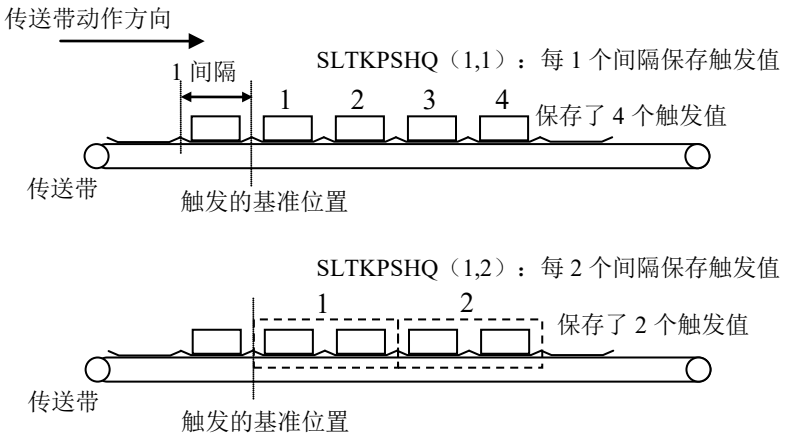


图 9.7.3.3 (b) SLTKPSHQ

SLTKPOPQ

用于取得保存的触发值的 KAREL 程序。如果运行，则取得保存的值中最老的触发值。取得的触发值被删除，无法再次取得。

参数 1: 指定使用伺服传送带的参数设定表的编号。
参数 2: 指定数值寄存器的编号。在指定编号的数值寄存器中设定触发值。
参数 3: 指定数值寄存器的编号。在指定编号的数值寄存器中设定状态。如果触发值的取得失败，则设定“0”以外的值。

下例，是保存了 4 个触发值的情况。

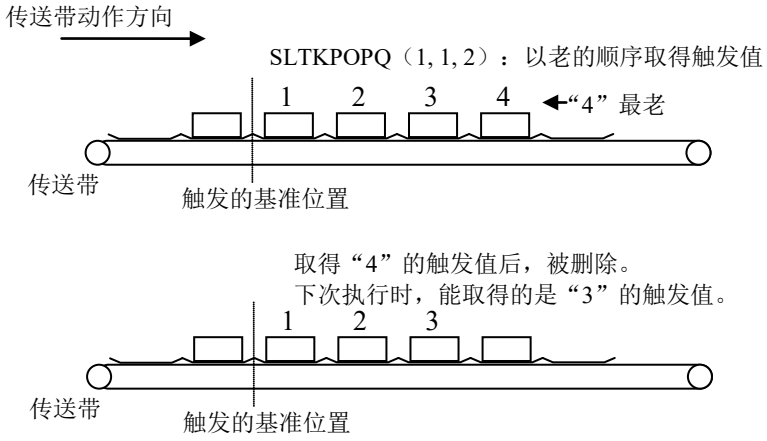


图 9.7.3.3 (c) SLTKPOPQ

SLTKDELQ

用于删除指定编号对应的触发值的 KAREL 程序。删除的触发值，无法通过 SLTKPOPQ 取得。

参数 1: 指定使用伺服传送带的参数设定表的编号。
参数 2: 指定需要删除的触发值的编号。

例如，参数 2 中设定“2”的情况，从最新的触发值起计算，去除第 2 个触发值。

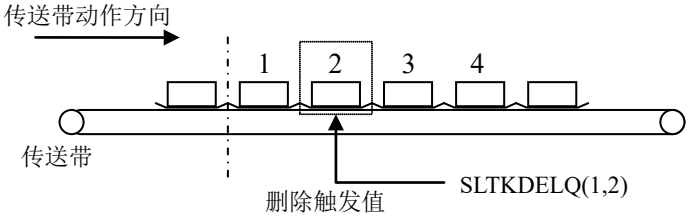


图 9.7.3.3 (d) SLTKDELQ

SLTKRSTQ

用于使停止/启动传送带（后面叙述）的功能中使用的 DO 为 OFF。也请参照停止/启动传送带的项。

参数 1: 指定使用伺服传送带的参数设定表的编号。

SLTKGTPP

将在保存的触发值中可取得的值中，对应最老的触发值的追踪坐标系上的 X 轴方向的位置返回的 KAREL 程序。

参数 1: 指定使用伺服传送带的参数设定表的编号。
参数 2: 指定数值寄存器的编号。设定计算触发值相对应的追踪坐标系上的 X 轴方向的现在位置的值。
参数 3: 指定数值寄存器的编号。在指定编号的数值寄存器中设定状态。如果位置的取得成功，则设定为“0”，如果失败，则设定为“0”以外的值。如果失败，参数 2 指定的数值寄存器的值不被更新。

例如下图，如果执行 SLTKGTPP，则计算最老的“4”的触发值相对应的位置。

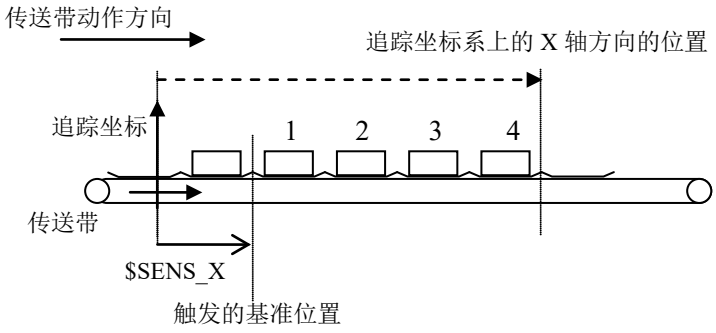


图 9.7.3.3 (e) SLTKGTPP

在可取得的触发中，使用最老一个进行位置计算。例如，如果通过 SLTKDELQ 删除“4”，则计算下一个的“3”的触发值相对应的位置。

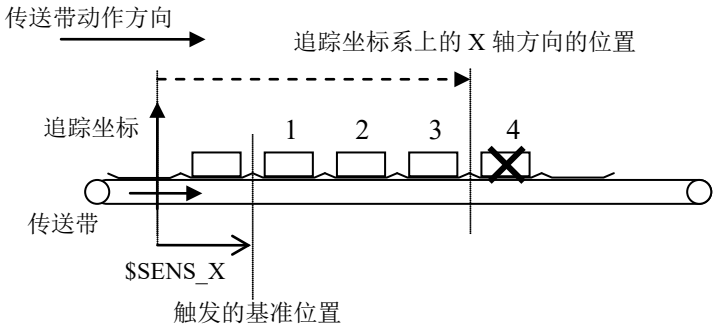


图 9.7.3.3 (f) SLTKGTPP 和 SLTKDELQ

如果设定了放弃线，例如“4”超过了放弃线的情况，则无法通过 SLTKPOPQ 取得“4”。这种情况，计算下一个的“3”的触发值相对应的位置。也请参照放弃线的项。

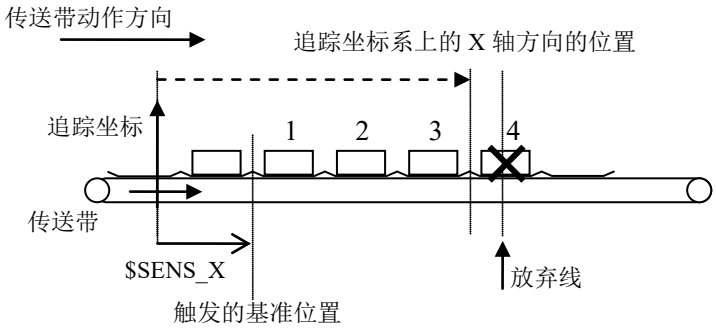


图 9.7.3.3 (g) SLTKGTPP 和放弃线

如果设定了停止/启动传送带，例如“4”超过了放弃线，则输出 DO，用于促进传送带的停止。停止/启动传送带的情况，即使超过了放弃线，也可通过 SLTKPOPQ 取得“4”的触发值。计算“4”的触发值相对应的位置。也请参照停止/启动传送带的项。

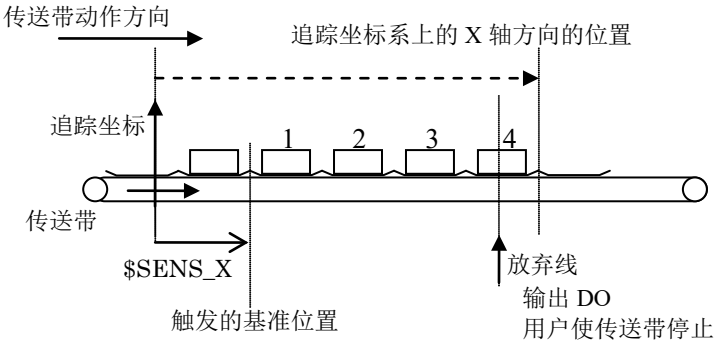


图 9.7.3.3 (h) SLTKGTPP 和停止/启动传送带

［放弃线］

触发值相对应的位置超过放弃线时，可跳过该触发值。放弃线将追踪范围的下方设定为基准。

使用时，需要设定以下系统变量。

$SLTKSCH［n］.$SLQ_ENABLE: 有效
$SLTKSCH［n］.$DISCARD_BND: 从追踪范围的下方的放弃线的位置［mm］
$SLTKSCH［n］.$SENS_X: 追踪坐标系 X 轴方向的触发的基准位置［mm］
n 是追踪参数设定表的编号

如果$SLTKSCH［n］.$DISCARD_BND 的值为负，则设定于下方的上游侧。通常，请设定为负值。
如果表示触发值的位置超过了放弃线，则无法通过 SLTKPOPQ 取得该触发值。用于判定是否跳过的触发值，使用保存的值中最早的值。
以下情况，触发值“4”所对应的位置超过了放弃线，因此跳过“4”。跳过“4”后，对触发值“3”相对应的位置进行跳过的判定。

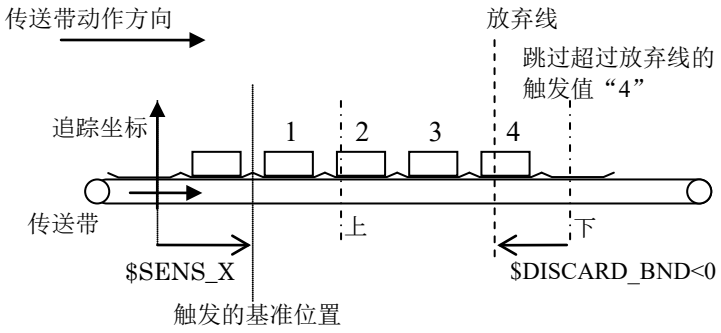


图 9.7.3.3 (i) 放弃线

［停止/启动传送带］

使停止/启动传送带有效的情况，如果触发值对应的位置超过了放弃线，则使指定的 DO 为 ON。请不要使用该信号，输入伺服传送带驱动用的 DO。请设定放弃线，使传送带能在停止的位置进行希望的追踪动作。

使用时，需要设定以下系统变量。

$SLTKSCH［n］.$SLQ_ENABLE:	有效
$SLTKSCH［n］.$DISCARD_BND:	从追踪范围的下方的放弃线的位置［mm］
$SLTKSCH［n］.$SENS_X:	追踪坐标系 X 方向的触发的基准位置［mm］
$SLTKSCH［n］.$STOPBELT:	有效
$SLTKSCH［n］.$STOP_DONUM:	使用的 DO 的编号

n 是追踪参数设定表的编号

使用停止/启动传送带的情况，即使超过了放弃线，停止/启动传送带的判定中使用的触发值被直接保存。DO 为 ON 期间（执行 SLTKRSTQ 之前），如果执行 SLTKPOPQ，则可以取得该触发值。如果 DO 为 OFF，则请务必使用 SLTKRSTQ。通过执行 SLTKRSTQ，下一个触发值被用于停止/启动传送带的判定。
以下情况，触发值“4”所对应的位置超过了放弃线，因此指定的 DO 为 ON。相应地使传送带停止。可通过 SLTKPOPQ 取得触发值“4”。在触发值“4”相对应的位置的处理结束后，执行 SLTKRSTQ，使指定的 DO 为 OFF。相应地使传送带的动作重新开始。SLTKRSTQ 执行后，有关触发值“3”相对应的位置，进行停止/启动传送带的判定。

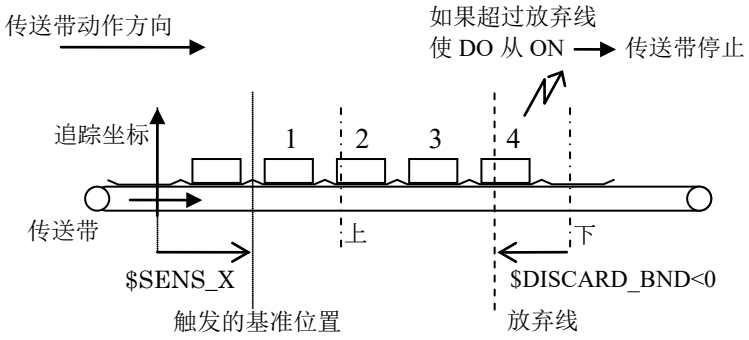


图 9.7.3.3 (j) 停止/启动传送带

注释
在相同的伺服传送带上，无法同时使用放弃线和停止/启动传送带。

分配开始线

触发值相对应的位置超过分配开始线之前，无法通过 SLTKPOPQ 取得该触发值。分配开始线将追踪范围的上方设定为基准。

$SLTKSCH［n］.$ALLOC_ENB:	分配开始线设定的有效/无效。通过标准值 FALSE 无效。
$SLTKSCH［n］.$ALLOC_BND:	从追踪范围的上方位置的分配开始线的位置［mm］
$SLTKSCH［n］.$SENS_X:	追踪坐标系 X 方向的触发的基准位置［mm］

n 是追踪参数设定表的编号。
如果$SLTKSCH［n］.$ALLOC_BND 的值为负，则在追踪范围的上方更上游侧设定分配开始线。通常，请设定为负值。在超过分配开始线之前，无法通过 SLTKPOPQ 取得该触发值。
例如，如果触发值“4”相对应的位置超过分配开始线，则可以通过 SLTKPOPQ 取得触发值“4”。

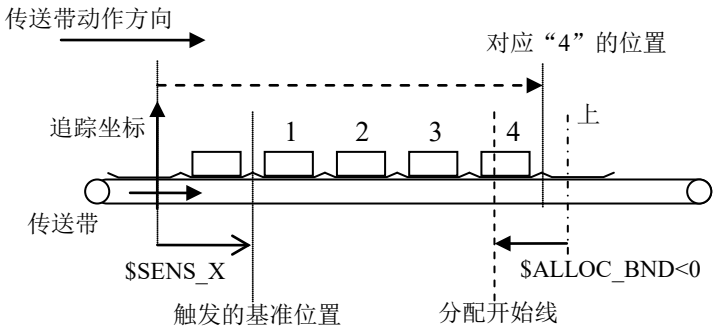


图 9.7.3.3 (k) 分配开始线

样本程序

以下表示使用停止/启动传送带的情况下，主程序、伺服传送带驱动用的程序（CONVEYOR.TP） 例。

- 追踪参数设定表使用“1”。
- 传送带的初始位置保存于 PR［1］。
- 传送带 2 次动作后，保存 1 次触发值。

主程序开头使用 SLTKINIT，使触发值的信息初始化。然后运行传送带驱动用的程序。

主程序的例

```
1:  J P[1] 100% FINE;
2:    LINE [1] =ON;
3:    CALLSLTKINIT(1,1,2) ;        初始化触发值信息。设定 PR[2]。
4:    RUN CONVEYOR ;               运行用于驱动传送带的程序
5:    LBL[1] ;
6:    WAIT    .10(sec) ;
7:    CALL SLTKPOPQ(1,1,2) ;             触发值的取得
8:    IF R[2]<>0,JMP LBL[1] ;      如果未能取得触发值，至 LBL[1]
9:    SETTRIG LNSCH [1] R[1] ;     作为触发值设定 R[1]
10:       CALL TRAK;                      调出追踪用程序
11:       CALL SLTKRSTQ(1) ;              使停止/启动传送带用的 DO 为 OFF
12:       CALL NORM;                      调出通常动作的程序
13:       JMP LBL[1] ;
```

- 在第 3 行运行 SLTKINIT，使触发值的信息初始化。此外，将从现在位置至 PR［1］中设定的初始位置的传送带的移动量，设定于 PR［2］。
- 在第 4 行“执行”（RUN）传送带驱动用的程序。
- 从第 5 行至第 8 行，每 100［ms］使用 SLTKPOPQ，进行触发值的取得。第 8 行确认 R［2］，对是否可取得触发值进行判定。如果未能取得触发值，则返回第 5 行。
- 如果在第 11 行，停止/启动传送带的 DO 为开启，则请进行关闭。

伺服传送带驱动用的程序，使用 SLTKPSHQ，传送带每动作 2 次，就取得触发值。R［3］用于对传送带的动作次数进行计数，根据 R［3］的值执行 SLTKPSHQ。

CONVEYOR.TP 的例子

```
1:  J PR[2] 50% CNT0 INC;                    至初始位置的动作
2:    CALL SLTKREST(1) ;                     设定触发值的基准值
3:    IF R[60:G2 Ready DO]=0, JMP LBL[3] ;
4:    DO[R[60]]=ON;                          使 R[60]中指定的 DO 为 ON
5:    LBL[3] ;
6:    R[61:G2 cur slot ID]=0;                以“0”复位 R[61]
7:    R[3]=0;                                R[3]用于 SLTKPSHQ 的计数
8:    LBL[1] ;
9:    $INDEXER[1].$INDEX_MV=1 ;              用于进行驱动用 DI 的输入等待的设定
10: J P[2] 180msec CNT100 INC ACC66;         1 间隔的动作
11:   $INDEXER[1].$INDEX_MV=0 ;              用于进行驱动用 DI 的输入等待的设定
12:   R[61:G2 cur slot ID]=R[61:G2 cur slot ID]+1;  传送带动作次数的计数
13:   R[3]=R[3]+1     ;                      传送带动作次数的计数
14:   IF R[61:G2 cur slot ID]=32, JMP LBL[2] ;
15:   JMP LBL[4] ;                           至第 19 行的 LBL[4]
16:   LBL[2] ;
17: J PR[1] 100% CNT100;                     用于解除误差的动作
18:   R[61:G2 cur slot ID]=0;                以“0”复位 R[61]
19:   LBL[4] ;
20:   IF R[3]<2, JMP LBL[1];                如果传送带未动作 2 次，则至 LBL[1]
21:   CALL SLTKPSHQ(1,2);                    如果传送带动作 2 次，则保存触发值
22:   R[3]=0;                                复位 R[3]
23:   JMP LBL[1] ;
```

```
/POS
P[2:“”]{
GP2:
UF : 0, UT : 1,
J1=    360.000 deg
};
```

- 从现在位置至 PR［1］中设定的初始位置的传送带的移动量被设定于 PR［2］，因此第 1 行，从现在位置向 PR［1］中设定的位置动作。
- 第 2 行，通过 SLTKREST 设定触发的基准值。
- 第 7、13、22 行进行计数（R［3］）的初始化、计数、复位。
- 第 17 行，传送带每动作 1 周，进行向用于消除误差的 PR［1］的动作。
- 第 20 行，确认传送带的动作次数，如果动作 2 次，则执行 SLTKPSHQ，保存触发值。
- 变更各触发的传送带的动作次数时，必须将第 20、21 行的“2”的位置变更为需要的数字。

9.7.4 伺服传送带追踪制约事项

- 驱动机器人和传送带的附加轴，需要成为其他的组。
- 对应多个伺服传送带。可最多追加至 4 轴（组 1：机器人、组 2～5：传送带轴）。
- 可以与使用原来的编码器的传送带同时使用。
- HDI 输入、高精度触发、DI 输入依旧可以使用。
- 可以和视觉追踪同时使用。但是对于被伺服的传送带，不能使用视觉追踪。
- 进行追踪的机器人中，不能使用连续回转功能。
- 不能使用原始路径恢复功能（通过暂时停止后的重新开始动作，恢复原来轨迹的功能）。
- 无法运行包括进行追踪的机器人和驱动传送带的附加轴的两者的动作组的程序。可通过 TP 程序的详细信息，确认动作组的设定。
- 在伺服传送带用的追踪程序中，请对所有的动作行附 Wjnt （无姿势）指令，或不附该指令。如果带 Wjnt 指令的动作行和不带 Wjnt 指令的动作行同时存在，则在追踪时，可能发生与目标位置的偏移。
- 在伺服传送带用的追踪程序中，请不要进行使用工具坐标系选择指令的工具的切换。进行工具的切换的情况下，可能发生追踪时与目标位置的偏移。
- 可以从设定画面变更伺服传送带功能中使用的标记。

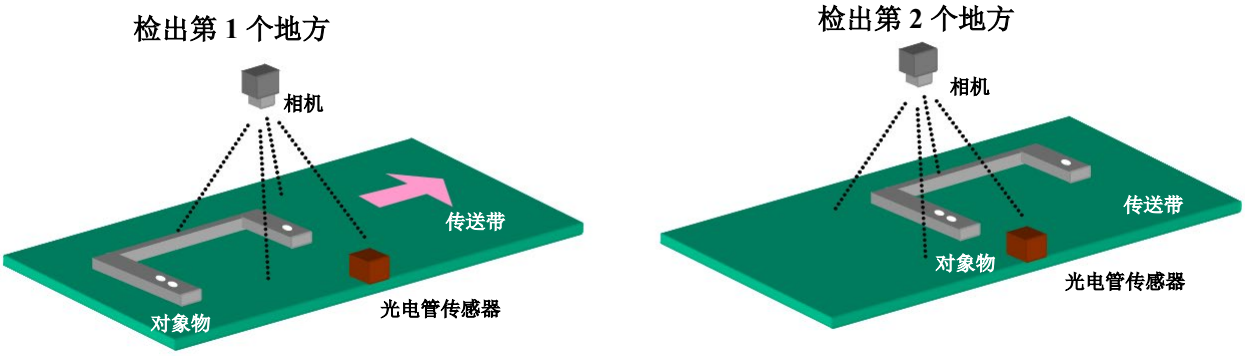
9.8 多视图

在本节中对使用多视图时的 iRPickTool 安装方法进行说明。

多视图用于在 1 次测量中不能检出全体对象物的情况。例如，车体面板等比较大的对象物被传送到传送带上的情况。

9.8.1 概要

使用多视图时，在下面所示的传送中检出对象物。
对象物通过光电管传感器检测到。检测到以后，如果对象物移动指定的距离，则自动进行第 1 次拍摄和检出。据此可检出对象物上的第 1 个特征。然后移动规定距离后，进行第 2 次拍摄和检出。据此可检出对象物上的第 2 个特征。传感器合成这 2 次的检出结果，计算对于对象物的补偿量。



9.8.2 制约事项

- 仅直线传送带可以使用。不能使用圆弧传送带和伺服传送带。
- 检出位置不在直线上，需要使用 2 台照相机用于多视图时，这些相机不能连接到分别的控制装置上。
- 多视图功能和托盘功能不能同时使用。
- 不能用 1 台控制装置同时运行单视图和多视图。
  但是，可以对多视图和单视图进行排他性切换使用。在 TP 程序中进行切换时，使用“6.8.4.1 PKSNENBSENS.PC”。例如，要将“SENS1”设为无效，将“SENS2”设为有效时，进行如下操作。

```
CALL PKSNENBSENS ('SENS1',Disable)
CALL PKSNENBSENS ('SENS2',Enable)
CALL PKSNSTART
```

  PKSNENBSENS.PC 中进行排他性地切换传感器后，运行 PKWCSTART.PC（或 PKSNSTART）。
- 不能用多视图同时检出 2 个以上的对象物。
- 1 个对象物的测量(相机视图 1 和相机视图 2 的测量)结束前，不能开始其他对象物的测量。

9.8.3 相机的准备

台数

使用 1 台或 2 台相机。
使用 1 台相机时，用相同的相机检出 2 次。此时，为了用 1 台相机检出第 1 个特征和第 2 个特征，需要将 2 个特征在相机的视野内传送。
使用 2 台相机时，其中一个相机检出第 1 个特征，另一个相机检出第 2 个特征。此时，可以根据 2 个特征的位置配置相机。

连接

使用 2 台相机时，将 2 台相机连接到相同机器人控制装置上。

配置

相机可以任意配置。

但是，按照“5.3 相机的设置和连接”中所描述的，光电管传感器检测到对象物后，相机拍摄之前，大约有 0.1 秒左右的延迟，因此，光电管传感器检测到对象物通过后，大约 0.1 秒后，使应该用相机检出的特征出现在相机视野的中心附近。

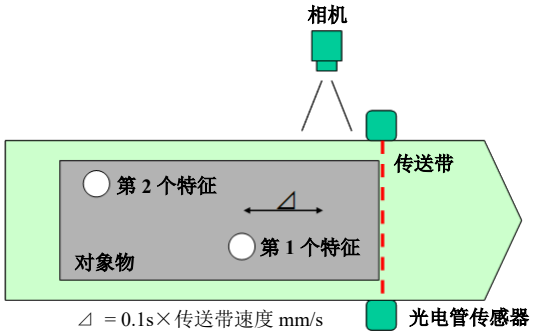


图 9.8.3(a)相机的配置示例

9.8.4 安装

传送带、相机数据的安装方法、机器人程序的创建和执行，与不是多视图时相同。此处对多视图固有的安装方法进行说明。

9.8.4.1 视觉设定

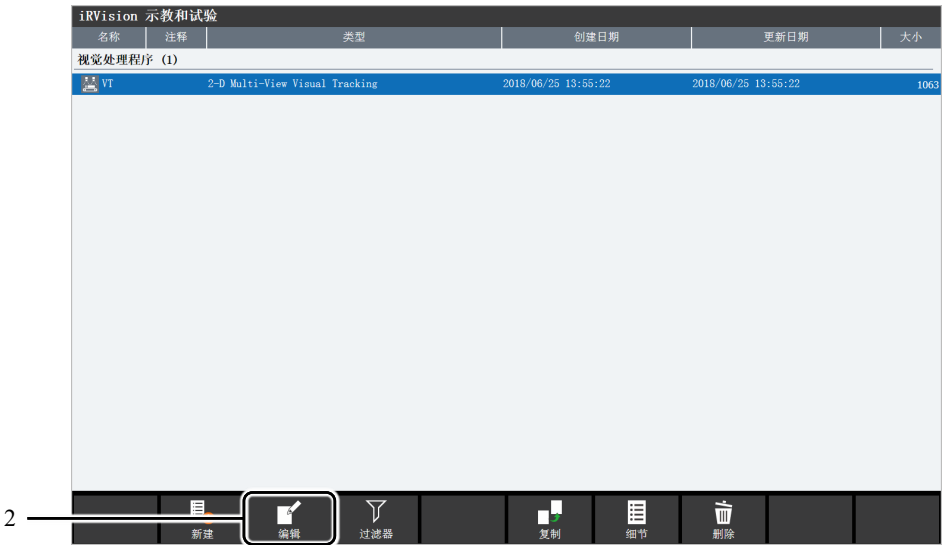
在多视图中，使用“2-D Multi-View Visual Tracking”的视觉程序。进行用于检出第 1 个特征的相机视图 1 和用于检出第 2 个特征的相机视图 2 的示教。

> **⚠注意**
> **需要将先检出的（第 1 个特征）设为相机视图 1，将之后检出的（第 2 个特征）设为相机视图 2。**

在以下传送中，按每个特征对相机视图进行示教。

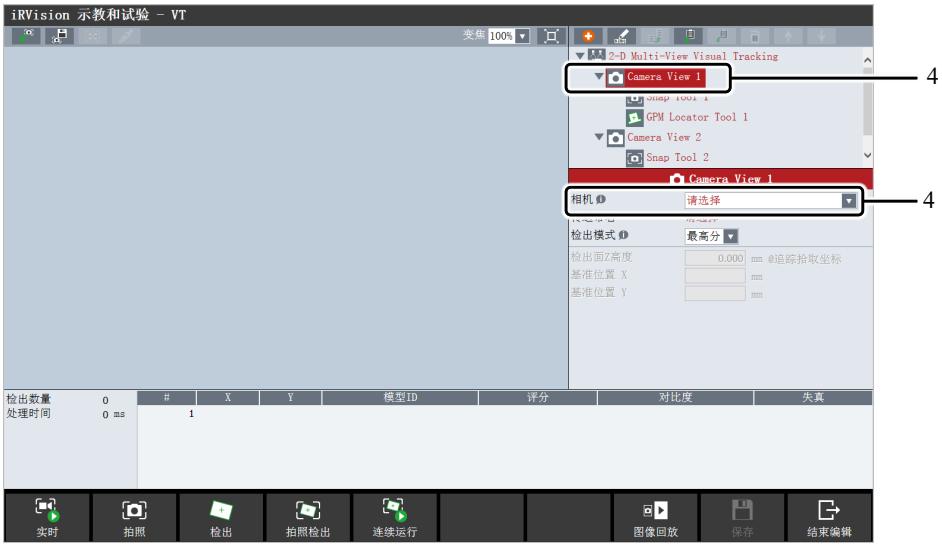
1 在 iRVision 的设定画面，新建“2-D Multi-View Visual Tracking”。

2　选择创建的视觉程序，按下 F3［编辑］。



3　安装工件时使第 1 个特征进入相机视图 1 的视野内。
　如果已供给方向已确定，将调整方向再安装。

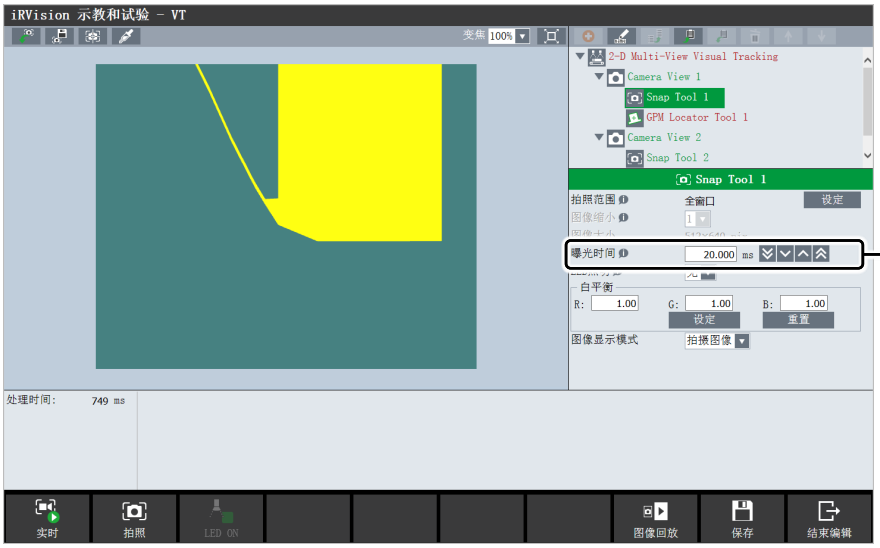
4　在［Camera View 1］中，从菜单中选择［相机］。



5 在［Camera View 1］的［Snap Tool 1］中输入［曝光时间］。
在设定曝光时间时，要使曝光中移动的工件图像偏差尽量小。
例如，考虑图像的偏差，传送带速度为 100 mm/s 时，为了将视觉检出时的位置偏差控制在 0.1 mm 以下，就要设定低于 1 ms 的值作为曝光时间。

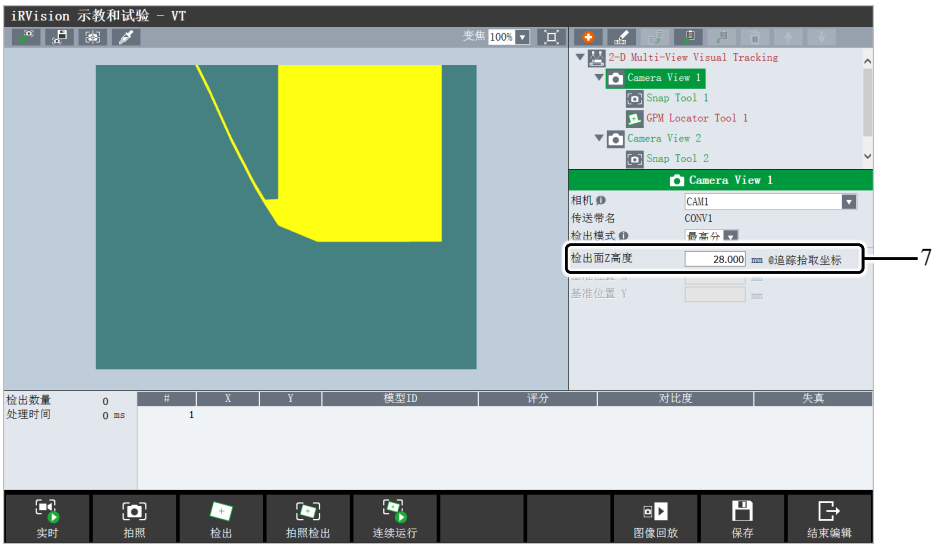
$$0.1(\text{mm}) \div 100(\text{mm/s}) = 1(\text{ms})$$

关于根据传送带速度计算合适的曝光时间，请参阅“2.5 适用探讨”中的“曝光时间”。



6 测量工件的厚度。
实测从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。

7 在［Camera View 1］中，在［检出面 Z 向高度］里输入工件厚度。
输入从示教追踪坐标系的面（点阵板表面）到具有视觉检出特征面的距离。



注释

［检出面 Z 向高度］是从追踪坐标系的视角看到的工件的高度。请用尺子等测量。
在点阵板表面对追踪坐标系进行示教时，“检出面 Z 向高度”如下图所示。

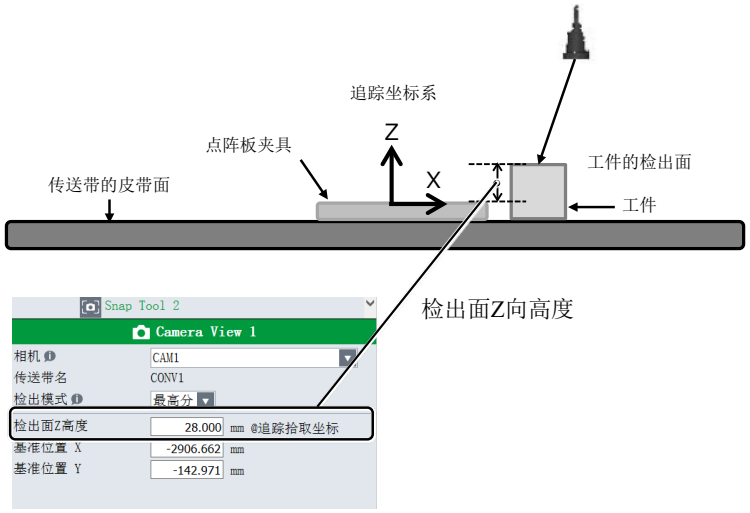


图 9.8.4.1(a)　系统与“检出面 Z 向高度”的关系

8　在［Camera View 1］中对［GPM Locator Tool 1］进行示教。关于图形匹配的设定步骤，请参照“3.2.7.2 视觉程序”。

9　安装工件，使第 2 个特征进入相机视图 2 的视野内。对于相机视图 2，进行与上述步骤 4 到 8 相同的设定。

9.8.4.2　传感器设定

从示教器的［MENU］键的［6 设定］中选择［iRPickTool］。
在显示的画面中，将光标与传感器对齐，在［触发条件］中选择［DI］，在［触发］中选择［使用多视图检出］，可见以下画面。

触发偏移 1、触发偏移 2

［触发偏移 1］是光电管传感器检测到对象物通过后，在相机视图 1 拍摄-检出第 1 个特征之前传送带移动的距离（mm 单位）。

［触发偏移 2］是光电管传感器检测到对象物通过后，在相机视图 2 拍摄-检出第 2 个特征之前传送带移动的距离（mm 单位）。

1 台相机时

例如，在 1 台相机中，进行下图所示的配置时，［触发偏移 1］和［触发偏移 2］将进行如下指定。

触发偏移 1 = D1(mm) - L1(mm) - 0.1(s)×传送带速度(mm/s)
触发偏移 2 = D2(mm) - L1(mm) - 0.1(s)×传送带速度(mm/s)

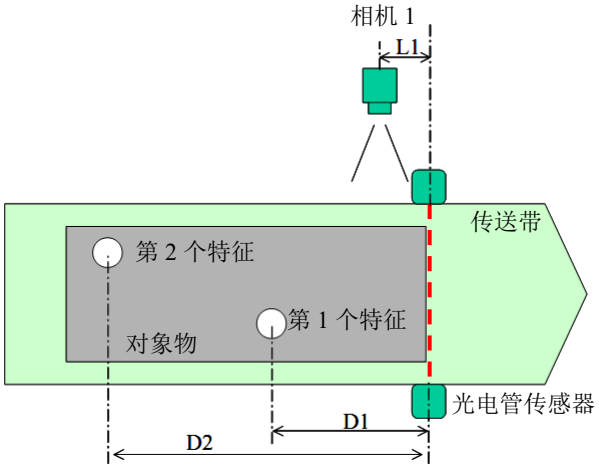


图 9.8.4.2(a)1 台相机时的配置示例

2 台相机时

在 2 台相机中进行如下配置，相机 1 检出第 1 个特征，相机 2 检出第 2 个特征时，［触发偏移 1］和［触发偏移 2］进行如下指定。

触发偏移 1 = D1(mm) - L1(mm) - 0.1(s)×传送带速度(mm/s)
触发偏移 2 = D2(mm) + L2(mm) - 0.1(s)×传送带速度(mm/s)

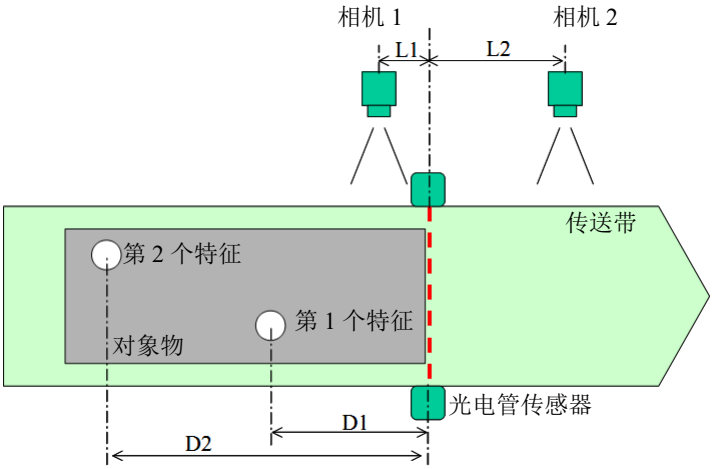


图 9.8.4.2(b)2 台相机时的配置示例

视觉程序

在［视觉程序］中，指定多台相机的二维视觉追踪的视觉程序名。

9.8.4.3 基准位置的设定、机器人的位置示教

根据向导的指示进行基准位置的设定。

1 与光电管传感器相比，将对象物放置于上游侧。
2 按下 F3 ［基准位置］，开始基准位置设定向导。则显示以下画面。

3 传送带运行，通过光电管传感器检测到对象物。对象物来到进行位置示教的机器人前面后，停止传送带。此时，成功检测到对象物后，对象物移动触发偏移 1 的量，自动执行第 1 个特征检出。此外，对象物移动触发偏移 2 的量后，自动执行第 2 个特征检出。
成功检测到 DI 后，显示 DI 检测后的对象物的移动距离。并且显示各相机视图的检出结果。双方的相机视图都检出成功后，显示合成的检出结果，F5 ［下一步］键有效。如按下 F5［下一步］，则显示以下画面。

便签
按照相机视图 1 中检出时的编码器值进行触发设定。

4 示教所有机器人的位置。基本上与不是多视图时的做法相同。
5 按下 F5［结束］，触发结束。

便签
PKCSGETQUE 中的工件获取位置是第 1 个特征和第 2 个特征的中点。另一方面，开始追赶工件是在机器人的示教位置超过上方位置之后。

9.8.5 生产状况

多视图传感器的状态如下所示。

视觉程序

显示传感器设定中指定的视觉程序的名称。

动作时间 1

显示执行相机视图 1 之后、相机视图 1 的拍摄结束之前的处理时间的最大、最小、平均值。

动作时间 2

显示执行相机视图 2 之后、结果被投入队列之前的处理时间的最大、最小、平均。

触发间隔 1

显示执行相机视图 1 之后、执行相机视图 2 之前的间隔时间的最大、最小、平均值。触发间隔 1 比动作时间 1 短时，处理就会不及时。

触发间隔 2

显示执行相机视图 2 之后、执行相机视图 1 之前的间隔的最大、最小、平均值。该时间比动作时间 2 短时，处理就会不及时。

便签

DI 输入后，在传送带只移动（触发偏移 2-触发偏移 1）的距离之间，下一个 DI 输入被忽略。这是为了防止只有相机视图 1 被连续执行。此时，重叠次数不被更新。

9.9 3D 视觉传感器

在本节中对使用 3D 视觉传感器时的 *i*RPickTool 的安装方法进行说明。

9.9.1 概要

使用 3D 视觉传感器可以取出上侧重叠的工件或倾斜的工件。此外，对于普通相机图像中难以检出的工件，可以使用 3D 视觉传感器获取的距离图像进行二维视觉追踪。

9.9.2 制约事项

- 在使用“3DV Single-View Visual Tracking”的传送带上，不能使用托盘功能。
- 在使用“3DV Single-View Visual Tracking”的传送带上，不能使用间隙检查功能。
- 在使用“3DV Single-View Visual Tracking”的传送带上，不能被选择为预整列功能的输入传送带。

9.9.3 设置安装位置

进行安装位置设置。

1 准备与 3D 传感器视野相符的点阵板。
2 在相机正下方安装点阵板。
为防止偏差，用胶带等固定。
3 点阵板的方向可以任意方向。例如，拾取追踪时，应使传送带的传送方向与点阵板的 X 方向保持同一方向。
4 在机器人控制装置的示教器上按下［MENU］键。
显示菜单。
5 从菜单选择［设置］→［坐标系］。

6 按下 F3［坐标］。
7 从菜单选择［工具］。
显示工具坐标系一览画面。

8 按下 F5［切换］。
9 输入在碰触销中设定的坐标系编号。
本步骤以使用工具 9 时的情况为例进行说明。

10 在机器人控制装置的示教器上按下［MENU］键。
显示菜单。
11 从菜单选择［*i*RVision］→［示教和试验］。

显示［*i*RVision 示教和试验］画面。

12 按下 F2［新建］。

13 选择传感器［3DV 传感器］，按下 F4［下一步＞］。

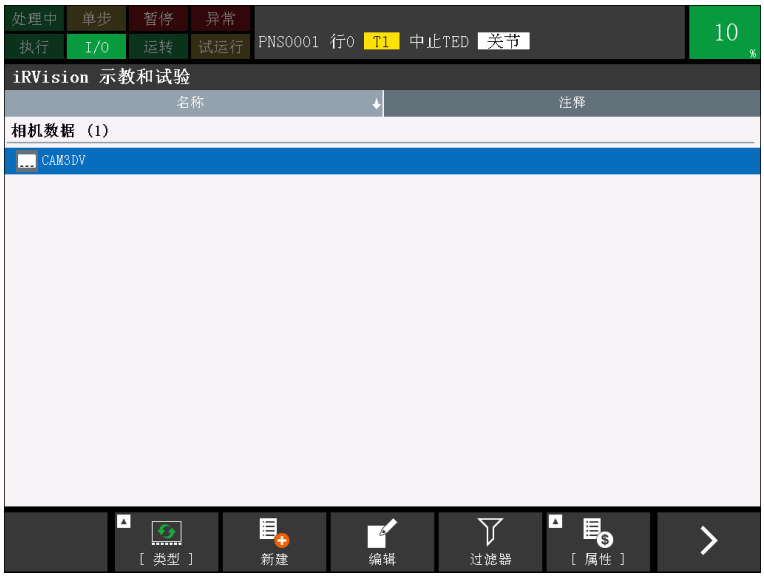
显示以下画面。

14 在［名称］中输入 3DV 传感器的名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例 ：CAM3DV

15 按下 F4［确定］。

在列表中显示创建的相机数据。

16 选择创建的相机数据，按下 F3［编辑］。

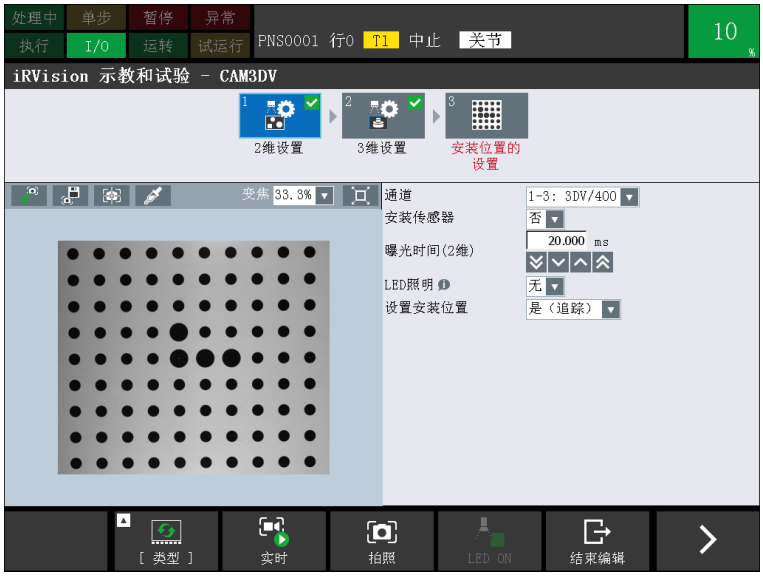


显示以下画面。

17 在导航区域选择了［2 维设置］的状态下，从菜单中选择［通道］。

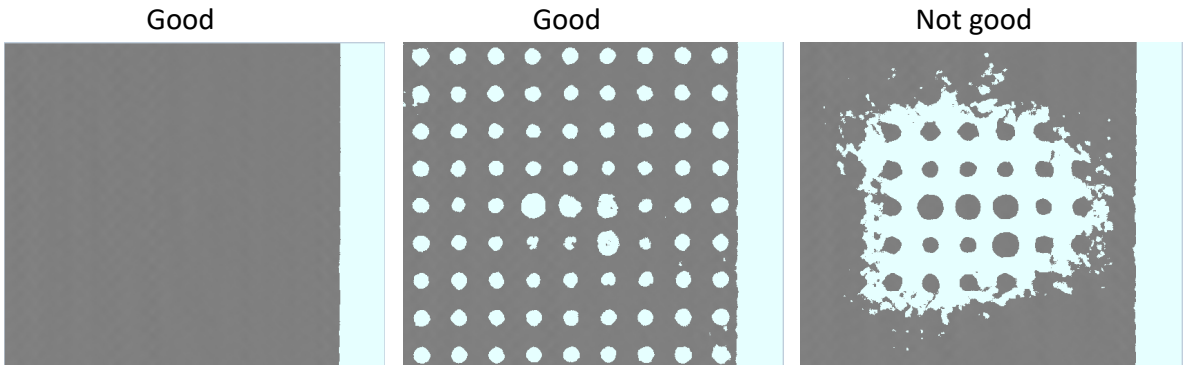
18 输入［曝光时间（2D）］。

19 在［设置安装位置］中，从菜单中选择［是（追踪）］。



20 在导航区域选择［3 维设置］。

21 为了获得点阵板表面的距离信息，请调整曝光时间、投影机光量、除噪强度。格子点上的距离不能测量也没有关系。请进行调整，以便获得格子点周围的距离信息。

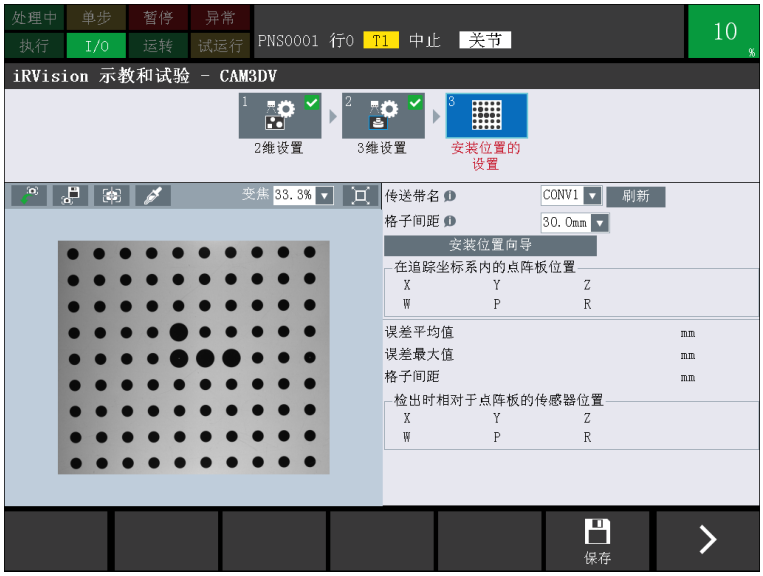


22 在导航区域选择［设置安装位置］。

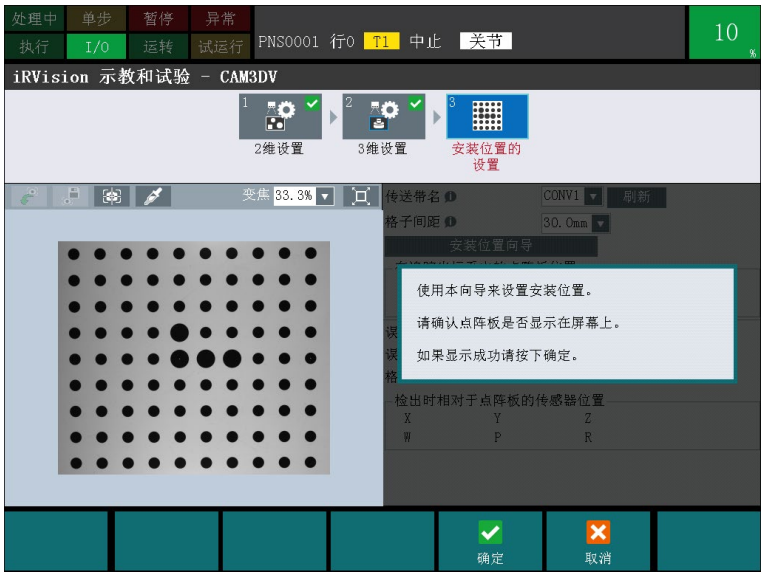
23 选择［传送带名］。

24 输入点阵的［格子间距］。

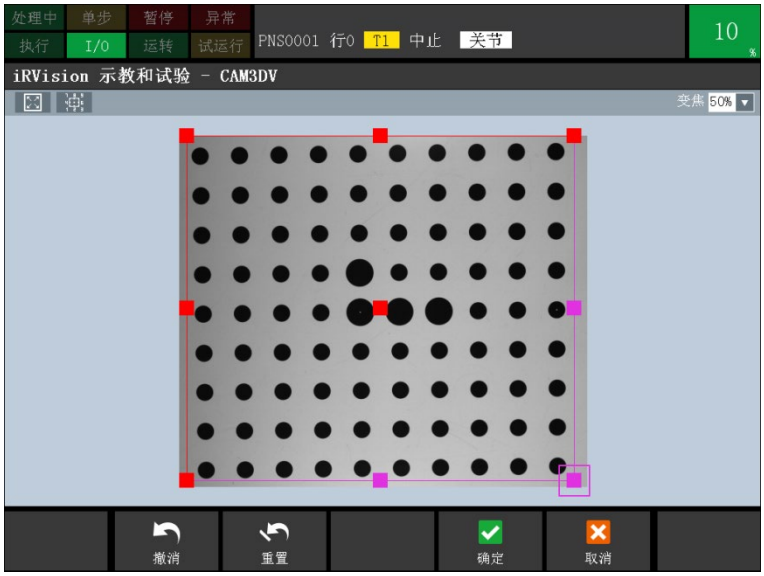
25 按下［＞（下一页）］，再按下 F5［保存］ 。



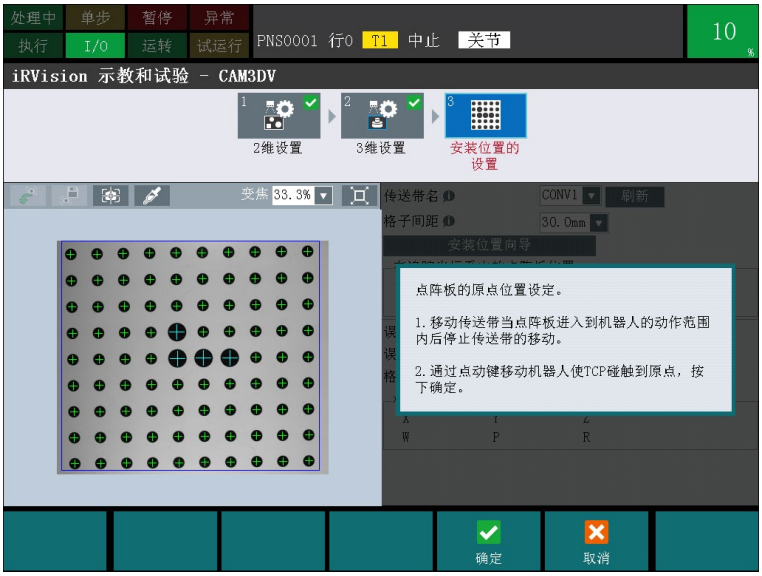
26 按下［安装位置向导］。
27 确认在画面上已显示点阵板，按下 F4［确定］。



28 用紫红色的长方形窗口圈出画面显示的点阵。
29 按下 F4［确定］。



30 运行传送带，当点阵板在机器人到达动作范围内中央附近后关停。



31 触碰点阵板的原点。

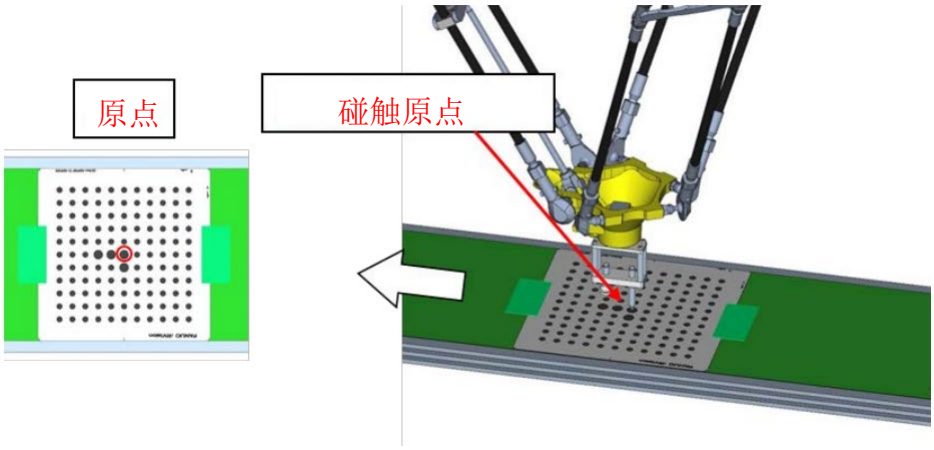


图 9.9.3(a) 触碰点阵板的原点

32 按下 F4［确定］。

33 在传送带停止的状态下触碰点阵板的 X 轴方向。

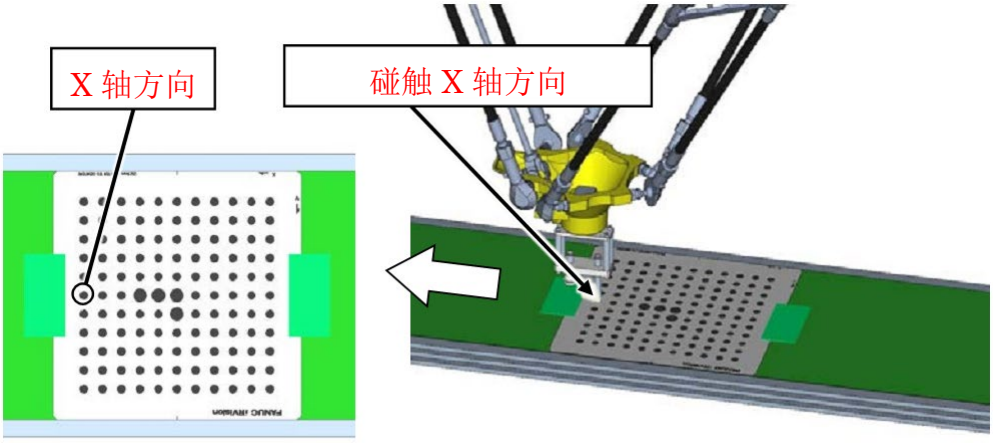


图 9.9.3(b) 触碰点阵板的 Y 轴方向

34 按下 F4［确定］。
35 在传送带停止的状态下触碰点阵板的 Y 轴方向。

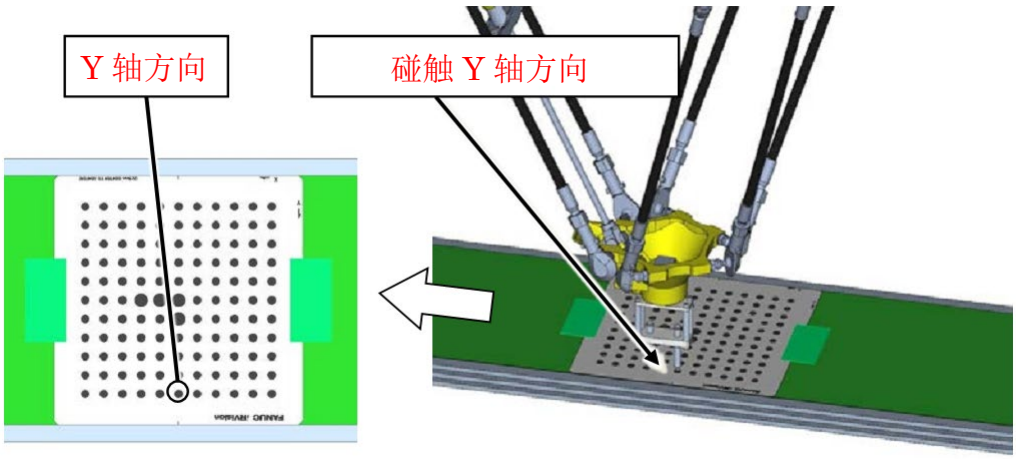
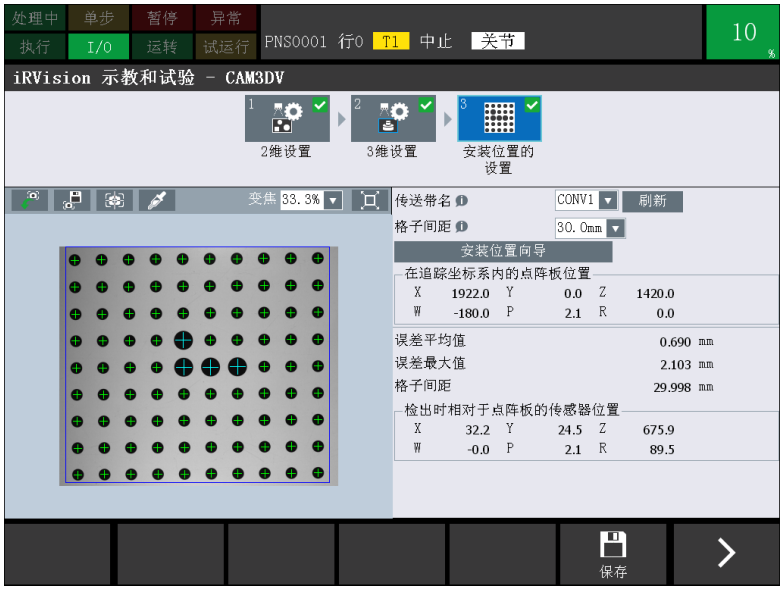


图 9.9.3(c) 触碰点阵板的 Y 轴方向

36 按下 F4［确定］。
计算安装位置。

37 按下 F4［确定］。
38 按下［＞（下一页）］，再按下 F5［保存］ 。

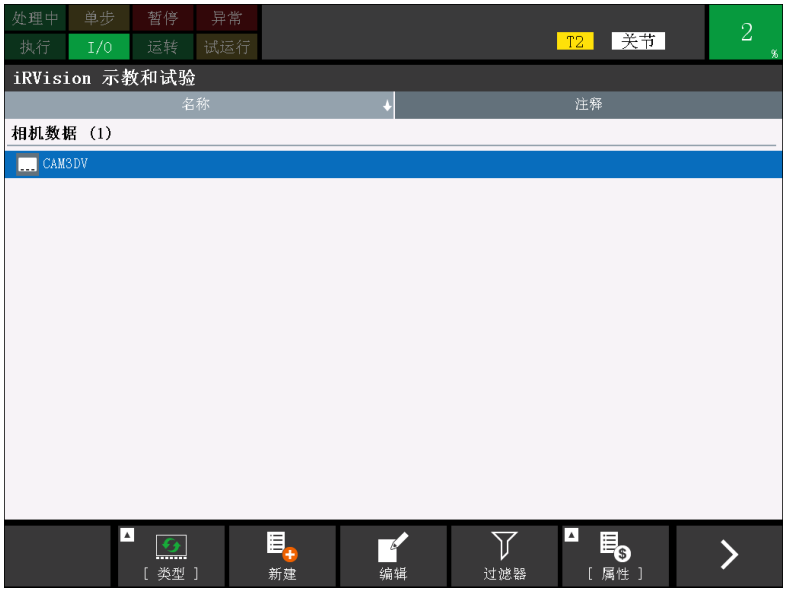


39 再次按下［>（下一页）］，按下 F5［结束编辑］。

9.9.4 视觉程序（三维）

此处针对为了取出上侧重叠的工件或倾斜的工件，对三维取出工件时的视觉程序设定进行说明。使用 3D 视觉传感器获取的距离图像进行二维视觉追踪时，请参照下一项的“视觉程序（二维）”。

1 按下 F2［新建］。



显示［创建新的视觉数据］的画面。

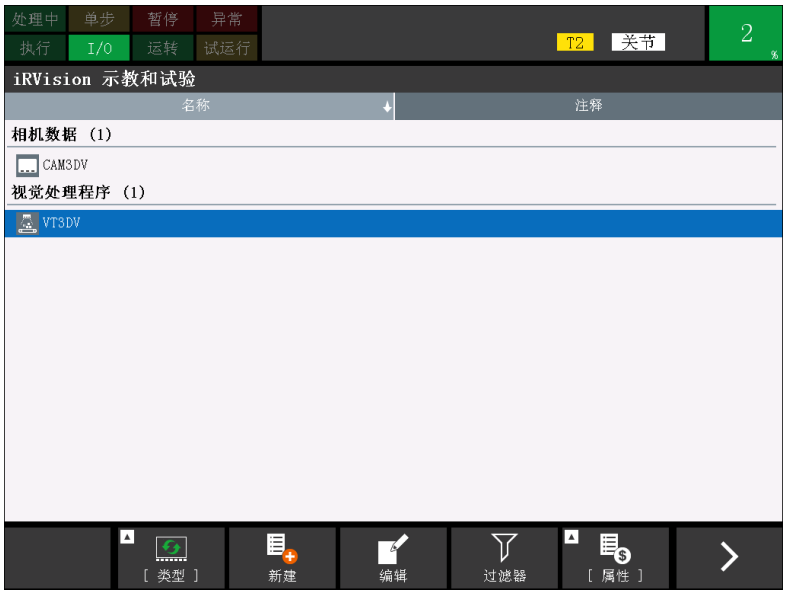
2 选择［3DV Single-View Visual Tracking］，按下 F4［下一个>］。

显示以下画面。

3 在［名称］中输入视觉程序名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例 ：VT3DV

4 按下 F4［确定］。

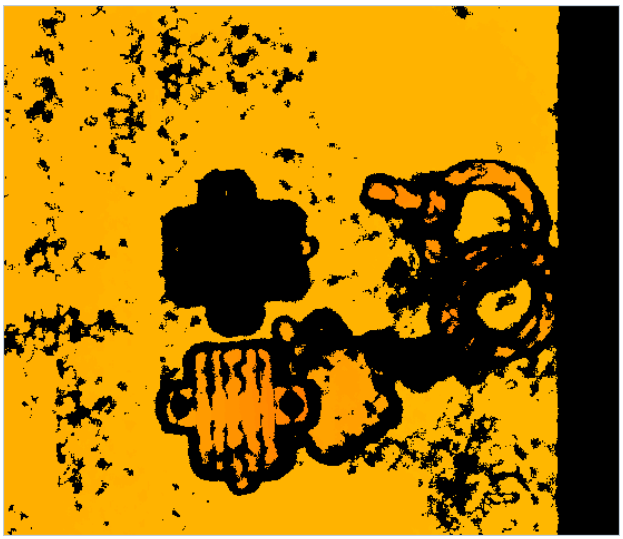
5　选择创建的视觉程序，按下 F3［编辑］。



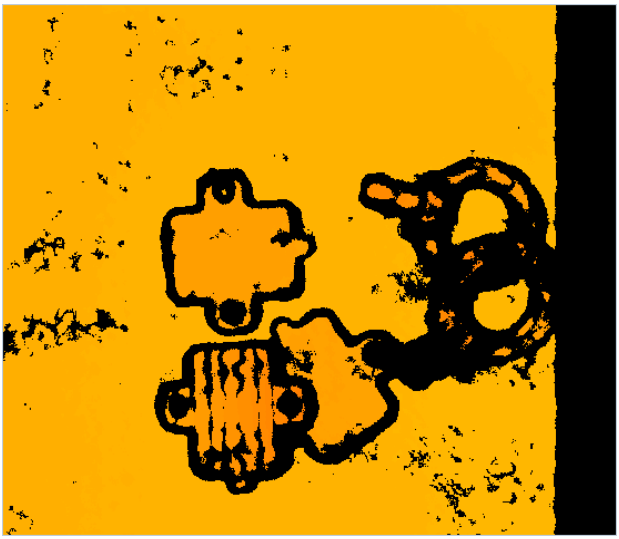
6　从菜单选择［3DV 传感器］。

7 按下树状图标签，选择［Snap Tool 1］。

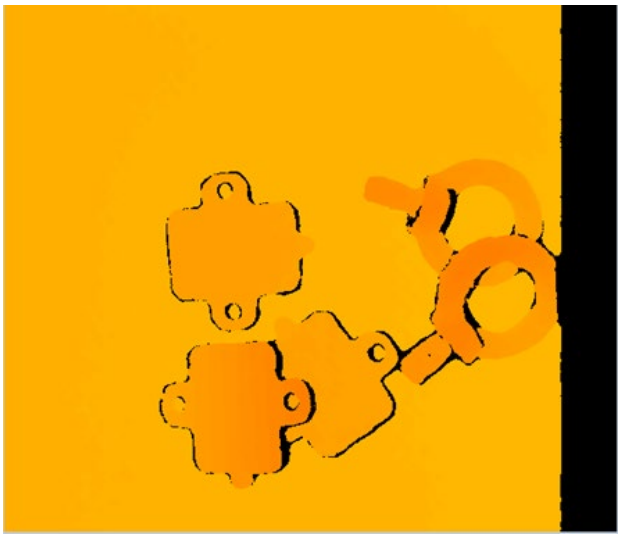
8 并非整体距离图像，而是想要拍摄部分距离图像时，设定［拍照范围］。
9 拍摄时可以缩小图像降低分辨率。通过［图像缩小］的下拉框选择缩小程度。
10 设定［距离范围］。变换成灰色镜头倍率图像的距离范围。指定的距离范围内的三维点被变换成灰色镜头倍率图像，输出为二维的命令工具。
11 按照以下步骤调整［曝光时间］、［投影机光量］、［除噪强度］。
12 将［彩色模式］设定为彩色。不能测量距离的像素显示为黑色。
13 将［投影机光量］设定为 16，将［除噪强度］设定为 10。这是用于正确调整［曝光时间］的临时设定。有时会临时增加黑色像素。



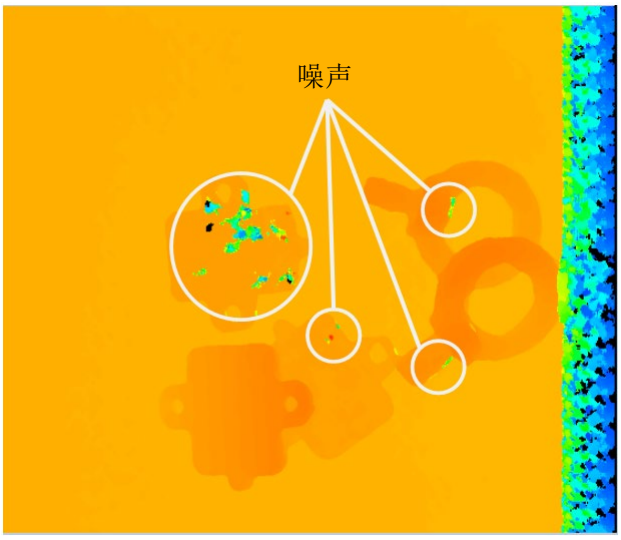
14 调整［曝光时间］，尽量减少黑色像素。
15 调整［投影机光量］，尽量减少黑色像素。即使更改投影机光量，图像也不会有大的变化时，请将光量设定为较大的数字。光量较多的抗扰乱光能力强。



16 在不增加噪声的范围内，下调［噪声除去强度］，以尽量减少黑色像素。



除噪强度设定较小时，能够拍摄到噪声较多的距离图像。下图为发生噪声时的示例。



17 按下树状图标签，选择［3DV Single-View Visual Tracking］。

18 输入［Z 取得范围］。输入追踪坐标系中的 Z 范围。

注释
追踪坐标系的 Z 轴朝下时，在［Z 取得范围］中输入“检出坐标系”中的 Z 范围。此处“检出坐标系”是使追踪坐标系绕 X 轴周围旋转 180 度后 Z 轴朝上的坐标系。
具体情况是，从设定安装位置时的追踪坐标系看的点阵板的位置朝下时，追踪坐标系的 Z 轴朝下。例如，顺时针旋转的圆弧传送带时，追踪坐标系的 Z 轴朝下。

各种颜色表示以下含义。
蓝～红： 在 Z 取得范围内，用于检出的三维点。
黑： 通过 3D 视觉传感器中的拍摄不能测量的三维点。对象工件显示为黑色时，请从 Depth Snap Tool 的设定中进行修正。
白： 在比 Z 取得范围的上限高的位置存在的三维点。不用于检出。
灰： 在比 Z 取得范围的下限低的位置存在的三维点。不用于检出。

请调整后使传送带面发蓝，工件最上面变为红色。

19 按下树状图标签，选择［GPM Locator Tool 1］。

20 放置工件，反映到测量区域。
21 按下 F3［拍照］，拍摄图像。
22 按下［模型示教］。
23 用紫红色的长方形窗口圈出工件。
24 按下 F4［确定］。
25 按下［遮蔽］中的［编辑］。
26 选择画面上方的工具，将不要匹配的部分涂成红色。
27 按下 F4［确定］。
28 根据需要设定搜索范围。
关于详细设定请参阅“*i*RVision 操作说明书 （参考篇）”中的“命令工具”章“图案匹配工具（GPM Locator Tool）”节。

29 按下树状图标签，选择［3D Plane Measurement Tool 1］。

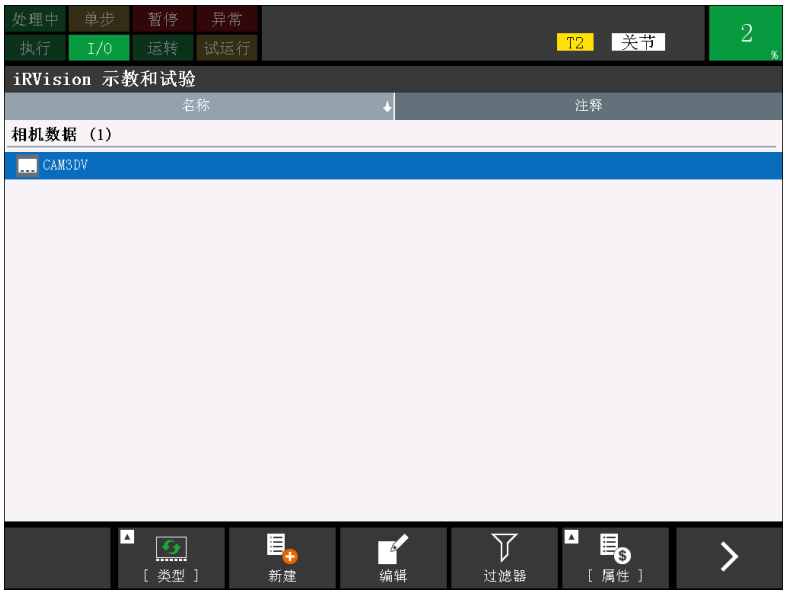
30 放置工件，反映到测量区域。
31 按下［范围示教］。
32 用紫红色的长方形窗口圈出测量平面的区域。
33 按下 F4［确定］。
34 输入［2D-3D 间隔］。
［2D-3D 间隔］是测量的平面与二维测量面的高低差。二维测量面比测量平面离相机近时，变为正数。
35 按下［遮蔽］中的［编辑］。
36 测量区域内存在高低差平面等，存在想要从测量区域排除的区域时，点击［遮蔽］的［编辑］按钮，进行遮蔽示教。即使进行遮蔽示教的情况下，如果不再勾选［使用］，遮蔽将被忽略。

37 按下［＞（下一页）］，再按下 F5［保存］ 。
38 再次按下［＞（下一页）］，按下 F5［结束编辑］。

9.9.5 视觉程序（二维）

此处对使用 3D 视觉传感器获取的距离图像进行二维视觉追踪的情况进行说明。

1 按下 F2［新建］。



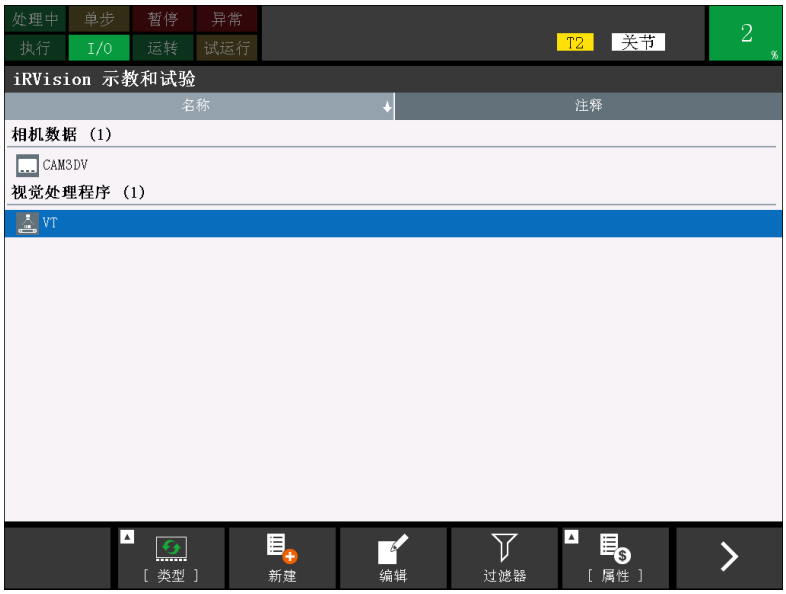
显示［创建新的视觉数据］的画面。

2 选择［2-D Single-View Visual Tracking］，按下 F4［下一个>］。

显示以下画面。

3 在［名称］中输入视觉程序名称。
以半角字母数字和半角片假名输入。不能使用下划线以外的记号和空格。开头的文字应是半角英文字母。
例 ：VT
4 按下 F4［确定］。

5 选择创建的视觉程序，按下 F3［编辑］。



6 从菜单选择［相机］。

7 在命令工具的新建画面选择［Depth Snap Tool］，新建距离图像拍照工具

8 按下树状图标签选择［Snap Tool 1］，按下 按钮进行删除。

9 按下树状图标签选择［GPM Locator Tool 1］，在［输入图像］中选择［Depth Snap Tool 1］。

10 与前款“视觉程序（三维）”相同，设定［Depth Snap Tool 1］、［GPM Locator Tool 1］。

11 按下［＞（下一页）］，再按下 F5［保存］ 。

12 再次按下［＞（下一页）］，按下 F5［结束编辑］。

9.10 间隙检查功能

9.10.1 概要

在视觉追踪中，要取出密集的工件，有时会移动了周围的工件。本功能就是在这种情况下，为工件设定任意范围（间隙），检查要取出工件的间隙中是否存在其他工件，只有在没有时才取出工件。

例如，下图情况下，关于工件 A、E，间隙内不存在其他工件，因此可以取出。关于工件 B、C、D，在间隙内存在其他工件，因此不能取出。但是，工件 E 已被取出时，工件 D 的间隙内没有了其他工件，因此工件 D 可以取出。

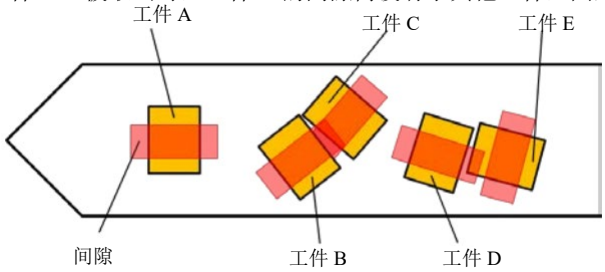


图 9.10.1(a) 间隙检查示例

9.10.2 制约事项

符合以下任一条件时，不能使用间隙检查。

- 对 1 个传送带工作站使用 2 个以上的传感器时。
- 在传送带的设定画面进行以下设定时。
  - 选择［托盘］。
  - 使用［预整列］。
  - 使用［伺服传送带］。
  - 使用［根据 Y 排序］。
- 在传感器的设定画面进行以下设定时。
  - 不是 ROBOGUIDE 时，
    ［触发条件］和［触发］并非［距离］和［通过视觉检出］。
  - 是 ROBOGUIDE 时，
    ［触发条件］和［触发］并非［距离］和［通过视觉检出］或［DI］和［RG 放入工件］。
  - ［3-D Single-View Visual Tracking］的视觉程序被选择。

9.10.3 注意事项

以下情况下有时无法正确进行间隙检查。

- 同时向传送带供给 2 种以上的工件。
  间隙检查仅可对 1 种工件进行设定。因此，如下图所示，同时向传送带供给 2 种工件时，设定了间隙检查的工件 A 的间隙中进入了种类不同的工件 B，但因工件 B 未设定间隙检查，因此不在间隙检查的对象范围内。所以可判断工件 A 能够取出。

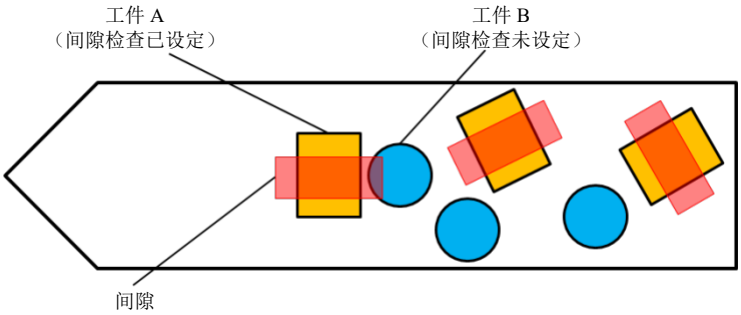


图 9.10.3(a) 2 种以上的工件同时被供给传送带的示例

注释
如果模型和模型原点相同，即使使用多个模型 ID，也可以使用间隙检查。

- 传感器的视野与最上游机器人的追踪范围的上方位置很近。
  下图情况下，工件 B 进入工件 A 的间隙中，但工件 B 未检出，因此不在间隙检查的对象范围内。所以判断工件 A 能够取出。

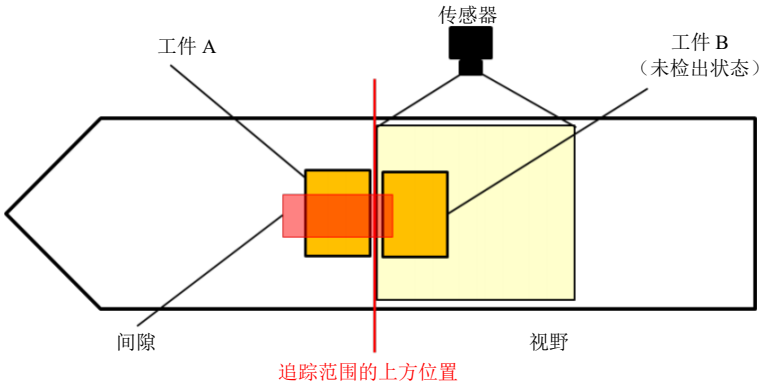


图 9.10.3 (b) 传感器视野和最上游机器人的追踪范围的上方位置较近的示例

- 上游机器人的发送位置（发信线）与下游机器人的追踪范围的上方位置很近。
  下图情况下，工件 B 进入工件 A 的间隙中，但工件 B 未超过机器人 1 的发送位置（发信线），工件 B 的数据未发送给机器人 2，因此，工件 B 不在间隙检查的对象范围内。所以判断工件 A 能够取出。

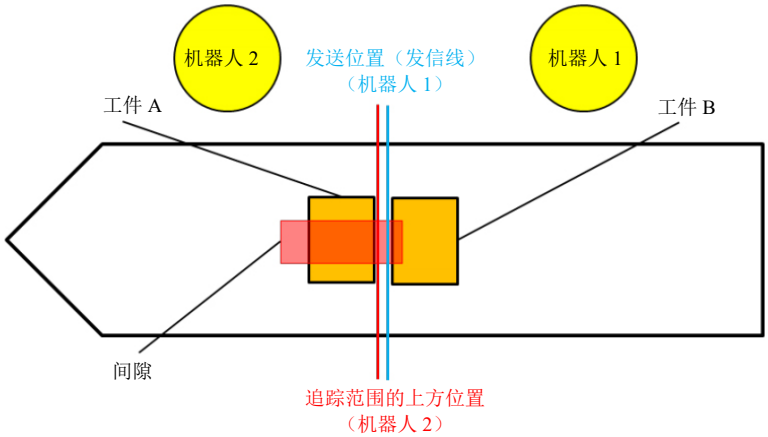


图 9.10.3 (c) 上游机器人的发送位置与下游机器人的追踪范围的上方位置较近的示例

9.10.4 设定

使用间隙检查功能时，首先完成“5 设定前的其他准备工作”。

9.10.4.1 工作单元、机器人设定

没有间隙检查情况下的特记事项。请参照“6.1 工作单元设定”、“6.2 机器人设定”。

9.10.4.2 托盘设定

没有间隙检查情况下的特记事项。
使用了间隙检查的传送带无法使用托盘，未使用间隙检查的传送带使用托盘时，或使用固定工作站时，请参照“6.3 托盘设定”。

9.10.4.3 传送带设定

在要使用间隙检查的传送带的设定画面中，勾选［间隙检查］中的［已使用］。
没有其他的特记事项。请参照“6.4 传送带设定”。

9.10.4.4 传感器设定

在母传送带上使用了间隙检查时，在传感器设定画面显示间隙检查的设定项目。

［工件形状］

选择设定了间隙检查的工件的形状。可选择的工件形状为［长方形］和［圆（无相位）］。根据工件形状，切换至如下设置画面。

［工件形状］是［长方形］的情况

显示以下画面。

［间隙位置］

结束“9.10.4.5 间隙向导”后用绿色表示［向导成功］。

［工件参数］

［X 方向长度］中设定工件的 X 方向长度。［Y 方向长度］中设定工件的 Y 方向长度。在“9.10.4.5 间隙向导”中进行设定。

［间隙参数］

［X 方向长度］中设定工件间隙的 X 方向长度。［Y 方向长度］中设定工件间隙的 Y 方向长度。在“9.10.4.5 间隙向导”中进行设定。

例如，［工件参数］的［X 方向长度］为 30mm，［Y 方向长度］为 50mm、［间隙参数］的［X 方向长度］为 15mm，［Y 方向长度］为-10mm 时，如下图所示。

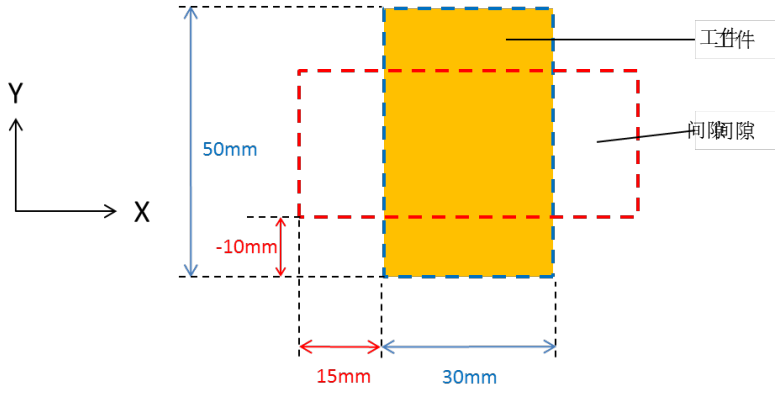


图 9.10.4.4 (a) 参数的设定示例(［工件形状］为［长方形］时)

［间隙设定］

在功能键的 F4 上。开始“9.10.4.5 间隙向导”。

> **⚠注意**
> **“9.10.4.5 间隙向导”在安装有碰触销的状态下进行。请在安装有机械手的状态下进行的基准位置设定前进行。**

［工件形状］是［圆（无相位）］的情况

显示以下画面。

［工件参数］

在［直径］中设定圆的直径。

［间隙参数］

在［长度］处设定间隙长度。

例如，［工件参数］为 30mm、［间隙参数］为 10mm 时，如下图所示。

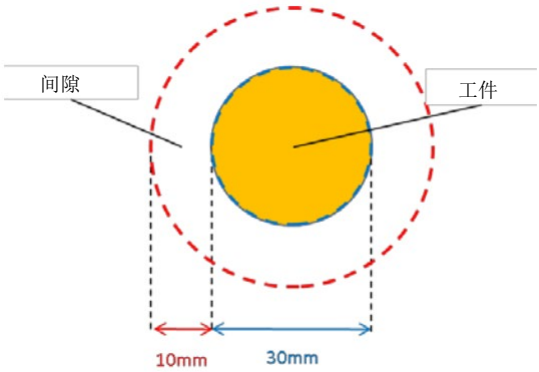
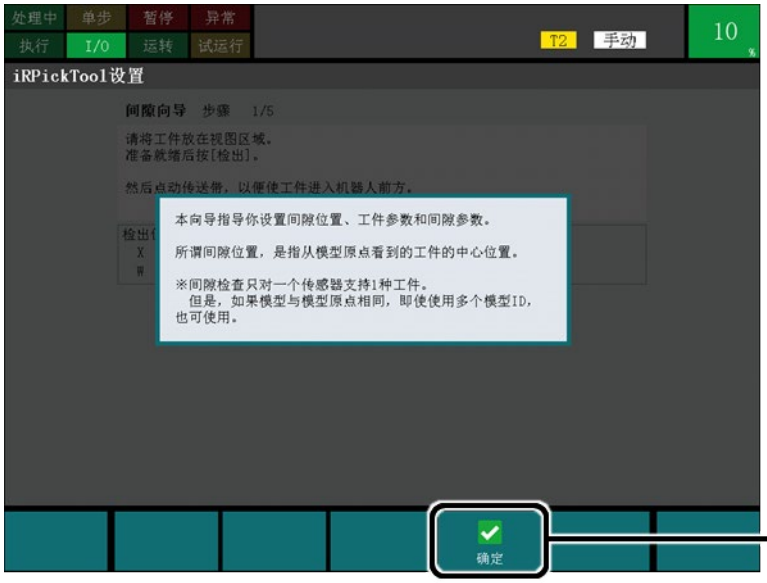


图 9.10.4.4(b) 参数的设定示例(［工件形状］为［圆形(无相位)］时)

9.10.4.5 间隙向导

仅在“9.10.4.4 传感器设定”的［工件形状］中选择了［长方形］时进行该设定。

1 使传送带停止。
2 按下 F4［间隙设定］。
则显示以下画面。



3 确认显示的消息的内容，按下 F4［OK］。
则显示以下画面。

4 工件盘放置于相机视野内，按下 F4［检出］。
如果工件盘的检出成功，显示检出位置，F5［下一步］有效。

5 按下 F5［下一步］。

则显示以下画面。

6 在工件来到机器人前面之前运行传送带。
7 点动运行机器人，碰触工件的原点。

注释

工件的碰触在工件的原点、X 方向点、Y 方向点 3 点进行。
碰触的点根据追踪坐标系的 Z 轴的方向不同而不同，如下所示。

- **追踪坐标系的+Z 轴相对于传送带面朝上时，**
  变成使 Y 方向点相对于 X 方向点逆时针旋转 90° 的方向。

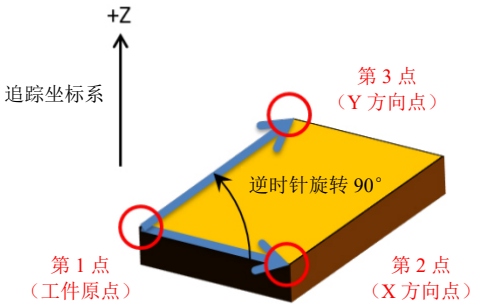


图 9.10.4.5 (a) 追踪坐标系的+Z 轴朝上时

- **追踪坐标系的+Z 轴相对于传送带面朝下时，**
  变成使 Y 方向点相对于 X 方向点顺时针旋转 90° 的方向。

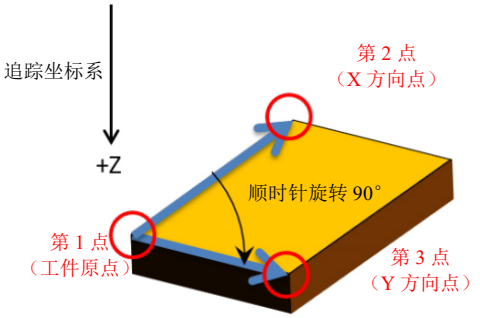


图 9.10.4.5 (b) 追踪坐标系的+Z 轴朝下时

8 按下 F4［记录］。
显示碰触时的编码器值、碰触位置，F5［下一步］有效。
9 按下 F5［下一步］。

则显示以下画面。

10 点动运行机器人，碰触工件的 X 方向点。
11 按下 F4［记录］。
显示碰触位置，F5［下一步］有效。
12 按下 F5［下一步］。

则显示以下画面。

13 点动运行机器人，碰触工件的 Y 方向点。

14 按下 F4［记录］。
显示碰触位置，F5［下一步］有效。

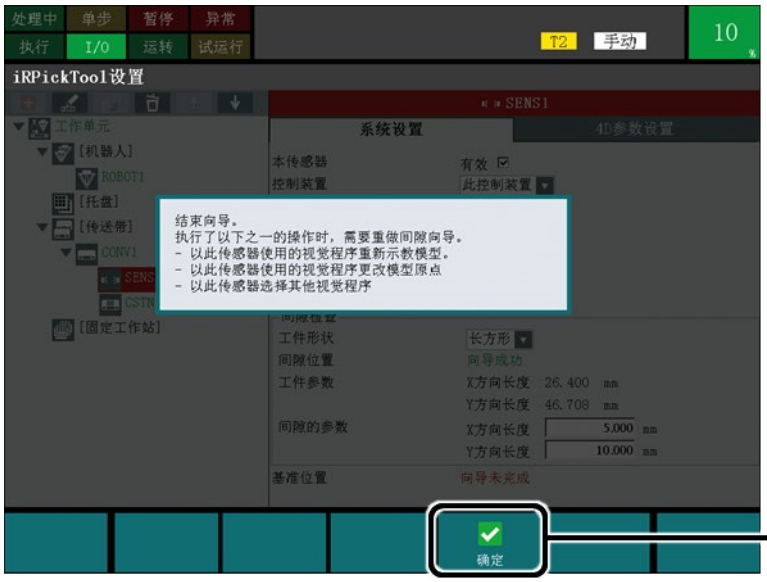
15 按下 F5［下一步］。

则显示以下画面。

16 分别输入［间隙参数］的［X 方向长度］和［Y 方向长度］。
F5［结束］变成有效。

17 按下 F5［结束］，

则显示以下画面。



18 确认显示的消息的内容，按下 F4［OK］。
返回原来的设置画面。
在［间隙位置］的旁边用绿色表示［向导成功］，在［工件参数］和［间隙参数］的［X 方向长度］、［Y 方向长度］中存在分别通过向导设定的值。

> **⚠注意**
> 在设定了间隙检查的传感器中使用的视觉程序中，进行了以下任意一项时，需要修改间隙向导。
> - 对模型进行重新示教
> - 更改模型原点

9.10.4.6 传送带工作站设定

线传送带的情况

显示［发送位置］，如下图所示。
发送位置是指通过此传送带工作站将工件的数据发送到下游工作站的位置，表示为追踪坐标系上的位置。
如果“9.10.4.4 传感器设定”中的间隙检查的设定、追踪范围（上游、下游）的设定、跳过距离的设定已结束，设定了间隙检查的传感器对应的所有传送带工作站的［发送位置］上自动设定值。
没有其他的特记事项。请参照“6.6 传送带工作站设定”。

圆弧传送带的情况

显示［发信线］，如下图所示。

发信线是指通过该传送带工作站将工件的数据发送到下游工作站的位置，表示为跳过距离的偏移。

发信线由用户输入。

没有其他的特记事项。请参照“6.6 传送带工作站设定”。

> **⚠注意**
> **根据工件从圆弧传送带的内侧传送还是从外侧传送，发信线的位置会发生变化。**
> **例如，工件从传送带的中心附近传送时，需要对发信线设定较大的值。**

9.10.4.7 固定工作站设定

没有间隙检查情况下的特记事项。请参照“6.7 固定工作站设定”。

10 常见问题处理方法

本章介绍 *i*RPickTool 追踪系统中，较容易碰到的问题及其对策。

10.1 机器人完全不去拾取工件

原因 1： 传送带工作站为无效。
对策 1： 在传送带工作站设定中，如果跳过此传送带工作站被设定为有效，工件的信息将完全不分配至该传送带工作站的机器人。请使跳过此传送带工作站无效。

原因 2： 负载平衡设定不合适。
对策 2： 确认该机器人的拾取比例是否为 0。如果放置方的负载平衡不正确，则拾取方也无法正确动作。请确认拾取方和放置方都正确设定负载平衡。

原因 3： 模型 ID 错误。
对策 3： 确认视觉程序检出的工件的模型 ID，和通过传送带的设定的负载平衡指定的模型 ID 是否相同。确认在托盘的情况下，单元的模型 ID 和负载平衡的模型 ID 是否相同。此外，确认在通过 PKCSGETQ、PKFSGETQ 或 PKFSGETBUF 的调出指定模型 ID 时，所指定的模型 ID，和视觉程序或单元中设定的模型 ID 是否相同。

原因 4： 视觉程序中的未检出。
对策 4： 通过 *i*RVision 的执行时监视确认是否正确检出工件。

原因 5： 尚未执行传感器任务。
对策 5： 确认是否忘记执行 KAREL 程序 PKWCSTART.PC（或者 PKSNSTART.PC）。可在生产状况画面中观察传感器和传送带工作站的状态来进行确认。详细信息请参照“9.2 确认生产状况”。

10.2 工件的跳过频发

原因 1： 机器人的处理过慢。
对策 1： 停电处理有效时，请使其无效。

> ⚠注意
> 使用 *i*RPickTool 的追踪，不对应停电处理。

使自动备份无效。在文件画面通过 F1［类型］→［自动备份］将自动备份变更为无效。

原因 2： 机器人的动作过慢。
对策 2： 如果机器人的动作慢，则无法拾取传送来的所有工件。调整动作程序的速度。如果对动作程序的动作速度指定 MAX_SPEED（最快速度），则可以自动指定该机器人中可设定的最快速度。此外，确认是否选择了适当的倍率。

原因 3： 视觉程序中的未检出。
对策 3： 如果视觉程序的参数不适当，可能无法检出工件。调整图形匹配工具的参数，使其能进行正确检出。

原因 4： 视觉的检出过慢。
对策 4： 如果视觉的处理赶不上传送带的速度，则无法在要求的时间运行下一个视觉程序，导致工件在未被检出的状态下通过视野。确认在 *i*RPickTool 的生产状况画面的传感器菜单中，视觉的执行是否赶得及（超限次数不是 1 以上），如果未能赶得及，则采取放慢传送带、放大相机视野等措施。此外，在 1 台机器人控制装置上，运行使用多个视觉的传感器时，使传感器分散于多个机器人控制装置。

原因 5： 视觉的检出时间变长。
对策 5： 由于视觉的检出处理的优先度比 TP 程序低，如果 TP 程序的运行繁忙，则视觉的检出处理会被推迟，结果导致检出时间延长的现象。特别是在传送带上没有可拾取的工件的情况，当发生反复调出包括至待机位置 0 距离动作的程序的环路时，TP 程序的运行变得繁忙。这种情况，需要使 TP 程序的运行暂停，进行视觉的检出处理。具体如“6.9.2.2 进行拾取动作的程序”中所述，对 PKCSGETQUE 的第 3 参数指定正值。如“6.8.5.3 PKCSGETQUE.PC”中所述，对 PKCSGETQUE 的第 3 参数指定正值时的动作如下。

- 如果有可取得的工件，则不待机。
- 如果没有可取得的工件，则只待机指定的时间。但是，在经过指定时间之前，有可取得的工件进入传送带工作站时，则取得该工件的信息，立即结束待机。

插入了普通的待机命令时，即使在待机时有可取得的工件进入传送带工作站，也必须待机指定的时间。根据情况，以 PKCSGETQ 的第 3 参数指定待机时间，使机器人待机必要的时间。通常以 100 左右为宜。
如果对 PKCSGETQ 的第 3 参数指定 0，在没有可取得的工件的情况下执行多次调出 PKCSGETQ 的环路（“6.9.2.2 进行拾取动作的程序”的第 5 行至第 10 行），其结果，至视觉检出结束的时间可能变长。

原因 6： 放弃线的设定值不合适。
对策 6： 如果放弃线设定于过于靠上游侧，则本来可以拾取的工件也被判断为无法拾取而跳过。实际边使系统镓动，边适当调整放弃线的位置。

原因 7： 追踪范围的设定不合适。
对策 7： 在传送带工作站设定中，对［上游边界］、［下游边界］进行合适的设定。

原因 8： 模型 ID 错误。
对策 8： 确认传送带的设定的负载平衡盒的［指定需处理的模型 ID］与通过视觉程序指定的模型 ID 是否一致。如果不一致，则发生仅该种类的工件不会被任何机器人拾取。

10.3 2 次去取相同的工件

原因 1： 未正确进行相机的校准。
对策 1： 确认进行校准时，碰触所使用的 TCP 是否正确。确认校准数据的格子间距、焦距等是否正确。

原因 2： ［同一物检查的阈值］不合适。
对策 2： 确认［同一物检查的阈值］是否太小。

原因 3： 工件的误检出。
对策 3： 确认是否错误找到了附近的相似工件。

10.4 如果工件旋转，则补偿偏移

原因： 未正确设定追踪坐标系。
对策： 请参照“6.12.2 追踪坐标系的设定误差的调整”。

10.5 传送带一端的工件补偿偏移

原因 1： 相机的校准中使用的点阵太小，可能未正确求得镜头的失真。
对策 1： 在点阵只显示到相机视野的一部分的状态下，进行相机校准时点阵未拍到的位置检出的工件位置计算不正确。请准备大的点阵，使点阵显示在画面整体，再次进行校准。

原因 2： 此外，如果视觉程序中指定的［检出面 Z 高度］不正确，也可能发生该现象。
对策 2： 请在［检出面 Z 高度］中正确设定追踪坐标系的工件检出部位的高度。

10.6 机器人动作不顺畅

原因： 平均速度（更新次数）值小。
对策： 如果使用的传送带不顺畅动作，则机器人也可能随着传送带的动作，发生动作不顺畅的问题。这种情况，请在示教器的编码器参数设定表画面，使［5.平均（更新次数）］的值变大。该值用于计算编码器平均速度的瞬时速度的累积个数。标准值是 1。起初，请设定 10 左右的值，确认机器人的动作。

10.7 停止/启动传送带不工作

原因： DO 的设定有可能矛盾。
对策： 在相同机器人控制装置上定义的多个传送带工作站上，将相同 DO 编号指定为停止/启动传送带用，并确认 DO 的极性是否反向（也就是说，一方面通过 DO＝ON 使传送带停止，另一方面，通过 DO＝OFF 使传送带停止）设定。

10.8 图像读入的时机延迟

原因： 由于图像等的执行履历保存在存储卡中，因此可能来不及进行视觉的处理。
对策： 请在 *i*RVision 的［系统设定］画面，解除［执行履历有效］的勾选，不保存执行履历。

10.9 超限次数不断增加

原因： 如果在生产状况的传感器画面上，超限次数不断增加，视觉的处理时间比视觉启动的触发间隔长，从而导致来不及处理的状态。即使在该状态下，系统也不会停止。但是，视觉在前一次检出处理结束后，进行下一个处理，该期间，工件可能穿过相机的视野。发生这种情况时，需进行改善。

对策： 每移动一定距离，就进行检出的情况下，打开传感器的设定画面，尽量将［触发距离（mm）］变更为较大值，确认现象是否变化。

每次输入 DI 信号，就进行检出的情况下，尽量调整工件的供给间隔，使 DI 的输入间隔变长。

对于视觉的设定等的变更，也可能使视觉的处理时间变短。可以采取以下对策。

- 如果执行履历为有效，则使其无效。
- 如果显示执行时监视，则在 *i*RVision 的［系统设定］中勾选［执行时监视无效］，在视觉程序的示教画面上使［处理结果显示］为［不显示］，从而不显示图像。
- 调整图形匹配的参数。特别在勾选了［扁平率］的情况下，请将该勾选解除。在工件的检出面无倾斜的视觉追踪中，不使用［扁平率］。将印刷在袋子的工件表面的图形作为模型时，如果图形的样子根据工件不同而不同，则并非通过［扁平率］，而是通过［失真的阈值］进行调整。

此外，有时也可以通过减少视觉以外的处理，使视觉的处理时间变短。例如，使用多任务功能的情况，则不使用。

10.10 生产中无法变更设定

原因： 有可能是在生产中设定不能变更的项目。

对策： *i*RPickTool 中，有关生产中的设定的变更，考虑到安全问题，根据设定项目的性质，分以下的 3 个等级。

1 可变更的对象。例如，负载平衡的设定在生产过程中也能更改。
2 在反映变更之前，确认是否可变更后，可进行变更的对象。例如，追踪范围的边界线位置在扩大区域方向有变化时，在确认是否适用更改后的值之后，进行实际变更。
3 不可变更的对象。例如，传感器的触发在生产过程中不能更改。要想更改时，将显示 IRPK-075“仅程序结束时可进行的操作。”的报警。

其中，有关［3. 不可变更的对象］，是指不能在生产过程中进行变更。停止生产，所有机器人的程序结束后，请重新更改设定。

另外，在工作单元内存在未退出程序的控制装置时，判断为“生产中”。

附录

A 系统变量文件
B 示教用电脑的设定
C HDI 设定
D αA1000S 脉冲编码器
E 量增脉冲编码器
F 另置型接口装置（SDU）
G 旧软件与 *i*RPickTool 的 KAREL 对照表
H 传感器任务的自定义
I 3rd Party 接口加载项

A 系统变量文件

*i*RPickTool 相关的系统变量文件如下所示。

SYSPKCFG.SV
　　*i*RPickTool 的设定数据被保存。

LNTKBCK.SV
　　追踪拾取的设定数据被保存。

B 示教用电脑的设定

*i*RPickTool 中，一般情况下，可通过对应触摸屏的示教器进行所有的设定，也可以连接电脑进行设定。本章中，就有关通过电脑进行设定的情况下的准备工作进行说明。

注释
通过对应触摸屏的示教器进行设定时，不需要进行示教用电脑的设定。

仅在 *i*RPickTool 的设定时，需要示教用电脑，在系统稼动中可以卸下电脑。

注释
示教用电脑与进行机器人间通信的局域网连接。

B.1 电脑与浏览器

动作确认完毕的电脑和浏览器，请参照下表。

OS	**Windows 7 Professional（32bit）** **Windows 7 Professional（64bit）**
浏览器	**Internet Explorer 9（32bit）** **Internet Explorer 10** **Internet Explorer 11**

⚠注意
1 语言仅支持日语版和英语版。
2 请在各版本的 Windows 中安装最新的服务软件组后使用。
3 如果以没有 Administrator 权限的用户登录您使用的电脑，则可能发生电脑和机器人无法正常通信的情况。因此请以有 Administrator 权限的用户重新登录。

B.2 通信电缆

连接机器人控制装置和 *i*RPickTool 的示教用电脑的电缆。请选择满足下表规格的 10BASE-T 或 100BASE-T 电缆。

电缆	**双绞线**
屏蔽	**有**

B.3 连接通信电缆

用以太网电缆连接机器人控制装置和示教用电脑。机器人控制装置侧与主板前面的以太网连接器连接。电脑侧与有 标记的连接器连接。

B.4 在电脑上设定 IP 地址

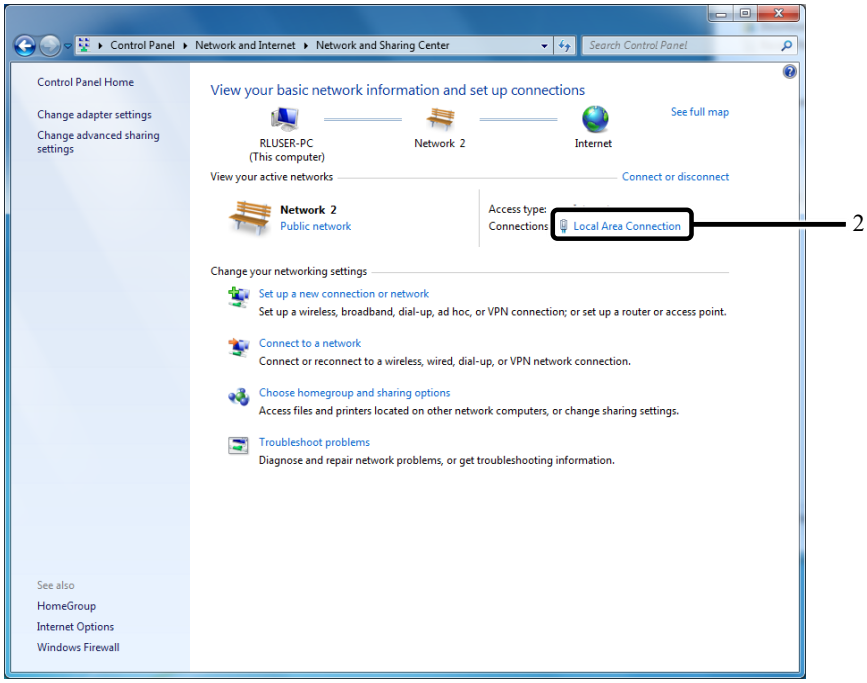
在示教用电脑上设定 IP 地址。3 台机器人控制装置与电脑连接，如果不与其他网络连接，则可以进行例如下面的 IP 地址设定。

机器人控制装置 1	**172.16.0.1**
机器人控制装置 2	**172.16.0.2**
机器人控制装置 3	**172.16.0.3**

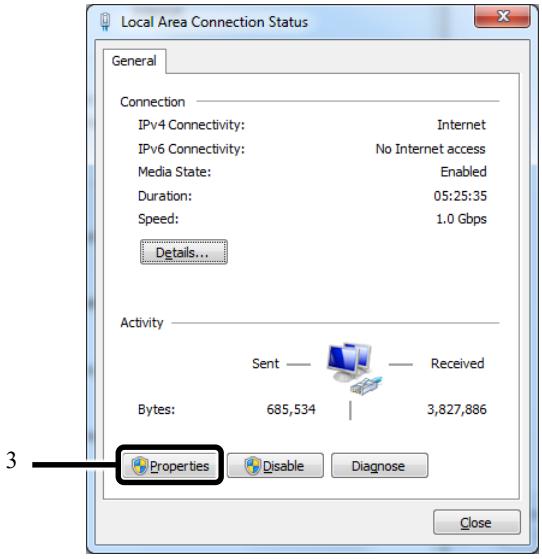
电脑	**172.16.0.4**
网关	**172.16.0.5**
子网掩码	**255.255.0.0**

按以下步骤设定 IP 地址。

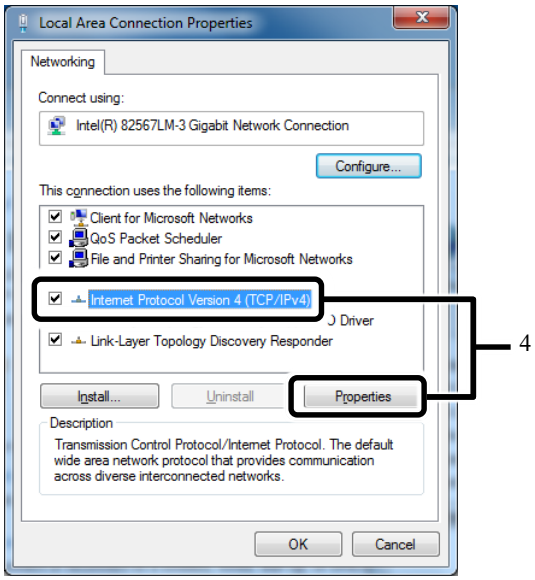
1 从电脑的［Start］（开始）→［Control Panel］（控制面板）打开［Network and Sharing Center］（网络与共享中心）。
2 点击［View your active networks］（查看活动网络）中的［Local Area Connection］（本地连接）。



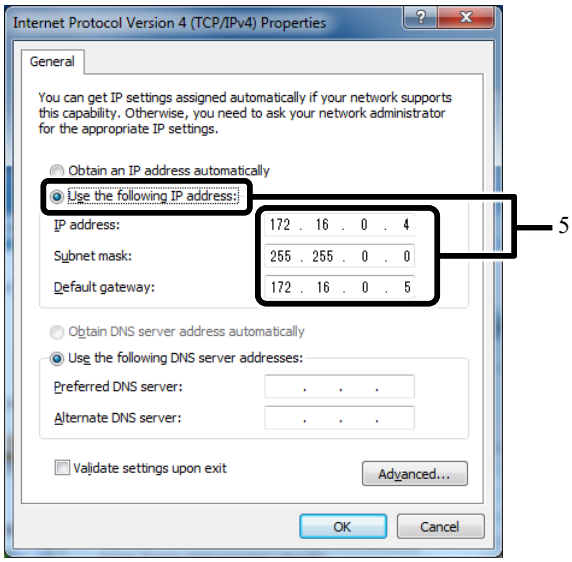
显示［Local Area Connection Status］（本地连接状态）画面。
3 点击［Properties］（属性）按钮。



4 选择［Internet Protocol Version 4（TCP/IPv4）］（Internet 协议版本 4（TCP/IPv4））， 点击［Properties］按钮。



5 勾选［Use the following IP address］（使用下面的 IP 地址），输入［IP address］（IP 地址）、［Subnet mask］（子网掩码）、［Default gateway］（默认网关）。

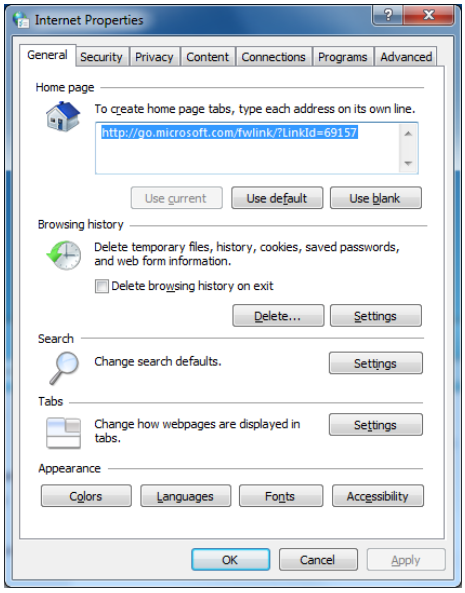


6 点击［OK］（确定）按钮，关闭对话框。

B.5 变更 Internet Explorer 的设定

变更 Internet Explorer 的设定，使与机器人控制装置的通信不被 Internet Explorer 限制。这里以 Windows7 的画面为例进行说明。

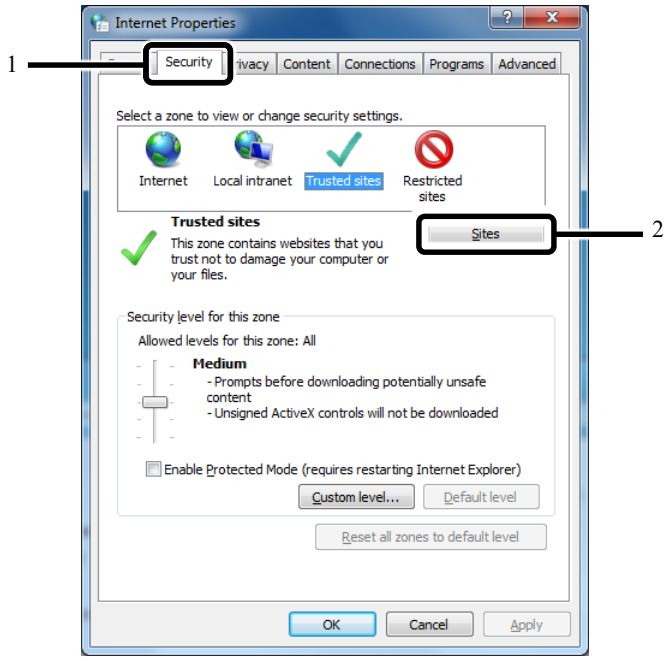
1 从电脑［Start］（开始）→［Control Panel］（控制面板）打开［Internet Options］（互联网选项）。
显示［Internet Properties］（互联网属性）画面。



受信任的站点

将机器人控制装置作为受信任的站点登录。

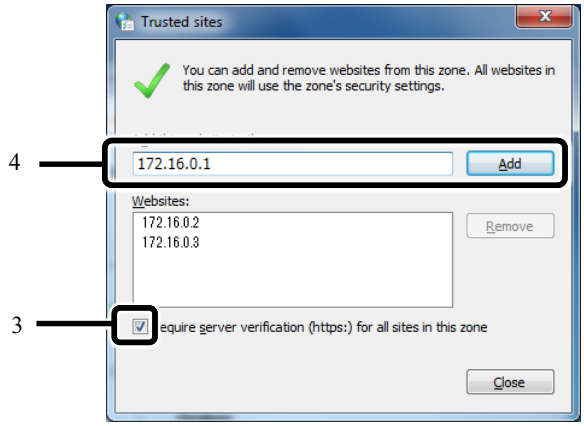
1 选择［Security］（安全）标签。
2 在区域内选择［Trusted sites］（受信任的站点），点击［Sites］（站点）按钮。



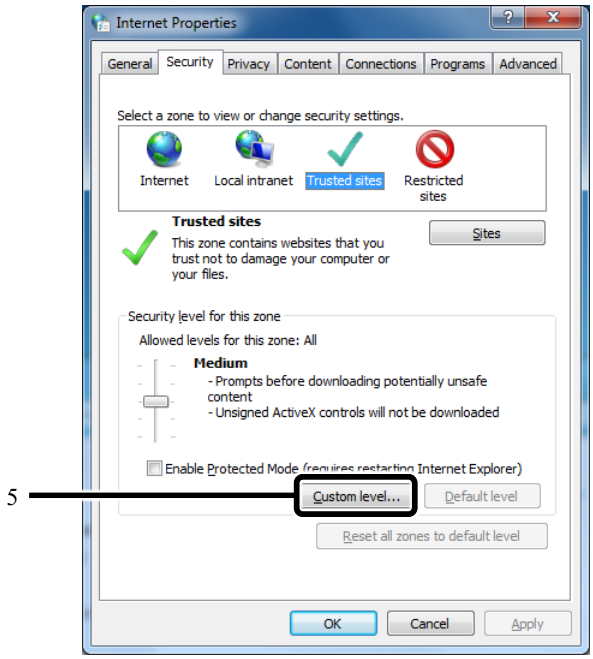
显示［Trusted Sites］（受信任的站点）画面。

3 去掉 Require server verification (https:) for all the sites in this zone］（对该区域中的所有站点要求服务器验证）的勾选。

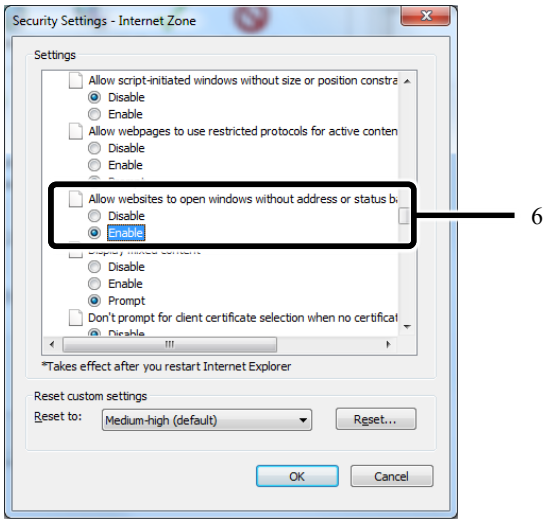
4 在［Add this website to the zone］（将该网站添加到区域）中输入机器人控制装置的 IP 地址，点击［Add］（添加）按钮。



5 返回［Security］（安全）标签的画面，点击［Custom level］（自定义级）按钮。



6 勾选［Miscellaneous］（其他）的［Allow websites to open windows without address or status bars］（允许网站打开没有地址或状态栏的窗口）的［Enable］（启用）。

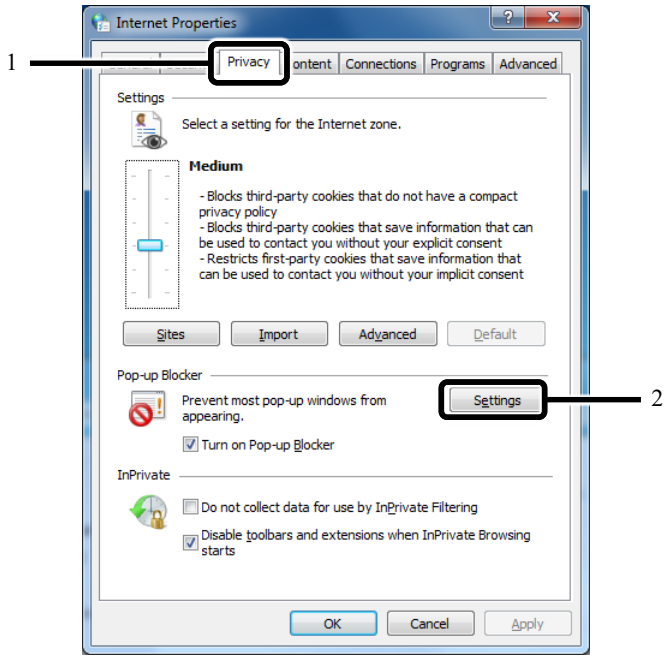


7 点击［OK］（确定）按钮，关闭对话框。
8 设定后，重新启动 Internet Explorer。

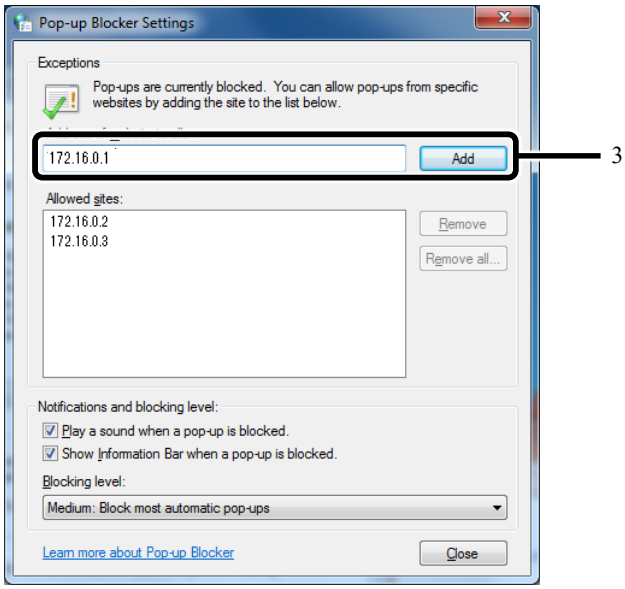
控制跳出的限制

访问机器人控制装置内的网页时，使跳出不被限制。

1 在［Internet Properties］（互联网属性）画面选择［Privacy］（隐私）标签。
2 点击［Pop-up Blocker］（弹出窗口阻止程序）的［Settings］（设定）按钮。



3 在［Address of website to allow］（允许的网站地址）中输入机器人控制装置的 IP 地址，点击［Add］按钮。

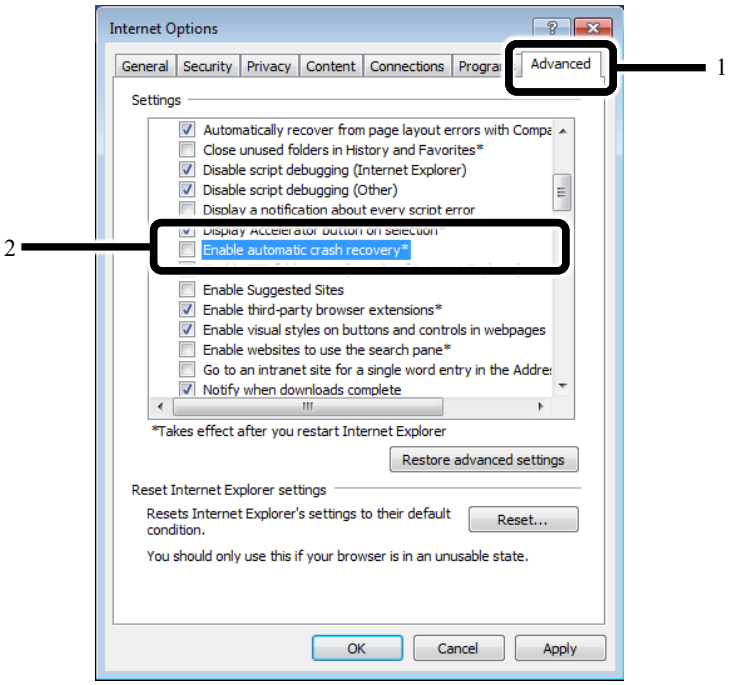


4 点击［Close］（关闭）按钮，关闭对话框。

使自动崩溃恢复功能无效

使 Internet Explorer 的自动崩溃恢复功能无效。要正确安装视觉界面控件，必须进行该设定。

1 选择［Advanced］（高级）标签。
2 去掉［Settings］（设定）的［Enable automatic crash recovery］（启用自动崩溃恢复）项目的勾选。

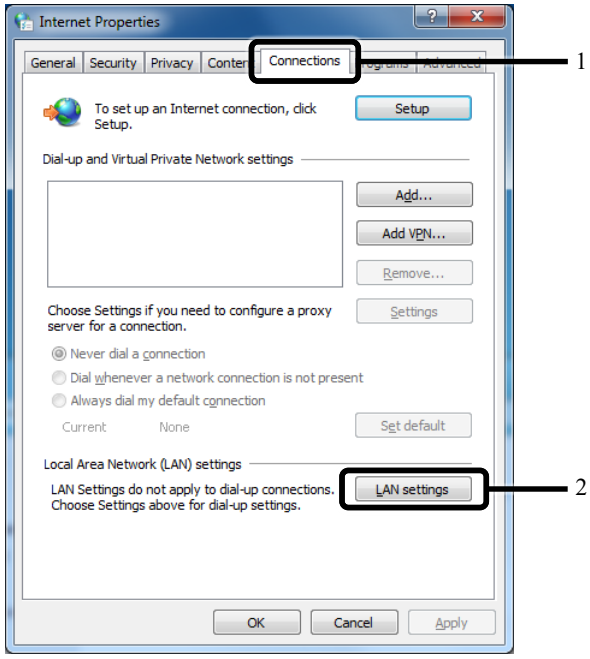


3 点击［OK］（确定）按钮，重新启动电脑。

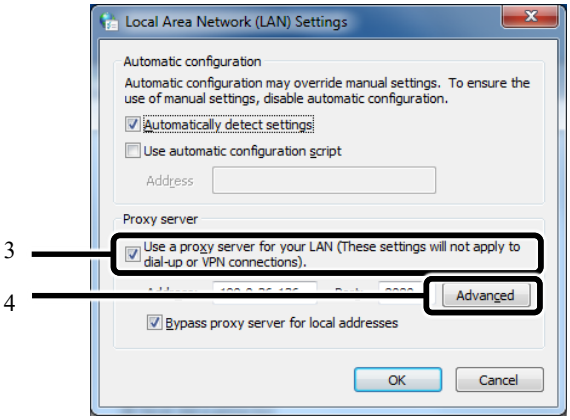
代理服务器

将机器人控制装置作为本地地址登录，与机器人控制装置的通信，不使用代理服务器。

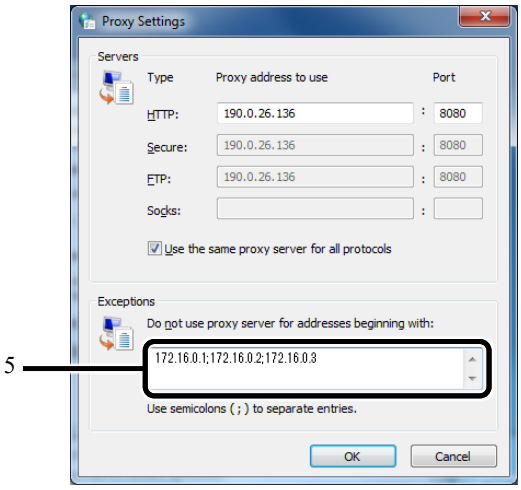
1 ［Internet Properties］（互联网属性）画面选择［Connections］（连接）标签。
2 点击［LAN settings］（局域网设定）按钮。



3 如果没有勾选［Use a proxy server for your LAN］（LAN 中使用代理），按下［OK］（确定）按钮，关闭画面。
如果勾选，则进行步骤 4～6。
4 点击［Proxy server］（代理服务器）的［Advanced］（高级）按钮。



5 在［Exceptions］（例外）中输入机器人控制装置的 IP 地址。

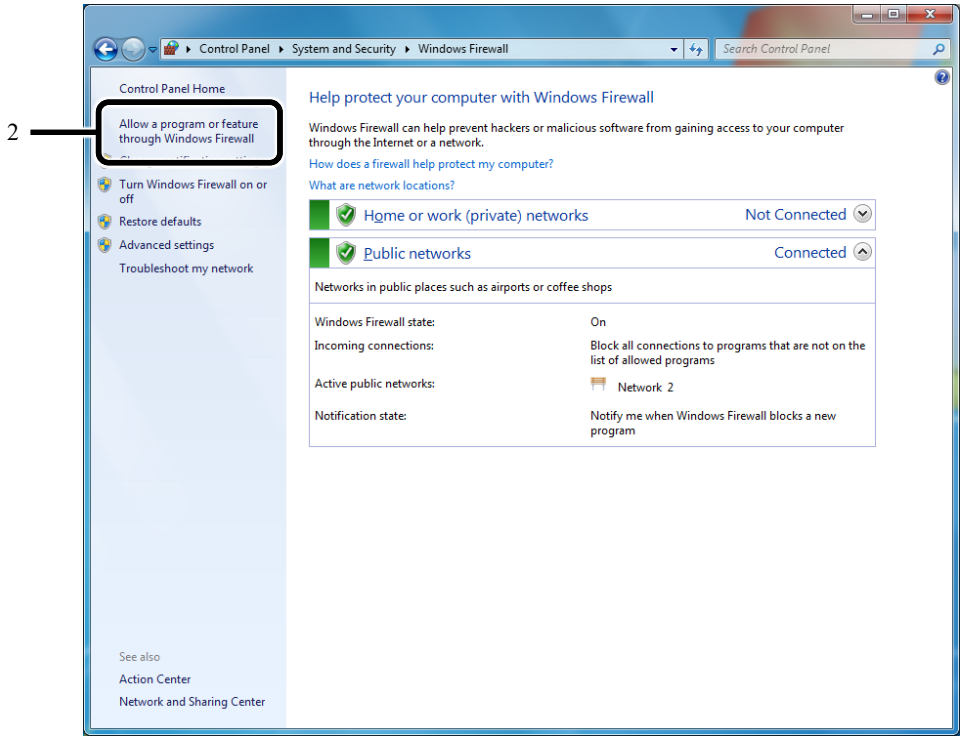


6 点击显示的所有［OK］（确定）按钮，关闭画面。

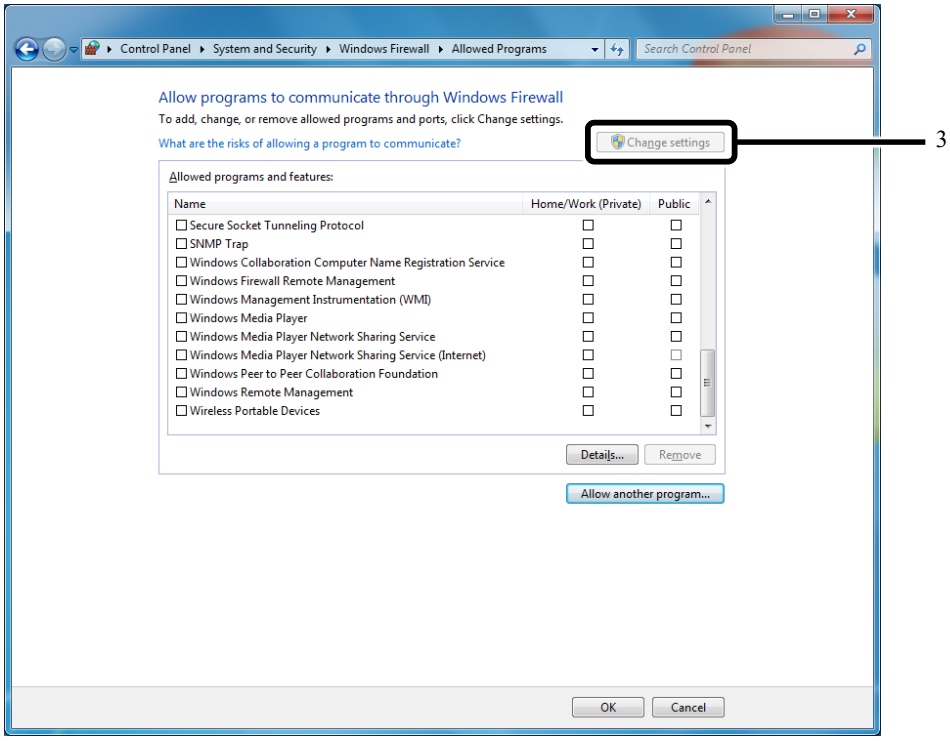
B.6 变更 Windows 防火墙的设定

变更 Windows 防火墙的设定，使与机器人控制装置的通信不被 Windows 防火墙限制。

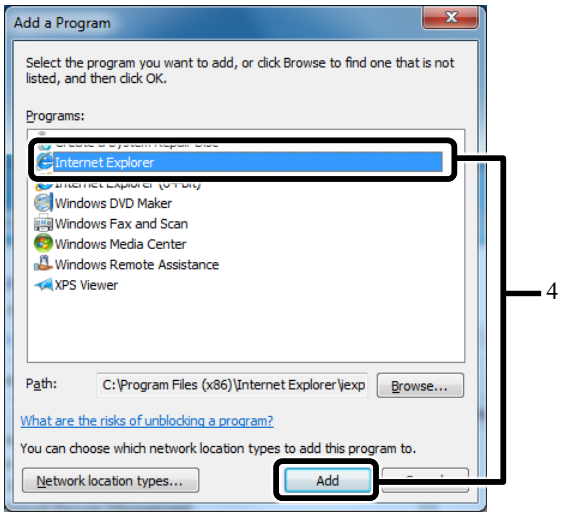
1 从控制面板打开［Windows Firewall］（Windows 防火墙）。
2 点击［Allow a program or feature through Windows Firewall］（允许程序或功能通过 Windows 防火墙）。



3 点击［Change settings］（更改设定）按钮。



4 从一览中选择［Internet Explorer］，点击［Add］按钮。



5 点击［OK］（确定）按钮，关闭对话框。

注释
此外，可能发生由于您使用的电脑中安装的 Internet Explorer 的挂件、安全软件的影响，导致无法正常与机器人控制装置通信的情况。如果在 *i*RPickTool 的设定中发生与电脑相关的问题，请首先参照“B.9 疑难解答”。

B.7 打开 *i*RPickTool 的画面

要打开 *i*RPickTool 的画面，首先以下面步骤显示机器人的网页。

1 从电脑的［Start］（开始）按钮启动 Internet Explorer。
2 在［Address］（地址）中输入机器人控制装置的 IP 地址。
显示机器人主页。

机器人的网页，是所有机器人控制装置都具备的功能。如果机器人控制装置中加入 *i*RPickTool 的选项，则机器人的网页上将显示 *i*RPickTool 相关的以下链接。

工作单元的设定

打开用于进行 *i*RPickTool 的设定的编辑画面。详细信息，请参照“6 基本功能参考”或者“7 Plug & Play 参考”。

工作单元生产状况

打开在 *i*RPickTool 中设定的，显示系统生产状况的画面。详细信息请参照“9.2 确认生产状况”。

B.8 打开 *i*RVision 的画面

视觉追踪中，需要在各个 *i*RVision 的设定画面进行相机设定、相机校准设定、视觉程序设定。带 *i*RPickTool 的队列管理的追踪，不需要这些设定。

安装视觉界面控件

要打开 *i*RVision 的设定画面，您使用的电脑中必须安装有“视觉界面控件”。第一次点击上述链接时，从机器人控制装置安装视觉界面控件。步骤如下所示。

1 在机器人主页点击 *i*RVision 的［示教和试验］。
如果您使用的电脑中已安装有视觉界面控件，则打开［*i*RVision 示教和试验］的画面。这种情况下，结束。
如果没有安装视觉界面控件，则显示以下画面。

2 点击［INSTALL］（安装）按钮。

显示确认以下文件的消息。

稍等片刻后，显示以下画面。

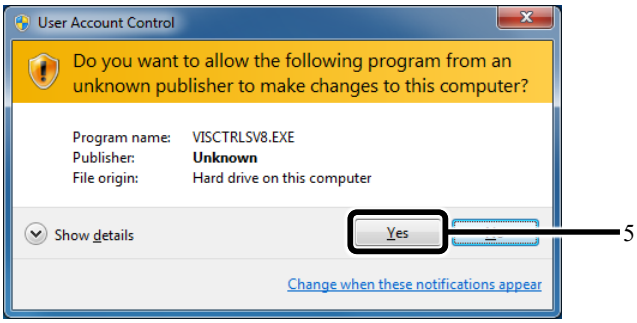
3 点击［Run］（运行）按钮。

稍等片刻后，显示以下画面。

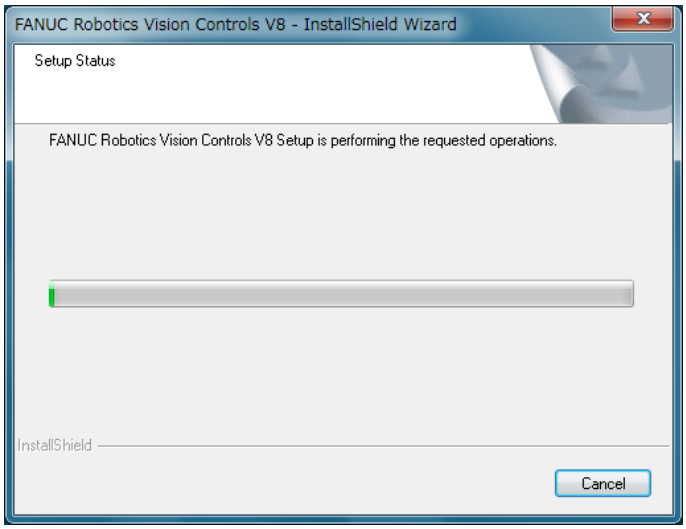
4 点击［Run］（运行）按钮。

显示如下画面。

5 点击［Yes］（是）按钮。



6 安装开始。



7 如果安装结束，则 Internet Explorer 的所有窗口关闭。

8 再次启动 Internet Explorer，打开机器人的网页。

B.9 常见问题处理方法

如果在电脑上设定 *i*RPickTool 的过程中遇到问题，请参考以下故障排除。

打不开机器人的网页

原因： 如果您使用的电脑的 Internet Explorer 被设定为使用代理服务器，则电脑和机器人控制装置之间可能无法正常通信。

对策： 请依照“B.5 变更 Internet Explorer 的设定”进行设定。

点击 *i*RPickTool 的［工作单元设置］或［工作单元生产状况］，则显示［登陆失败］

原因： Internet Explorer 的设定可能不正确。

对策： 请依照“B.6 变更 Windows Fire Wall（防火墙）的设定”进行设定。

如果点击 *i*RPickTool 的［工作单元设置］或［工作单元生产状况］，则显示［在 Internet Explorer 将弹出方式设定为有效］

原因： Internet Explorer 的设定可能不正确。

对策： 请依照“B.5 变更 Internet Explorer 的设定”进行设定。

新建工作单元的各对象时，发生运行时间错误

原因： Internet Explorer 的设定可能不正确。

对策： 请依照“B.5 变更 Internet Explorer 的设定”进行设定。

如果点击 *i*RPickTool 的［工作单元设置］或［工作单元生产状况］，则显示［70:Cannot write］（无法写入）

原因： Internet Explorer 的设定可能不正确。

对策： 请依照“B.5 变更 Internet Explorer 的设定”进行设定。

即使点击 *i*RPickTool 的［工作单元设置］或［工作单元生产状况］，窗口也不会打开

原因 1： Internet Explorer 的设定可能不正确。

对策 1： 请依照“B.6 变更 Windows Fire Wall（防火墙）的设定”进行设定。

原因 2： 此外，如果您使用的电脑中安装有安全软件，则通信可能被安全软件限制。

对策 2： 请使安全软件无效。

原因 3： 有报告说，Windows 7 中发生 Internet Explorer 无法打开新窗口的问题，该问题可能属于此问题。

对策 3： 请通过以下步骤，确认是否属于该问题。

1 用 Internet Explorer 打开机器人的网页。

2 从 Internet Explorer 的“Tools”（工具）菜单打开“Developer Tools”（开发人员工具）。（使用 F12 键也能打开）

3 选择 Developer Tools 的“Script”（脚本）标签，通过右侧的窗格选择“Console”（控制台）。

4 在机器人的主页点击 *i*RVision 的“示教和试验”。

5 控制台上显示有类似“script 16386 no such interface supported”（不支持这样的界面）的信息时，属于这种情况。

网络上有若干解决该问题的对策方案，如果您向本公司咨询，我们将告知您曾经奏效的方法。

虽然已安装了视觉界面控件，但仍然被多次要求进行视觉界面控件的安装

请确认以下项目。

原因 1： 如果机器人控制装置的 IP 地址未登录于 Internet Explorer 的“Trusted sites”（受信任的站点），则可能发生这种情况。
对策 1： 请确认/变更“B.5 变更 Internet Explorer 的设定”的受信任的站点的设定，重新启动 Internet Explorer。

原因 2： 在 Internet Explorer 的“Manage add-ons”（管理加载项）中，如果使视觉界面控件无效化，可能发生这种情况。
对策 2： 请通过以下步骤，确认视觉界面控件是否被无效化。
1 用 Internet Explorer 显示机器人的网页。
2 从 Internet Explorer 的［Tools］按钮（齿轮的图标）选择“Manage add-ons”。
3 在“Toolbars and Extensions”（工具栏和扩展）的“Show”（显示）选择“All add-ons”（所有加载项）。
4 从一览中寻找“FRImageDisplay Control”，确认“Status”（状态）为“Enable”（已启用）。
5 如果是“Disable”（已禁用），则点击右下方的［Enable（已启用）］按钮。
6 用［Close］（关闭）按钮关闭对话框，暂时结束 Internet Explorer 后重新启动。

原因 3： Internet Explorer 的“ActiveX Filtering”（ActiveX 过滤器）如果有效，则可能发生这种情况。
对策 3： 请从 Internet Explorer 的［Tools］按钮（齿轮的图标）打开“Safety”（安全），确认没有勾选“ActiveX Filtering”项。

工件单元的设定中，示教器上出现［PMON-001 通告到 PC 监视器失败］报警

原因： Windows Fire Wall（防火墙）的设定可能不正确。
对策： 请依照“B.6 变更 Windows Fire Wall（防火墙）的设定”进行设定。
此外，如果您使用的电脑中安装有安全软件，则通信可能被安全软件限制。请使安全软件无效。

如果点击 *i*RPickTool 的［工作单元设置］或［工作单元生产状况］，则显示［A problem occurred］（发生问题），Internet Explorer 关闭

原因： 由于 Internet Explorer 的挂件软件的影响，可能导致与机器人控制装置无法正常通信。
对策： 通过 Internet Explorer 的“Tools”菜单的“Manage add-ons”，发行商暂时使 FANUC Robotics North America 以及 FRNA 以外的挂件全部无效。在该状态下，可确认是否可进行工作单元的设定。如果没有问题，在确认对工作单元的设定操作没有影响后，尝试将无效的挂件一个个复原。

*i*RVision 的示教画面上，不显示图像

请确认以下项目。

原因 1： 有报告说，重新启动电脑后，会发生显示图像的情况。
对策 1： 重新启动电脑，重新打开 *i*RVision 的示教画面。

原因 2： 如果以没有 Administrator 权限的用户登录您使用的电脑，则可能发生电脑和机器人无法正常通信的情况。
对策 2： 请以有 Administrator 权限的用户重新登录。

原因 3： 如果您使用的电脑中安装有 Microsoft Internet Information Server，且 Worldwide Web Publishing Service 有效，则电脑和机器人控制装置之间可能无法正常通信。
对策 3： 请使 Worldwide Web Publishing Service 无效。

原因 4： 对于被设定为相同 IP 地址的不同软件版本的多个机器人控制装置，如果以 1 台电脑进行 *i*RVision 的示教，则可能发生不正确显示示教画面，示教作业无法正常进行等情况。这是因为 Internet Explorer 将一端的控制装置的示教画面页高速缓存，还将该页数据用于其他控制装置的画面显示。

对策 4： 这种情况下，请从 Internet Explorer 的“Tools”菜单选择“Internet Options”（Internet 选项），用“General”（常规）标签进行“Delete”（删除），删除 Internet Explorer 的浏览履历。进行控制装置的软件交换时，也会发生相同情况。

> 注释
>
> 最新的 Internet Explorer 的“Delete Browsing History”（删除浏览履历）对话框中，有“Preserve favorite website data”（保留收藏夹网站数据）复选框。如果勾选，好像可以实现不删除登录于收藏夹的网页浏览履历。例如，将某机器人的网页添加至收藏夹，则该机器人的 *i*RVision 画面的浏览履历将不会被删除。请去掉该勾选后，删除浏览履历。

读入图像文件时，发生［Runtime error '0'］（运行时间错误'0'）

原因： 如果互联网信息服务（IIS）有效，则可能无法与机器人控制装置正常通信。

对策： 从控制面板打开“Add or Remove Program”（添加/删除程序），在“Windows Components”（Windows 组件）的一览中去掉“Internet Information Services （IIS）”（Internet 信息服务（IIS））的勾选。

遮罩的设定按钮被隐藏，无法操作

请确认以下项目。

原因 1： 如果 Internet Explorer 的受信任的站点的设定不正确，可能使遮罩的设定按钮被隐藏，导致无法操作。

对策 1： 请根据“B.5 变更 Internet Explorer 的设定”，确认机器人的 IP 地址登录于受信任的站点。

原因 2： 如果在 *i*RVision 的示教窗口显示 Internet Explorer 的地址栏、状态栏，则可能发生这种情况。

对策 2： 请根据“B.5 变更 Internet Explorer 的设定”，确认勾选了［Allow websites to open windows without address or status bars］（允许网站打开没有地址或状态栏的窗口）的［Enable］（启用）。

示教画面上显示地址栏

原因： 如果 Internet Explorer 的受信任的站点的设定不正确，可能在示教画面上会显示地址栏，无法操作。

对策： 请确认/变更“B.5 变更 Internet Explorer 的设定”的受信任的站点的设定，重新启动 Internet Explorer。

如果进行遮罩的设定，会发生“CVIS-005 文件访问被拒”报警

原因： 如果互联网信息服务（IIS）有效，则可能无法与机器人控制装置正常通信。

对策： 从控制面板打开“Add or Remove Program”（添加/删除程序），在“Windows Components”（Windows 组件）的一览中去掉“Internet Information Services （IIS）”（Internet 信息服务（IIS））的勾选。

无法安装视觉界面控件

对策： 请在您使用的控制装置上，确认安装了“*i*RVision UIF Controls”（*i*RVision UIF 控制）选项。只有在安装有该选项的控制装置中，可安装视觉界面控件。如果参加视觉追踪的所有机器人控制装置都没有安装本选项，请联系本公司销售负责人员。

C HDI 设定

本章就有关 HDI 的硬件连接方法以及软件的设定方法进行说明。

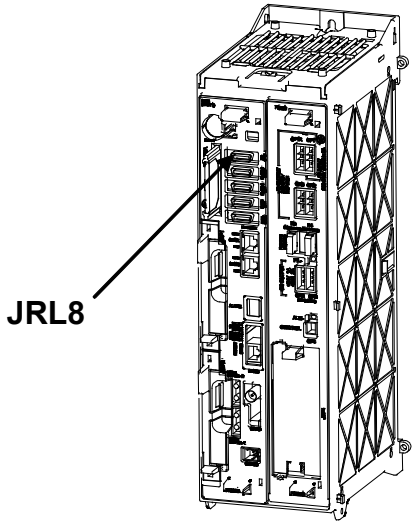
注释
使用 HDI，需要追踪拾取 HDI 功能选项。

⚠注意
HDI 信号不能作为通用的 DI 使用。

C.1 接线图

本节就将 HDI 接口和编码器的接线图进行说明。

C.1.1 R-30*i*B Plus 时



R-30*i*B Plus

主板
JRL8

1	Reserved	11	HDI0
2	Reserved	12	0V
3	Reserved	13	HDI1
4	Reserved	14	0V
5	Reserved	15	HDI2
6	Reserved	16	0V
7	Reserved	17	HDI3
8	Reserved	18	0V
9	Reserved	19	HDI4
10	Reserved	20	0V

图 C.1.1 (a) R-30*i*B Plus HDI 接口

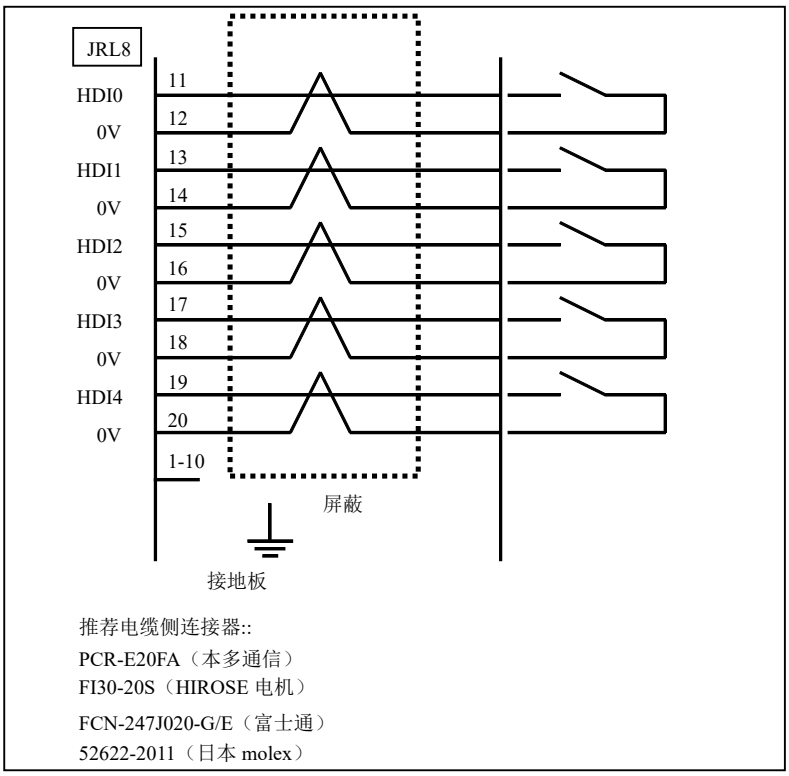
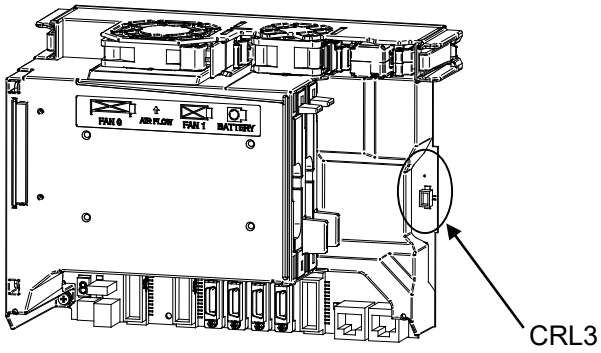


图 C.1.1 (b) R-30*i*B Plus HDI 电缆连接

C.1.2 R-30*i*B Mate Plus 时



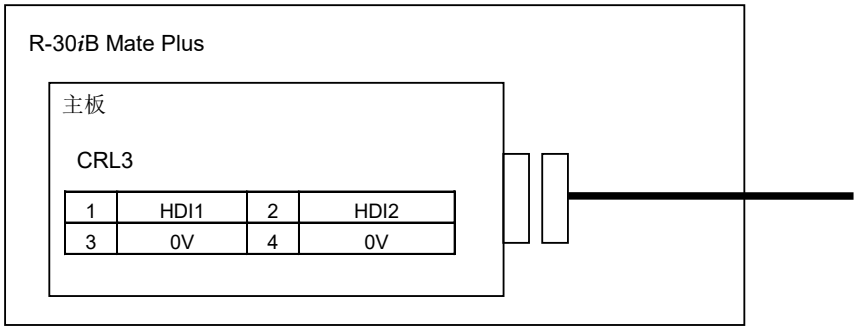


图 C.1.2 (a) R-30*i*B Mate Plus HDI 接口

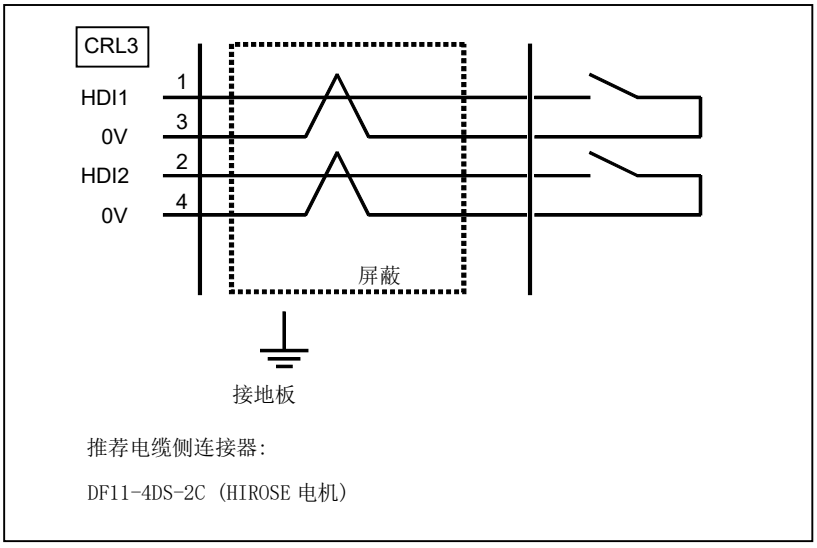
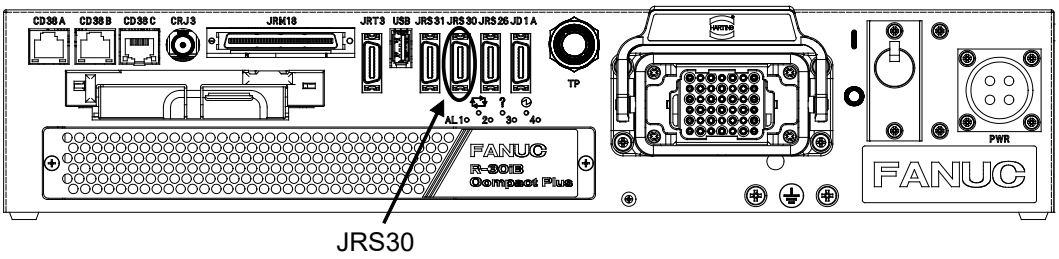
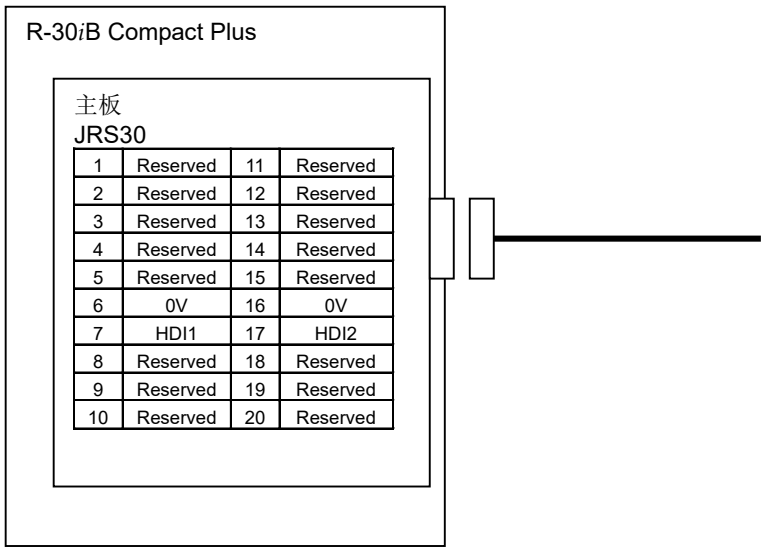


图 C.1.2 (b) R-30*i*B Mate Plus HDI 电缆连接

C.1.3 R-30*i*B Compact Plus 时





1	Reserved	11	Reserved
2	Reserved	12	Reserved
3	Reserved	13	Reserved
4	Reserved	14	Reserved
5	Reserved	15	Reserved
6	0V	16	0V
7	HDI1	17	HDI2
8	Reserved	18	Reserved
9	Reserved	19	Reserved
10	Reserved	20	Reserved

图 C.1.3 (a) R-30*i*B Compact Plus HDI 接口

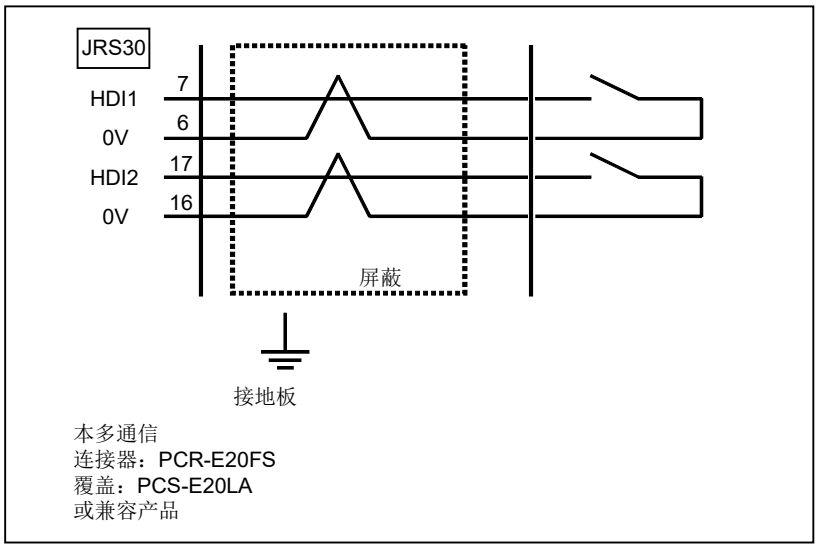
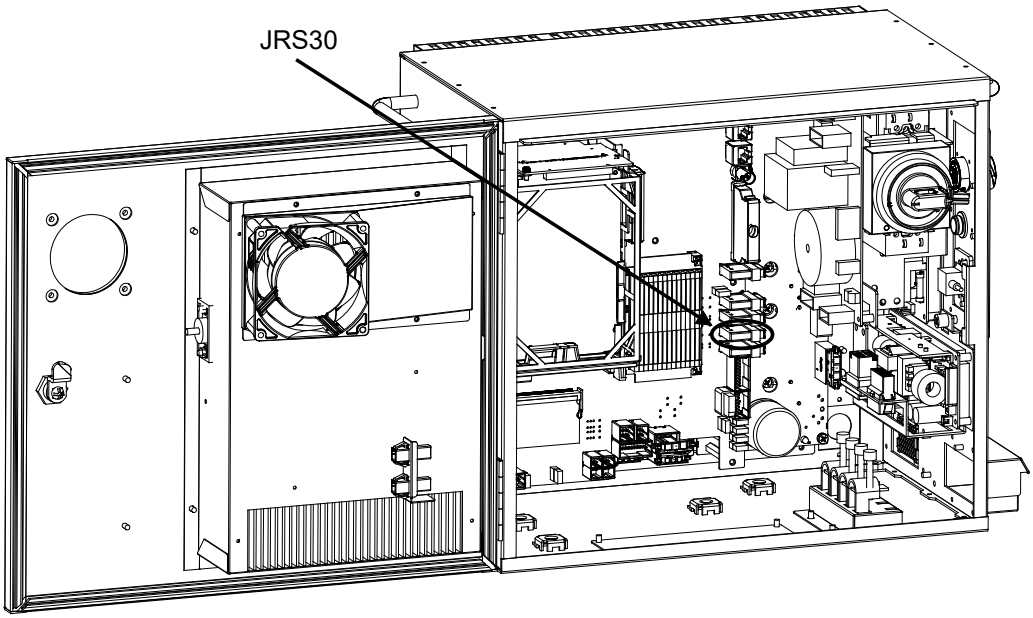


图 C.1.3 (b) R-30*i*B Compact Plus HDI 电缆连接

C.1.4 R-30*i*B Mini Plus 时



R-30*i*B Mini Plus

主板
JRS30

1	Reserved	11	Reserved
2	Reserved	12	Reserved
3	Reserved	13	Reserved
4	Reserved	14	Reserved
5	Reserved	15	Reserved
6	0V	16	0V
7	HDI1	17	HDI2
8	Reserved	18	Reserved
9	Reserved	19	Reserved
10	Reserved	20	Reserved

图 C.1.4 (a) R-30*i*B Mini Plus HDI 接口

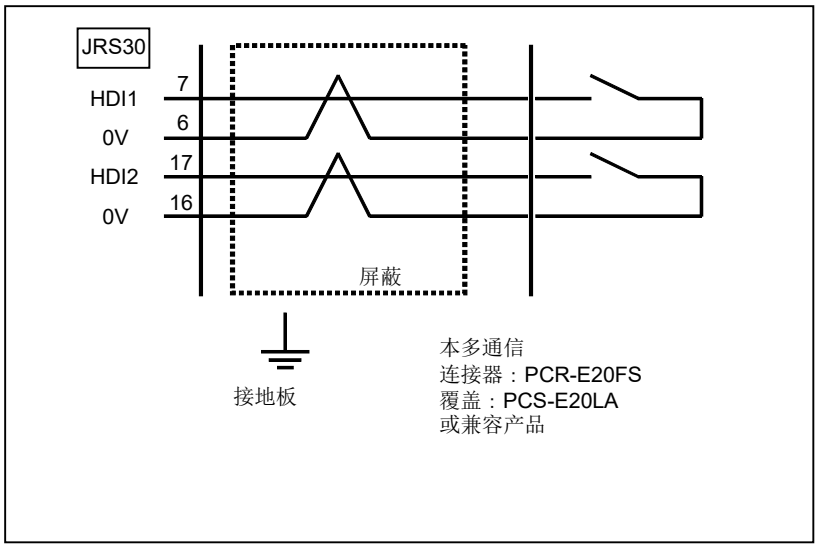


图 C.1.4 (b) R-30*i*B Mini Plus HDI 电缆连接

C.2 有关输入信号

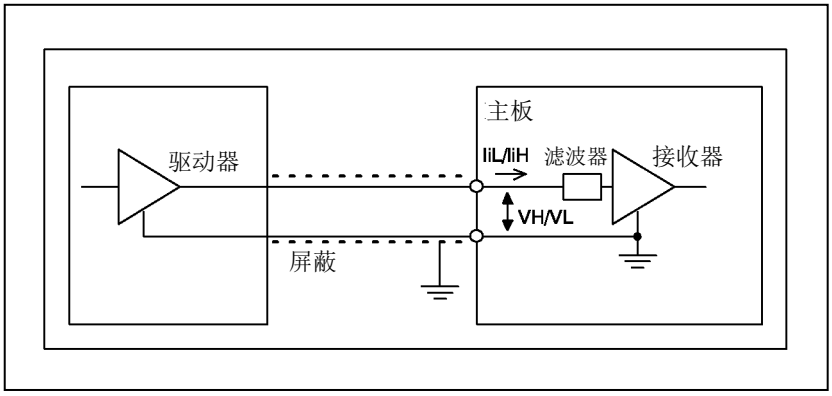


图 C.2(a) HDI 输入信号

绝对最大额定值
输入电压范围 Vin：-3.6V～+10V
输入特性

项目	记号	规格	单位	备注
高输入电压	VH	3.6～11.6	V	
低输入电压	VL	0～1.0	V	
高输入电流	IiH	2 max	mA	Vin=5V
		11 max	mA	Vin=10V
低输入电流	IiL	-8.0 max	mA	Vin=0V
输入信号脉冲宽度		20 min	μs	
输入信号的延迟、差异		0.02(max)	ms	

注释
1 IiH、IiL 的符号，+表示流入接收器的方向，－表示流出方向。
2 快速跳过信号的输入电压为低位时视为“1”，高位时视为“0”。

C.3 设定方法

C.3.1 概要

追踪拾取 HDI 功能（J831），是用于传送带的速度为高速时，也能进行正确的工件位置检出的功能。本功能中，作为工件检出用的输入信号，不使用通常的触发输入信号（DI），而使用 HDI（应用程序用的高速 DI）。
为了使用追踪拾取 HDI 功能，需要以下操作。

- 在编码器设定画面上设定 HDI 端口编号。

C.3.2 追踪拾取 HDI 功能的有效

要使追踪拾取 HDI 功能有效时，请依照下述步骤 C-1 进行设定。

步骤 C-1 使触发输入周期的高速化功能有效的设定

准备

- 准备追踪拾取 HDI 功能选项。
- 将工件检出硬件与 HDI 连接。（有关硬件的连接方法，请参照“附录 C.1”、“C.2”。）
- R-30*i*B Plus 的情况，可使用的 HDI 端口是#1～#5。从主板上的 JRL8 连接器输入。

步骤

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
2 选择［下页］→［系统］→［系统变量］。
3 使光标移动至以下的系统变量的位置。
4 确认系统变量值是否已被设正确设定为以下值。
订购了追踪拾取用 HDI 功能选项时，默认情况下已被设定为以下值。
- $HSLTENBL = TRUE
- $LNCFG.$HSDI_ENABLE = TRUE

5 变更了系统变量值时，为了使设定有效，将控制装置的电源 OFF/ON。
6 打开编码器设定画面，在 HDI 端口编号的项目中输入使用的 HDI 端口编号。
7 为了使设定有效，重新接通机器人控制装置的电源。

⚠注意
如果安装了追踪拾取 HDI 功能，在使用通常的 DI 输入示教追踪子程序时，请设定为系统变量 $LNCFG.$HSDI_ENABLE=FALSE。否则，可能无法编辑追踪程序。

D αA1000S 脉冲编码器

本章对αA1000S 脉冲编码器（A860-0372-T001）的使用方法进行说明。

D.1 必要的结构

表 D.1(a) R-30*i*B Plus: 使用αA1000S 脉冲编码器 A860-0372-T001 时的结构

装置	R-30*i*B Plus 控制装置 A-控制柜	R-30*i*B Plus 控制装置 B-控制柜	注释
硬件			
追踪拾取用接口板	A20B-8101-0421（迷你宽槽） 或 A20B-8101-0601（迷你槽）	A20B-8101-0421（迷你宽槽） 或 A20B-8101-0601（迷你槽）	• 从以下的规格，准备左记印刷电路板。 A05B-2600-J035, A05B-2660-J035（A/B-控制柜共通、迷你宽槽） A05B-2600-J036, A05B-2660-J036（A-控制柜、迷你槽） A05B-2600-J037（B-控制柜、迷你槽） • 另置型接口装置（SDU） A02B-0323-C205 可代替追踪拾取用接口板使用。 但是，这种情况下，需要进行用于装置安装的加工。（详细信息，请参照“附录 F 另置型接口装置（SDU）”。 • 也可将αA1000S 脉冲编码器连接到主 CPU 板上的编码器端子(JD17)。此时，虽然不需要生产线追踪用接口板，但不可与学习减振功能(A05B-2600-J573) (选项)同时使用。
光纤（FSSB）	A66L-6001-0023	A66L-6001-0023	• 从以下的规格，准备左记光纤。 A05B-2600-J035, A05B-2660-J035（A/B-控制柜共通、迷你宽槽） A05B-2600-J036, A05B-2660-J036（A-控制柜、迷你槽） A05B-2600-J037（B-控制柜、迷你槽）

装置	R-30*i*B Plus 控制装置 A-控制柜	R-30*i*B Plus 控制装置 B-控制柜	注释
硬件			
脉冲编码器电缆 (增量使用 αA1000S 时)	A20B-8101-0421 连接迷你宽槽的情况： A05B-2601-J220（7M） A05B-2601-J221 （14M） A05B-2601-J222 （20M） A05B-2601-J223 （30M）	A20B-8101-0421 连接迷你宽槽的情况： A05B-2603-J220（7M） A05B-2603-J221 （14M） A05B-2603-J222 （20M） A05B-2603-J223 （30M）	• 增量使用αA1000S 脉冲编码器时，本表中的电缆不可使用。 • 使用脉冲分配器时，αA1000S 脉冲编码器不可使用。希望使用αA1000S 脉冲编码器，用多个机器人追踪工件时，使用以太网编码器功能(A05B-2600-R762) (选项)。
	A20B-8101-0601 连接迷你槽的情况（1 个脉冲编码器）： A05B-2601-J210（7M） A05B-2601-J211 （14M） A05B-2601-J212 （20M） A05B-2601-J213 （30M）	A20B-8101-0601 连接迷你槽的情况（1 个脉冲编码器）： A05B-2603-J210（7M） A05B-2603-J211 （14M） A05B-2603-J212 （20M） A05B-2603-J213 （30M）	
	A20B-8101-0601 连接迷你槽的情况（2 个脉冲编码器）： A05B-2601-J260（7M） A05B-2601-J261 （14M） A05B-2601-J262 （20M） A05B-2601-J263 （30M）	A20B-8101-0601 连接迷你槽的情况（2 个脉冲编码器）： A05B-2603-J260（7M） A05B-2603-J261 （14M） A05B-2603-J262 （20M） A05B-2603-J263 （30M）	
	※连接到主 CPU 板时： A05B-2601-J270（7M） A05B-2601-J271 （14M） A05B-2601-J272 （20M） A05B-2601-J273 （30M）	※连接到主 CPU 板时： A05B-2603-J270（7M） A05B-2603-J271 （14M） A05B-2603-J272 （20M） A05B-2603-J273 （30M）	

表 D.1 (b) 需要条件(R-30*i*B Mate Plus:使用αA1000S 脉冲编码器 A860-0372-T001 时)

装置	R-30*i*B Mate Plus 控制装置	R-30*i*B Mate Plus 控制装置 (外气导入型)	注释
硬件			
生产线追踪用接口板	A20B-8101-0601 (迷你槽)	A20B-8101-0601 (迷你槽)	• 根据以下准备规格准备左边的印刷板。 A05B-2650-J035, A05B-2661-J035 (R-30*i*B Mate Plus、迷你槽) A05B-2655-J035, A05B-2662-J035 (R-30*i*B Mate Plus (外气导入型)、迷你槽) • 也可将αA1000S 脉冲编码器连接到主 CPU 板上的编码器端子(CRS41)。此时，虽然不需要生产线追踪用接口板，但不可与学习减振功能(A05B-2660-J573) (选项)同时使用。
光纤(FSSB)	A66L-6001-0026	A66L-6001-0023	• 根据以下准备规格准备左边的光纤。 A05B-2650-J035, A05B-2661-J035 (R-30*i*B Mate Plus 迷你槽) A05B-2655-J035, A05B-2662-J035 (R-30*i*B Mate Plus (外气导入型)、迷你槽)
脉冲编码器电缆(增量使用αA1000S 脉冲编码器时)	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 1 个)： A05B-2650-J205 (7M) A05B-2650-J206 (14M) A05B-2650-J207 (20M) A05B-2661-J205 (7M) A05B-2661-J206 (14M) A05B-2661-J207 (20M)	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 1 个)： A05B-2655-J205 (7M) A05B-2655-J206 (14M) A05B-2655-J207 (20M) A05B-2662-J205 (7M) A05B-2662-J206 (14M) A05B-2662-J207 (20M)	• 以绝对使用αA1000S 脉冲编码器时，本表中的电缆不可使用。 • 使用脉冲分配器时，αA1000S 脉冲编码器不可使用。希望使用αA1000S 脉冲编码器，用多个机器人追踪工件时，使用以太网编码器功能(A05B-2660-R762) (选项)。 • 在 R-30*i*B Mate Plus 用主 CPU 板 A (A20B-8201-0750)中，不可使用将αA1000S 脉冲编码器连接到主 CPU 板的接口。关于αA1000S 脉冲编码器可连接的主 CPU 板，请参照表 D.1 (d)。
	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 2 个)： A05B-2650-J215 (7M) A05B-2650-J216 (14M) A05B-2650-J217 (20M) A05B-2661-J215 (7M) A05B-2661-J216 (14M) A05B-2661-J217 (20M)	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 2 个)： A05B-2655-J215 (7M) A05B-2655-J216 (14M) A05B-2655-J217 (20M) A05B-2662-J215 (7M) A05B-2662-J216 (14M) A05B-2662-J217 (20M)	
	连接到主 CPU 板时： A05B-2650-J220 (7M) A05B-2650-J221 (14M) A05B-2650-J222 (20M) A05B-2661-J220 (7M) A05B-2661-J221 (14M) A05B-2661-J222 (20M)	连接到主 CPU 板时： A05B-2655-J220 (7M) A05B-2655-J221 (14M) A05B-2655-J222 (20M) A05B-2662-J220 (7M) A05B-2662-J221 (14M) A05B-2662-J222 (20M)	

表 D.1 (c) 需要条件 (R-30*i*B Compact Plus /R-30*i*B Mini Plus:使用αA1000S 脉冲编码器 A860-0372-T001 时)

装置	R-30*i*B Compact Plus 控制装置	R-30*i*B Mini Plus 控制装置	注释
硬件			
脉冲编码器电缆 (增量使用 αA1000S 脉冲编码器时)	连接到主 CPU 板时（脉冲编码器 1 个）： A05B-2690-J300 (7M) A05B-2690-J301 (14M) A05B-2690-J302 (20M)	连接到主 CPU 板时（脉冲编码器 1 个）： A05B-2696-J300 (7M) A05B-2696-J301 (14M) A05B-2696-J302 (20M)	• 以绝对使用αA1000S 脉冲编码器时，本表中的电缆不可使用。 • 希望使用αA1000S 脉冲编码器，用多个机器人追踪工件时，使用以太网编码器功能 (A05B-2660-R762) (选项)。 • 在 R-30*i*B Compact Plus 用主 CPU 板 A (A17B-8100-0800)中，不可使用将 αA1000S 脉冲编码器连接到主 CPU 板的接口。关于αA1000S 脉冲编码器可连接的主 CPU 板，请参照表 D.1 (e)。
	连接到主 CPU 板时（脉冲编码器 2 个）： A05B-2690-J310 (7M) A05B-2690-J311 (14M) A05B-2690-J312 (20M)	连接到主 CPU 板时（脉冲编码器 2 个）： A05B-2696-J310 (7M) A05B-2696-J311 (14M) A05B-2696-J312 (20M)	

关于脉冲编码器信号信息、规格、尺寸，请参照图 D.2 (a) ～ (b)。
与连接及安装相关的信息，请参照以下内容。

- R-30*i*B Plus 时： 图 D.3.1 (a) ～ 图 D.3.1 (e)
- R-30*i*B Mate Plus 时： 图 D.3.2 (a) ～ 图 D.3.2 (d)
- R-30*i*B Compact Plus 时： 图 D.3.3 (a) ～ 图 D.3.3 (c)
- R-30*i*B Mini Plus 时： 图 D.3.4 (a) ～ 图 D.3.4 (c)

αA1000S 脉冲编码器根据控制装置的印刷板图号不同，无法连接以主 CPU 板。详情如下所示，请在讨论构成时注意。

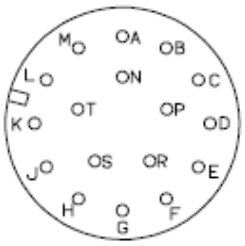
表 D.1 (d) R-30*i*B Mate Plus 主 CPU 板的所需条件 (αA1000S 脉冲编码器)

主 CPU 板	印刷板图号	是否可以连接αA1000S 脉冲编码器	注释
主 CPU 板 A	A20B-8201-0750 (R-30*i*B Mate Plus)	不可连接	• 根据以下准备规格准备左边的主 CPU 板。 A05B-2680-H001 (R-30*i*B Mate Plus) A05B-2685-H001 (R-30*i*B Mate Plus 外气导入)
主 CPU 板 B	A20B-8201-0751 (R-30*i*B Mate Plus)	可连接	• 根据以下准备规格准备左边的主 CPU 板。 A05B-2680-H002 (R-30*i*B Mate Plus) A05B-2685-H002 (R-30*i*B Mate Plus 外气导入)
主 CPU 板 C	A20B-8201-0752 (R-30*i*B Mate Plus)	可连接	• 根据以下准备规格准备左边的主 CPU 板。 A05B-2680-H001 (R-30*i*B Mate Plus) A05B-2685-H003 (R-30*i*B Mate Plus 外气导入)

表 D.1 (e) R-30*i*B Compact Plus 主 CPU 板的所需条件 (αA1000S 脉冲编码器)

主 CPU 板	印刷板图号	是否可以连接αA1000S 脉冲编码器	注释
主 CPU 板 A	A17B-8100-0800	不可连接	• 根据以下准备规格准备左边的主 CPU 板。 A05B-2690-H001
主 CPU 板 B	A17B-8100-0801	可连接	• 根据以下准备规格准备左边的主 CPU 板。 A05B-2690-H002

D.2 概略图



信号名称	定位销名称 αA1000S 3102A20 - 29P
SD *SD	A D
REQ *REQ	F G
+5V 0V	J,K N,T
屏蔽	H
+6VA 0VA	R S

图 D.2 (a) αA1000S 脉冲编码器 （A860-0372-T001） 的连接用信号信息

表 D.2 αA1000S 脉冲编码器 （A860-0372-T001） 规格

项目		规格
电源电压		5 [V]±5%
消耗电流		最大 0.3 [A]
使用温度范围		0～+60 [℃]
分辨率		1,000,000 [/rev.] （但是，在追踪拾取功能中使用的分辨率为 65,536 [/rev.]）
最高转速		4000 [min^{-1}]
输入轴惯性		最大 1×10^{-4}($kg\cdot m^2$)
输入轴启动扭矩		最大 0.1[N・m]
输入轴容许荷重	径向负载	98 [N]
	轴向负载	49 [N]
轴旋转下降		0.02×10^{-3} [m]
构造		防尘、防滴结构 （相当于 IP55：嵌合防水连接器时）
耐振加速度		5 [G] （50～2,000[Hz]）
重量		约 0.75 [kg]

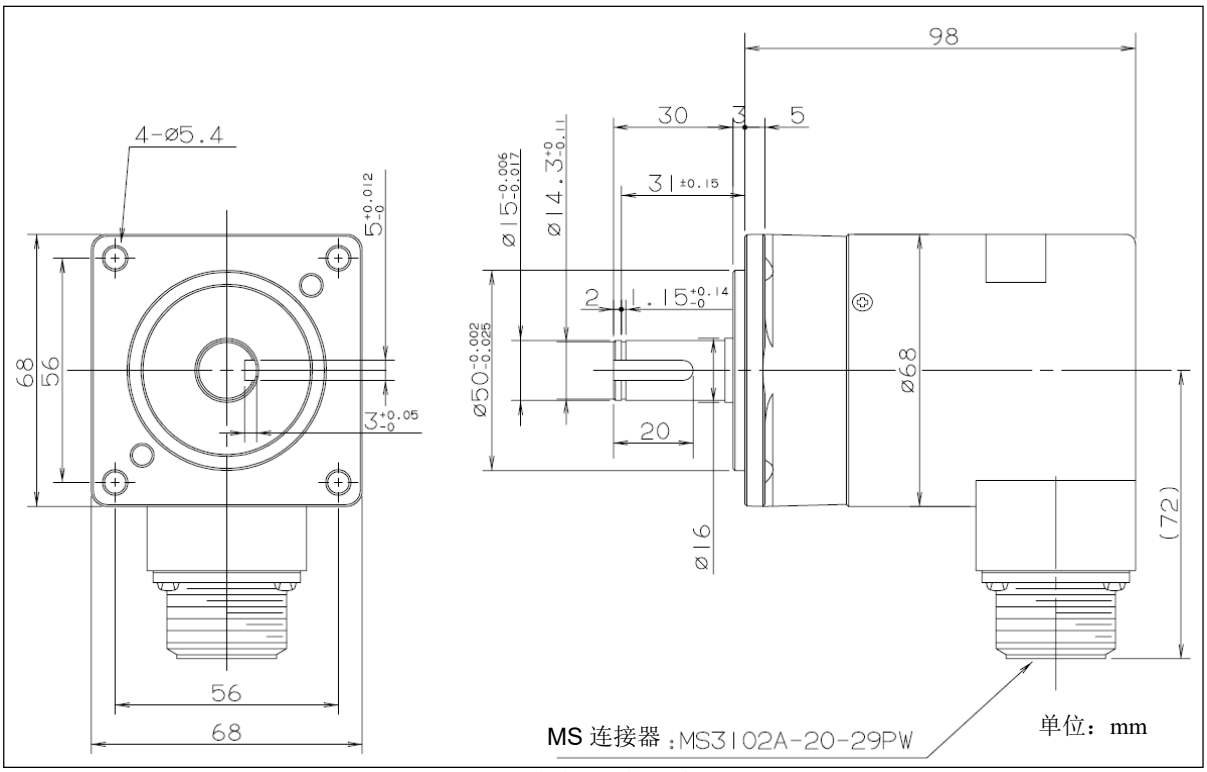


图 D.2 (b) αA1000S 脉冲编码器的尺寸（A860-0372-T001）

D.3 连接方法

在使用 R-30*i*B Plus / R-30*i*B Mate Plus / R-30*i*B Compact Plus / R-30*i*B Mini Plus 的线性跟踪中，可以采用表 D.3 所示的编码器构成。

注释
可以连接到主 CPU 板编码器端子的编码器仅为αA1000S 脉冲编码器 (A860-0372-T001)。

表 D.3 编码器构成例子 (使用αA1000S 脉冲编码器时)

编码器数	编码器构成例子	注释
1	• **编码器编号 1 :** 连接到主 CPU 板的编码器端子 (R-30*i*B Plus: JD17、R-30*i*B Mate Plus: CRS41(*注释 1) / R-30*i*B Compact Plus: JRS31(*注释 2) / R-30*i*B Mini Plus: JRS31)	
2	• **编码器编号 1, 2 :** R-30*i*B Plus / R-30*i*B Mate Plus 时，连接到线性跟踪用接口板的编码器端子 JF21～JF22 R-30*i*B Compact Plus(*注释 2) / R-30*i*B Mini Plus 时，连接到主 CPU 板的编码器端子 JRS31	
3	• **编码器编号 1, 2 :** 连接到线性跟踪用接口板的编码器端子 JF21～JF22 • **编码器编号 3 :** 连接到主 CPU 板的编码器端子 (R-30*i*B Plus: JD17、R-30*i*B Mate Plus: CRS41(*注释 1))	• 组合生产线追踪用接口板和主 CPU 板的接口构成时，请务必将连接到主 CPU 板的编码器设定成最后的编码器编号。在左边的构成示例中，将连接到生产线追踪用接口板的脉冲编码器设定成编码器编号 1,2,将连接到主 CPU 板的脉冲编码器设定成编码器编号 3。
4	• **编码器编号 1～4 :** 连接到分离型接口装置 SDU 基本单元的编码器端子 JF101～JF104	• 使用分离型接口装置(SDU) 时，需要进行加工用以安装装置。请参照“附录 F 另置型接口装置（SDU）”。
5	• **编码器编号 1～4 :** 连接到分离型接口装置(SDU)基本单元的编码器端子 JF101～JF104 • **编码器编号 5 :** 连接到主 CPU 板的编码器端子 (R-30*i*B Plus: JD17、R-30*i*B Mate Plus: CRS41(*注释 1))	• 组合分离型接口装置(SDU)和主 CPU 板的接口构成时，请务必将连接到主 CPU 板的编码器设定成最后的编码器编号。在左边的构成示例中，将连接到分离型接口装置的脉冲编码器设定成编码器编号 1～4，将连接到主 CPU 板的脉冲编码器设定成编码器编号 5。 • 使用分离型接口装置(SDU) 时，需要进行加工用以安装装置。请参照“附录 F 另置型接口装置（SDU）”。
6	• **编码器编号 1～4 :** 连接到分离型接口装置(SDU)基本单元的编码器端子 JF101～JF104 • **编码器编号 5, 6 :** 连接到分离型接口装置(SDU)扩展单元的编码器端子 JF105～JF106	• 使用分离型接口装置(SDU)时，需要进行加工用以安装装置。请参照“附录 F 另置型接口装置（SDU）”。
7	• **编码器编号 1～4 :** 连接到分离型接口装置(SDU)基本单元的编码器端子 JF101～JF104 • **编码器编号 5～7 :** 连接到分离型接口装置(SDU)扩展单元的编码器端子 JF105～JF107	• 使用分离型接口装置(SDU)时，需要进行加工用以安装装置。请参照“附录 F 另置型接口装置（SDU）”。
8	• **编码器编号 1～4 :** 连接到分离型接口装置(SDU)基本单元的编码器端子 JF101～JF104 • **编码器编号 5～8 :** 连接到分离型接口装置(SDU)扩展单元的编码器端子 JF105～JF108	• 使用分离型接口装置(SDU)时，需要进行加工用以安装装置。请参照“附录 F 另置型接口装置（SDU）”。

注释
1 R-30*i*B Mate Plus 用主 CPU 板 A(A20B-8201-0750)中不可使用将αA1000S 脉冲编码器连接到主 CPU 板的接口。此时，请使用生产线追踪用接口板。关于αA1000S 脉冲编码器可连接的主 CPU 板，请参照表 D.1 (d)。
2 R-30*i*B Compact Plus 用主 CPU 板 A(A17B-8100-0800)中不可使用将αA1000S 脉冲编码器连接到主 CPU 板的接口。关于αA1000S 脉冲编码器可连接的主 CPU 板，请参照表 D.1 (e)。
3 不能使用生产线追踪用接口板、主 CPU 板时，请使用附录 F 所示的分离型接口装置（SDU）。

D.3.1 R-30*i*B Plus 时

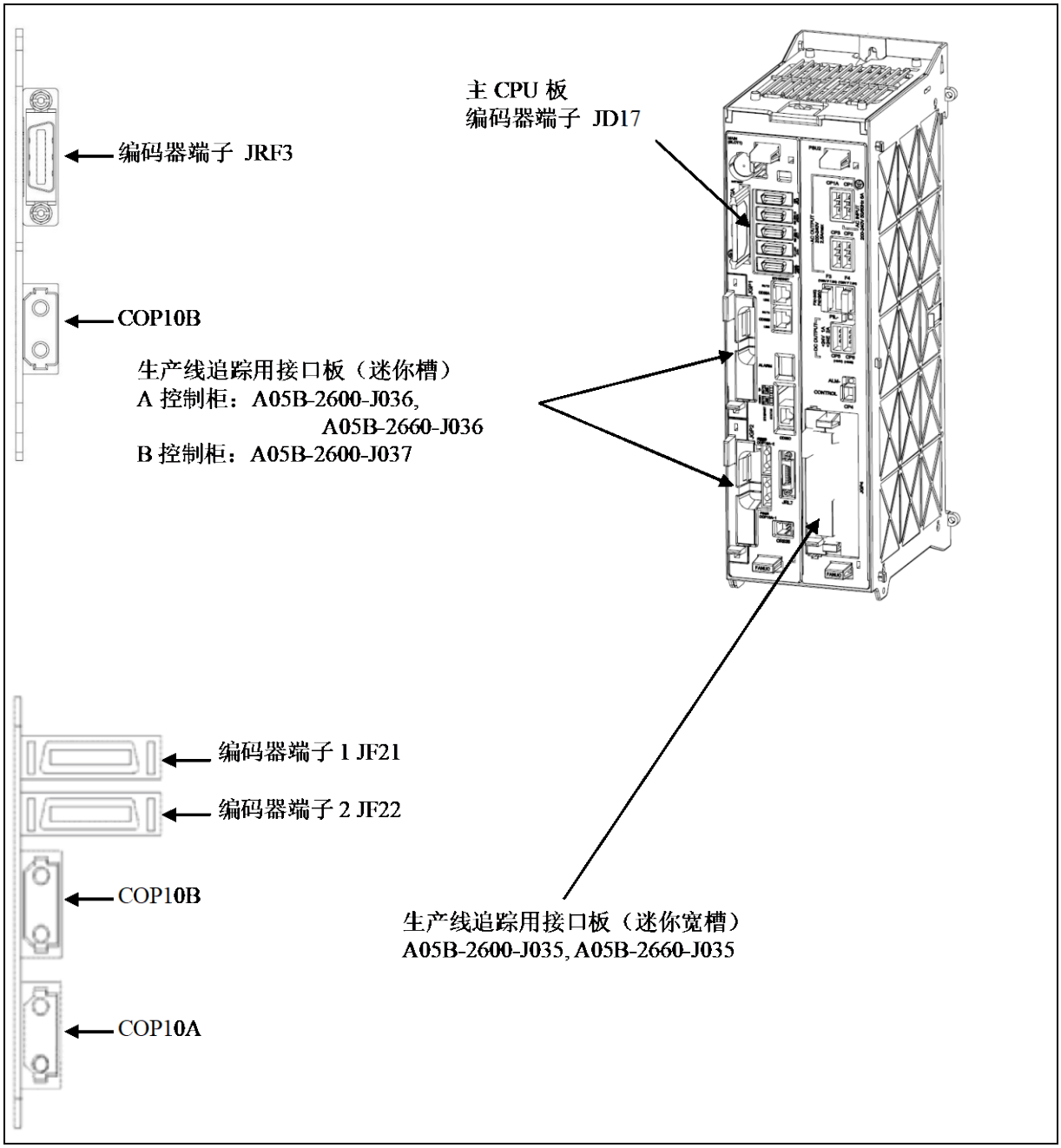


图 D.3.1 (a) 连接编码器(脉冲编码器)的 R-30*i*B Plus 控制装置

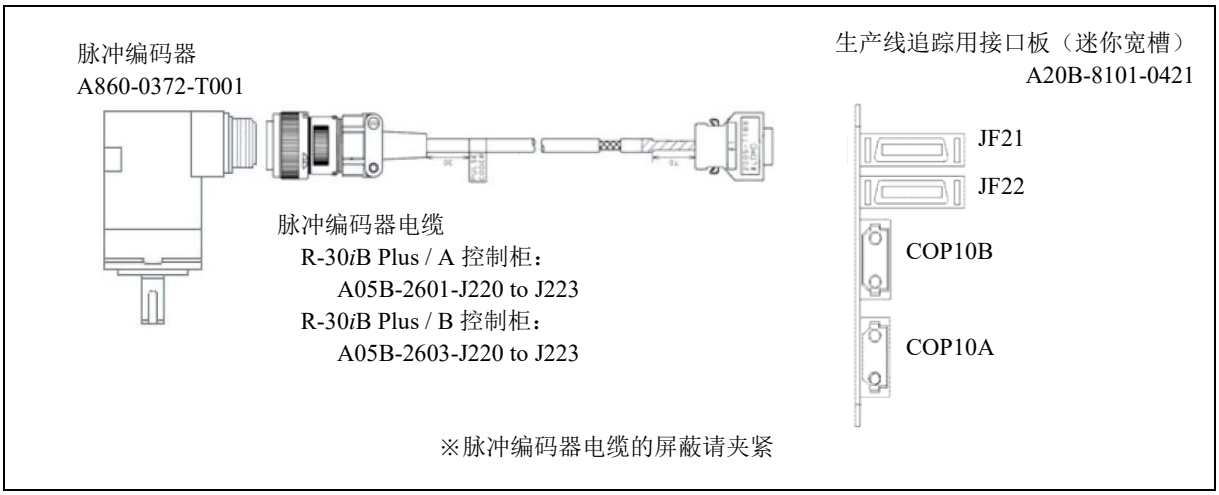


图 D.3.1 (b) 使用生产线追踪用接口板 A05B-2600-J035, A05B-2660-J035 时的电缆连接方法（αA1000S 脉冲编码器 1 个）

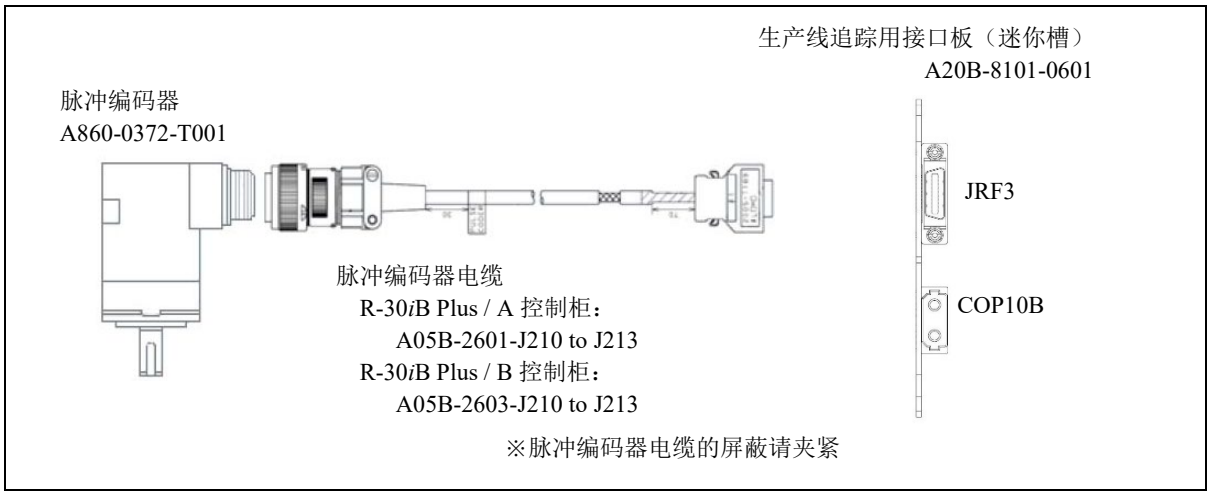


图 D.3.1 (c) 使用生产线追踪用接口板 A05B-2600-J036, A05B-2660-J036, A05B-2600-J037 时的电缆连接方法（αA1000S 脉冲编码器 1 个）

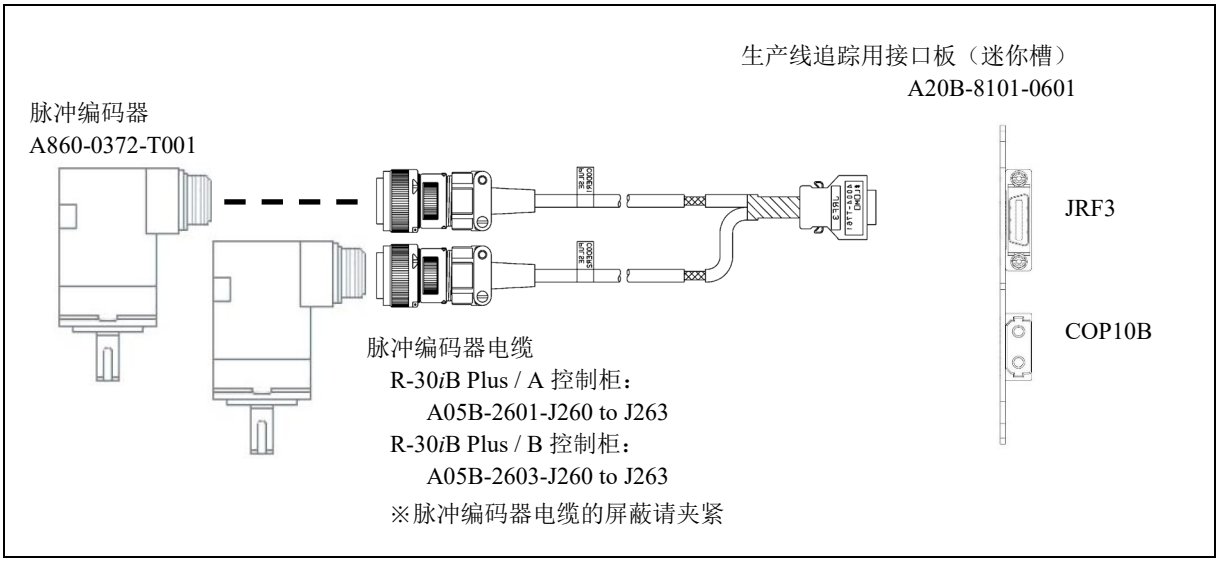


图 D.3.1 (d) 使用生产线追踪用接口板 A05B-2600-J036, A05B-2660-J036, A05B-2600-J037 时的电缆连接方法（αA1000S 脉冲编码器 2 个）

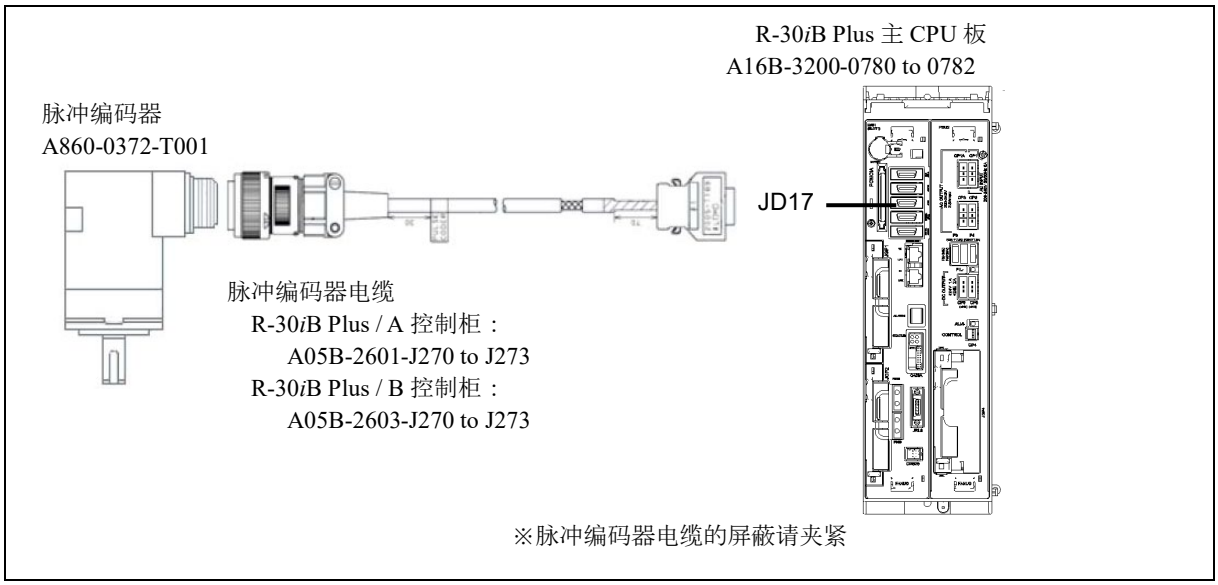


图 D.3.1 (e) 使用 R-30*i*B Plus 主 CPU 板时的电缆连接方法（αA1000S 脉冲编码器 1 个）

D.3.2 R-30*i*B Mate Plus 时

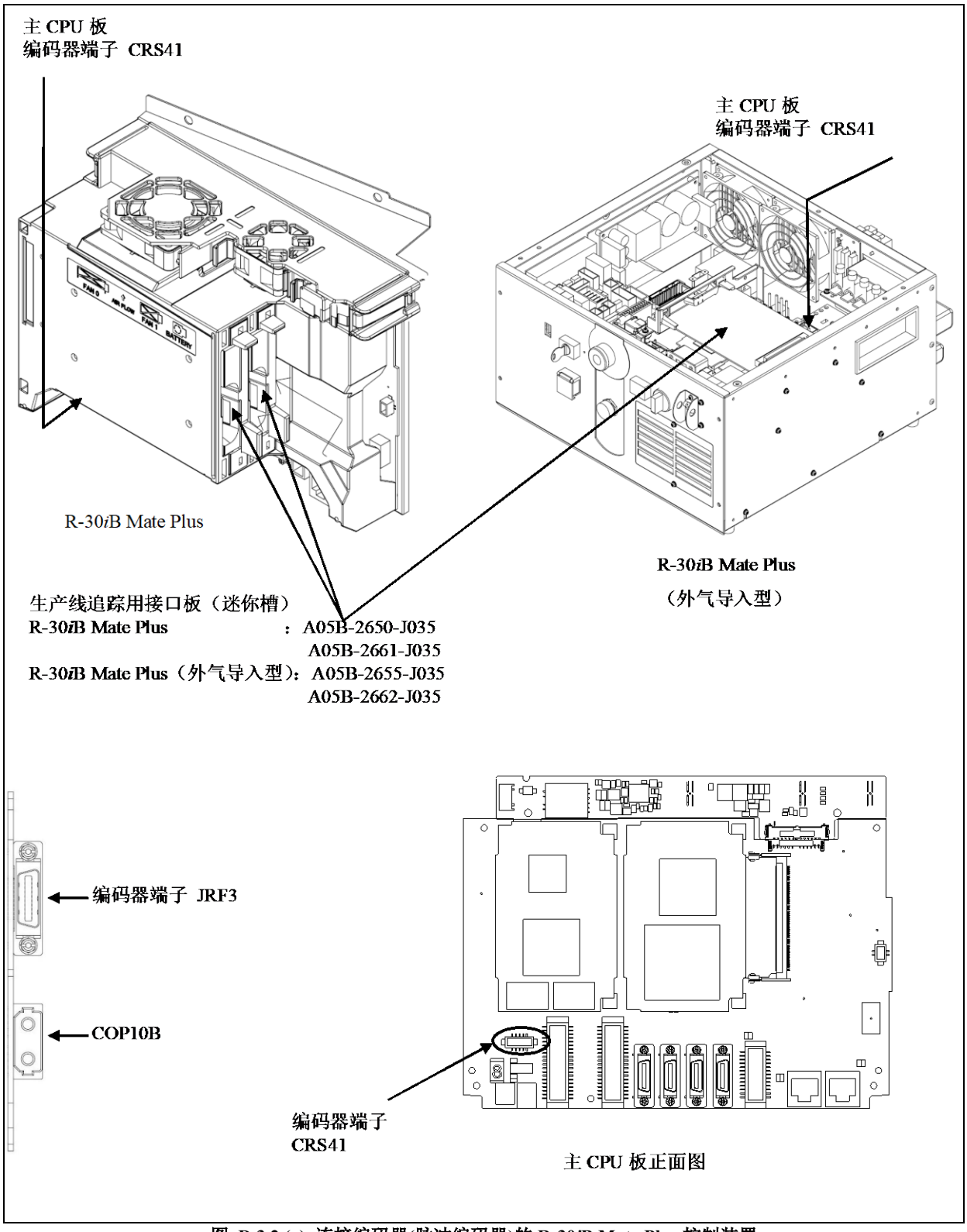


图 D.3.2 (a) 连接编码器(脉冲编码器)的 R-30*i*B Mate Plus 控制装置

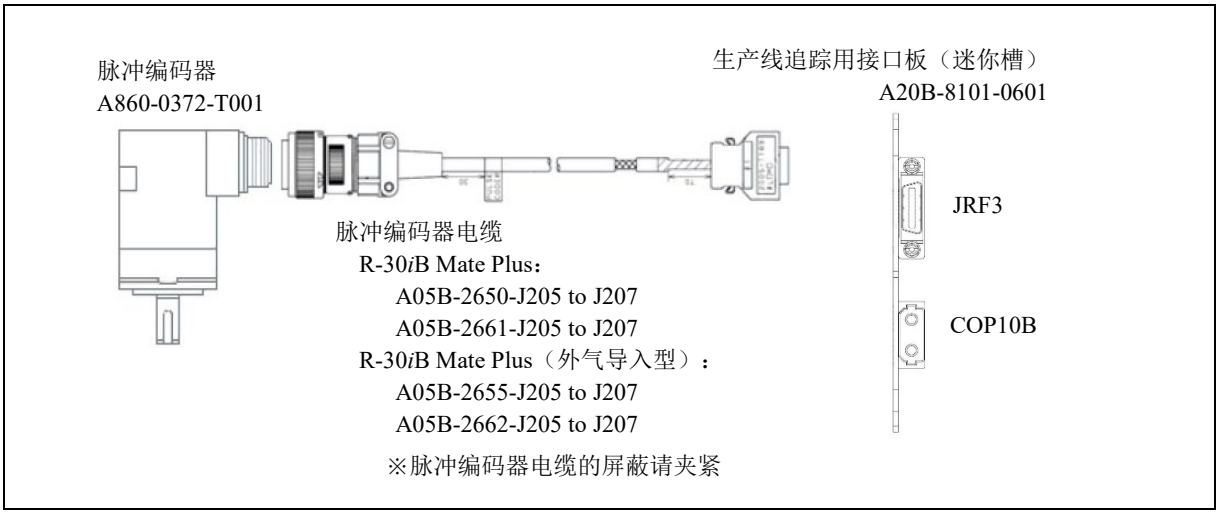


图 D.3.2 (b) 使用生产线追踪用接口板 A05B-2650-J035, A05B-2655-J035, A05B-2661-J035, A05B-2662-J035 时的电缆连接方法（αA1000S 脉冲编码器 1 个）

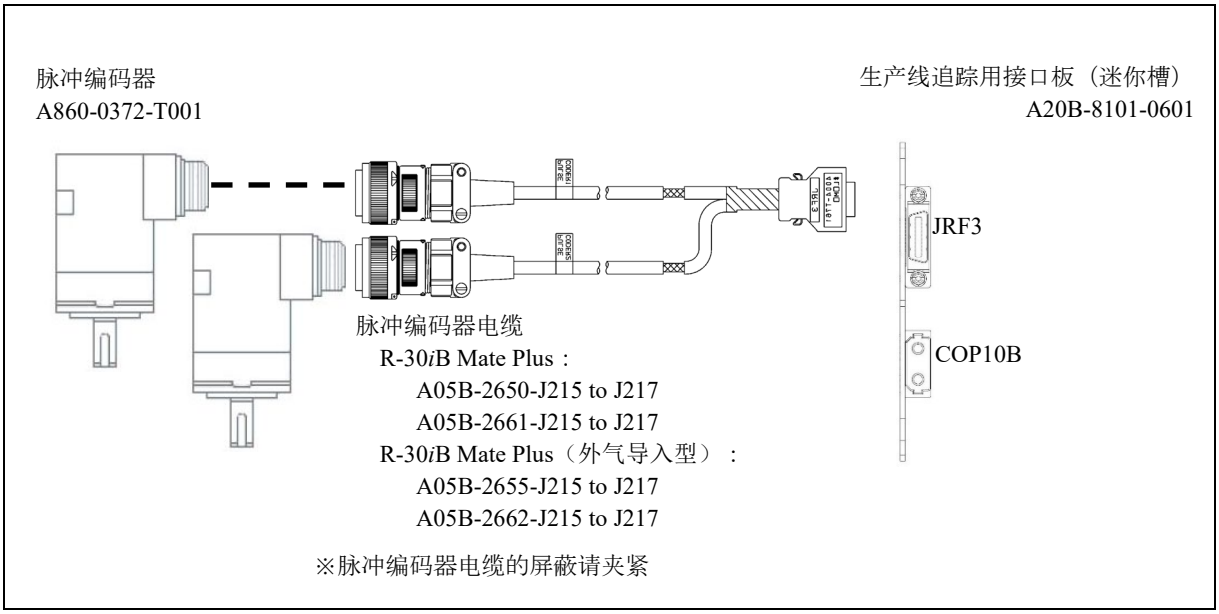


图 D.3.2 (c) 使用生产线追踪用接口板 A05B-2650-J035, A05B-2655-J035, A05B-2661-J035, A05B-2662-J035 时的电缆连接方法（αA1000S 脉冲编码器 1 个）

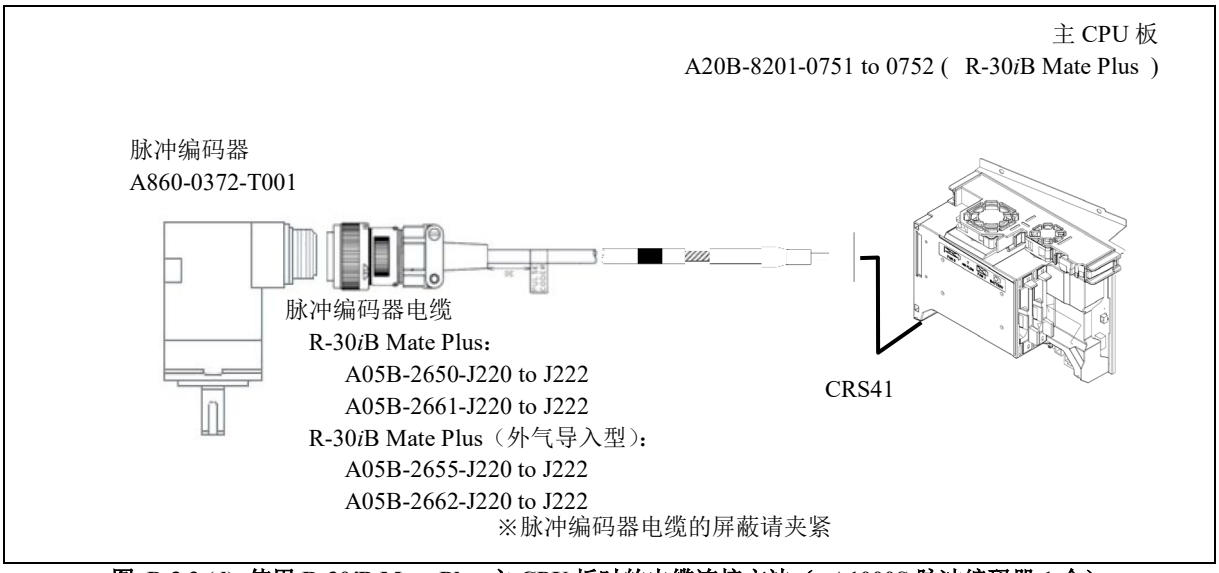


图 D.3.2 (d) 使用 R-30*i*B Mate Plus 主 CPU 板时的电缆连接方法（αA1000S 脉冲编码器 1 个）

D.3.3 R-30*i*B Compact Plus 时

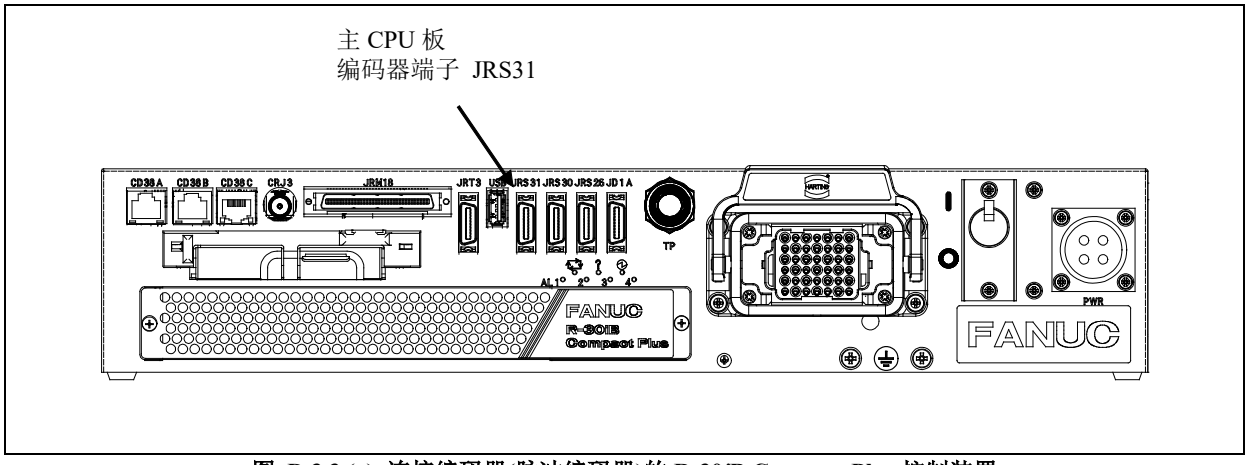


图 D.3.3 (a) 连接编码器(脉冲编码器)的 R-30*i*B Compact Plus 控制装置

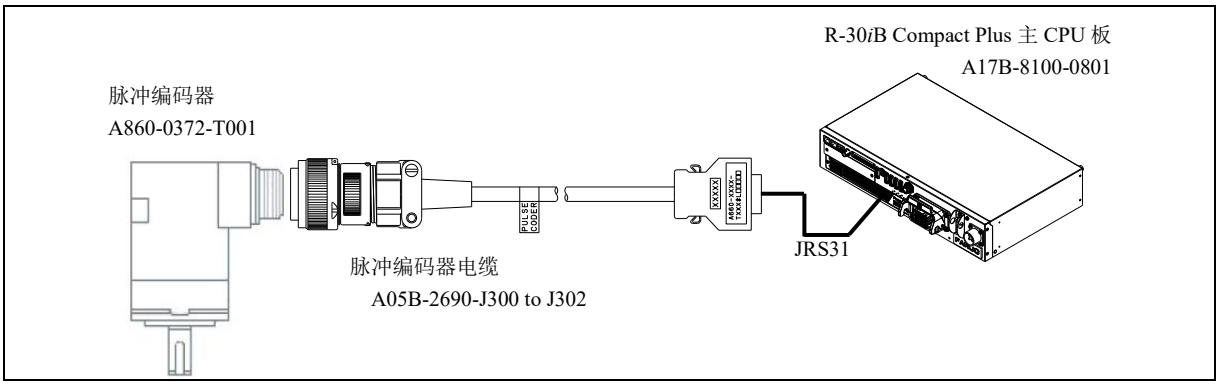


图 D.3.3 (b) 使用 R-30*i*B Compact Plus 主 CPU 板时的电缆连接方法（αA1000S 脉冲编码器 1 个）

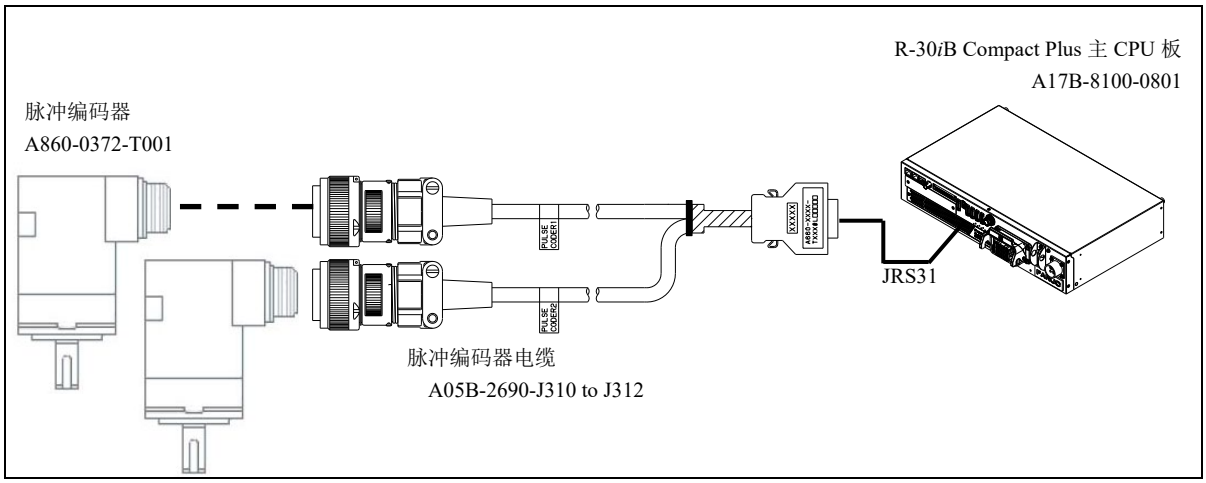


图 D.3.3 (c) 使用 R-30*i*B Compact Plus 主 CPU 板时的电缆连接方法（αA1000S 脉冲编码器 2 个）

D.3.4 R-30*i*B Mini Plus 时

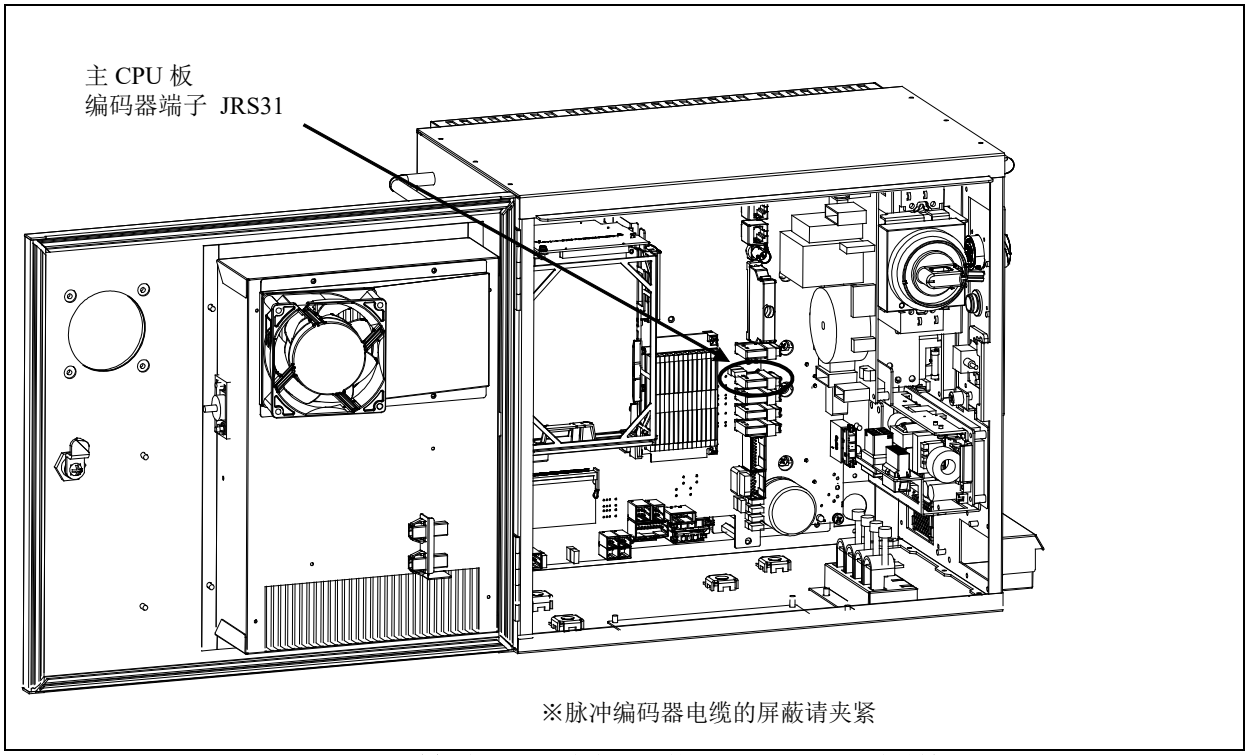


图 D.3.4 (a) 连接编码器(脉冲编码器)的 R-30*i*B Mini Plus 控制装置

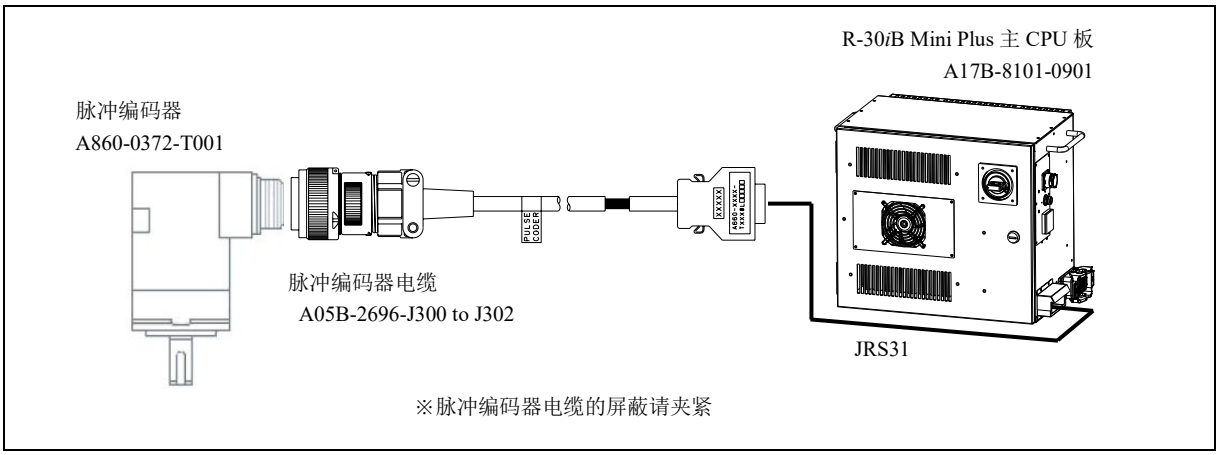


图 D.3.4 (b)R-30*i*B Mini Plus 的电缆连接方法（αA1000S 脉冲编码器 1 个）

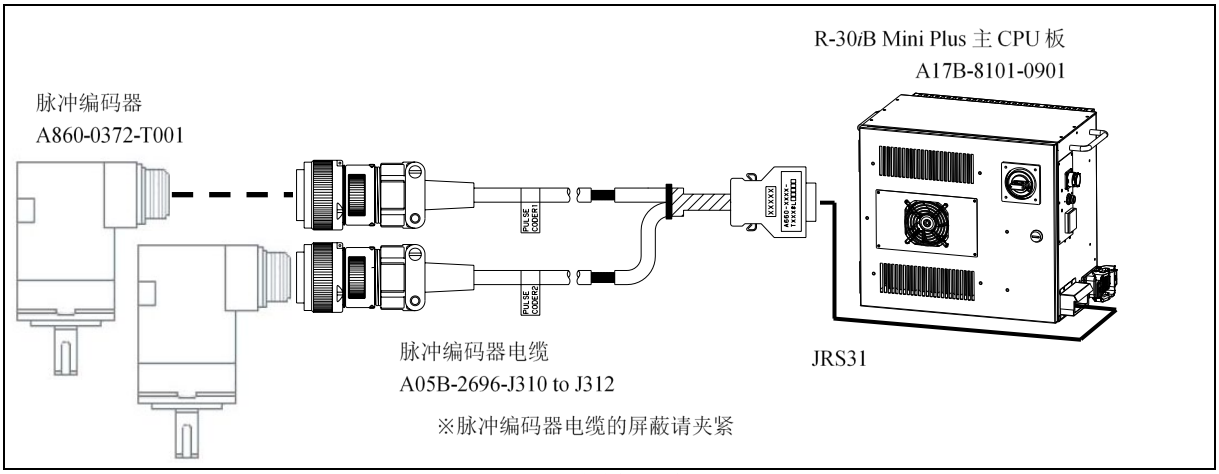


图 D.3.4 (c)R-30*i*B Mini Plus 的电缆连接方法（αA1000S 脉冲编码器 2 个）

E 量增脉冲编码器

本章对量增脉冲编码器（A860-0301-T001～T004）的使用方法进行说明。

> ⚠注意
> **不可将量增脉冲编码器连接到 R-30*i*B Compact Plus / R-30*i*B Mini Plus。**

E.1 必要的结构

表 E.1(a)R-30*i*B Plus: 使用量增脉冲编码器 A860-0301-T001～T004 时的结构

装置	R-30*i*B Plus 控制装置 A-控制柜	R-30*i*B Plus 控制装置 B-控制柜	注释
硬件			
追踪拾取用接口板	A20B-8101-0421（迷你宽槽）或 A20B-8101-0601（迷你槽）	A20B-8101-0421（迷你宽槽）或 A20B-8101-0601（迷你槽）	• 从以下的规格，准备左记印刷电路板。 A05B-2600-J035, A05B-2660-J035（A/B-控制柜共通、迷你宽槽） A05B-2600-J036, A05B-2660-J036（A-控制柜、迷你槽） A05B-2600-J037（B-控制柜、迷你槽） • 另置型接口装置（SDU） A02B-0323-C205 可代替追踪拾取用接口板使用。但是，这种情况下，需要进行用于装置安装的加工。（详细信息，请参照“附录 F 另置型接口装置（SDU）”。）
光纤（FSSB）	A66L-6001-0023	A66L-6001-0023	• 从以下的规格，准备左记光纤。 A05B-2600-J035, A05B-2660-J035（A/B-控制柜共通、迷你宽槽） A05B-2600-J036, A05B-2660-J036（A-控制柜、迷你槽） A05B-2600-J037（B-控制柜、迷你槽）

装置	R-30*i*B Plus 控制装置 A-控制柜	R-30*i*B Plus 控制装置 B-控制柜	注释
硬件			
脉冲编码器 电缆	A20B-8101-0421 连接迷你宽槽 的情况： A05B-2601-J380 （7M） A05B-2601-J381 （14M） A05B-2601-J382 （20M） A05B-2601-J383 （30M） A05B-2660-J380 （7M） A05B-2660-J381 （14M） A05B-2660-J382 （20M） A05B-2660-J383 （30M）	A20B-8101-0421 连接迷你宽槽的情况： A05B-2603-J380 （7M） A05B-2603-J381 （14M） A05B-2603-J382 （20M） A05B-2603-J383 （30M）	• 量增脉冲编码器 A860-0301-T001～004 不可连接到主 CPU 板上的编码器端子。
	A20B-8101-0601 连接迷你槽的情况 （1 个脉冲编码器）： A05B-2601-J370 （7M） A05B-2601-J371 （14M） A05B-2601-J372 （20M） A05B-2601-J373 （30M） A05B-2660-J370 （7M） A05B-2660-J371 （14M） A05B-2660-J372 （20M） A05B-2660-J373 （30M）	A20B-8101-0601 连接迷你槽的情况 （1 个脉冲编码器）： A05B-2603-J370 （7M） A05B-2603-J371 （14M） A05B-2603-J372 （20M） A05B-2603-J373 （30M）	

装置	R-30*i*B Plus 控制装置 A-控制柜	R-30*i*B Plus 控制装置 B-控制柜	注释
硬件			
脉冲分配器用电缆	A20B-8101-0421 连接迷你宽槽的情况： A05B-2601-J385 （7M） A05B-2601-J386 （14M） A05B-2601-J387 （20M） A05B-2601-J388 （30M） A05B-2660-J385 （7M） A05B-2660-J386 （14M） A05B-2660-J387 （20M） A05B-2660-J388 （30M）	A20B-8101-0421 连接迷你宽槽的情况： A05B-2603-J385 （7M） A05B-2603-J386 （14M） A05B-2603-J387 （20M） A05B-2603-J388 （30M）	• 用多个机器人追踪工件时，需要使用脉冲分配器或者是以太网编码器功能(A05B-2600-R762) (选项)。
	A20B-8101-0601 连接迷你槽的情况： A05B-2601-J375 （7M） A05B-2601-J376 （14M） A05B-2601-J377 （20M） A05B-2601-J378 （30M） A05B-2660-J375 （7M） A05B-2660-J376 （14M） A05B-2660-J377 （20M） A05B-2660-J378 （30M）	A20B-8101-0601 连接迷你槽的情况： A05B-2603-J375 （7M） A05B-2603-J376 （14M） A05B-2603-J377 （20M） A05B-2603-J378 （30M）	

表 E.1 (b) 需要条件 (使用 R-30*i*B Mate Plus: 量增脉冲编码器 A860-0301-T001～T004 时)

装置	R-30*i*B Mate Plus 控制装置	R-30*i*B Mate Plus 控制装置 外气导入控制柜	注释
硬件			
生产线追踪用接口板	A20B-8101-0601 (迷你槽)	A20B-8101-0601 (迷你槽)	• 根据以下准备规格准备左边的印刷板。 A05B-2650-J035, A05B-2661-J035 (R-30*i*B Mate Plus、迷你槽) A05B-2655-J035, A05B-2662-J035 (R-30*i*B Mate Plus(外气导入型)、迷你槽)

装置	R-30*i*B Mate Plus 控制装置	R-30*i*B Mate Plus 控制装置 外气导入控制柜	注释
硬件			
光纤(FSSB)	A66L-6001-0026	A66L-6001-0023	• 根据以下准备规格准备左边的光纤。 A05B-2650-J035, A05B-2661-J035 (R-30*i*B Mate Plus、迷你槽) A05B-2655-J035, A05B-2662-J035 R-30*i*B Mate Plus(外气导入型)、迷你槽)
脉冲编码器电缆	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 1 个)： A05B-2650-J200 (7M) A05B-2650-J201 (14M) A05B-2650-J202 (20M) A05B-2661-J200 (7M) A05B-2661-J201 (14M) A05B-2661-J202 (20M)	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 1 个)： A05B-2655-J200 (7M) A05B-2655-J201 (14M) A05B-2655-J202 (20M) A05B-2662-J200 (7M) A05B-2662-J201 (14M) A05B-2662-J202 (20M)	• 量增脉冲编码器 A860-0301-T001～004 不可连接到主 CPU 板上的编码器端子。
	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 2 个)： A05B-2650-J210 (7M) A05B-2650-J211 (14M) A05B-2650-J212 (20M) A05B-2661-J210 (7M) A05B-2661-J211 (14M) A05B-2661-J212 (20M)	A20B-8101-0601 连接到(迷你槽)时(脉冲编码器 2 个)： A05B-2655-J210 (7M) A05B-2655-J211 (14M) A05B-2655-J212 (20M) A05B-2662-J210 (7M) A05B-2662-J211 (14M) A05B-2662-J212 (20M)	
脉冲分配器用电缆	A20B-8101-0601 连接到(迷你槽)时(脉冲分配器 1 个)： A05B-2650-J225 (7M) A05B-2650-J226 (14M) A05B-2650-J227 (20M) A05B-2650-J228 (30M)	-	• 用多个机器人追踪工件时，需要使用脉冲分配器或者是以太网编码器功能 (A05B-2600-R762) (选项)。
	A20B-8101-0601 连接到(迷你槽)时(脉冲分配器 2 个)： A05B-2650-J230 (7M) A05B-2650-J231 (14M) A05B-2650-J232 (20M) A05B-2650-J233 (30M)	-	

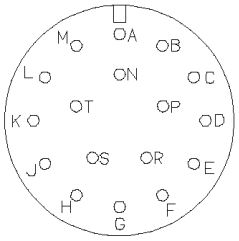
表 E.1 (c) 需要条件 (R-30*i*B Plus / R-30*i*B Mate Plus: 使用脉冲分配器 A16B-1212-0290 时)

装置	R-30*i*B Plus / R-30*i*B Mate Plus 控制装置	注释
硬件		
脉冲分配器	A16B-1212-0290	• 用多个机器人追踪工件时，除了使用本脉冲分配器，还可以使用以太网编码器功能 (A05B-2600-R762) (选项)。
脉冲分配器-脉冲编码器用电缆	A05B-2451-K102 (7M) A05B-2451-K103 (14M)	
脉冲分配器-脉冲分配器 连接电缆	A05B-2600-K155(7M) A05B-2600-K156(14M) A05B-2660-K155(7M) A05B-2660-K156(14M)	• 串联 3 台以上的脉冲分配器时，需要另行商讨。
脉冲分配器用 5V 连接器	A02B-0061–K203	• 准备 1 个外壳和 6 个触点（锡焊）。
要使用脉冲分配器，需要 5V 电源（额定输出 10W 以上）。		

有关脉冲编码器的信号信息、规格、尺寸，请参照图 E.2(a)、表 E.2(a)、图 E.2 (b)。关于脉冲分配器的使用环境条件、尺寸，请参照表 E.2 (b)、图 E.2 (c)。
与连接及安装相关的信息，请参照图 E.3.1 (a) ～ 图 E.3.1 (d) 以及图 E.3.2 (a) ～ 图 E.3.2 (c) 。

E.2 概略图

A860-0301-T001



3102A20-29P

信号名称	定位销名称 310A20-29P
A *A	A D
B *B	B E
Z *Z	F G
C1 C2 C4 C8	- - - -
+5V 0V	C,J,K N,P,T
Shield	H
OH1 OH2	
REQ	
+6VA 0VA	

图 E.2(a) 量增脉冲编码器 （A860-0301-T001） 的连接用信号信息

表 E.2(a) 量增脉冲编码器 （A860-0301-T001） 规格

项目		规格
电压		5（V）5%
电流		最大 0.35（A）
运转温度		0～60 （℃）
最大应答频率		100 x 10^3（Hz）
输入轴惯性		最大 5×10^{-3}（kg • m^2）
输入轴启动扭矩		最大 0.8（Nm）
额定负载	半径方向	20（N）
	轴方向	10（N）
轴直径摆动		0.02×10^{-3} （m）
重量		约 2.0（kg）

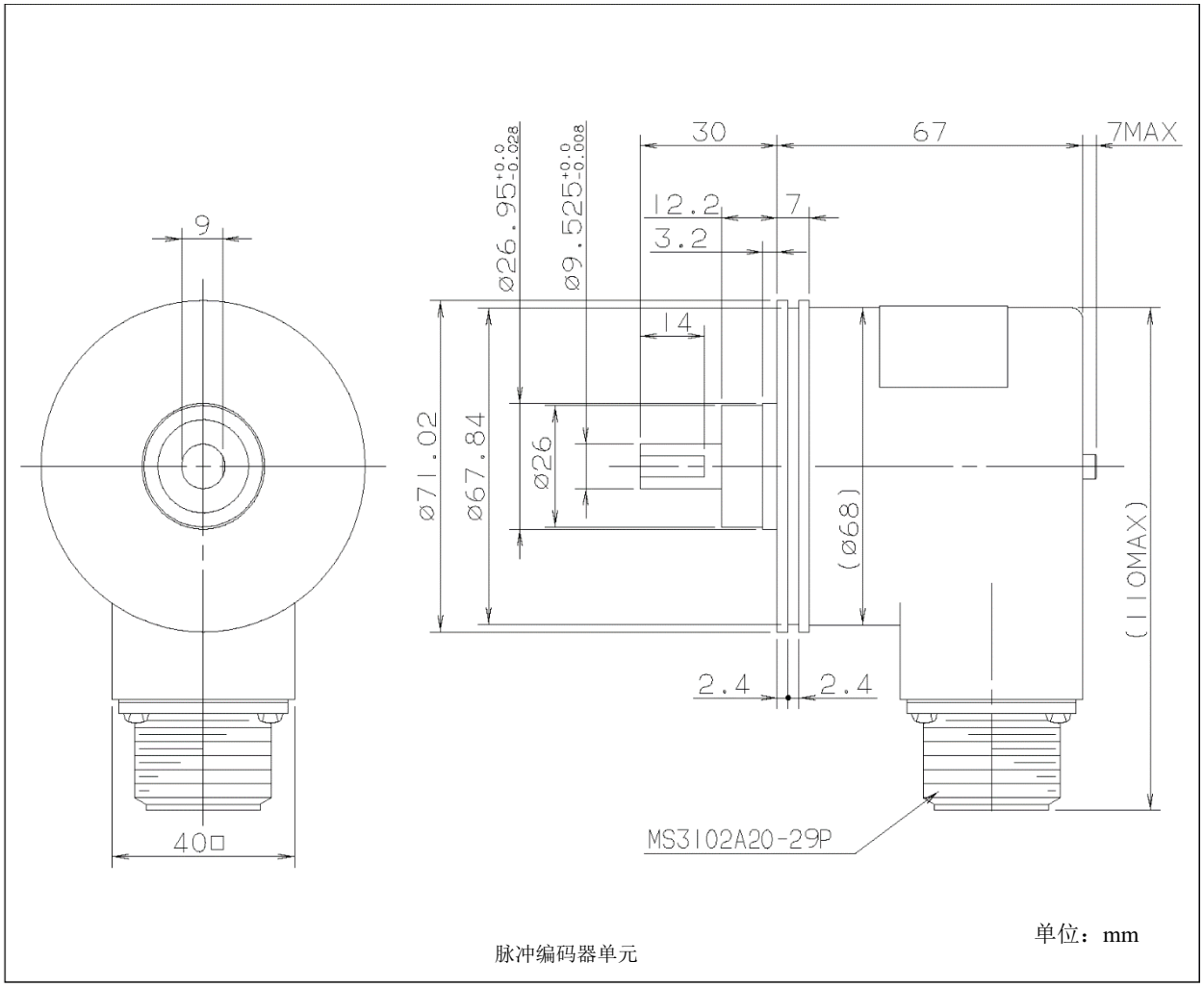


图 E.2(b) 脉冲编码器的尺寸（A860-0301-T001）

表 E.2 (b) 脉冲分配器的使用环境条件(A16B-1212-0290)

项目	条件
容许环境温度	动作时 0～55℃ 运输、储藏时 -20～+60℃
容许环境湿度	95%RH 以下 无结露
振动加速度	4.9m/s^2 (0.5G) 以下
环境	存放在保持通风清洁的控制柜中，以避免暴露在存在导电性杂质、切削液、有机溶剂等的环境中。

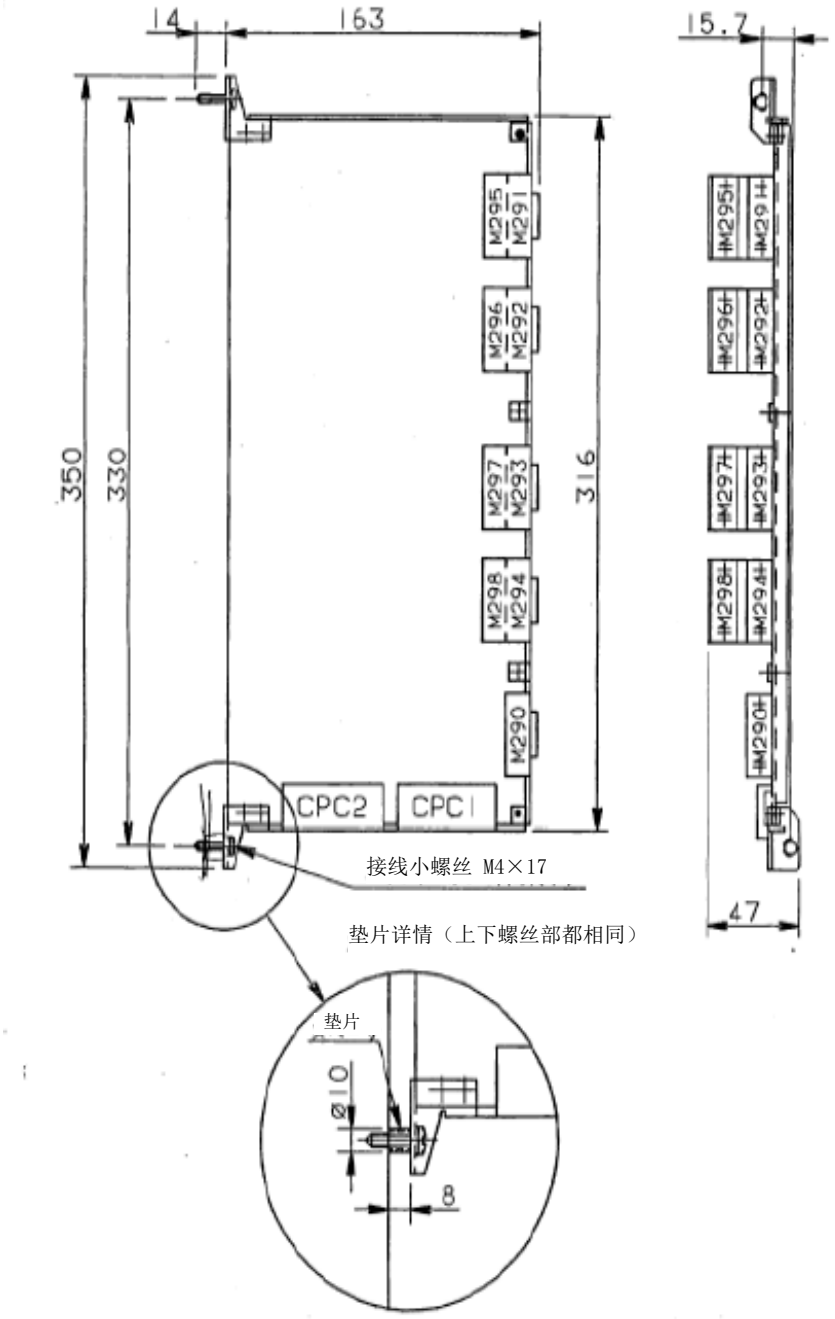


图 E.2 (c) 脉冲分配器的尺寸 (A16B-1212-0290)

E.3 连接方法

E.3.1 与机器人控制装置直接连接

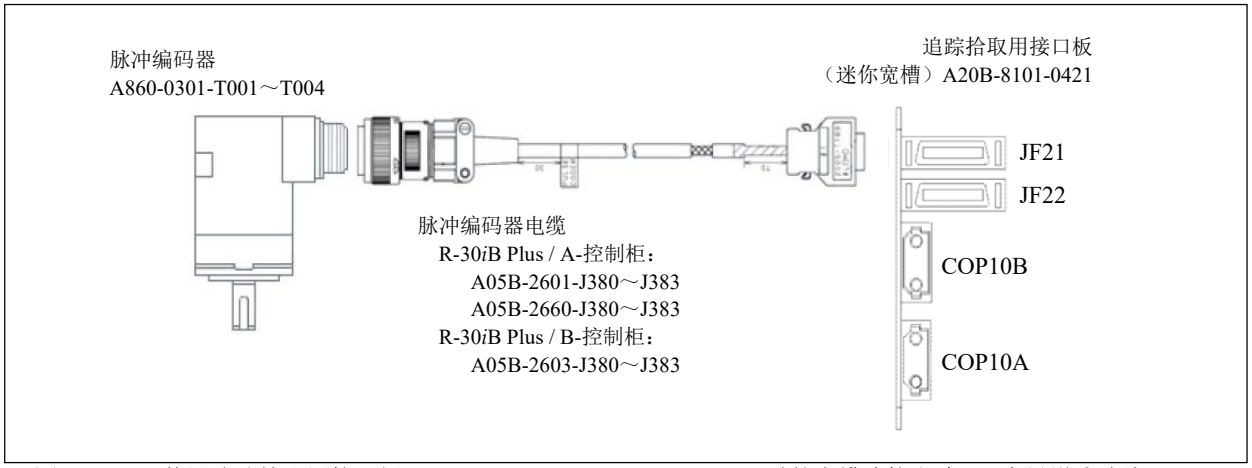


图 E.3.1(a) 使用追踪拾取用接口板 A05B-2600-J035, A05B-2660-J035 时的电缆连接方法（1 个量增脉冲编码器）

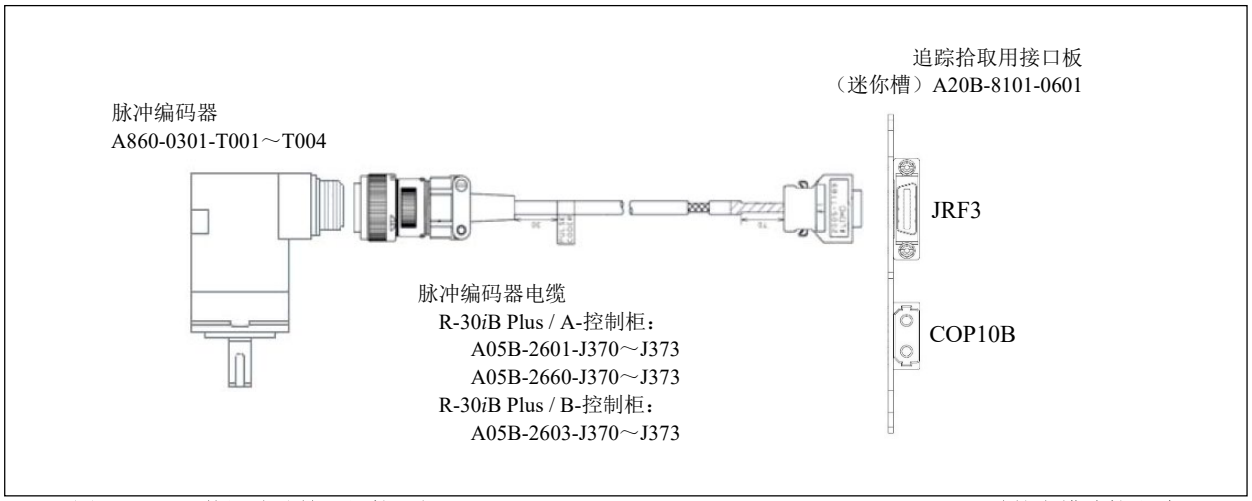


图 E.3.1(b) 使用追踪拾取用接口板 A05B-2600-J036, A05B-2660-J036, A05B-2600-J037 时的电缆连接方法（1 个量增脉冲编码器）

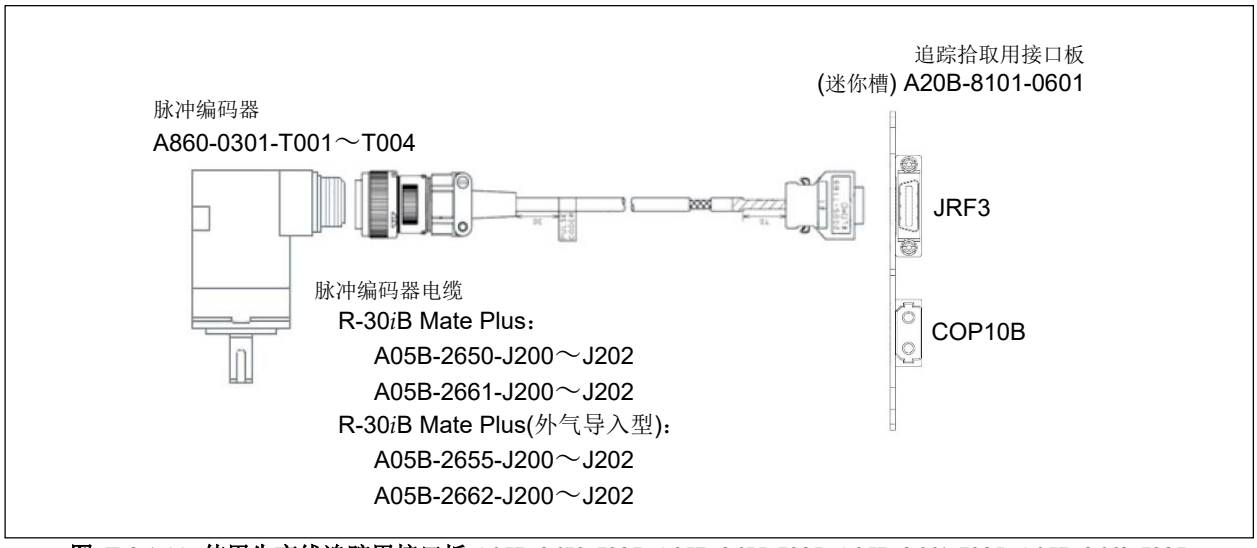


图 E.3.1 (c) 使用生产线追踪用接口板 A05B-2650-J035, A05B-2655-J035, A05B-2661-J035, A05B-2662-J035 时的电缆连接方法（增量脉冲编码器 1 个）

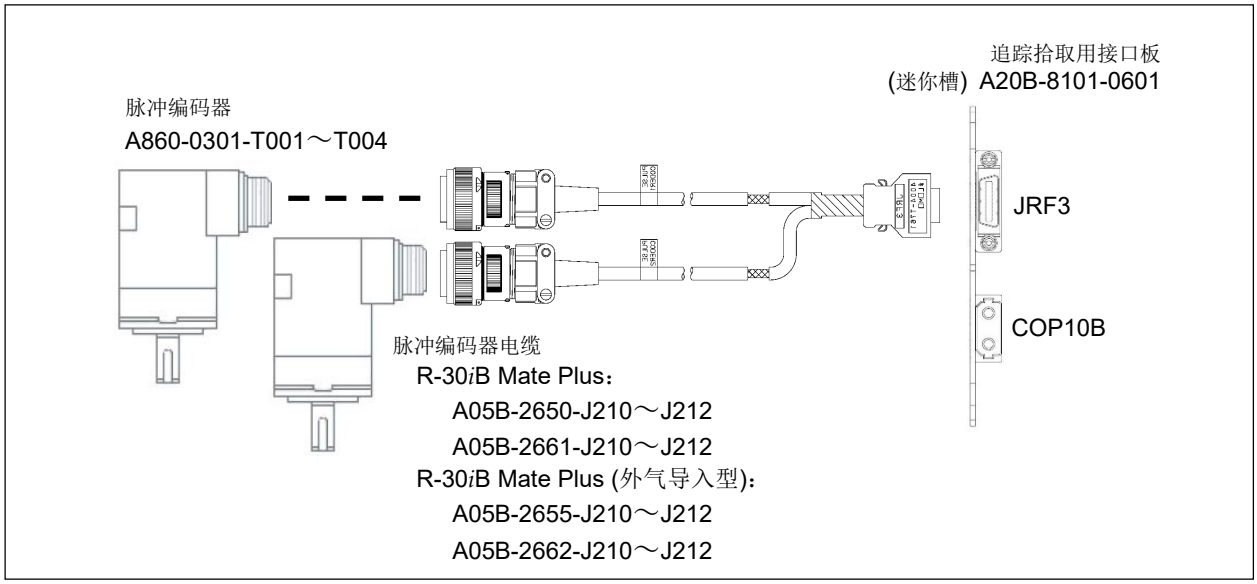


图 E.3.1 (d) 使用生产线追踪用接口板 A05B-2650-J035, A05B-2655-J035, A05B-2661-J035, A05B-2662-J035 时的电缆连接方法（量增脉冲编码器 2 个）

E.3.2 与脉冲分配器连接

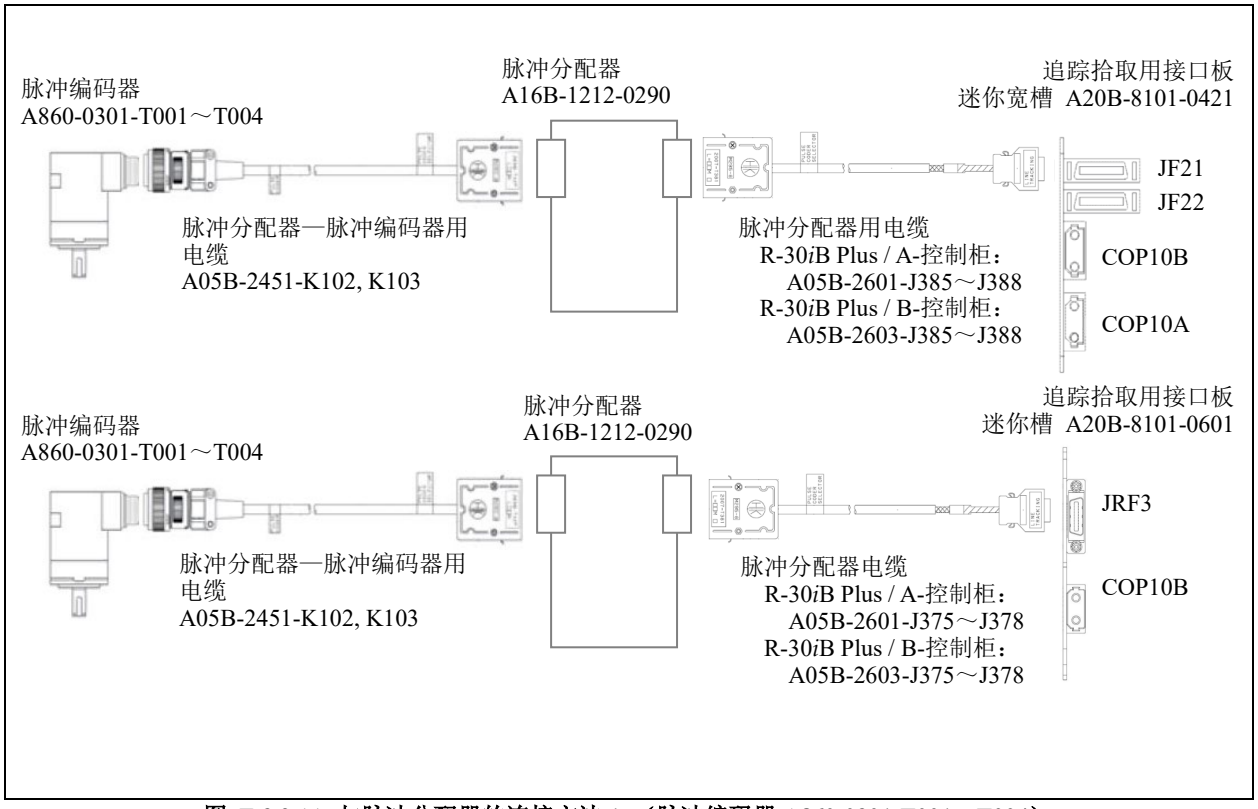


图 E.3.2 (a) 与脉冲分配器的连接方法 1 （脉冲编码器 A860-0301-T001～T004）

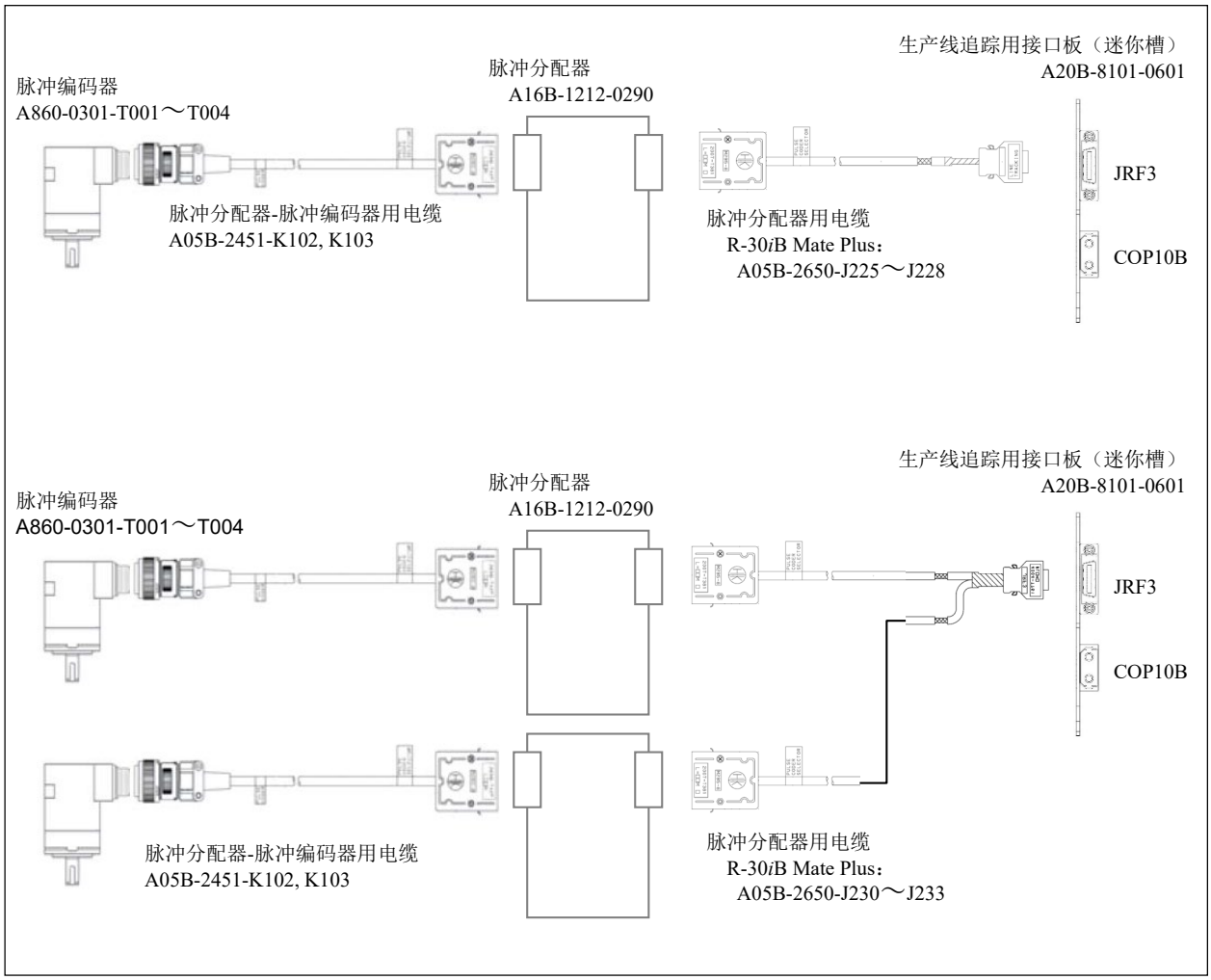


图 E.3.2 (b) 脉冲分配器和连接方法 2(脉冲编码器 A860-0301-T001～T004)

注释
无法使用追踪拾取用接口板时，请使用附录 F 所示的另置型接口装置（SDU）。

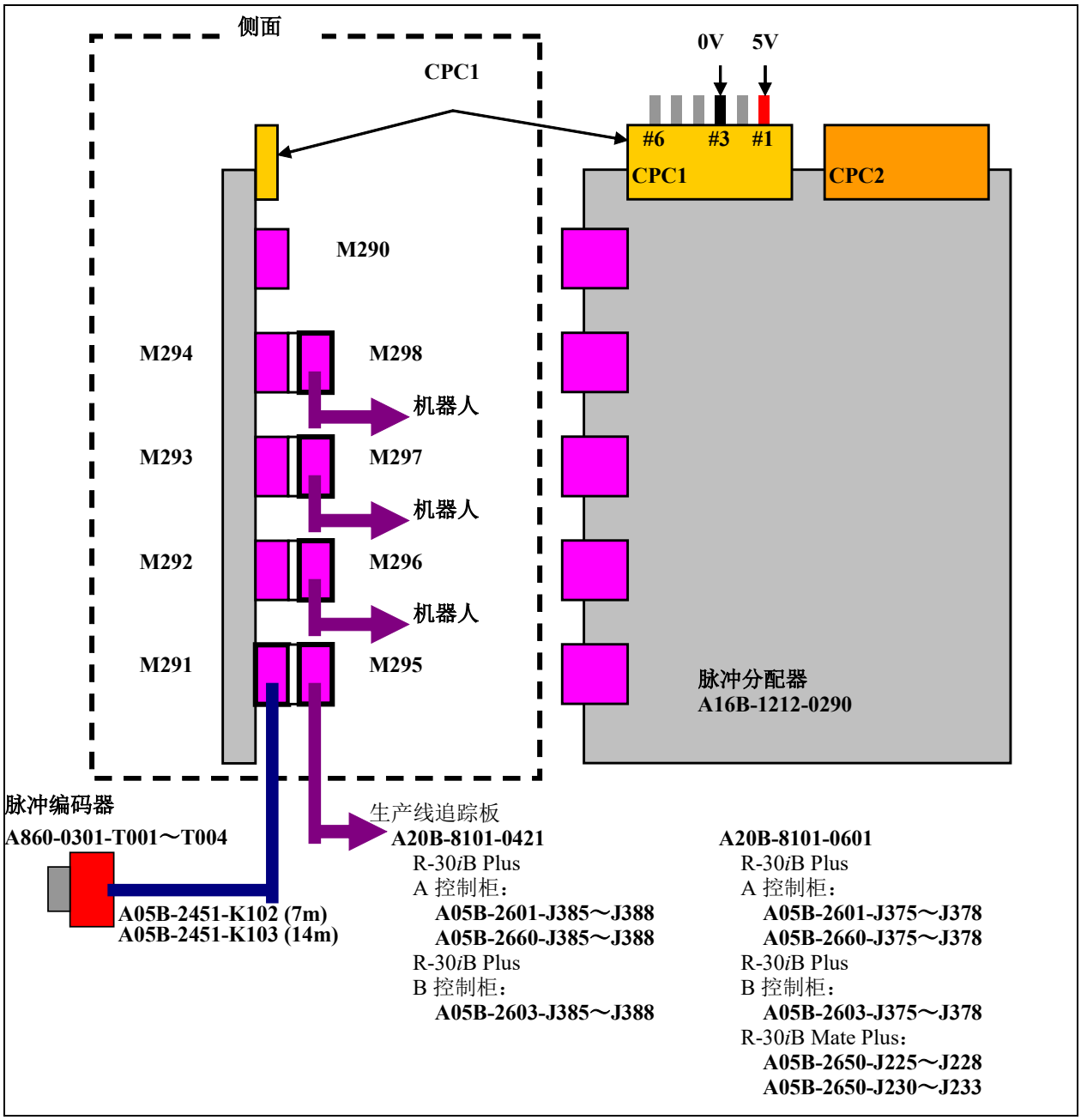


图 E.3.2 (c) 与脉冲分配器的连接方法 3 （脉冲编码器 A860-0301-T001～T004）

注释
1 个脉冲分配器最多可连接 4 根电缆。

F 另置型接口装置（SDU）

由于需在 1 台控制装置上连接 4 个以上的脉冲编码器等原因，无法连接追踪拾取用接口板的情况，请使用另置型接口装置（SDU）。但是，这种情况下，需要进行用于装置安装的加工。请参照图 F(a)～图 F (i)，进行适当加工后使用。

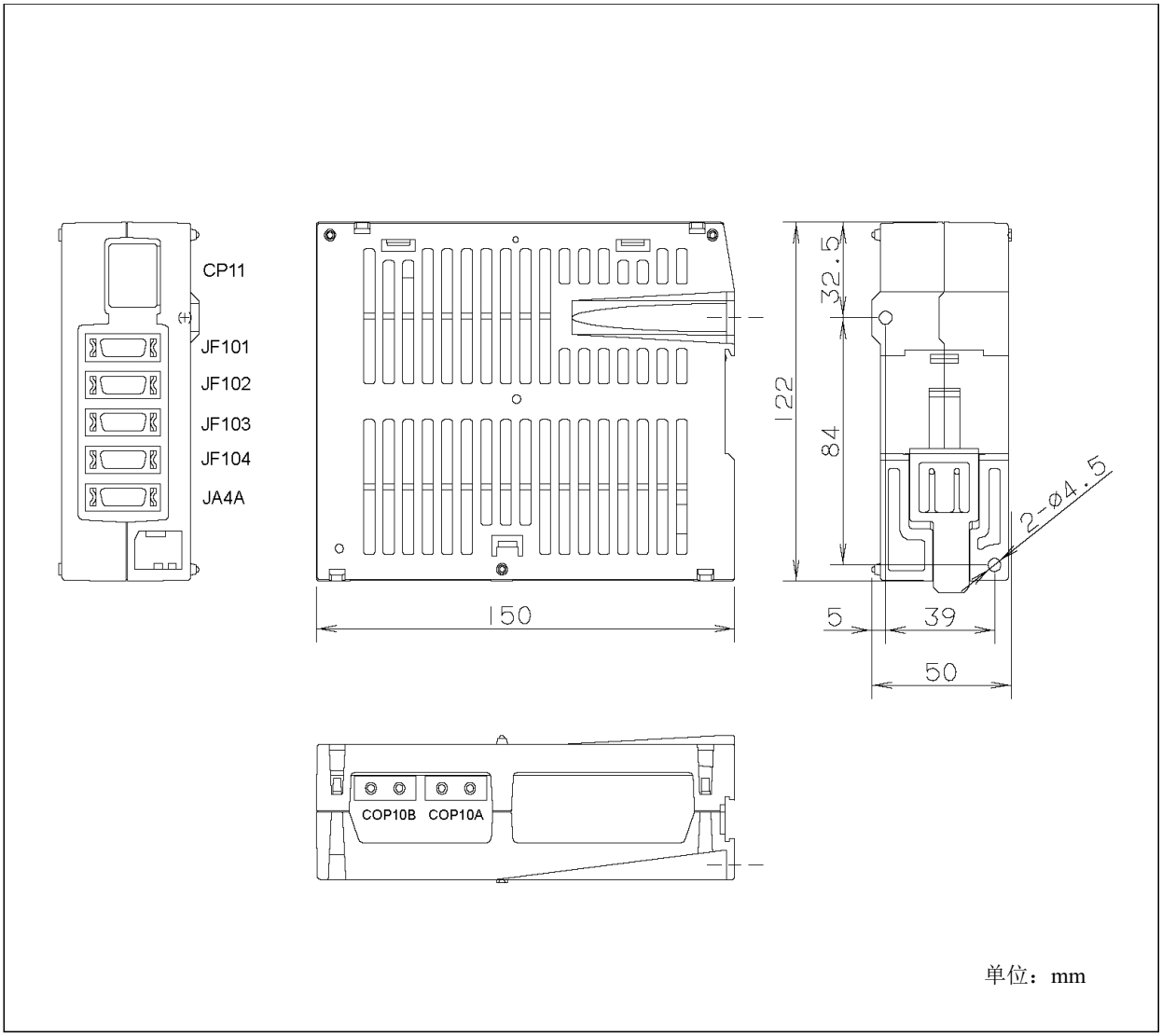


图 F(a) 另置型接口装置的尺寸

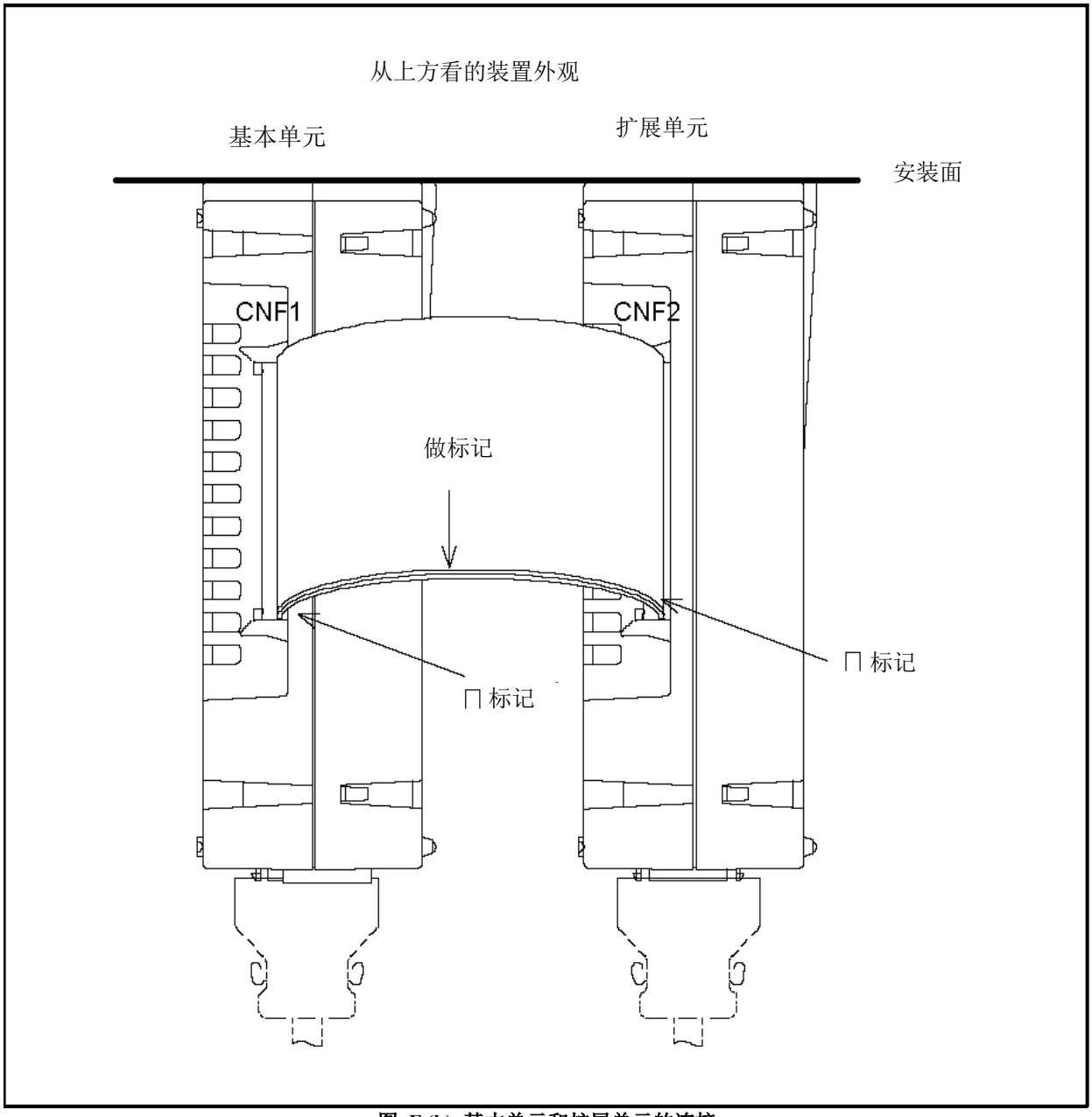


图 F (b) 基本单元和扩展单元的连接

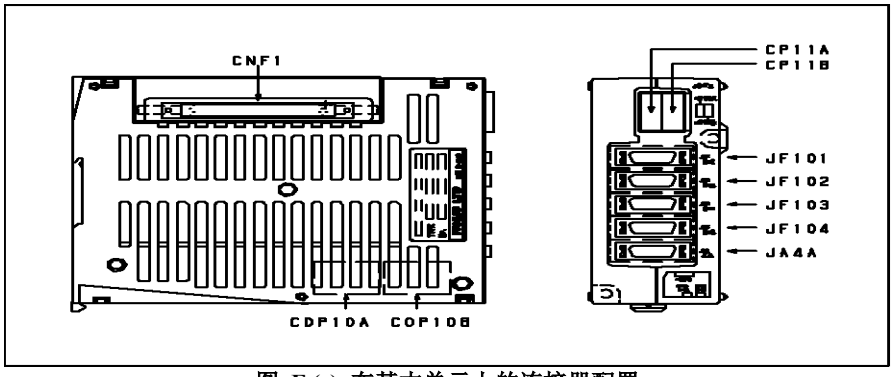


图 F (c) 在基本单元上的连接器配置

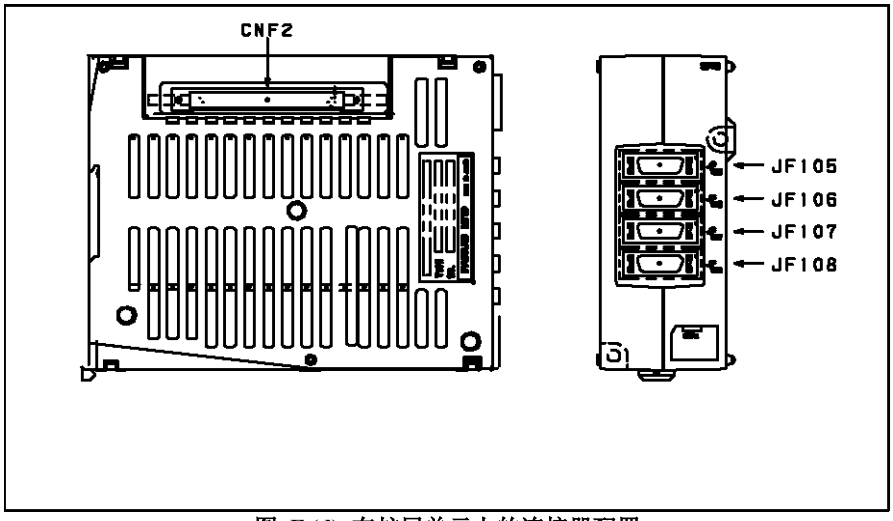


图 F (d) 在扩展单元上的连接器配置

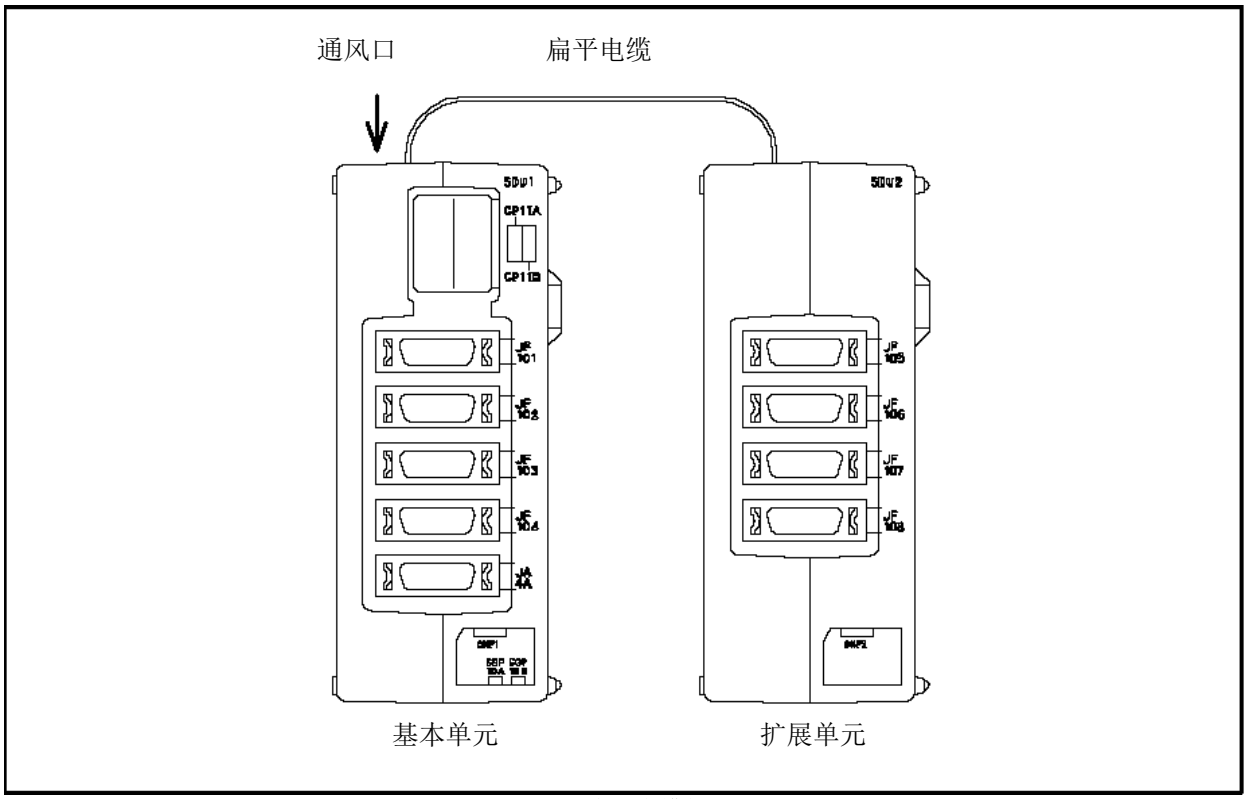


图 F (e) 扁平电缆的配置

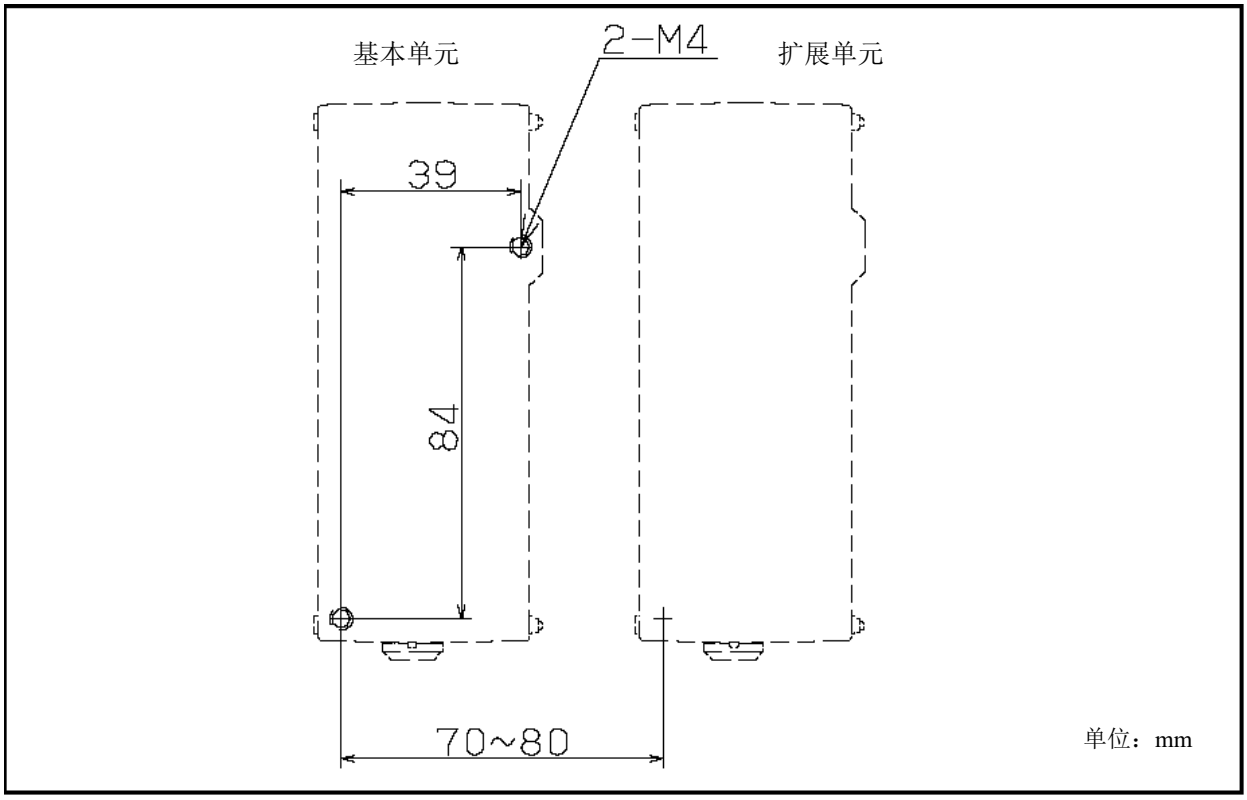


图 F (f) 安装孔的配置

⚠注意

安装或拆卸装置时，需要斜向插入螺丝刀。 因此，请在装置两侧预留足够间隙。
作为一般的指南，相邻单元的前表面面板（有 JF105 等连接器的面）的面的高度基本相同时，请与相邻单元之间留 20mm 以上的空隙。
如果相邻的单元较高，相邻单元的前表面面板的位置向前突起，则与相邻单元之间留 70mm 以上的空隙。在机器人控制装置的控制柜壁附近安装时，也请使与壁面之间留 70mm 以上的空隙。

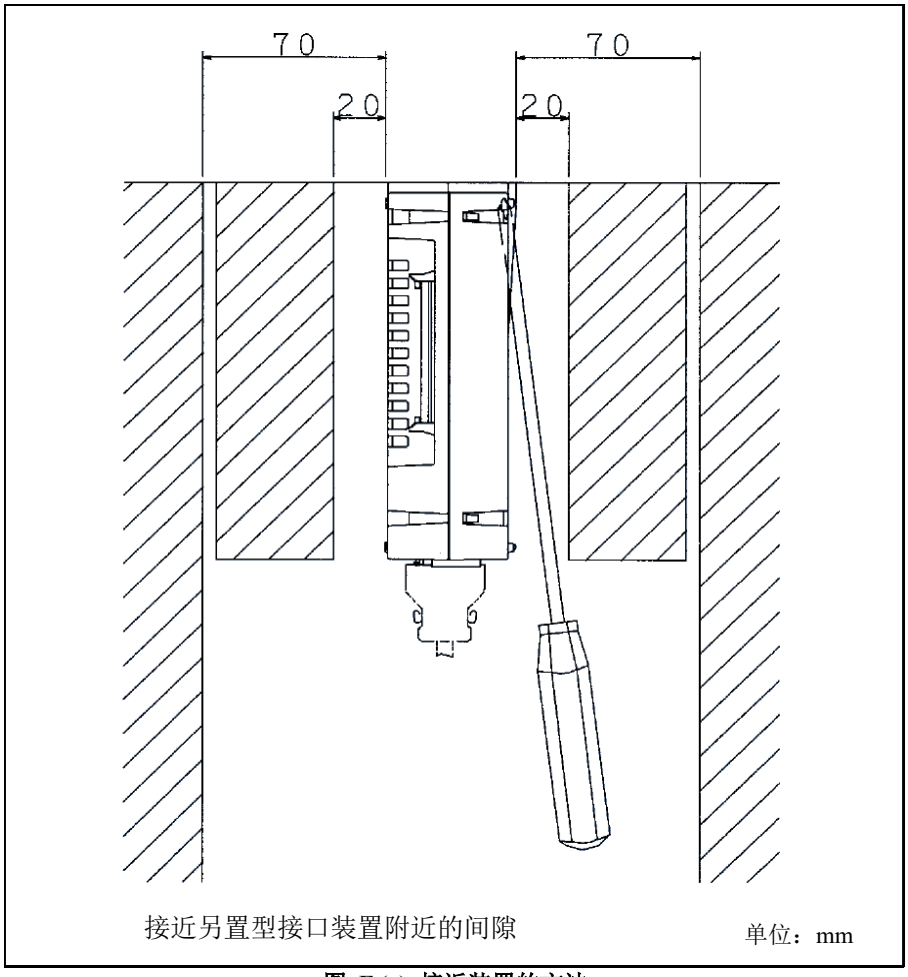


图 F (g) 接近装置的方法

⚠注意
拆卸装置时，可能会令指甲破损，请注意不要过度用力。安装或卸下装置时，请不要对抓握装置上端和下段的装置侧面（有狭缝的面）施加力量。

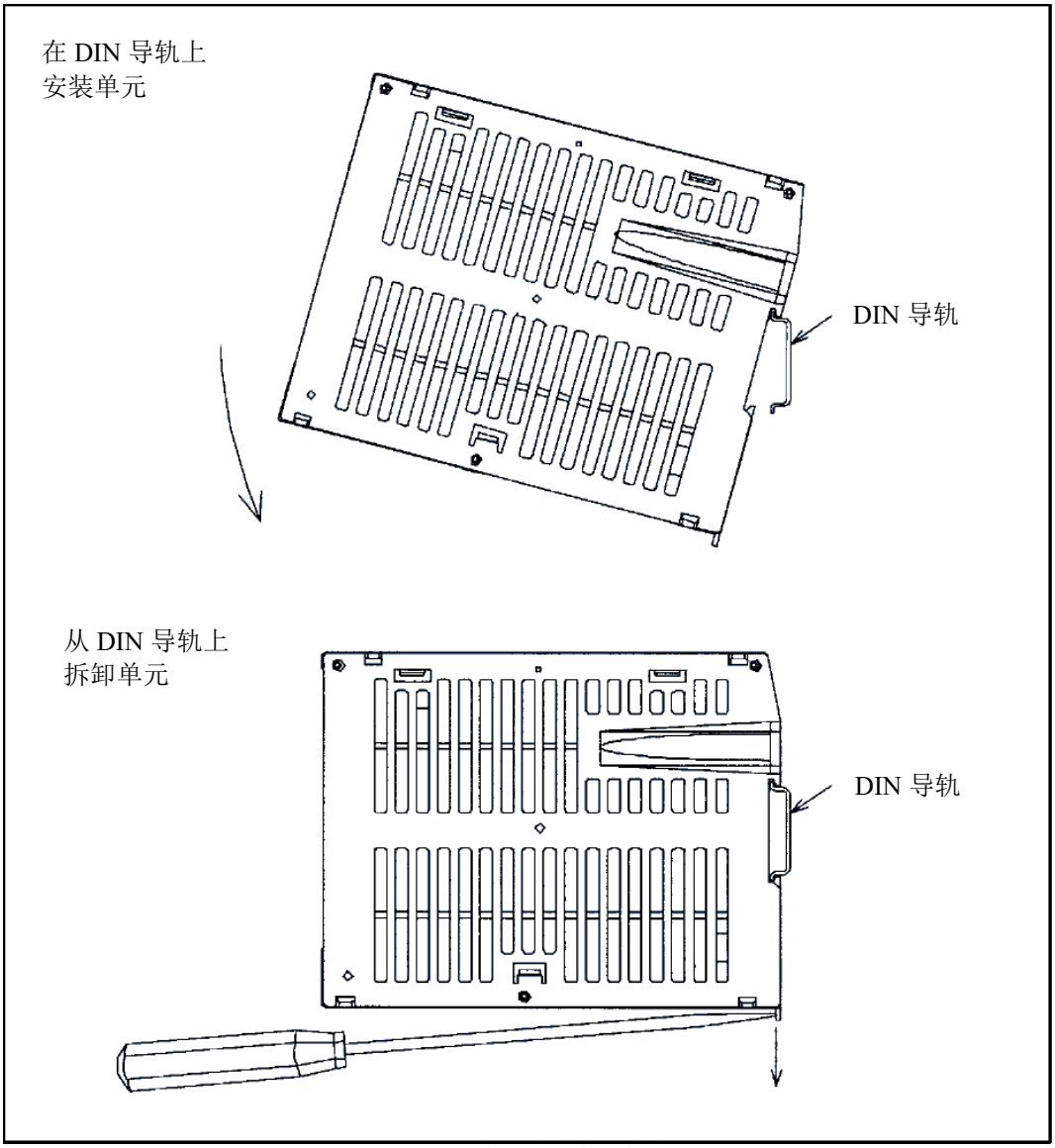


图 F (h) 装置的安装和拆卸

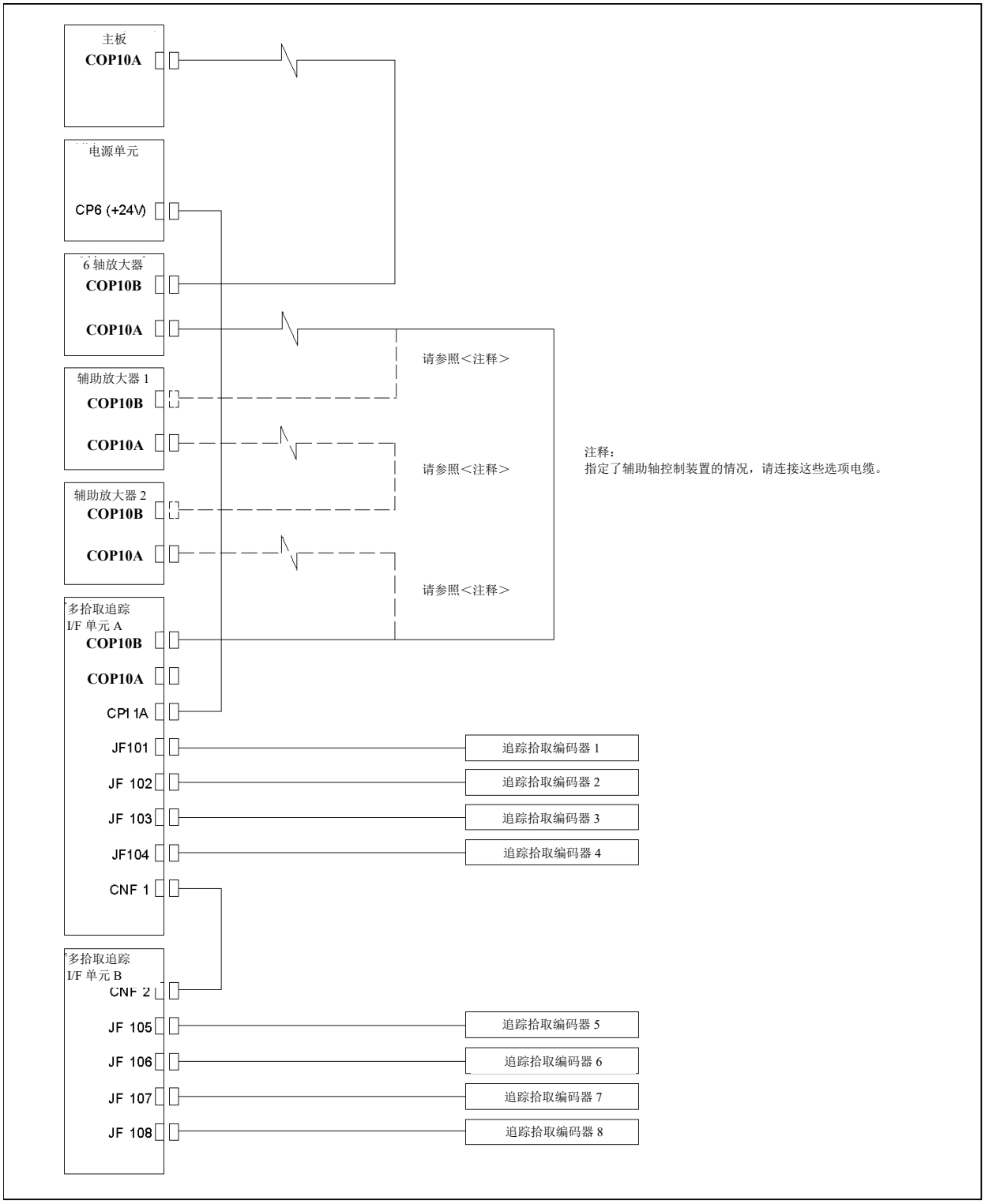

主板
COP10A
电源单元
CP6 (+24V)
6 轴放大器
COP10B
COP10A
辅助放大器 1
COP10B
COP10A
辅助放大器 2
COP10B
COP10A
多拾取追踪
I/F 单元 A
COP10B
COP10A
CP11A
JF101
JF 102
JF 103
JF104
CNF 1
多拾取追踪
I/F 单元 B
CNF 2
JF 105
JF 106
JF 107
JF 108
请参照<注释>
请参照<注释>
请参照<注释>
注释：
指定了辅助轴控制装置的情况，请连接这些选项电缆。
追踪拾取编码器 1
追踪拾取编码器 2
追踪拾取编码器 3
追踪拾取编码器 4
追踪拾取编码器 5
追踪拾取编码器 6
追踪拾取编码器 7
追踪拾取编码器 8

图 F (i) 连接图

G 旧软件与 *i*RPickTool 的 KAREL 对照表

软件版本 7DC1 系列以前的旧视觉追踪软件与 *i*RPickTool 的 KAREL 程序的对照表。

No.	旧KAREL	内容	参数	新KAREL	参数	备注
1	VSTKCLRQ.PC	清除队列。	1：工作区域名	PKCSCLRQUE.PC	1：传送带工作站ID	通过PKCSGETID取得ID。
2	VSTKGETQ.PC	从队列取得工件。	1：工作区域名 2：用于存储工件信息的视觉寄存器序号 3：待机时间毫秒 4：状态用寄存器序号 5：连续取得的标记 6(可省略)：模型ID	PKCSGETQUE.PC	1：传送带工作站ID 2：连续取得的标记 3：待机时间毫秒 4：用于存储工件信息的视觉寄存器序号 5：状态用寄存器序号 6(可省略)：模型ID	通过PKCSGETID取得ID。 参数的顺序已变更。
3	VSTKACKQ.PC	返回从队列取得工件的处理结果。	1：工作区域名 2：用GETQ指定的视觉寄存器序号 3：处理的结果	PKCSACKQUE.PC	1：传送带工作站ID 2：处理的结果	不用参数指定视觉寄存器序号。
4	VSTKSTRT.PC	启动传感器。	无	PKSNSTART.PC	无	
5	VSTKSTOP.PC	停止传感器。	无	PKSNSTOP.PC	无	
6	VSTKSTLB.PC	变更负载平衡。	1：Line名 2：用于指定负载平衡的寄存器序号 3：工作区域数 4：“跳过”值 5(可省略)：种类编号	PKCVSETLBD.PC	1：传送带名 2：用于指定负载平衡的寄存器序号 3：传送带工作站数 4：“跳过”值 5(可省略)：种类编号	
7	VSTKENLB.PC	切换负载平衡的有效无效。	1：Line名 2：有效无效	PKCVENBLB.PC	1：传送带名 2：有效无效	
8	VSTKENSC.PC	切换停止/启动传送带的有效无效。	1：Line名 2：用于指定有效无效的寄存器序号 3：工作区域数	PKCVENBSC.PC	1：传送带名 2：用于指定有效无效的寄存器序号 3：传送带工作站数	
9	VSTKSTVN.PC	变更执行的视觉程序。	1：传感器编号 2：视觉程序名	PKSNSETVPNAM.PC	1：传感器名 2：视觉程序名	
10	VSTKSTTN.PC	变更托盘名。	1：传感器编号 2：托盘名	PKCVSETTRNAM.PC	1：传送带名 2：托盘名	
11	VSTKGETT.PC	取得到达上游边界所需的预估时间。	1：工作区域名 2：用于存储预估时间的寄存器序号 3：连续取得标记 4：(可省略)：模型ID	PKCSGETTIME.PC	1：传送带工作站ID 2：顺序 3：连续取得的标记 4：多个拾取数 5：用于存储预估时间的寄存器序号 6：用于存储工件信息的视觉寄存器序号 7：用于存储状态的寄存器序号 8(可省略)：模型ID	通过PKCSGETID取得ID。 参数增加，可以的项都增加。
12	VSTKPFRT.PC	取得刚调出后，每分钟处理的工件的数量（预计搬送率）。	1：工作区域名 2：用于存储预计搬送率的寄存器序号	PKCSGETPFRT.PC	1：传送带工作站ID 2：用于存储预计搬送率的寄存器序号	通过PKCSGETID取得ID。
13	VSTKNPRT.PC	取得队列内的工件数。	1：工作区域名 2：用于存储保持工件数的寄存器序号	PKCSGETNPRT.PC	1：传送带工作站ID 2：用于存储保持工件数的寄存器序号	通过PKCSGETID取得ID。

H 传感器任务的自定义

H.1 概要

传感器任务在 *i*RPickToo 的工作单元中监视传送带的行程或外部信号（DI、RI、HDI），条件成立后，发挥执行视觉程序等动作的作用。

可以在 *i*RPickTool 中，跟以往的视觉追踪一样自定义此动作部分。

譬如，在检测对象物候直至将该信息放入到队列中为止的期间，可以追加应用固有的处理。

这里，说明自定义传感器任务的步骤、和动作部分的自定义要点。

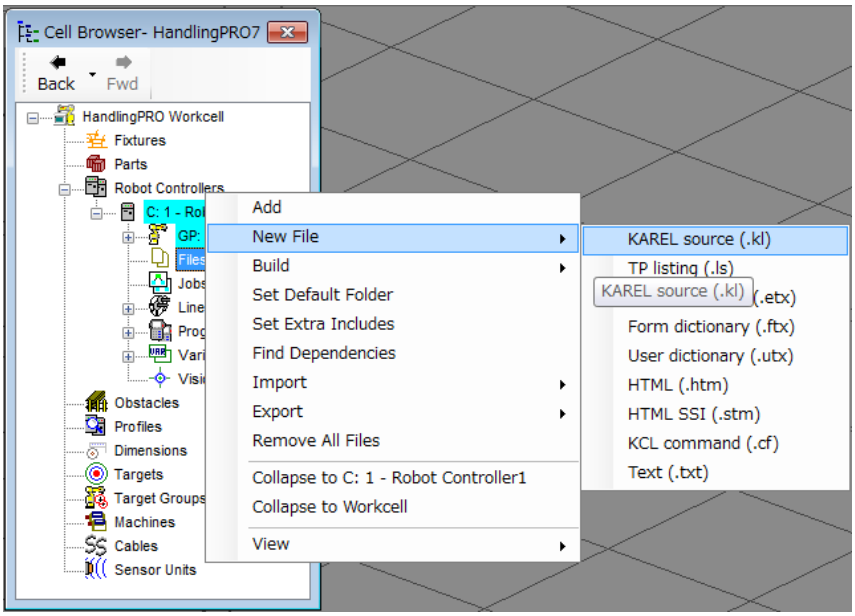
> ⚠ **注意**
> **使用自定义程序时由客户承担责任。使用前请通过实机充分确认动作。**

H.2 自定义传感器任务的步骤

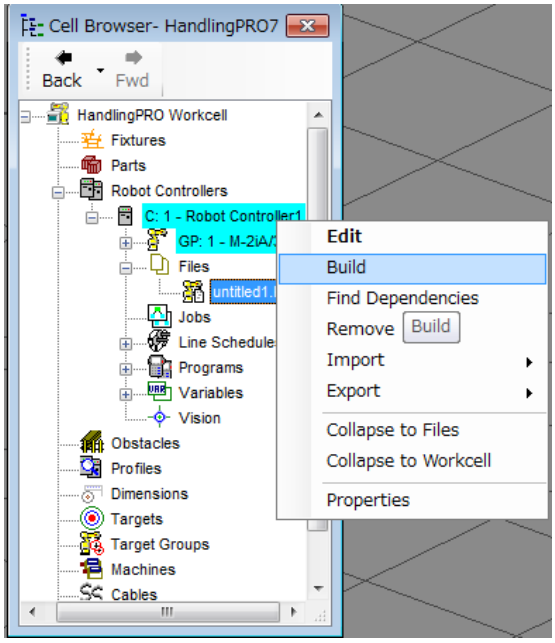
传感器任务利用 ROBOGUIDE 予以自定义。
以下操作通过 ROBOGUIDE 而进行。

1 利用 ROBOGUIDE 创建工作单元。
在创建虚拟机器人时，至少要指定如下选项。
- *i*RPickTool Basic (J944)

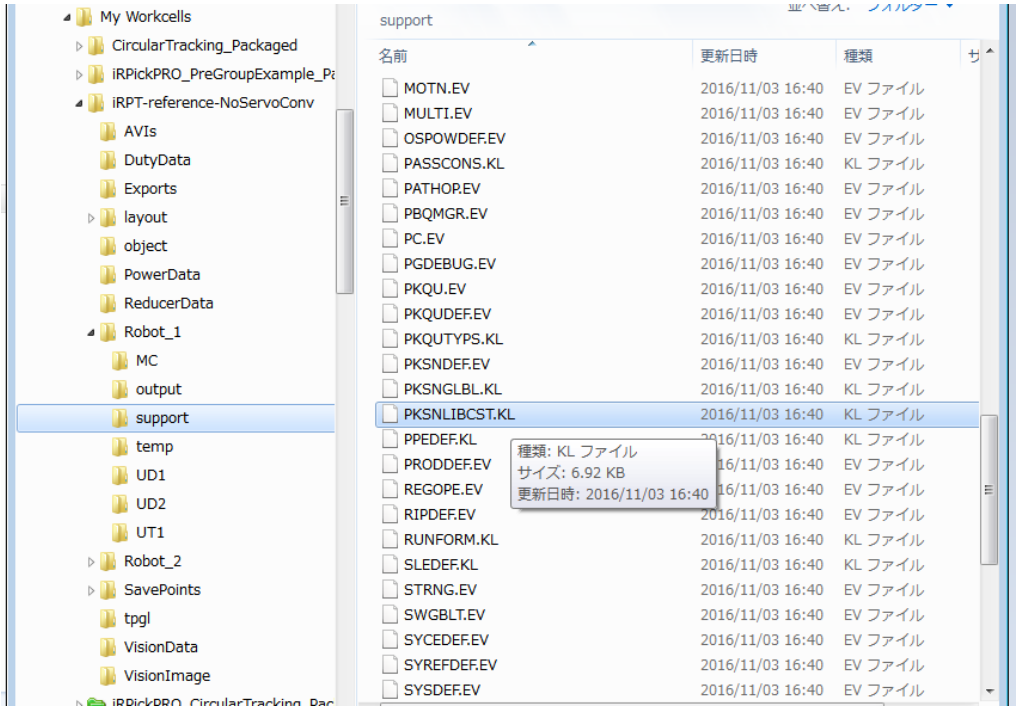
2 从右击单元浏览器中的［Files］后显示的菜单中选择［New File］→［KAREL source (.kl)］。



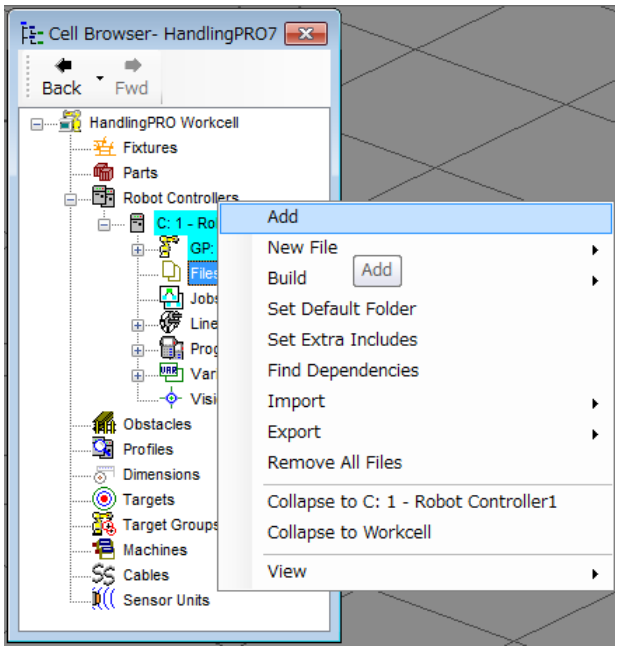
3 不进行任何编辑，创建已新建的 KAREL 资源。
在别的窗口，显示“File cannot be opened or created. …”（文件无法打开或创建），可予以忽略。



由此，在工作单元文件夹的 Robot_1 文件夹内生成 support 文件夹。



4 在单元浏览器中添加 support 文件夹内的 pksnlibcst.kl 。
pksnlibcst.kl 是自定义用的范例 KAREL 程序。



5 自定义 pksnlibcst.kl。

⚠ 注意
无法变更 KAREL 程序的名称和函数名。

6 创建已自定义的 pksnlibcst.kl。

至此为止是利用 ROBOGUIDE 进行的作业。

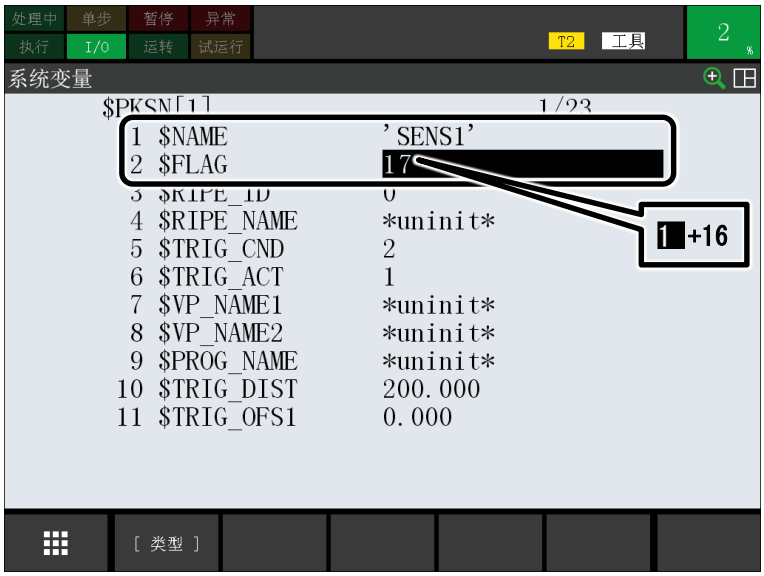
以下作业利用实机(机器人控制装置)进行。
将已经创建而生成的 KAREL 程序（.PC）加载到实机中利用。

7 在冷启动了实机的状态下，从 TP 的文件菜单加载已创建的 pksnlibcst (.PC)。

注释
在冷启动的状态下，以全部备份的方式保存从 TP 的文件菜单加载的 KAREL 程序。

8 进行用于自定义的 KAREL 程序动作中执行的设定。
如果处于已注册到 *i*RPickTool 的设定页的状态，则暂时注销。

传感器已创建完毕时，在存储有希望对$PKSN[n].$NAME 进行 “SENS1”等自定义的传感器名称的排列的$PKSN[n].$FLAG 中设定 Bit4(=16)。
初始状态下，尚未设定 Bit4(=16)，因而只要在$PKSN[n].$FLAG 加 16 即可。



尚未创建传感器时，首先请创建传感器。而后，如前所述那样在$PKSN[n].$FLAG(系统变量)中设定 Bit4(=16)。

> ⚠ **注意**
> **在 *i*RPickTool 的设定页，在已注册的状态下进行了设定上述位的作业时，在设定页变更传感器的设定时，所设定的位恐会被复位。这种情况下，请再次进行设定位的作业。**

9 跟以往一样，利用 TP 程序调用 PKWCSTART/PKWCEND 或 PKSNSTART/PKSNSTOP，启动/停止传感器。
只有进行了上述设定的传感器，通过已自定义的 KAREL 程序执行动作。

是否已执行所自定义的 KAREL，可通过如下所示方式进行确认。

10 将动作执行 1 次以上。
在机器人控制装置的示教器上按下［MENU（菜单）］键，从菜单中选择［下页］→［状态］→［程序］。

动作为［放入工件］时，若［程序］的项目中有 PKSNPUTCSTn (n = 1, 2, 3, 4)，则所自定义的 KAREL 程序已将被执行。通常，执行 PKSNPUTn (n = 1,2,3,4)的程序。

动作为［通过视觉检出］时，［程序］的项目中如有 PKSNFINDCSTn (n = 1, 2, 3, 4)，则所自定义的 KAREL 程序已被执行。通常，执行 PKSNFINDn (n = 1, 2, 3, 4)的程序。

PKSNPUTCSTn (n = 1, 2, 3, 4) 和 PKSNFINDCSTn (n = 1, 2, 3, 4) 的 n，是系统软件为了识别有效传感器而附加的 id。
譬如，SENS1 有效、SENS2 无效、SENS3 有效，它们的设定已被按如下所示方式存储在系统变量的情况下，系统软件从上按照顺序对有效的传感器赋予编号。

$PKSN[1].$NAME=“SENS1”
$PKSN[2].$NAME=“SENS2”
$PKSN[3].$NAME=“SENS3”
$PKSN[4].$NAME=“”

上述情况下，SENS1 为 1 号，SENS3 为 2 号。
在执行各自的传感器的动作时，调用与这里赋予的编号对应的 KAREL 程序。
SENS1 的动作若是［放入工件］则为 PKSNPUTCST1。
SENS3 的动作若为［通过视觉检出］则为 PKSNFINDCST2。
进而，各 KAREL 程序调用与 pksnlibcst 中所定义各动作对应的后述程序。

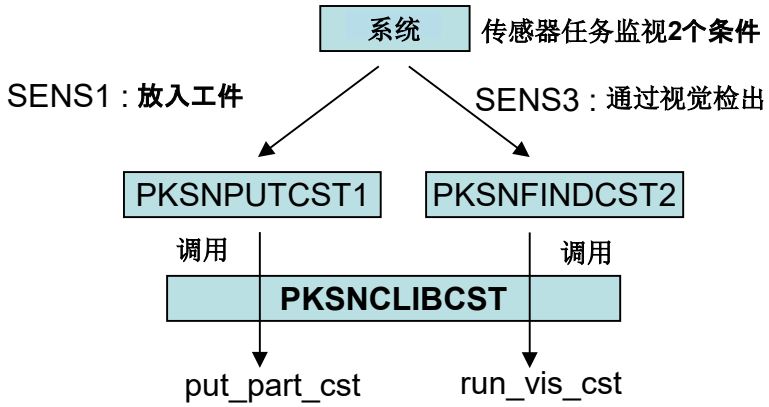


图 H.2(a) SENS1 和 SENS3 有效时条件成立情形下调用的流程

H.3 自定义的要点

pksnlibcst.kl 中，记述有执行［放入工件］和［通过视觉检出］各自的动作的函数。在设定了前述的旗标的传感器上，从系统软件调用这里记述的函数。各函数的详情，请参照后述的“H.4 KAREL 函数”和“H.5 内嵌函数”。这里示出自定义的要点。

定义部分：

（请勿变更此部分。）

```
-----------------------------------------------------------------------------------------------------------
PROGRAM pksnlibcst
-- You cannot change this KAREL program name
-----------------------------------------------------------------------------------------------------------
%COMMENT='CustomSenLib'
%SYSTEM
%NOBUSYLAMP
%NOLOCKGROUP
%NOPAUSESHFT
%NOPAUSE=ERROR+COMMAND+TPENABLE
%NOABORT=ERROR+COMMAND+TPENABLE

%ENVIRONMENT cvis          -- Builtins for vision
%ENVIRONMENT ltrk          -- Builtins for line tracking
%ENVIRONMENT pksndef       -- Variable for sensor

%INCLUDE pksnglbl
%INCLUDE vierrdef  -- Vision error definition. VEXE_E_NOMOR


-----------------------------------------------------------------------------------------------------------
ROUTINE get_ecnt_idx(indexer_id : INTEGER) : INTEGER FROM pksnlib
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
ROUTINE update_nfnd(cond_id : INTEGER; num_found : INTEGER) FROM pksnlib
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
ROUTINE start_srv(cond_id : INTEGER) FROM pksnlib
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
ROUTINE incre_wid(work_id : INTEGER) FROM pksnlib
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------
ROUTINE put_queue(cnv_id : INTEGER; stn_id : INTEGER; work_id : INTEGER; enc_count : INTEGER;
model_id : INTEGER; offset : XYZWPR; found_pos : ARRAY OF XYZWPR; meas : ARRAY OF REAL; duplicated :
INTEGER; stat : INTEGER) FROM pksnlib
-----------------------------------------------------------------------------------------------------------
```

动作［放入工件］处理部：

```
-----------------------------------------------------------------------------------------------------------
ROUTINE put_part_cst(cond_id : INTEGER; stat : INTEGER)
-- Action: “Put part in queue”
-- You cannot change this function name
-----------------------------------------------------------------------------------------------------------
VAR
  model_id          : INTEGER
  enc_count         : INTEGER
  offset    : XYZWPR
  I         : INTEGER
  dmy_ofs           : XYZWPR
```

```
  dmy_meas          : ARRAY[MAX_MEAS] OF REAL
  duplicated        : INTEGER
  nfound  : INTEGER

BEGIN
  -- Initialize number of found part
  nfound = 0

  -- Read the triggered encoder count
  IF sens_cfg[cond_id].trig_cnd = PKSN_CND_FLG THEN
    -- For servo conveyor (indexer)
    enc_count = get_ecnt_idx(sens_cfg[cond_id].indexer_id)
  ELSE
    IF sens_cfg[cond_id].trig_cnd = PKSN_CND_DST THEN
      -- For belt conveyor where part is put every the specified distance
      enc_count = $pksnact[cond_id].$enc_count
    ELSE
      -- For belt conveyor where part is put when DI or HDI is input
      enc_count = LINE_COUNT(sens_cfg[cond_id].enc_id)
    ENDIF
  ENDIF

  -- Start the service to read accurate encoder for the next action
  start_srv(cond_id)

  -- Set dummy results. You can customize them before they are put
dmy_ofs.x = 0.0
  dmy_ofs.y = 0.0
  dmy_ofs.z = 0.0
  dmy_ofs.w = 0.0
  dmy_ofs.p = 0.0
  dmy_ofs.r = 0.0
  FOR i = 1 TO MAX_MEAS DO
    dmy_meas[i] = 0.0
  ENDFOR
  model_id = 0

  -- Put part information to queue
  put_queue(sens_cfg[cond_id].cnv_id,
            sens_cfg[cond_id].stn_id,
            sens_cfg[cond_id].work_id,
            enc_count,
            model_id,
            dmy_ofs,
            sens_cfg[cond_id].sens_pos,
            dmy_meas,
            duplicated,
            stat)
  IF (stat = ER_SUCCESS) THEN
    nfound = 1
    -- Update work id. Work id is updated as below. Work id = 1,2,3...
    incre_wid(sens_cfg[cond_id].work_id)
  ENDIF

  -- Notify of the number of found part. You must call this function in the end of this routine.
  update_nfnd(cond_id, nfound)
END put_part_cst
```

有希望追加到队列中的信息时，在这里存储值。

H

动作［通过视觉检出］处理部：

```
---------------------------------------------------------------------------------------------------------
ROUTINE run_vis_cst(cond_id : INTEGER; stat : INTEGER)
-- Action: “Find part by vision”
-- You cannot change this function name
---------------------------------------------------------------------------------------------------------
VAR
   model_id            : INTEGER
   enc_count           : INTEGER
   offset    : XYZWPR
   tray_pos            : XYZWPR
   found_pos           : ARRAY[MAX_NUM_VIEW] OF XYZWPR
   meas    : ARRAY[MAX_MEAS] OF REAL
   nfound  : INTEGER
   duplicated          : INTEGER

BEGIN
   -- Initialize number of found parts
   nfound = 0

   -- Find parts
   V_RUN_FIND(sens_cfg[cond_id].vp_name1, -1, stat)
   WHILE (stat = ER_SUCCESS) DO

      -- Get information of found parts
      VT_GET_FOUND(sens_cfg[cond_id].vp_name1, model_id, enc_count, offset, found_pos, meas, stat)
      IF (stat = ER_SUCCESS) THEN

         -- If tray, compute tray position on track frame
         IF (sens_cfg[cond_id].tray_id <> 0) THEN
            found_pos[1] = found_pos[1] : sens_cfg[cond_id].tray_pos
         ENDIF
```

有希望追加到队列中的信息时，在这里存储值。

```
         -- Put part information to queue
         put_queue(sens_cfg[cond_id].cnv_id,
                   sens_cfg[cond_id].stn_id,
                   sens_cfg[cond_id].work_id,
                   enc_count,
                   model_id,
                   offset,
                   found_pos,
                   meas,
                   duplicated,
                   stat)
         IF (stat = ER_SUCCESS) THEN
            nfound = nfound + 1
            -- Update work id. Work id is updated as below. Work id = 1,2,3...
            incre_wid(sens_cfg[cond_id].work_id)
         ENDIF
      ENDIF
   ENDWHILE
   IF stat = VEXE_E_NOMOR THEN
      stat = ER_SUCCESS
   ENDIF

   -- Notify of the number of found part. You must call this function in the end of this routine.
```

希望对应模型 ID 将工件放入别的传送带上时，在这里进行条件转移。传送带 ID 和传送带工作站 ID 的取得，利用后述的内嵌函数 P_CVGETID、P_CSGETID。

```
  update_nfnd(cond_id, nfound)
END run_vis_cst
```

H.4 KAREL 函数

H.4.1 put_part_cst

目的：

执行动作［投入工件］时从系统软件调用的函数。

句法：

put_part_cst(cond_id, status)
返回值型: 无
[in] cond_id : INTEGER
[out] status : INTEGER

详细：

- cond_id：是系统软件为了识别有效的传感器而赋予的 ID。传感器编号并非必然一致。传感器编号可从 sens_cfg[cond_id].sens_id 进行参照。cond_id 所指的传感器，可从$PKSN[sens_cfg[cond_id].sens_id].$NAME 进行参照。
- 在 status 中存储处理的状态。若处理没有正常结束就存储 0(ER_SUCCESS)，否则就存储报警代码。

H.4.2 run_vis_cst

目的：

执行动作[视觉检出］时从系统软件调用的函数。

句法：

run_vis_cst(cond_id, status)
返回值型: 无
[in] cond_id : INTEGER
[out] status : INTEGER

详细：

- cond_id：是系统软件为了识别有效的传感器而赋予的 ID。传感器编号并非必然一致。传感器编号可从 sens_cfg[cond_id].sens_id 进行参照。
- 在 status 中存储处理的状态。若处理没有正常结束就存储 0(ER_SUCCESS)，否则就存储报警代码。

H.4.3 get_ecnt_idx

目的：

使用了伺服传送带时，取得编码器值。

句法：

get_ecnt_idx(indexer_id)
返回值型: INTEGER
[in] indexer_id : INTEGER

详细：

- indexer_id：指定伺服传送带编号。所使用的伺服传送带编号，已被存储在 sens_cfg[cond_id].indexer_id 中。
- 返回值是编码器值。

H.4.4 update_nfnd

目的：

用于向系统软件通知检出个数的函数。在动作中必须要调用。

句法：

update_nfnd(cond_id, num_found)
返回值型: 无
[in] cond_id : INTEGER
[in] num_found : INTEGER

详细：

- cond_id：是系统软件为了识别有效的传感器而赋予的 ID。用户自身无法赋予。此外，传感器的编号也不同。传感器编号可从 sens_cfg[cond_id].sens_id 进行参照。
- num_found：是检测个数。

H.4.5 start_srv

目的：

启动实施高精度触发器的服务。在动作中必须要调用。

句法：

start_srv(cond_id)
返回值型: 无
[in] cond_id : INTEGER

详细：

- cond_id：是系统软件为了识别有效的传感器而赋予的 ID。用户自身无法赋予。此外，传感器的编号也不同。传感器编号可从 sens_cfg[cond_id].sens_id 进行参照。

H.4.6 incre_wid

目的：

这是使得工件 ID 增加的函数。工件 ID 超过 13421771 时，返回至 0。使用托盘时，在放入到队列中后，作为工件 ID 向各单元分配（托盘的工件 ID×160＋单元编号－1）。为了避免该值超过 INTEGER 的最大值，通过传感器任务分配的工件 ID 的最大值为 13421771。

句法：

incre_wid(work_id)
返回值型: 无
[in/out] work_id : INTEGER

详细：

- work_id：是上次的工件 ID。
- 增加的工件 ID 被存储在 work_id 中。

H.4.7 put_queue

目的：

这是将工件信息放入到队列中的函数。同时进行同一物检查，在同一工件的情况下，作为状态（= 下述“句法”中的 status）返还。作为对象的传送带工作站的机器人离线时，自动地将工件信息放入到后续的传送带工作站中。所有的机器人都离线时，在 TP 上显示表示该内容的信息，为了继续后续的处理，作为 SUCCESS (= 0)结束状态。

句法：

put_queue(cnv_id, stn_id, work_id, enc_count, model_id, offset, found_pos, meas, duplicated, stat)
返回值型: 无
[in] cnv_id : INTEGER
[in] stn_id : INTEGER
[in] work_id : INTEGER
[in] enc_count : INTEGER
[in] model_id : INTEGER
[in] offset : XYZWPR
[in] found_pos : ARRAY[4] OF XYZWPR
[in] meas : ARRAY[10] OF REAL
[out] duplicated : INTEGER
[out] status : INTEGER

详细：

- 在 cnv_id 中指定对象物流动的传送带的传送带 ID。传送带 ID 可从 sens_cfg[cond_id].cnv_id 取得。
- 在 cstn_id 中指定放入工件信息的传送带工作站 ID。传送带工作站 ID 可从 sens_cfg[cond_id].stn_id 取得。
- 在 work_id 中指定用来识别工件的工件 ID。如没有必要区别工件 ID，则指定 0。
- 在 enc_count 中指定编码器值。编码器值，是在拍摄对象物的瞬间，或者传感器检测到对象物的瞬间的编码器值。
- 在 model id 中指定检测到的对象物的模型 ID。
- 在 offset 中指定检测到的对象物的补偿数据。
- 在 found_pos[1] ～ found_pos [4]中分别指定相机视图 1～4 检测到的对象物的检测位置。
- 在 meas_val[1] ～ meas_val[10] 中指定检测的到的对象物的测量值。
- 从 enc_count 至 meas_val，通常指定从后述的内嵌函数 VT_GET_FOUND 取得的值。
- 被判断为新放入的工件信息相同的工件信息已经存在于队列中时，在 duplicated 中存储 1，工件就不会被放入到队列中。不存在时，在 duplicated 中存储 0，工件就不会被放入到队列中。
- status：在 status 中存储处理的状态。若处理没有正常结束就存储 0(ER_SUCCESS)，否则就存储报警代码。

H.5 内嵌函数

H.5.1 LINE_COUNT

目的：

取得编码器值。

句法：

LINE_COUNT(encoder_id)
返回值型: INTEGER
[in] encoder_id : INTEGER
%ENVIRONMENT GROUP: LTRK

详细：

- 在 encoder_id 中指定编码器编号。
- 返回值是编码器值。通常返还执行此内嵌时的编码器值。
- 预先执行 ACCU_TRIG 内嵌函数时，返还检测到 DI 信号接通（OFF→ON）瞬间的正确的编码器值。在 start_srv 例行程序内调用 ACCU_TRIG。
- 利用 HDI 时，返还检测到 HDI 信号接通（OFF→ON）瞬间的正确的编码器值。

H.5.2 V_RUN_FIND

目的：

执行视觉程序。

句法：

V_RUN_FIND(vp_name, camera_view, status)
[in] vp_name: STRING
[in] camera_view: INTEGER
[out] status : INTEGER
%ENVIRONMENT GROUP: CVIS

详细：

- 在 vp_name 中指定视觉程序名。
- 在 camera_view 中指定相机视图的编号。若指定 0，则执行所有的相机视图。
- 在 status 中存储处理的状态。若处理没有正常结束就存储 0(ER_SUCCESS)，否则就存储报警代码。
- 此内嵌函数在视觉程序完成图像的加载时刻进入后续行，视觉程序进行的图像处理，在后台继续。因此，可以并列执行 KAREL 程序的执行和图像处理。在图像处理执行中利用 KAREL 程序执行 VT_GET_FOUND 内嵌后，等待图像处理的完成。

H.5.3 VT_GET_FOUND

目的：

取得视觉程序检测到的对象物的信息。

句法：

VT_GET_FOUND(vp_name, model_id, enc_count, offset, found_pos, meas_val, status)
[in] vp_name : STRING
[out] model_id : INTEGER
[out] enc_count : INTEGER
[out] offset : XYZWPR
[out] found_pos : ARRAY[4] OF XYZWPR
[out] meas_val : ARRAY[10] OF REAL
[out] status : INTEGER
%ENVIRONMENT GROUP: CVIS

详细：

- 在 vp_name 中指定视觉程序名。
- 在 model_id 中存储检测到的对象物的模型 ID。
- 在 enc_count 中存储拍摄检测到的对象物图像之瞬间的编码器值。
- 在 offset 中存储检测到的对象物的补偿数据。
- 在 found_pos[1] ～ found_pos [4]中存储相机视图 1～4 检测到的对象物的检测位置。*i*RPickTool 中只使用 1 个相机视图。
- 在 meas_val[1] ～ meas_val[10]中存储检测到的对象物的测量值。测量值是通过视觉程序的测量值输出工具指定的值。

- 在 status 中存储处理的状态。若处理没有正常结束就存储 0(ER_SUCCESS)，否则就存储报警代码。
- 此内嵌函数，在内嵌函数 V_RUN_FIND 后使用。
- 视觉程序检测到 2 个以上的对象物时，反复调用此内嵌函数，直至 status 不再成为 0(ER_SUCCESS)。没有可取得的对象物时，在 status 中存储 117151(VEXE_E_NOMOR)。

H.5.4 P_CSGETID

目的：

从传送带名和机器人的组编号中，取得机器人在该传送带上进行作业的传送带工作站的 ID。

句法：

P_CSGETID(conv_name, group_num)
返回值型: INTEGER
[in] conv_name: STRING
[in] group_num: INTEGER
%ENVIRONMENT Group: PKQU

详细：

- 在 conv_name 中定传送带名。
- 在 group_num 中指定机器人的组编号。
- 作为返回值，返还通过 group_num 指定的机器人进行作业的传送带工作站的 ID。

H.5.5 P_CVGETID

目的：

从传送带名取得该传送带的 ID。

句法：

P_CVGETID(conv_name)
返回值型: INTEGER
[in] conv_name: STRING
%ENVIRONMENT Group: PKQU

详细：

- 在 conv_name 中定传送带名。
- 作为返回值，返还传送带的 ID。

I 3rd Party 接口加载项

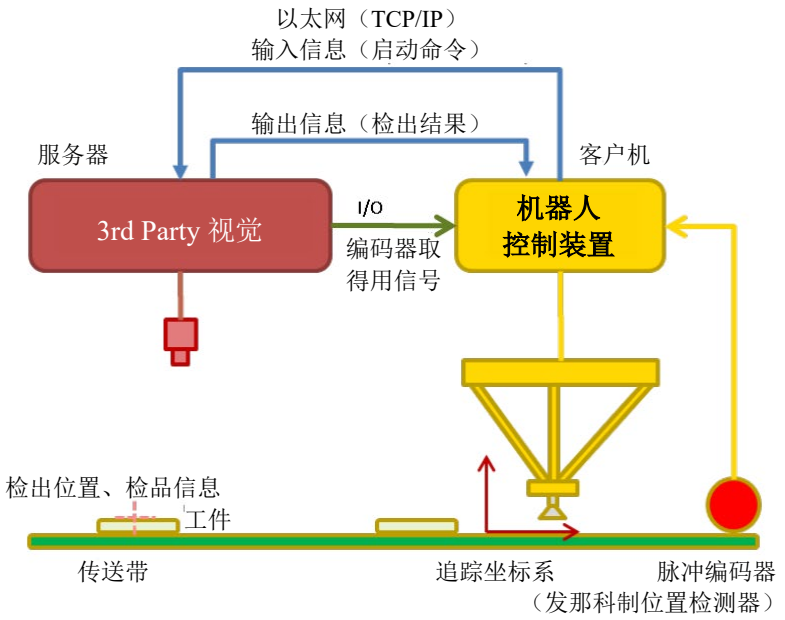
对 *i*RPickTool 3rd Party 接口加载项进行说明。

I.1 概要

*i*RPickTool 3rd Party 接口加载项是根据通过使用 TCP/IP 的通信从 3rd Party 视觉取得的工件的检出结果进行二维视觉追踪的功能。作为 *i*RPickTool 的传感器的触发，执行与 3rd Party 视觉进行通信的 TP 程序。此 TP 程序进行以下动作。

- 向 3rd Party 视觉发送启动命令，接收检出结果。
- 3rd Party 视觉收到拍摄时输出的信号（编码器取得用信号），取得编码数。
- 将检出位置的坐标转换成追踪坐标系上的位置。
- 将检出结果加入队列中。

与 3rd Party 视觉进行信息的接收和发送时，使用 KAREL 程序。信息格式由客户结合 3rd Party 视觉和系统来决定。



为了使用 *i*RPickTool 3rd Party 接口加载项，需要以下软件选项。

- J944（*i*RPickTool 基本）或 J945（*i*RPickTool）
- R648（用户套件信息功能）
- S521（*i*RPickTool / 3rd Party 接口加载项）

> **注释**
> **本功能只能在 7DF1 系列 17 版以后的软件下可以订购。**

I.2 用语

对 *i*RPickTool 3rd Party 接口加载项特有的用语进行说明。

数据
用分隔符进行分割的信息的各要素。

输入信息
机器人控制装置向 3rd Party 视觉发送的信息。信息的末尾带有换行字符。输入信息包含以下内容。

启动命令
3rd Party 视觉是执行拍摄和检出，发送检出结果的命令。详细内容请参考各视觉厂商的操作说明书。

输出信息
机器人控制装置收到 3rd Party 发送的信息。信息的末尾带有换行字符。输出信息包含以下内容。

检出结果
是 3rd Party 视觉向机器人控制装置输出的工件的信息。客户根据系统选择必要信息，决定信息格式。

响应
是 3rd Party 视觉收到命令时输出的状态。详细内容请参考各视觉厂商的操作说明书。

触发程序
与 3rd Party 视觉进行通信时，将检出结果加入队列的程序。

锁存程序
取得 3rd Party 视觉拍摄的瞬间编码数的程序。

编码器取得用信号
锁存使用的信号。输入此信号时，机器人控制装置取得编码数。信号中使用与频闪观测器信号等的拍摄同步的信号。

I.3 制约

对可使用 *i*RPickTool 3rd Party 接口加载项的机器人系统及 3rd Party 视觉的制约进行说明。

I.3.1 机器人系统的制约

- 仅可适用于二维视觉追踪。
- 3rd Party 视觉仅可适用于直线传送带。
- 在相同机器人控制装置中使用多个 3rd Party 视觉时，使用 3rd Party 视觉的传感器对应的传送带工作站中使用的编码器必须完全不同。
- 不能对适用 3rd Party 视觉的传送带使用托盘。
- 不可对双手臂机器人连接 3rd Party 视觉。
- 不可对预整列的输入传送带使用 3rd Party 视觉。

I.3.2 3rd Party 视觉的制约

- 需要能够通过经由以太网的 TCP/IP 协议进行通信。
- 需要可根据 TCP/IP 的 ASCII 形式进行通信。
- 3rd Party 视觉需要适应 TCP/IP 的服务器。不能将机器人控制装置作为服务器。
- 关于输入信息
  - 3rd Party 视觉应能够根据机器人控制装置发送的输入信息执行拍摄和检出。
  - 输入信息的各数据的字符串长应在 34 个字符以下。
  - 应对应 LF、CR、CR+LF、无中的任一换行字符。
- 关于输出信息
  - 执行完检出时或接收取得检出结果的输入信息时，应可以将输出信息（检出结果）发送给机器人控制装置。
  - 作为检出结果，应能够发送最低工件检出位置（X,Y,R）。
  - 输出信息的各数据的字符串长应在 254 个字符以下。
  - 检出结果的各数据应能够用‘，’（逗号）或‘ ’（空格）等分隔符进行分割。
  - 检出结果的末尾应可以带 LF、CR、CR+LF 中任一一种换行字符。
- 应可以按照拍摄时机从 3rd Party 视觉向机器人控制装置输出信号。

I.4 关于支持范围

对支持范围进行说明。

软件部分
- 本公司的支持范围只有机器人控制装置上的功能。
- 对于相机拍摄及图像中的对象检出和检查部分等的支持，请向各视觉厂商进行确认。
- 3rd Party 视觉向本公司的机器人控制装置发送正确的检出结果，机器人控制装置在接收正确时机的信号时将进行动作。

硬件部分
- 本公司的支持范围只有机器人和机器人控制装置部分。
- 相机、镜头、电缆等视觉系统部分请向各视觉厂商进行确认。

支持范围的详细内容请参照图 I.4（a）。

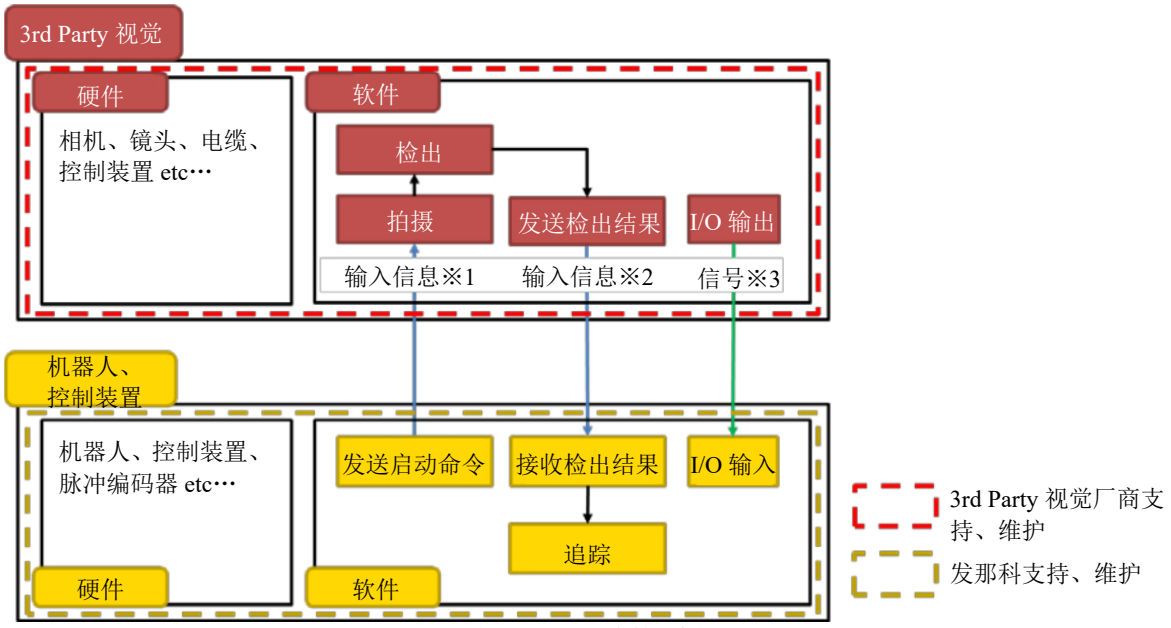


图 I.4（a） 支持范围

※1
本公司对将决定的输入信息用 TCP/IP 发送到 3rd Party 视觉之前的部分进行支持。
3rd Party 视觉根据收到的输入信息执行检出等处理的部分是 3rd Party 视觉厂商的支持范围。

※2
输出信息的格式由操作机器人的客户结合系统来决定。
结合格式创建输出信息，3rd Party 视觉用 TCP/IP 进行发送的部分是 3rd Party 视觉厂商的支持范围。
本公司从机器人控制转装置收到的输出信息中取出检出结果，对追踪部分进行支持。

※3
拍摄时用于取得传送带位置的信号（与频闪观测器信号等的拍摄同步的信号）的发送和发送时机是 3rd Party 视觉厂商的支持范围。
本公司支持信号的接收及取得该时点传送带位置的部分。

I.5 启动步骤

对 *i*RPickTool 3rd Party 接口加载项的启动步骤进行说明。
本节对与“3 *i*RPickTool 基本（基本功能）的启动步骤”使用相同构成的系统例进行说明。为了使用 *i*RPickTool 3rd Party 接口加载项，对供给传送带的组合进行如下更改。

供给：［距离］＋［程序的执行］

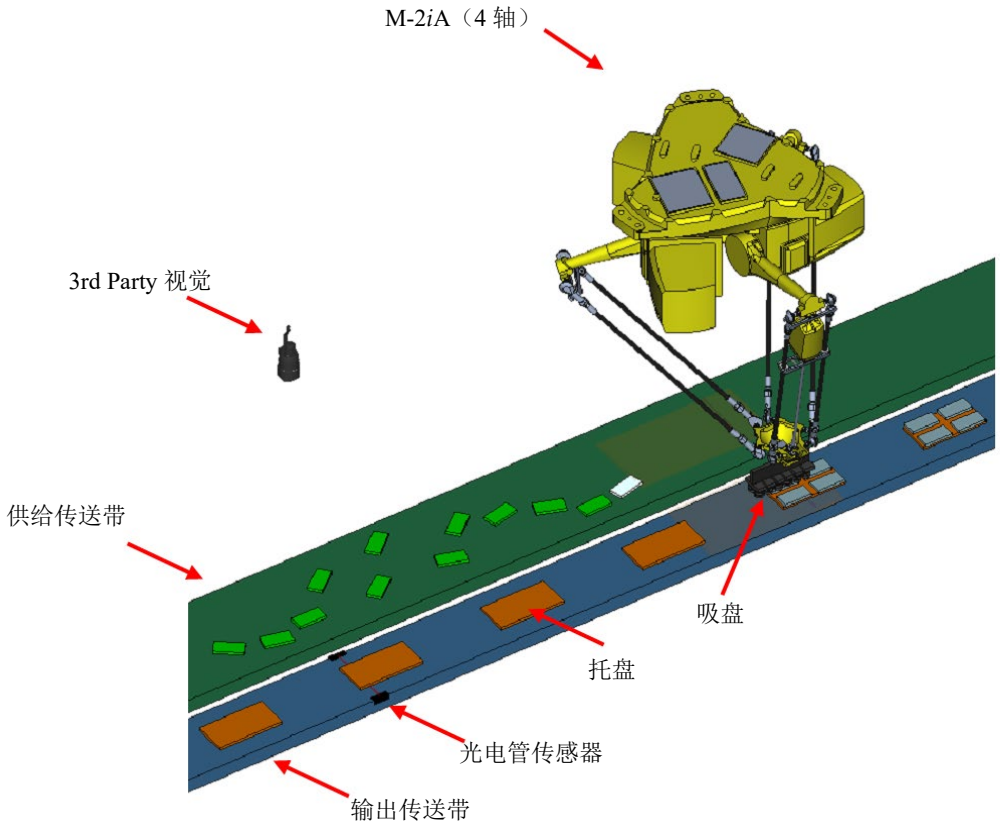


图 I.5（a） 系统构成

此处作为示例，对 3rd Party 视觉和机器人控制装置使用表 I.5（a）的信息格式进行通信的情况进行说明。进行实际适用时，请结合使用的 3rd Party 视觉的规格更改信息格式。

表 I.5（a） 信息格式

输入信息 (控制装置→3rd Party 视觉)	**RUNFIND(字符串)<CR>**
对输入信息的 响应 (3rd Party 视觉→控制装置)	**正常时 ：OK(字符串)<CR> 异常时 ：ER(字符串)<CR>**
输出信息 (3rd Party 视觉→控制装置)	**检出数(整数)、模型 ID(整数)、X(实数)、Y(实数)、R(实数)、测量值 1(实数)<CR>**

I.5.1 3rd Party 视觉的连接和设定

进行连接和设定，使 3rd Party 视觉和机器人控制装置能够进行通信。设定方法请参照各视觉厂商的操作说明书。

I.5.1.1 3rd Party 视觉的连接

- 通过以太网电缆连接 3rd Party 视觉和机器人控制装置。
- 将与频闪观测器信号等的拍摄同步的信号作为编码器取得用信号连接到机器人控制装置的 DI 或 RI。

注释
与 3rd Party 视觉连接的以太网端口请使用不进行其他通信的独立端口。

I.5.1.2 3rd Party 视觉的设定

- 设定时确保 3rd Party 视觉和机器人控制装置可以通过经由以太网的 TCP/IP 协议进行连接。
- 进行示教，以确保可以检测到工件。
- 设定时确保可以结合输出信息的格式将检出结果发送到机器人控制装置。
- 设定编码器取得用信号。

且记录以下项目，用于机器人控制装置的设定。
- TCP/IP 中使用的 IP 地址、端口号。
- 是 3rd Party 视觉执行拍摄、检出和检出结果发送的启动命令。
- 连接编码器取得用信号的机器人控制装置的 DI 编号或 RI 编号。

I.5.2 通信设定

进行与 3rd Party 视觉进行通信相关的设定。

I.5.2.1 IP 地址的设定

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
2 从菜单选择［设置］→［主机通讯］。
3 选择 TCP/IP，按下 F3［详细］。
4 设定 IP 地址及子网掩码。

I.5.2.2 客户标签的设定

主机通讯画面的设定

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
2 从菜单选择［设置］→［主机通讯］。
3 选择 F4［显示］→［2 客户］。
4 将光标移动到使用的客户标签，按下［输入］键。
5 将光标移动到［协议］，按下［输入］键。
6 选择［SM］。
7 将光标移动到［服务器 IP/主机名］，按下［输入］键。
8 输入 3rd Party 视觉的 IP 地址。
9 将光标移动到［启动时的状态］，按下［输入］键。
10 选择［定义］。
11 按下 F2［操作］，选择［定义］。

系统变量的设定

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
2 从菜单中选择［系统］→［系统变量］。
3 选择$HOSTC_CFG。
4 选择在“主机通讯画面的设定”中设定的客户标签编号。
5 确认$STRT_REMOTE 中显示了“主机通讯画面的设定”的步骤 8 中输入的 IP 地址。
6 向$SERVER_PORT 输入 3rd Party 视觉端口号。

I.5.3 *i*RPickTool 的设定

设定与“3 *i*RPickTool 基本（基本功能）的启动步骤”共通的项目。以下项目在使用 *i*RPickTool 3rd Party 接口加载项时也使用相同的设定。设定时请参照“3 *i*RPickTool 基本（基本功能）的启动步骤”的各项目。

- “3.1.1 脉冲编码器的连接与设定”

- “3.1.3 触发传感器的输入确认”
- “3.1.4 设定碰触销的工具坐标系”
- “3.1.6 负载设定（动作性能）”
- “3.2.1 工作单元设定”
- “3.2.2 托盘设定”
- “3.2.3 传送带设定”
- “3.2.4 追踪坐标系设定”
- “3.2.5.2 排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时）”或“3.2.5.3 排出传送带传感器的设定（［HDI］＋［放入工件］时）”
- “3.2.6 传感器位置的设定”
- “3.2.8 传送带工作站设定”
- “3.2.9 机械手工具坐标系的设定”
- “3.2.10.2 排出传送带基准位置的设定、机器人的位置示教”

I.5.4 机器人程序的创建

创建机器人程序程序的创建方法请参照“3.1.5 机器人程序的创建”。

机器人程序创建以下程序。关于追踪程序和吸附 ON/OFF 用程序，与“3.1.5 机器人程序的创建”中创建的程序相同，因此此处不进行说明。请参照“3.1.5 机器人程序的创建”来创建程序。

- 主程序
- 追踪程序
- 吸附 ON/OFF 用程序

在 *i*RPickTool 3rd Party 接口加载项中，需要与 3rd Party 视觉进行通信，取得检出结果的程序。此程序作为 *i*RPickTool 的触发，每次满足触发条件时，从传感器任务中执行。与主程序的关系请参照图 I.5.4（a）。

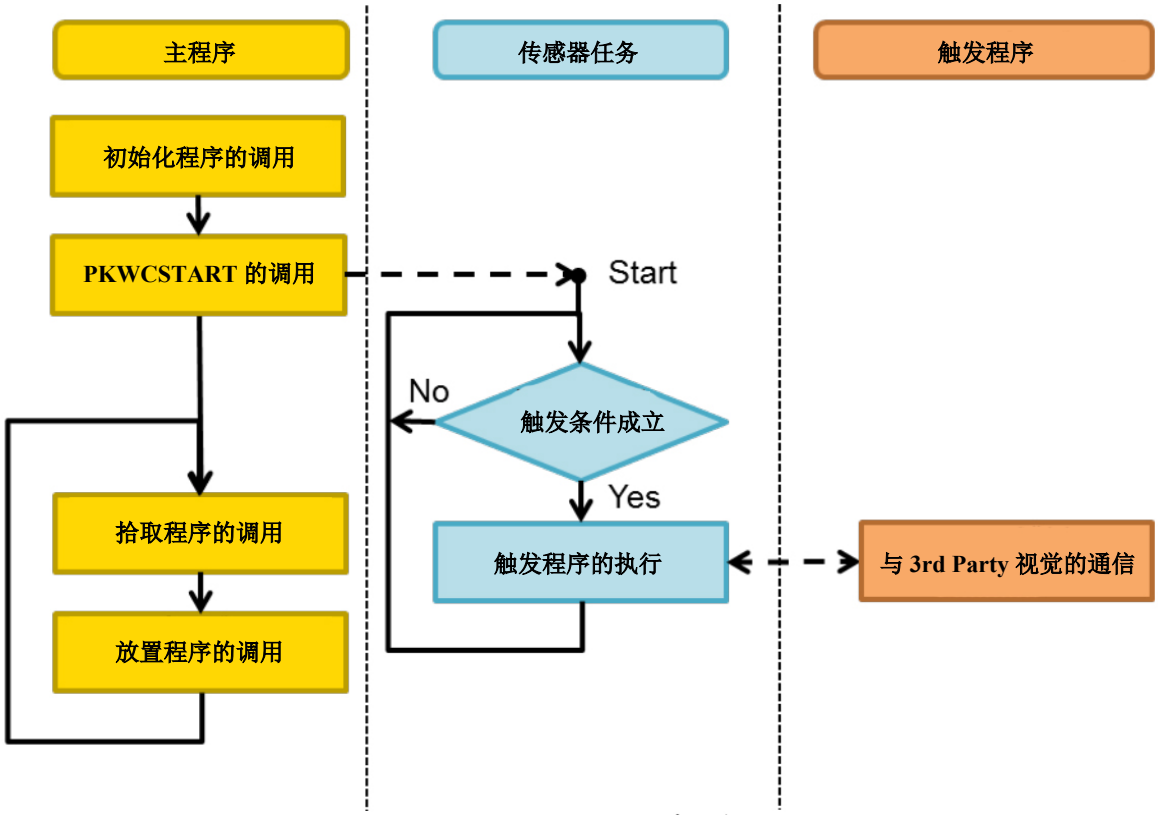


图 I.5.4（a） 被执行的程序

作为与 3rd Party 视觉进行通信，取得检出结果的程序，创建以下程序。

- 触发程序
- 锁存程序

- 检出程序

注释
在上述程序中，请将忽略暂停设为有效。如果在通信过程中因急停等原因造成程序退出，工件信息有可能会在投入之前消失。

在这些程序中进行以下动作。程序流程请参照图 I.5.4（b）。
1　向 3rd Party 视觉发送启动命令，接收检出结果。
2　3rd Party 视觉收到拍摄时输出的信号（编码器取得用信号），取得编码数。
3　将检出位置的坐标转换成追踪坐标系上的位置。
4　将检出结果加入队列中。

此外，为了对与 3rd Party 视觉的连接等进行初始化，创建以下程序。
- 初始化程序

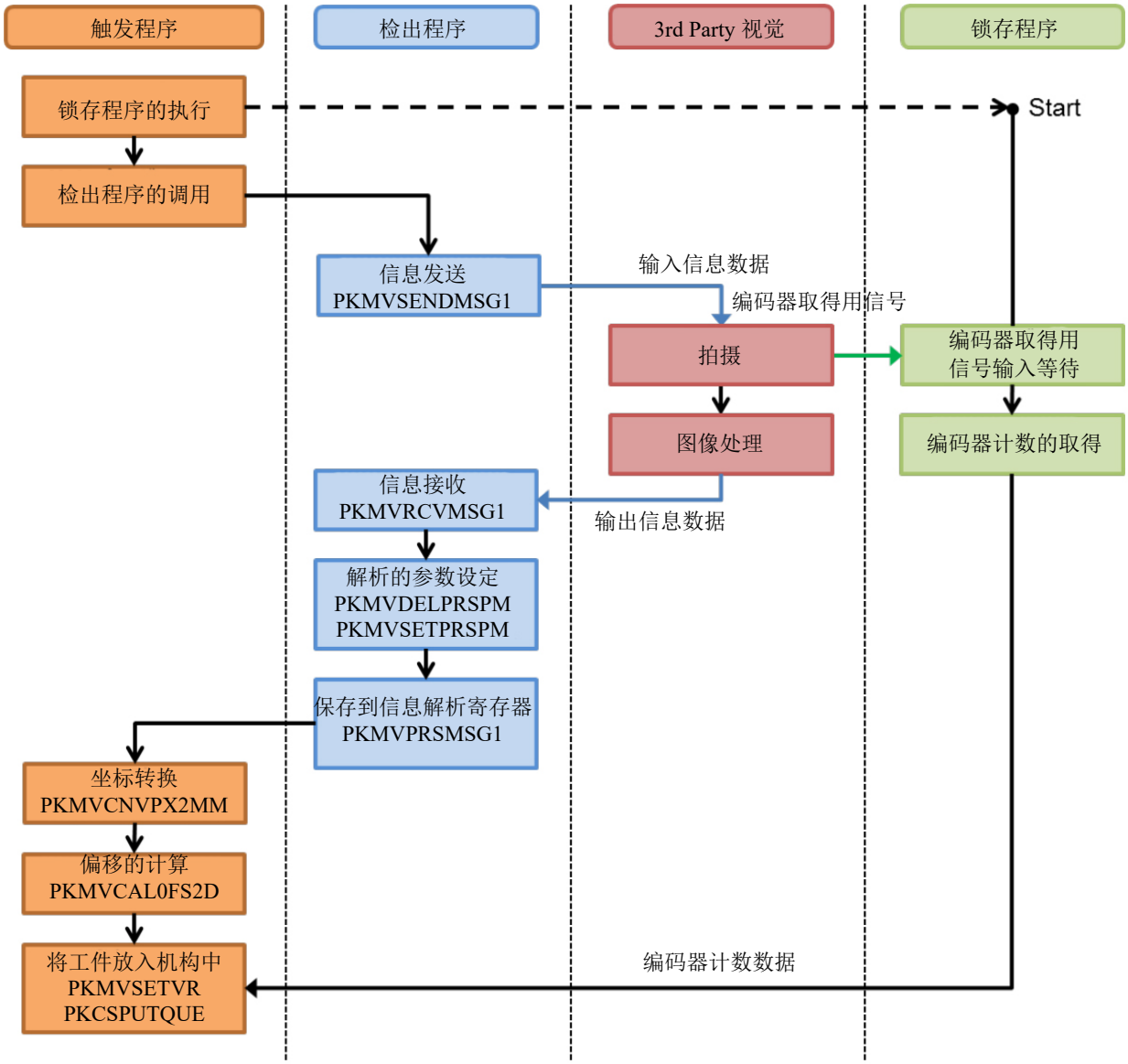


图 I.5.4（b） 触发程序的流程图

在 *i*RPickTool 3rd Party 接口加载项中，制作用于坐标转换参数计算和基准位置设定的程序。
- 坐标转换参数计算程序
- 基准位置设定程序

数值寄存器的定义

此程序中使用以下数值寄存器。请根据需要，适当更改数值寄存器编号。

表 I.5.4（a） 数值寄存器

R[2:GETQ 状态 Pick	**]=0**	**拾取工件处理状态**
R[3:GETQ 状态 Drop	**]=0**	**放置工件处理状态**
R[11:CONV1 ID	**]=1**	**传送带 1(拾取）ID**
R[12:CONV2 ID	**]=2**	**传送带 2(放置）ID**
R[21:Pick AP_LD	**]=30**	**拾取接近距离指定 [mm]**
R[22:Pick RT_LD	**]=30**	**拾取缩回距离指定 [mm]**
R[23:Pick Wait	**]=.05**	**在拾取位置的待机时间 [sec]**
R[24:Pick CNT	**]=0**	**拾取位置的平滑指定**
R[25:Pick TB	**]=.1**	**先应用时间(VACUUMON) [sec]**
R[26:Pick Speed1	**]=4000**	**拾取程序速度 1 [mm/sec]**
R[27:Pick Speed2	**]=4000**	**拾取程序速度 2 [mm/sec]**
R[28:Pick Speed3	**]=2000**	**拾取程序速度 3 [mm/sec]**
R[31:Drop AP_LD	**]=30**	**放置接近距离指定 [mm]**
R[32:Drop RT_LD	**]=30**	**放置缩回距离指定 [mm]**
R[33:Drop Wait	**]=.05**	**放置位置的待机时间 [sec]**
R[34:Drop CNT	**]=0**	**放置位置的平滑指定**
R[35:Drop TB	**]=.1**	**先应用时间(VACUUMOFF) [sec]**
R[36:Drop Speed1	**]=2000**	**放置程序速度 1 [mm/sec]**
R[37:Drop Speed2	**]=2000**	**放置程序速度 2 [mm/sec]**
R[38:Drop Speed3	**]=4000**	**放置程序速度 3 [mm/sec]**
R[40:编码器取得用编号	**]=2**	**编码器取得用 DI/RI 编号(请指定使用的编号)**
R[41:客户编号	**]=1**	**客户标签编号(请指定使用的编号)**
R[43:SENDMSG_STAUTS	**]=0**	**PKMVSENDMSG 的状态**
R[44:RCVMSG_STAUTS	**]=0**	**PKMVRCVMSG 的状态**
R[45:PRSMSG_STAUTS	**]=0**	**PKMVPRSMSG 的状态**
R[46:MVSFIND1_STAUTS	**]=0**	**3rd Party 视觉的检出结果的状态※1**
R[47:MVSACTION1 结束	**]=0**	**表示 MVS ACTION1 结束的标记**
R[48:LATCH 成功	**]=0**	**表示锁存成功的标记**
R[49:重复结果用 R	**]=0**	**将检出结果加入队列时的重复结果**
R[51:x0	**]=0**	**x0(坐标转换参数计算的检出位置)**
R[52:y0	**]=0**	**y0(坐标转换参数计算的检出位置)**
R[53:x1	**]=0**	**x1(坐标转换参数计算的检出位置)**
R[54:y1	**]=0**	**y1(坐标转换参数计算的检出位置)**
R[55:x2	**]=0**	**x2(坐标转换参数计算的检出位置)**
R[56:y2	**]=0**	**y2(坐标转换参数计算的检出位置)**
R[60:LINECOUNT	**]=0**	**编码数(检出结果)**
R[61:检出数	**]=0**	**检出数(检出结果)**
R[62:模型 ID	**]=0**	**模型 ID(检出结果)**
R[63:x	**]=0**	**x(检出结果)**
R[64:y	**]=0**	**y(检出结果)**
R[65:r	**]=0**	**r(检出结果)**
R[66:测量值 1	**]=0**	**测量值 1(检出结果)**

※1 3rd Party 视觉的检出结果的状态在 TP 程序中进行如下定义。

0 : PUTQUE 成功

1 : 在与 3rd Party 视觉的通信中发生错误

2 : 在 PKMVPRSMSG 中解析的信息的数据数不同

3 : 响应异常

4 : 检出数 0

5 : 未输入编码器取得用信号

表 I.5.4（b） 位置寄存器

PR[1:HOME POS]=R	原位置
PR[7:Pick Pos]=R	拾取位置
PR[8:Drop Pos]=R	放置位置
PR[11:Pick Apr Z]=R	从拾取位置偏移
PR[12:Drop Apr Z]=R	从放置位置偏移
PR[21:ADJ_OFS]=R	拾取位置补偿量调整用
PR[26:Ref Pos]=R	基准位置
PR[27:Detection Pos]=R	检出位置
PR[28:Offset]=R	偏移

表 I.5.4（c）字符串寄存器

SR[6 :Response]	响应
SR[7 :OK Response]	正常时的响应（请指定正常时的响应）

表 I.5.4（d）视觉寄存器

VR[1]	拾取位置补偿用
VR[2]	放置位置补偿用
VR[5]	检出结果存储用

I.5.4.1 主程序

对系统的主程序示例进行说明。

PNS0001

```
 1:  !PKCSGETID(Get conveyor station ID)
 2:  !*1:Conveyor name
 3:  !*2:Register number for conveyor station ID
 4:  !*3:Group number(Can be omitted)
 5:  CALL PKCSGETID('CONV1',ID Reg=11)
 6:  CALL PKCSGETID('CONV2',ID Reg=12)
 7:
 8:  ! 3rd Party 视觉的初始化
 9:  CALL MVSINIT1
10:
11:  UFRAME_NUM=0
12:  UTOOL_NUM=1
13:  PAYLOAD[1]
14:  L PR[1:HOME POS] 500mm/sec FINE Wjnt
15:
16:!--- System startup ---
17:  !PKWCSTART(System startup，including check and clear)
18:  CALL PKWCSTART
19:
20:  LBL[101]
21:  !--- Pick program ---
22:  CALL PT_PICKCS
23:  IF DI[1:CYCLE STOP]=ON,JMP LBL[901]
24:  !--- Placement program ---
25:  CALL PT_DROPCS
26:  IF DI[1:CYCLE STOP]=ON,JMP LBL[901]
27:  JMP LBL[101]
28:
29:  LBL[901]
30:  UFRAME_NUM=0
31:  UTOOL_NUM=1
```

KAREL 程序

KAREL 程序

初始化

KAREL 程序

```
 32:   L PR[1:HOME POS] 500mm/sec FINE Wjnt
 33:
 34:   !PKWCEND(System end)
 35:   CALL PKWCEND
[END]
```

进行 *i*RPickTool 和 3rd Party 视觉的初始化，进行追踪动作。

在第 9 行调出对与 3rd Party 视觉的连接等进行初始化的 TP 程序。详细信息请参照“I.5.4.3 初始化程序”。

其他行与“3.1.5 机器人程序的创建”的主程序相同。此外，在第 22、25 行调出的程序与“3.1.5 机器人程序的创建”的拾取程序和放置程序相同。请参照“3.1.5 机器人程序的创建”，创建拾取程序和放置程序。

I.5.4.2 触发程序

对作为 *i*RPickTool 的触发被执行的程序示例进行说明。

MVSACTION1

```
  1:   R[47:MVSACTION1 End]=0
  2:
  3:   !Use ACCUTRIG
  4:   ACCUTRIG LNSCH[1]
  5:
  6:   R[48:LATCH Success]=0
  7:   RUN LATCH
  8:
  9:   CALL MVSFIND1
 10:   IF R[46:MVSFIND1 Status]<>0,JMP LBL[990]
 11:
 12:   IF R[48:LATCH Success]=0,JMP LBL[900]
 13:
 14:   !Store found position in PR
 15:   PR[27,1:Found Pos]=R[63:x]
 16:   PR[27,2:Found Pos]=R[64:y]
 17:   PR[27,3:Found Pos]=0
 18:   PR[27,4:Found Pos]=0
 19:   PR[27,5:Found Pos]=0
 20:   PR[27,6:Found Pos]=R[65:r]
 21:
 22:   !Conv fnd pos into tracking frm
 23:   CALL PKMVCNVPX2MM(“Action ID”=1,“Fnd Pos PR”=27,“Cnv Pos PR”=27)
 24:   !Calculate offset
 25:   CALL PKMVCALOFS2D(“Fnd Pos PR”=27,“Ref Pos PR”=26,“Offset PR”=28)
 26:
 27:   !Put result into queue
 28:   CALL PKMVSETVR(“VR”=5,“Fnd Pos PR”=27,“Offset PR”=28,“Model ID Reg”=62,“Enc
Count Reg”=60,66)
 29:   CALL PKCSPUTQUE(“CStn ID”=R[11:CONV1 ID],“Offset VR”=5,“OverlapStat Reg”=49)
 30:   JMP LBL[990]
 31:
 32:   LBL[900:LATCH ERORR]
 33:   R[46:MVSFIND1 Status]=5
 34:
 35:   LBL[990]
 36:   R[47:MVSACTION1 End]=1
 37:   CALL PKMVWAITTASK(“Task Name”='LATCH')
[End]
```

使用高精度触发

执行锁存程序

取得检出结果

确认锁存是否成功

存储到位置寄存器中

检出位置的坐标变换

将检出结果加入队列

错误处理

作为 *i*RPickTool 的触发被执行的程序。从 3rd Party 视觉取得检出结果，加入到队列中。

在第 4 行进行高精度触发的设定。使用 1 号以外的传送带追踪参数设定表编号时，请更改成使用此参数的传送带追踪参数设定表编号。
在第 7 行从 3rd Party 视觉输入信号时，请执行取得编码数的锁存程序。详细内容请参照“I.5.4.5 锁存程序”。

在第 9 行调出从 3rd Party 视觉取得检出结果的程序。详细内容请参照“I.5.4.4 检出程序”。

确认在第 12 行是否成功锁存。编码器取得用信号的输出时机延迟时或者输出宽度较长时，可能会被判定为在此判定之前未检测到信号，锁存失败。此时，对于编码器取得用信号，请使用与拍摄同时输出的信号，并将信号的输出宽度设置得短些。或者在该行的前面添加待机命令。

在第 15~20 行将检出位置存储到位置寄存器中。
在第 23 行转换检出结果的坐标，在第 25 行计算偏移。
在第 28 行将结果存储到视觉寄存器，在第 29 行将工件放入队列中。

在第 32~33 行发生错误时，定义了跳过水平。进行错误处理时在此处进行描述。

在第 36 行将 1 存储到 R［47］，将触发程序结束的情况传达给锁存程序，使锁存程序结束。在第 37 行，在锁存程序结束之前进行待机。

⚠注意
1 动作程序的 PKMVSETVR 视觉寄存器编号请指定不同于拾取程序或放置程序 PKCSGETQUE 视觉寄存器编号的值。如果指定的值相同，有可能会被覆盖而取到错误位置。
2 除动作程序之外，请勿执行 PKMVSENDMSG 和 PKMVRSVMSG。
多个程序试图同时通信时，发生“INTP-326（PKMVLIB、207) 文件变量已使用”的警报。但是，只要不在动作程序中使用相符的通信端口，就不会有问题。

I.5.4.3 初始化程序

对进行初始化的程序示例进行说明。

MVSINIT1

```
   1:  CALL PKMVCONNECT(“Client ID”=R[41:Client ID])
   2:
   3:  CALL PKMVINIACT(“Action ID”=1,
                       “Client ID”=R[41:Client ID],
                       “In Msg Delimiter”=',',
                       “In Msg Newline char”='CR',
                       “In Msg Max num data”=10,
                       “Out Msg Delimiter”=',',
                       “Out Msg Newline char”='CR',
                       “Out Msg Max num data”=20)
   4:
   5:  CALL PKMVSETSNDPM(“Action ID”=1,“Stat Reg”=43,'RUNFIND')
   6:  CALL PKMVSETRCVPM(“Action ID”=1,“Timeout (ms)”=5000,“Stat Reg”=44)
[End]
```

与 3rd Party 视觉连接

触发的初始化

发送参数的设定

接收参数的设定

在第 1 行与 3rd Party 视觉进行连接。请事先输入在 R［41］中使用的客户编号。

在第 3 行进行触发的初始化。参数的详细内容请参照“I.6.2.1 PKMVINIACT.PC”。

在第 5 行设定输入信息的设定。请在输入信息中指定 3rd Party 视觉的启动命令。

请在第 6 行进行接收时的参数设定。请为超时指定比 3rd Party 视觉拍摄和检出的时间更长的时间。

注释

有的 3rd Party 视觉，在连接时需要登录等，只在上述程序中未完成连接。此时，请结合 3rd Party 视觉添加必要的处理。

I.5.4.4 检出程序

对与 3rd Party 视觉进行通信，取得检出结果的程序示例进行说明。

MVSFIND1

```
  1:  R[46:MVSFIND1 Status]=1
  2:
  3:  !Send run command
  4:  CALL PKMVSENDMSG1
  5:
  6:  !Receive response
  7:  CALL PKMVRCVMSG1
  8:
  9:  CALL PKMVDELPRSPM(“Action ID”=1)
 10:  CALL PKMVSETPRSPM(“Action ID”=1,“Stat Reg”=45,
                        3,6)
 11:  CALL PKMVPRSMSG1
 12:  IF R[45:PRSMSG Status]<>0,JMP LBL[900]
 13:  IF SR[6]<>SR[7],JMP LBL[910]
 14:
 15:  !Receive results
 16:  CALL PKMVRCVMSG1
 17:
 18:  CALL PKMVDELPRSPM(“Action ID”=1)
 19:  CALL PKMVSETPRSPM(“Action ID”=1,“Stat Reg”=45,
                        1,61,1,62,2,63,2,64,2,65,2,66)
 20:  CALL PKMVPRSMSG1
 21:  IF R[45:PRSMSG Status]<>0,JMP LBL[900]
 22:
 23:  IF R[61:Num of found]<=0,JMP LBL[920]
 24:
 25:  R[46:MVSFIND1 Status]=0
 26:  END
 27:
 28:  LBL[900:PKMVPRSMSG ERROR]
 29:  R[46:MVSFIND1 Status]=2
 30:  JMP LBL[990]
 31:
 32:  LBL[910:RESPONSE ERROR]
 33:  R[46:MVSFIND1 Status]=3
 34:  JMP LBL[990]
 35:
 36:  LBL[920:NUM OF FND IS 0]
 37:  R[46:MVSFIND1 Status]=4
 38:
 39:  LBL[990]
[End]
```

启动命令的发送
接收响应
解析参数的设定
响应的解析
接收检出结果
解析参数的设定
检出结果的解析
错误处理

在第 4 行向 3rd Party 视觉发送启动命令。此处发送的启动命令在“I.5.4.3 初始化程序”中进行设定。

在第 7 行接收对启动命令的响应。PKMVRCVMSG 接收每次执行时通过换行字符进行分割的 1 个信息。3rd Party 视觉即使在发送响应和检出结果两方面的信息时，在此处理中也只能接收响应。
在第 10 行设定响应的解析处理参数。参数的详细内容，请参照“I.6.5.1 PKMVSETPRSPM.PC”。
在第 11 行对响应进行解析，存储到 SR［6］中。
在第 13 行判定响应是否正常。请事先向 SR［7］中输入 3rd Party 视觉正常接收启动命令时的响应。

在第 16 行接收检出结果。
在第 19 行对检出结果的解析处理进行参数设定。

在第 20 行解析检出结果，存入在 PKMVSETPRSPM 中设定的各数值寄存器中。

在第 28~37 行对各 KAREL 程序中发生错误时进行跳过的水平进行了定义。进行错误处理时在此处进行描述。

注释
有的 3rd Party 视觉通过其他命令执行拍摄、检出和检出结果的发送。此时请添加发送各个命令的处理。

I.5.4.5 锁存程序

对进行锁存的程序示例进行说明。

LATCH

```
  1:  ! Detect signal input
  2:  WAIT DI[R[40]]=ON OR R[47:MVSACTION1 End]=1
  3:  WAIT DI[R[40]]=OFF OR R[47:MVSACTION1 End]=1
  4:
  5:  !Acquire encoder count
  6:  IF (R[47:MVSACTION1 End]=0) THEN
  7:  LINECOUNT[1] R[60]
  8:  R[48:LATCH Success]=1
  9:  ENDIF
[End]
```

检测出信号的输入

编码数的取得

从 3rd Party 视觉输入信号时，将编码数存储到数值寄存器。

在第 2、3 行检出编码器取得用信号的输入之前进行待机。请事先在 R［40］中输入使用的编码器取得用 DI/RI 编号。
此外，在编码器取得用 DI/RI 编号中使用 RI 时，请将此待机命令的条件更改为 RI。
因为在无信号输入的情况下结束了触发程序，因此表示触发程序结束的 R［47］为 1 时也结束待机。

在第 7 行将编码数存储在 R［60］。
为了表示在第 8 行锁存成功，在 R［48］中存入 1。

I.5.4.6 坐标转换参数计算程序

对计算 3rd Party 视觉的坐标转换参数的程序示例进行说明。

CALC_EXTPM

```
   1:  CALL PKWCEND
   2:
   3:  CALL PKCSGETID('CONV1',“CStn ID Reg”=11)
   4:
   5:  PAUSE
   6:  !Set x0,y0,x1,y1 and resume program
   7:  CALL PKMVSCLPUTP1(“Action ID”=1,“CStn ID”=R[11:CONV1 ID],“X0”=R[51:x0],“Y0”
=R[52:y0])
   8:
   9:  CALL PKMVSCLPUTP2(“Action ID”=1,“CStn ID”=R[11:CONV1 ID],“X1”=R[53:x1],“Y1”
=R[54:y1])
  10:
  11:  PAUSE
  12:  !Move conveyor
  13:  !Set x2,y2 and resume program
  14:  CALL PKMVSCLPUTP3(“Action ID”=1,“CStn ID”=R[11:CONV1 ID],“X2”
=R[56:y2])
  15:
  16:  PAUSE
  17:  !Touchup target1 and resume program
  18:  CALL PKMVSCLSETP1(“Action ID”=1,“CStn ID”=R[11:CONV1 ID])
  19:
  20:  PAUSE
  21:  !Touchup target2 and resume program
  22:  CALL PKMVSCLSETP2(“Action ID”=1,“CStn ID”=R[11:CONV1 ID])
  23:
  24:  CALL PKMVSCLCALPM(“Action ID”=1)
  [End]
```

指定检出位置

指定移动后的位置

碰触对象 1

碰触对象 2

计算参数

计算 3rd Party 视觉的坐标转换参数。坐标转换参数被用于将 3rd Party 视觉的检出位置转换成在追踪坐标系中的位置。在此程序中，设想手动将对象的检出位置存储到数值寄存器中。

在第 1 行终止传感器，以免示教过程中触发被执行。

在第 7、9 行将对象 1 的检出位置 P0（x0,y0）和对象 2 的检出位置 P1（x1,y1）传给程序。在执行该行之前取得检出位置，并存储到数值寄存器中。
之后，在对象不出 3rd Party 视觉的视野的范围内移动传送带，将移动后的对象 2 的检出位置 P2（x2,y2）存储到数值寄存器中，并执行第 13 行。

将传送带移动到可以碰触对象的位置，在碰触对象 1 的状态下执行第 17 行，在碰触对象 2 的状态下执行第 21 行，对对象在追踪坐标系中的位置进行示教。

根据第 23 行中指定的位置信息计算 3rd Party 视觉的坐标转换参数。

I.5.4.7 基准位置设定程序

对进行基准位置设定的 TP 程序示例进行说明。

SET_REFPOS

```
 1:  CALL PKWCEND
 2:
 3:  CALL MVSINIT1
 4:
 5:  CALL MVSFIND1
 6:  IF R[46:MVSFIND1 Status]<>0,JMP LBL[900]
 7:  IF R[61:Num of found]<=0,JMP LBL[900]
 8:
 9:  LINECOUNT[1] R[60]
10:  SETTRIG LNSCH[1] R[60]
11:
12:  !Store found position in PR
13:  PR[27,1:Found Pos]=R[63:x]
14:  PR[27,2:Found Pos]=R[64:y]
15:  PR[27,3:Found Pos]=0
16:  PR[27,4:Found Pos]=0
17:  PR[27,5:Found Pos]=0
18:  PR[27,6:Found Pos]=R[65:r]
19:
20:  !Conv fnd pos into tracking frm
21:  CALL PKMVCNVPX2MM(“Action ID”=1,“Fnd Pos PR”=27,“Cnv Pos PR”=26)
22:
23:  END
24:  LBL[900]
25:  UALM[1]
[End]
```

初始化

取得检出结果

触发设置

存储到位置寄存器

基准位置的坐标转换

从 3rd Party 视觉中取得检出位置，进行坐标转换，作为基准位置存储到位置寄存器中。
应用此程序后，通过进行拾取动作的程序进行机器人的位置示教。

在第 1 行终止传感器，以免示教过程中触发被执行。

在第 3 行调出对与 3rd Party 视觉的连接等进行初始化的 TP 程序。详细内容请参照前面所述的“I.5.4.3 初始化程序”。

在第 5 行调出从 3rd Party 视觉取得检出结果的 TP 程序。详细内容请参照“I.5.4.4 检出程序”。
在第 7 行检出数为 0 时跳至第 24 行，发生用户报警。

在第 9、10 行取得编码数，为传送带追踪参数设定表设定触发。使用 1 号以外的编码器编号、1 号以外的传送带追踪参数设定表编号时，请更改成使用此参数的编号。

在第 13~18 行将取得的结果存储到位置寄存器中。
在第 21 行转换检出位置的坐标，作为基准位置存储到 PR［26］。

I.5.5 供给传送带传感器的设定

设定供给传送带的传感器。

1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
2 从菜单中选择［设置］→［*i*RPickTool］。
3 选择［SENS1］。

4 设定以下项目。
［本传感器］ ： 勾选［有效］
［控制装置］ ：［此控制装置］
［触发条件］ ：［距离］
［触发］ ：［程序的执行］

如果更改［触发条件］和［触发］，设定项目就会发生变化。

5 设定以下项目。
［触发距离］ ：输入传送带每前进多少 mm 执行触发程序
［编码器取得用 DI/RI 编号］ ：对输入编码器取得用信号的 DI 编号或 RI 编号进行输入
［程序名］ ：［MVSACTION1］

注意
该项目设定后，传感器变成已示教的状态，但设定尚未完成。在后面所述的“I.5.7 坐标转换参数计算”和“I.5.8 供给传送带的基准位置设定、机器人的位置示教”结束之前，请不要启动传感器。

I.5.6 位置寄存器操作的设定

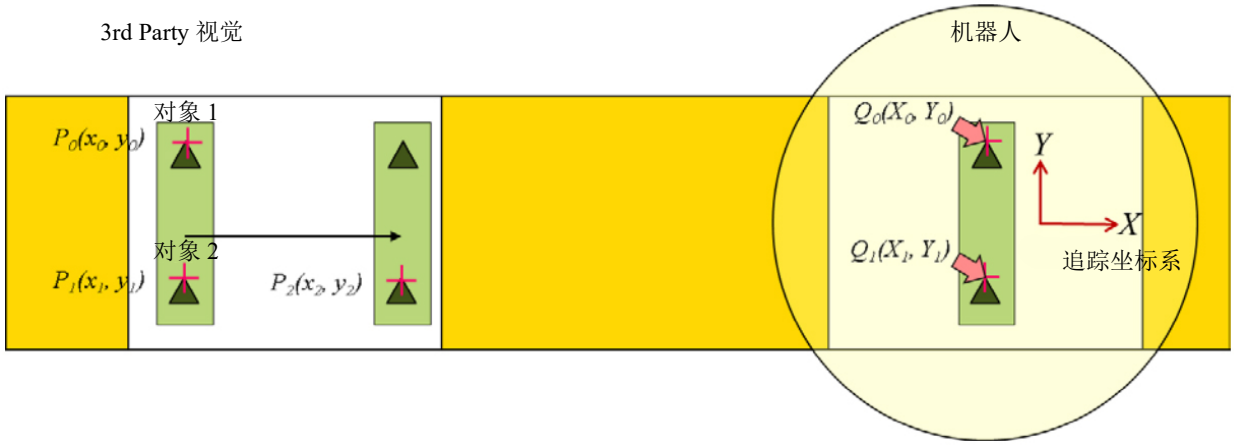
1 在机器人控制装置的示教器上按下［MENU（菜单）］键。
2 从菜单中选择［系统］→［系统设定］。
3 将［无动作组位置寄存器操作］设为有效。

注意
［无动作组位置寄存器操作］有效时，可以从未设定动作组的 TP 程序更改位置寄存器的值。为了在触发程序中进行位置寄存器操作，需要此设定。从未设定触发程序以外的动作组的 TP 程序更改位置寄存器的值时，请确认该变更不会对机器人的动作产生影响。

I.5.7 坐标转换参数计算

计算用于将 3rd Party 视觉的检出位置转换为追踪坐标系上的位置的参数。请根据以下步骤执行坐标转换参数计算程序。坐标转换参数计算程序请参照“I.5.4.6 坐标转换参数计算程序”。

1 切换到碰触销顶端设定的工具坐标系。
2 固定 2 个对象，使在 3rd Party 视觉的视野内位置关系与传送带的传送方向垂直。
3 根据以下步骤，执行坐标转换参数计算程序。
  1 记录 3rd Party 视觉中的对象 1 的检出位置 P0（x0,y0）和对象 2 的检出位置 P1（x1,y1）。
  2 为 PKMVSCLPUTP1（检出位置手动输入 KAREL 程序）的参数指定 P0 后进行执行。
  3 不移动传送带，为 PKMVSCLPUTP2 的参数指定 P1 后进行执行。
  4 在对象不出 3rd Party 视觉的视野的范围内使传送带朝传送方向移动，记录移动后的对象 2 的检出位置 P2（x2,y2）。
  5 为 PKMVSCLPUTP3 的参数指定 P2 后进行执行。
  6 将传送带移动到可以碰触对象的位置，碰触对象 1 的检出位置 Q0（X0, Y0）。
  7 执行 PKMVSCLSETP1（机器人位置取得用 KAREL 程序）。
  8 不移动传送带，碰触对象 2 的检出位置 Q1（X1, Y1）。
  9 执行 PKMVSCLSETP2。
  10 执行 PKMVSCLCALPM。



注释
1 为了提高精度，请使用高度与工件拾取位置较近的对象来计算坐标转换参数。
2 机器人碰触的位置偏离 3rd Party 视觉中的检出位置时，未正确计算坐标转换参数。请使用容易判断检出位置的对象。

I.5.8 供给传送带的基准位置设定、机器人的位置示教

设定基准位置，示教机器人位置。基准位置设定程序请参照“I.5.4.7 基准位置设定程序”。

1 在 3rd Party 视觉的视野内固定工件。

2 执行基准位置设定程序。进行以下设定。
- 从 3rd Party 视觉中取得检出结果，存储到位置寄存器中。
- 使用 PKMVCNVPX2MM 对检出结果进行坐标转换。
- 应用触发设定，为传送带追踪参数设定表设置触发值。

3 运行传送带，工件到达机器人动作范围内后关停。
4 将机器人点动移动到工件拾取位置。
5 显示进行拾取动作的程序，将光标对准示教拾取位置行。
6 ［SHIFT］键的同时按下 F5［TOUCHUP］。

I.6 KAREL 程序

对 *i*RPickTool 3rd Party 接口加载项中使用的 KAREL 程序进行说明。

注释
将 iRPickTool 3rd Party 接口加载项的 KAREL 程序中发生的本功能特有的报警作为“IRPK-201”来显示。此报警的原因被显示在报警信息文本中。请参照发生报警的 KAREL 程序和报警信息文本来修正问题。

I.6.1 通信相关程序

I.6.1.1 PKMVCONNECT.PC

使用被指定的客户标签，与 3rd Party 视觉进行连接。执行此程序时因为会暂时切断通信，再次连接时也不需要事先执行 PKMVDISCO。

参数 1：
指定进行 3rd Party 视觉设定的客户标签编号。

使用例：

```
CALL PKMVCONNECT(“Client ID”=2)
```

I.6.1.2 PKMVDISCO.PC

使用指定的客户标签来切断与连接的 3rd Party 视觉的通信。通过未连接的客户标签执行此 KAREL 程序时会发生报警。

参数 1：
指定进行 3rd Party 视觉设定的客户标签编号。

使用例：

```
CALL PKMVDISCO(“Client ID”=2)
```

I.6.2 初始化相关程序

I.6.2.1 PKMVINIACT.PC

对指定的触发进行初始设定。对与 3rd Party 视觉进行通信，取得检出结果的程序进行首次执行。以后的 KAREL 程序将把此程序中设定的触发 ID 指定为参数，并进行执行。

参数 1：
在 1~4 中指定设定的触发的 ID。为每个通信的 3rd Party 视觉指定不同的触发 ID。

参数 2：
指定进行 3rd Party 视觉设定的客户标签编号。不能指定其他触发 ID 中已经设定的客户 ID。

参数 3：
指定分割输入信息的各数据的字符。

参数 4：
指定输入信息的末尾附带的换行字符。
换行字符从以下各项中指定。
“LF”
“CR”
“CRLF”
“”（无）

参数 5:
指定输入信息的最大数据数。发送多个输入信息时，指定最大的数据数。在 PKMVSETSNDPM 中设定的数据数超过此数时，将发生“IRPK-201 max_data_in is insufficient”警告。可设定的最大数据数为 500。
例如，设定以下输入信息时，则指定为 9。
“SUCCESS,FIND,0,0,0,0,0,0,0<LF>”

参数 6：
指定分割输出信息的各数据的字符。请不要指定数据中包含的字符。

参数 7：
指定输出信息的末尾附带的换行字符。
换行字符从以下各项中指定。
“LF”
“CR”
“CRLF”

参数 8：
指定输出信息的最大数据数。接收多个输出信息时，指定最大的数据数。执行 PKMVRCVMSG 时接收的数据数超过此数时，将发生“IRPK-201 max_data_out is insufficient”警告。可设定的最大数据数为 500。

使用例：

```
CALL PKMVINIACT(“Action ID”=1,“Client ID”=2,“In Msg Delimiter”=',',“In Msg Newline char”='LF',“In Msg Max num data”=1,“Out Msg Delimiter”=',',“Out Msg Newline char”='LF',“Out Msg Max num data”=11)
```

I.6.3 输入信息发送相关程序

I.6.3.1 PKMVSETSNDPM.PC

设定 PKMVSENDMSG 中发送的输入信息的数据。设定的数据多于指定数据的参数数 28 时，通过继续执行该程序，将接着上次设定的数据进行设定。设定的数据多于 PKMVINIACT 中设定的输入信息的最大数据数时，将发生“IRPK-201 max_data_in is insufficient”警告。在此程序中设定的数据即使执行 PKMVSENDMSG，也将被删除。只发送相同输入信息时，此程序在最初只执行 1 次是没有问题的。设定不同的输入信息时，在执行 PKMVDELSNDPM 删除设定的输入信息之后，请执行此程序。

参数 1：
指定触发 ID。

参数 2：
指定存储 PKMVSENDMSG 状态的数值寄存器编号。

参数 3~30（参数 4~30 可省略）：
指定输入信息的数据。可通过整数、实数、字符串进行指定。指定实数时，发送至小数点第 3 位。指定字符串时，可设定的最大字符串长为 34。

使用例：
执行以下程序时，“RUNFIND”和“1.234”被设定为输入信息的数据。

```
CALL PKMVSETSNDPM(“Action ID”=1,“Stat Reg”=10,'RUNFIND',1.2345)
```

I.6.3.2 PKMVDELSNDPM.PC

删除通过 PKMVSENDMSG 发送的输入信息。设定不同的输入信息时，请在 PKMVSETSNDPM 之前执行此程序。

参数 1：
指定触发 ID。

使用例：

```
CALL PKMVDELSNDPM(“Action ID”=1)
```

I.6.3.3 PKMVSENDMSG1~4.PC

将输入信息发送到 3rd Party 视觉。程序名末尾的数字与触发 ID 是对应的。
用分隔符将 PKMVSETSNDPM 中设定的各数据进行分割，在末尾加上换行字符后进行发送。分隔符和换行字符均在 PKMVINIACT 中设定。

执行后，作为状态，以下任一值将被存储在数值寄存器中。
0 ：成功
12 ：通信错误

无参数。

使用例：

```
CALL PKMVSENDMSG1
```

I.6.4 输出信息接收相关程序

I.6.4.1 PKMVSETRCVPM.PC

设定 PKMVRCVMSG 的超时时间。在此程序中设定的数据即使执行 PKMVRCVMSG，也将被删除。不更改超时时间时，此程序在最初只执行 1 次是没有问题的。

参数 1:
指定触发 ID。

参数 2:
指定接收输出信息的超时时间。单位为［ms］。

参数 3:
指定存储 PKMVRCVMSG 状态的数值寄存器编号。

使用例：

```
CALL PKMVSETRCVPM(“Action ID”=1,“Timeout (ms)”=5000,“Stat Reg”=11)
```

I.6.4.2 PKMVRCVMSG1~4.PC

接收 3rd Party 视觉发送的输出信息。程序名末尾的数字与触发 ID 是对应的。

如果执行此程序，将接收通过换行字符分割的 1 个输出信息。换行字符在 PKMVINIACT 中进行设定。执行时上次接收的信息将被删除。为了将接收的输出信息保存到数值寄存器中，请执行 PKMVPRSMSG。接收的数据多于 PKMVINIACT 中设定的输出信息的最大数据数时，将发生“IRPK-201 max_data_out is insufficient”警告。执行此程序后，在超时时间内无法接收输出信息时，将发生“IRPK-201 Receive message timeout”警告。接收到不正确的换行字符或 255 个字符以上的数据时，将发生“IRPK-201 Receive data is invalid”警告。

执行后，作为状态，将存储以下任一值。
0 ：成功
11 ：发生超时
12 ：通信错误

无参数。

使用例：

```
CALL PKMVRCVMSG1
```

I.6.5 输出信息解析相关程序

I.6.5.1 PKMVSETPRSPM.PC

设定在 PKMVPRSMSG 中解析的参数。为参数设定接收的输出信息的数据形式和存储的数值寄存器编号的组。设定的数据多于指定数据的参数数 14 时，通过继续执行此程序，将接着上次设定的数据进行设定。可设定的数据的最大数为 500（可设定的数据的最大数与 PKMVINIACT 中设定的输出信息的最大数据数不同。）。在此程序中设定的数据即使执行 PKMVPRSMSG，也将被删除。设定不同的参数时，在执行 PKMVDELPRSPM 删除设定的参数之后，请执行此程序。

参数 1：
指定触发 ID。

参数 2：
指定存储 PKMVPRSMSG 状态的数值寄存器编号。

参数 3,5,7, … ,29 （可省略）：
指定数据的形式。从以下值中指定。该参数和下面指定的数值寄存器编号是相对于 1 个数据的参数组。
1：整数
2：实数
3：字符串

参数 4,6,8, … ,30 （可省略）：
指定存储数据的数值寄存器编号。前面参数指定的数据形式为整数或实数时，将被存储于数值寄存器中，前面参数指定的数据形式为字符串时，被存储在字符串寄存器中。

使用例：
设定将输出信息“SUCCESS（字符串）,模型 ID（整数）,X（实数）<LF>”的数据存储到 SR［5］、R［10］、R［11］中时的参数。

```
CALL PKMVSETPRSPM(“Action ID”=1,“Stat Reg”=12,3,5,1,10,2,11)
```

I.6.5.2 PKMVDELPRSPM.PC

删除在 PKMVSETPRSPM 中设定的参数。对不同接收信息设定参数时，请在 PKMVSETPRSPM 之前执行此程序。

参数 1:
指定触发 ID。

使用例：

```
CALL PKMVDELPRSPM(“Action ID”=1)
```

I.6.5.3 PKMVPRSMSG1~4.PC

对 PKMVRCVMSG 中接收的输出信息进行解析，并存储到数值寄存器中。程序名末尾的数字与触发 ID 是对应的。将每次执行时 PKMVRCVMSG 中接收的输出信息的数据从头按顺序取出，按照 PKMVSETPRSPM 中设定的参数的顺序转换形式，并存储到数值寄存器中。设定的数据形式和解析的数据形式不同时，将发生“IRPK-201 Data type is invalid”警告。解析空数据时，将发生“IRPK-201 Receive data is invalid”警告。

注意
进行解析的输出信息的数据数比 PKMVSETPRSPM 中设定的参数数少时，下面在执行 PKMVRCVMSG 后再执行 PKMVPRSMSG 时，将接着上次继续使用参数进行解析。因此，因发生错误解析中途终止时等，有时会因解析使用的参数偏离而发生错误。基本上在执行 PKMVPRSMSG 之前，执行 PKMVDELPRSPM 和 PKMVSETPRSPM，进行参数的再次设定。

执行后，作为状态，将存储以下任一值。
0 ： 成功
1 ： 进行解析的输出信息的数据数比 PKMVSETPRSPM 中设定的参数数多
2 ： 进行解析的输出信息的数据数比 PKMVSETPRSPM 中设定的参数数少
3 ： 进行解析的输出信息的数据 1 个也不存在
4 ： 失败

无参数。

使用例：

```
CALL PKMVPRSMSG1
```

I.6.6 坐标转换参数计算相关程序

I.6.6.1 PKMVSCLPUTP1.PC

输入第 1 个对象的检出位置。

参数 1：
指定触发 ID。

参数 2：
指定传送带工作站的 ID。

参数 3：
指定 x0 的值。

参数 4：
指定 y0 的值。

使用例：

```
CALL PKMVSCLPUTP1(“Action ID”=1,“CStn ID”=R[6:CSTN1 ID], “X0”=R[51:x0],“Y0”=R[52:y0])
```

I.6.6.2 PKMVSCLPUTP2.PC

输入第 2 个对象的检出位置。执行 PKMVSCLPUTP1 后不移动传送带，执行此程序。

参数 1：
指定触发 ID。

参数 2：
指定传送带工作站的 ID。

参数 3：
指定 x1 的值。

参数 4：
指定 y1 的值。

使用例：

```
CALL PKMVSCLPUTP2(“Action ID”=1,“CStn ID”=R[6:CSTN1 ID], “X1”=R[53:x1],“Y1”=R[54:y1])
```

I.6.6.3 PKMVSCLPUTP3.PC

输入传送带移动后的第 2 个对象的检出位置。执行 PKMVSCLPUTP2 后，在对象不出 3rd Party 视觉的视野的范围内移动传送带，将第 2 个对象的检出位置指定为参数，并执行此程序。

参数 1：
指定触发 ID。

参数 2：
指定传送带工作站的 ID。

参数 3：
指定 x2 的值。

参数 4：
指定 y2 的值。

使用例：

```
CALL PKMVSCLPUTP3(“Action ID”=1,“CStn ID”=R[6:CSTN1 ID], “X2”=R[55:x2],“Y2”=R[56:y2])
```

I.6.6.4 PKMVSCLSETP1.PC

对第 1 个对象的追踪坐标系中的位置进行示教。在碰触第 1 个对象的检出位置的状态下执行。

参数 1：
指定触发 ID。

参数 2：
指定传送带工作站的 ID。

使用例：

```
CALL PKMVSCLSETP1(“Action ID”=1,“CStn ID”=R[6:CSTN1 ID])
```

I.6.6.5 PKMVSCLSETP2PC

对第 2 个对象的追踪坐标系中的位置进行示教。应用 PKMVSCLSETP1 后不移动传送带，在碰触第 2 个对象的检出位置的状态下进行执行。

参数 1：
指定触发 ID。

参数 2：
指定传送带工作站的 ID。

使用例：

```
CALL PKMVSCLSETP2(“Action ID”=1,“CStn ID”=R[6:CSTN1 ID])
```

I.6.6.6 PKMVSCLCALPM.PC

计算 3rd Party 视觉的坐标转换参数。将坐标转换参数计算相关的程序全部执行后，执行此程序。

参数 1：
指定触发 ID。

使用例：

```
CALL PKMVSCLCALPM(“Action ID”=1)
```

I.6.7 检出位置转换相关程序

I.6.7.1 PKMVCNVPX2MM.PC

将 3rd Party 视觉的检出位置转换为追踪坐标系上的位置。在完成坐标转换参数计算后可执行。

参数 1：
指定触发 ID。

参数 2：
指定存储了 3rd Party 视觉中的检出位置的位置寄存器编号。

参数 3：
指定存储坐标转换后位置的位置寄存器编号。指定与参数 2 相同的值时，指定的检出位置被坐标转换后的值覆盖。

使用例：

```
CALL PKMVCNVPX2MM(“Action ID”=1,“Fnd Pos PR”=2,“Cnv Pos PR”=1)
```

I.6.7.2 PKMVCALOFS2D.PC

根据检出位置和基准位置计算偏移量。指定使用 PKMVCNVPX2MM 进行坐标转换的位置。不能计算 *i*RPickTool 3rd Party 接口加载项以外的偏移。

参数 1：
指定存储了检出位置的位置寄存器编号。

参数 2：
指定存储了基准位置的位置寄存器编号。

参数 3：
指定存储偏移的位置寄存器编号。

使用例：

```
CALL PKMVCALOFS2D(“Fnd Pos PR”=2,“Ref Pos PR”=1,“Offset PR”=3)
```

I.6.7.3 PKMVSETVR.PC

将指定的值存储到视觉寄存器。用于使用 PKCSPUTQUE 将工件放入队列中。

参数 1：
指定存储值的视觉寄存器编号。

参数 2：
指定存储了检出位置的位置寄存器编号。

参数 3：
指定存储了偏移的位置寄存器编号。

参数 4：
指定存储了模型 ID 的数值寄存器编号。

参数 5：
指定存储了编码数的数值寄存器序号。

参数 6 ~15（可省略）：
指定存储了测量值 1~10 的数值寄存器编号。0.0 被存储到省略的测量值中。

使用例：
为视觉寄存器 1 设定以下值。
检出位置: PR［2］
偏移 : PR［3］
模型 ID : R［31］
编码数 : R［28］
测量值 1: R［38］

```
CALL PKMVSETVR(“VR”=1,“Fnd Pos PR”=2,“Offset PR”=3,“Model ID Reg”=31,“Enc Count Reg”
=28,38)
```

I.6.8 其他程序

I.6.8.1 PKMVWAITTASK.PC

在指定的任务结束之前进行待机。用于等待后台执行的程序结束。不存在指定的任务时，此程序立即结束。

参数 1：
等待结束的任务名称。

使用例：

```
CALL PKMVWAITTASK(“Task Name”=' LATCH')
```

I.7 对应用途的设定例子

I.7.1 一次拍摄检测出多个工件

如以下消息所示，针对一次检测发送最多 2 个检测结果的 3rd Party 视觉系统启动步骤进行说明。

表 I.7.1 消息格式

输入消息 **(控制装置→3rd Party 视觉)**	**RUNFIND(字符串)<CR>**
对输入消息的响应 **(3rd Party 视觉→控制装置)**	**正常时 ：OK(字符串)<CR>** **异常时 ：ER(字符串)<CR>**
输出消息 **(3rd Party 视觉→控制装置)**	**检测数(整数)，模型 ID1(整数)，X1(实数)，Y1(实数)，R1(实数)，测量值 1.1(实数)，模型 ID2(整数)，X2(实数)，Y2(实数)，R2(实数)，测量值 2.1(实数)<CR>**

在该消息格式中，即使检测数为 1 个时，不要省略第 2 个检测结果数据，需要发送输出消息。因此，当检测数为 1 个时，需要进行如下处理。

- 在“检测数”中存储 1
- 在“模型 ID2”以后的数据中存储 0

整体的启动步骤和“I.5 启动步骤”相同，但动作程序和检测程序请分别更改成以下。

动作程序

MVSACTION1

```
  1:  R[47:MVSACTION1 End]=0      ;
  2:   ;
  3:  !Use ACCUTRIG ;
  4:  ACCUTRIG LNSCH[1] ;
  5:   ;
  6:  R[48:LATCH Success]=0      ;
  7:  RUN LATCH ;
  8:   ;
  9:  CALL MVSFIND1      ;
 10:  IF R[46:MVSFIND1 Status]<>0,JMP LBL[990] ;
 11:   ;
 12:  IF R[48:LATCH Success]=0,JMP LBL[900] ;
 13:   ;
 14:  !Store found position1 in PR ;
 15:  PR[27,1:Found Pos]=R[63:x1]      ;
 16:  PR[27,2:Found Pos]=R[64:y1]      ;
 17:  PR[27,3:Found Pos]=0      ;
 18:  PR[27,4:Found Pos]=0      ;
 19:  PR[27,5:Found Pos]=0      ;
 20:  PR[27,6:Found Pos]=R[65:r1]      ;
 21:   ;
 22:  !Conv fnd pos into tracking frm ;
 23:  CALL PKMVCNVPX2MM(“Action ID”=1,“Fnd Pos PR”=27,“Cnv Pos PR”=27) ;
 24:  !Calculate offset ;
 25:  CALL PKMVCALOFS2D(“Fnd Pos PR”=27,“Ref Pos PR”=26,“Offset PR”=28) ;
 26:   ;
 27:  !Put result into queue ;
 28:  CALL PKMVSETVR(“VR”=5,“Fnd Pos PR”=27,“Offset PR”=28,“Model ID Reg”=62,“Enc
Count Reg”=60,66) ;
 29:  CALL PKCSPUTQUE(“CStn ID”=R[11:CONV1 ID],“Offset VR”=5,“OverlapStat Reg”=49) ;
 30:   ;
 31:  IF R[61:Num of found]<2,JMP LBL[990] ;
```

将检测结果 1 投入队列

检测数为 1 个时结束

```
  32:   !Store found position2 in PR ;
  33:   PR[27,1:Found Pos]=R[73:x2]      ;
  34:   PR[27,2:Found Pos]=R[74:y2]      ;
  35:   PR[27,3:Found Pos]=0      ;
  36:   PR[27,4:Found Pos]=0      ;
  37:   PR[27,5:Found Pos]=0      ;
  38:   PR[27,6:Found Pos]=R[75:r2]      ;
  39:    ;
  40:   !Conv fnd pos into tracking frm ;
  41:   CALL PKMVCNVPX2MM(“Action ID”=1,“Fnd Pos PR”=27,“Cnv Pos PR”=27) ;
  42:   !Calculate offset ;
  43:   CALL PKMVCALOFS2D(“Fnd Pos PR”=27,“Ref Pos PR”=26,“Offset PR”=28) ;
  44:    ;
  45:   !Put result into queue ;
  46:   CALL PKMVSETVR(“VR”=5,“Fnd Pos PR”=27,“Offset PR”=28,“Model ID Reg”=62,“Enc
Count Reg”=60,66) ;
  47:   CALL PKCSPUTQUE(“CStn ID”=R[11:CONV1 ID],“Offset VR”=5,“OverlapStat Reg”=49) ;
  48:   JMP LBL[990] ;
  49:    ;
  50:   LBL[900:LATCH ERORR] ;
  51:   R[46:MVSFIND1 Status]=5      ;
  52:    ;
  53:   LBL[990] ;
  54:   R[47:MVSACTION1 End]=1      ;
  55:   CALL PKMVWAITTASK(“Task Name”='LATCH') ;
```

将检测结果 2 投入队列

检测程序

MVSFIND1

```
   1:   R[46:MVSFIND1 Status]=1      ;
   2:    ;
   3:   !Send run command ;
   4:   CALL PKMVSENDMSG1      ;
   5:    ;
   6:   !Receive response ;
   7:   CALL PKMVRCVMSG1      ;
   8:    ;
   9:   CALL PKMVDELPRSPM(“Action ID”=1) ;
  10:   CALL PKMVSETPRSPM(“Action ID”=1,“Stat Reg”=45,3,6) ;
  11:   CALL PKMVPRSMSG1      ;
  12:   IF R[45:PRSMSG Status]<>0,JMP LBL[900] ;
  13:   IF SR[6]<>SR[7],JMP LBL[910] ;
  14:    ;
  15:   !Receive results ;
  16:   CALL PKMVRCVMSG1      ;
  17:    ;
  18:   !Parse found position1 ;
  19:   CALL PKMVDELPRSPM(“Action ID”=1) ;
  20:   CALL PKMVSETPRSPM(“Action ID”=1,“Stat Reg”=45,1,61,1,62,2,63,2,64,2,65,2,66) ;
  21:   CALL PKMVPRSMSG1      ;
  22:    ;
  23:   !Parse found position2 ;
  24:   CALL PKMVDELPRSPM(“Action ID”=1) ;
  25:   CALL PKMVSETPRSPM(“Action ID”=1,“Stat Reg”=45,1,72,2,73,2,74,2,75,2,76) ;
  26:   CALL PKMVPRSMSG1      ;
  27:   IF R[45:PRSMSG Status]<>0,JMP LBL[900] ;
  28:    ;
  29:   IF R[61:Num of found]<=0,JMP LBL[920] ;
  30:    ;
```

解析第 1 个检测结果

解析第 2 个检测结果

```
31:   R[46:MVSFIND1 Status]=0      ;
32:   END ;
33:    ;
34:   LBL[900:PKMVPRSMSG ERROR] ;
35:   R[46:MVSFIND1 Status]=2      ;
36:   JMP LBL[990] ;
37:    ;
38:   LBL[910:RESPONSE ERROR] ;
39:   R[46:MVSFIND1 Status]=3      ;
40:   JMP LBL[990] ;
41:    ;
42:   LBL[920:NUM OF FND IS 0] ;
43:   R[46:MVSFIND1 Status]=4      ;
44:    ;
45:   LBL[990] ;
```

I.8 常见问题处理方法

通过使用 *i*RPickTool 3rd Party 接口项加载的系统，对较容易遇到的问题及其应对方法进行说明。

I.8.1 不能与 3rd Party 视觉进行连接

原因 1： 客户标签的设定错误。
对策 1： 参照“I.5.2.2 客户标签的设定”，确认为客户标签设定了正确的 3rd Party 视觉的 IP 地址和端口号。此外，确认客户标签的［当前状态］为［定义］。
原因 2： 未进行以太网端口的设定。
对策 2： 确认进行了与 3rd Party 视觉连接的以太网端口的 TCP/IP 的设定。

I.8.2 在 3rd Party 视觉中未执行拍摄和检出

原因 1： 输入信息（启动命令）错误。
对策 1： 确认 PKMVSETSENDPM 中指定的输入信息与 3rd Party 视觉的启动命令相同。此外，确认 3rd Party 视觉可以识别 PKMVINIACT 中指定的输入信息的分隔符和换行字符。
原因 2： 3rd Party 视觉未处于可接收命令的状态。
对策 2： 3rd Party 视觉未变成可接收命令的状态。设定方法请参照各视觉厂商的操作说明书。

I.8.3 PKMVRCVMSG 中发生超时

原因 1： 输出信息的换行字符错误。
对策 1: 确认输出信息的换行字符与 PKMVINIACT 中指定的输出信息的换行字符一致。

原因 2： 超时时间过短。
对策 2： 请将超时时间设定得比 3rd Party 视觉的检出时间要长。

原因 3： 用于发送检出结果的输入信息未发送。
对策 3： 有的 3rd Party 视觉将分别通过其他命令执行拍摄、检出和检出结果的发送。请确认可以通过机器人控制装置发送的输入信息发送检出结果。

I.8.4 PKMVPRSMSG 失败

原因 1： 输出信息的解析参数错误。

对策 1： 确认 PKMVSETPRSPM 中设定的参数与 3rd Party 视觉发送的输出信息的格式相同。此外，确认 PKMVINIACT 中设定的输出信息的分隔符正确。

原因 2： 解析参数偏离。
对策 2: 进行解析的输出信息的数据数比 PKMVSETPRSPM 中设定的参数数少时，下面在执行 PKMVRCVMSG 后再执行 PKMVPRSMSG 时，将接着上次继续使用参数进行解析。因此，因发生错误解析中途终止时等，有时会因解析使用的参数偏离而发生错误。基本上在执行 PKMVPRSMSG 之前，执行 PKMVDELPRSPM 和 PKMVSETPRSPM，进行参数的再次设定。

原因 3： 收到的输出信息与设定参数的输出信息不同。
对策 3： 有时 3rd Party 视觉在接收到命令时作为输出信息将返回响应等，机器人控制装置将接收到检出结果以外的输出信息。请确认 3rd Party 视觉发送的输出信息。

I.8.5 锁存失败

原因 1： 编码器取得用 DI/RI 编号错误。
对策 1： 请确认编码器取得用信号的编号和通过锁存程序检测到信号输入的编号正确。

原因 2： 未输出编码器取得用信号。
对策 2： 请确认拍摄时从 3rd Party 视觉输出编码器取得用信号。此外，请确认 3rd Party 视觉与机器人控制装置之间的配线正确。

原因 3： 到编码器取得用信号的下降沿的时间过长。
对策 3： 有时编码器取得用信号的输出时机延迟，或者编码器取得用信号的输出宽度较长时，在锁存程序中检出信号的下降沿之前检出结果的解析结束，将判定锁存失败。对于编码器取得用信号，请使用与拍摄同时输出的信号，并将输出宽度设置得短些。或者，请在判定触发程序的锁存是否成功的命令之前添加待机命令。

I.8.6 抓住位置不正确

原因 1： 坐标转换参数计算不正确。
对策 1： 请参照“I.5.7 坐标转换参数计算”，再次应用坐标转换参数计算。此外，在应用坐标转换参数计算后，请再次应用基准位置设定。

原因 2： 基准位置设定不正确。
对策 2： 请参照“I.5.8 供给传送带的基准位置设定、机器人的位置示教”，再次应用基准位置设定。此外，请确认在基准位置的最初执行 PKWCEND 后传感器结束。

原因 3： 编码器取得用信号的时机不均衡。
对策 3： 编码器取得用信号的输出时机不均衡时，不能取得拍摄时的正确传送带位置。请讨论使用时机上不存在不均衡情况的其他信号。

原因 4： 高精度触发未变为有效。
对策 4： 发送输入信息前，需要执行高精度触发命令，并将高精度触发设置为有效。此外，请确认传感器对象的编码器取得用 DI/RI 编号正确。

原因 5： 跟踪坐标系未能正确设定。
对策 5： 详细内容请参照“6.12.2 追踪坐标系的设定误差的调整”。

索引

＜α＞

αA1000S 脉冲编码器 683

＜数字＞

2 次去取相同的工件 652
3D 视觉传感器 618
3rd Party 接口加载项 734
3rd Party 视觉的连接 737
3rd Party 视觉的连接和设定 737
3rd Party 视觉的设定 738
3rd Party 视觉的制约 735
4D 拾取边界 585
4D 工作单元的删除 586
4D 工作单元的生成 575
4D 工作单元的显示 575
4D 拾取坐标系 586

＜A＞

安全使用须知 s-1
安装 611

＜B＞

本说明书的使用方法 1
必要的结构 683,699
变更 Internet Explorer 的设定 661
变更 Windows 防火墙的设定 667
［标记］＋［放入工件］的情况 320
标准 TP 程序 467,494
标准 TP 程序的详细 469
标准 TP 程序下的机器人的动作 449
标准 TP 程序一览 469
不能与 3rd Party 视觉进行连接 762
不使用托盘的情况 325,334
不同用途的设定例子 519

＜C＞

操作的设定 238,441,493
层操作 296
常见问题处理方法 651,672,762
超出领域时，跳过追踪动作 571
超限次数不断增加 654
初始化程序 745
初始化相关程序 752
触发程序 743
触发传感器的输入确认 39,162
传感器任务的自定义 720
传感器设定 94,204,312,418,614,640
传感器位置设定 97,207,324
传感器详细设定 312
传送带—传送带(相同，下游)的构成的样本程序 431
传送带—传送带(相同，最上游)的构成的样本程序 428
传送带—传送带的构成的样本程序 383,432
传送带工作站 400
传送带工作站的操作 442
传送带工作站的操作的详细设定 443
传送带工作站设定 124,233,341,419,649
传送带—固定工作站的构成的样本程序 387
传送带—缓冲区传送带的构成的样本程序 390
传送带设定 84,193,300,412,640
传送带一端的工件补偿偏移 653
从 2 个以上的供给传送带上拾取 538
从箱子一端依次放置工件 531
伺服传送带设定 594
伺服传送带用 TP 程序的生成 597
伺服传送带用的动态误差的微调整参数 601
伺服传送带用功能 601
伺服传送带专用 KAREL 程序 602
伺服传送带追踪拾取功能 587
伺服传送带追踪制约事项 609

＜D＞

［DI］/［RI］＋［通过视觉检出］ 397
［DI］+［程序的执行］ 318
［DI］＋［放入工件］的情况 316
［DI］＋［通过视觉检出］的情况 317,331
打开 *i*RPickTool 的画面 668
打开 *i*RVision 的画面 669
带队列管理的追踪 6
单元 295
等待伺服传送带停止的功能 602
点阵板夹具 278
电脑与浏览器 658
对象数量设定 289
对应用途的设定例子 760
多层托盘中的固定接近高度 514
多视图 610

＜F＞

法兰盘直线指令 405
方案 291
方案的详细设定 546
方案设定 545
负载设定 452
负载设定（动作性能） 68,172
附加轴设定 588

＜G＞

get_ecnt_idx 729
概略图 688,704
概要 4,587,610,618,638,682,720,734
根据传送带上的位置，由不同的机器人拾取 519
根据工件的部分特征，由不同的机器人拾取 523
根据工件种类不同，切换放置位置 533
根据种类，放置于其他的输出传送带 536
工件的跳过频发 651
工件数据 VR 452
工具坐标系 452

工具坐标系的设定.....278
工作单元、机器人设定.....410,640
工作单元设定.....436
工作单元设置.....71,175,288
供给传送带传感器的设定.....94,204,750
供给传送带的基准位置设定、机器人的位置示教...751
供给传送带的设定.....84,193
供给基准位置的设定、机器人的位置示教.....133,244
构造网络时的注意事项.....260
固定工作站.....400
固定工作站的操作.....445
固定工作站的操作的详细设定.....446
固定工作站设定.....353,425,447,650
关于 Microsoft®产品.....2
关于本说明书.....1
关于其它说明书.....2
关于支持范围.....736

<*H*>

［HDI］＋［放入工件］的情况.....319
HDI 设定.....675
缓冲区的设定.....502
缓冲区功能中使用的 KAREL 程序.....503
缓冲区功能中使用的 TP 程序.....504

<*I*>

incre_wid.....730
IP 地址的设定.....260,738
*i*RPickTool Basic（基本功能）启动调试步骤.....27
*i*RPickTool（基本功能+Plug & Play）的启动调试步骤.....151
*i*RPickTool 的 4D 图形显示.....574
*i*RPickTool 的设定.....738

<*J*>

机器构成.....6
机器人程序的创建.....50,739
机器人的程序.....382,428,448
机器人的动态误差的调整.....403,454
机器人的位置示教.....400,433,453
机器人的原位置示教.....258
机器人动作不顺畅.....653
机器人间的通信设定.....265
机器人设定.....291
机器人完全不去拾取工件.....651
机器人系统的制约.....735
机械手工具坐标系的设定.....129,239
基本功能参考.....279
基准单元的设定.....299
基准位置的设定、机器人的位置示教.....133,244,616
基准位置设定.....396,433,453
基准位置设定程序.....749
间隙检查功能.....638
间隙向导.....643
检出程序.....746
检出位置转换相关程序.....758
将工件多层堆起放入箱中.....528
接线图.....676
仅在单元都已满的情况下，取出托盘.....541
进行放置动作的程序.....387,389,394
进行放置动作的程序(缓冲区).....395
进行拾取动作的程序.....385,389,393
进行拾取动作的程序(缓冲区).....395
旧软件与 *i*RPickTool 的 KAREL 对照表.....719
［距离］＋［放入工件］的情况.....313
［距离］+［程序的执行］.....315
［距离］＋［通过视觉检出］的情况.....314
［距离］＋［通过视觉检出］的情况。.....328

<*K*>

KAREL 程序.....354,425,455,560,752
KAREL 函数.....729
可同时执行的传感器的限制.....340
客户标签的设定.....738

<*L*>

LINE_COUNT.....731
利用 PLC 的方案操作.....555
连接方法.....690,707
连接脉冲编码器.....275
连接通信电缆.....658
量增脉冲编码器.....699
另置型接口装置（SDU）.....712

<*M*>

脉冲编码器的连接.....28,152
脉冲编码器的连接与设定.....28,152
脉冲编码器的设定.....29,153
脉冲编码器的设定确认.....33,157
脉冲编码器的设置和设定.....274
每次触发条件和动作组合的必要设定.....324
目标位置设定.....334

<*N*>

内嵌函数.....731

<*P*>

P_CSGETID.....733
P_CVGETID.....733
PK_BF_DROP1.TP.....506
PK_BF_GET_DATA1.TP.....504
PK_BF_OK2DROP1.TP.....509
PK_BF_OK2PICK1.TP.....508
PK_BF_PICK1.TP.....505
PK_BUF_DROP1.TP.....512
PK_BUF_PICK1.TP.....512
PK_CV_DROP11.TP.....482
PK_CV_GET_DROP_DATA11.TP.....476
PK_CV_GET_PICK_DATA11.TP.....475
PK_CV_OK2DROP1.TP.....511
PK_CV_OK2PICK1.TP.....510
PK_CV_PICK11.TP.....479
PK_CV_WAITPOS1.TP.....485
PK_CYCSTOP1.TP.....493
PK_DROP1.TP.....473
PK_FS_DROP11.TP.....488

PK_FS_GET_DROP_DATA11.TP 478
PK_FS_GET_PICK_DATA11.TP 477
PK_FS_INDEX.TP 490
PK_FS_PICK11.TP 486
PK_FS_READY.TP 491
PK_GR_GET_ENDZONE1.TP 492
PK_GR_GET_UTOOLS1.TP 474
PK_INIT1.TP 470
PK_MAIN1.TP: 组 1 的主程序 469
PK_PERCH1.TP 472
PK_PICK1.TP 473
PK_PRGR_CV_PICKDROP1.TP 496
PK_PRGR_CV_PICKVTRAY1.TP 498
PK_PRGR_INIT1.TP 496
PK_PRGR_MAIN1.TP 495
PKABO.PC 357
PKCOPYPR2POS.PC 358
PKCSACKQPRGR.PC 427
PKCSACKQUE.PC 369
PKCSCALWPOS.PC 461
PKCSCLRACK.PC 375
PKCSCLRQUE.PC 367
PKCSFIXAPRZ.PC 516
PKCSGENVRTRY.PC 425
PKCSGETAPPR.PC 461
PKCSGETDIST.PC 371
PKCSGETENC.PC 374
PKCSGETID.PC 366
PKCSGETNPRT.PC 373
PKCSGETOPPR.PC 462
PKCSGETPFRT.PC 372
PKCSGETPMUL.PC 462
PKCSGETPPCHK.PC 462
PKCSGETQCELL.PC 368
PKCSGETQPRGR.PC 426
PKCSGETQUE.PC 367
PKCSGETTIME.PC 370
PKCSGETTRID.PC 374
PKCSGETVR.PC 463
PKCSGETWPOS.PC 463
PKCSPUTQUE.PC 369
PKCSSETDSLIN.PC 373
PKCSTRGPRGR.PC 426
PKCVENBLB.PC 361
PKCVENBSC.PC 361
PKCVGETNAME.PC 362
PKCVSETLBD.PC 359
PKCVSETTRNAM.PC 360
PKFSACKBUF.PC 379
PKFSACKQUE.PC 377
PKFSCLRACK.PC 380
PKFSCLRBUF.PC 377
PKFSCLRQUE.PC 376
PKFSFIXAPRZ.PC 517
PKFSGETAPPR.PC 463
PKFSGETBUF.PC 378
PKFSGETBUFDT.PC 504
PKFSGETBUFPT.PC 504
PKFSGETID.PC 375
PKFSGETIXDLY.PC 464
PKFSGETIXODO.PC 464
PKFSGETIXSIM.PC 464
PKFSGETOPPR.PC 465
PKFSGETPMUL.PC 465
PKFSGETPPCHK.PC 465
PKFSGETQUE.PC 376
PKFSGETRDY.PC 465
PKFSGETRDYDI.PC 466
PKFSGETTRID.PC 379
PKFSGETUF.PC 380
PKFSGETVR.PC 466
PKFSMONIDX.PC 466
PKFSPUTBUF.PC 377
PKFSPUTQUE.PC 375
PKFSSETRDY.PC 467
PKGETVAR.PC 355
PKGRCHKPP.PC 456
PKGRCLOSE.PC 456
PKGRGETDPDLY.PC 457
PKGRGETID.PC 457
PKGRGETPKDLY.PC 457
PKGRGETZNCNT.PC 457
PKGRGETZNUT.PC 458
PKGRINIT.PC 458
PKGROPEN.PC 458
PKLABO.PC 357
PKMVCALOFS2D.PC 758
PKMVCNVPX2MM.PC 758
PKMVCONNECT.PC 752
PKMVDELPRSPM.PC 755
PKMVDELSNDPM.PC 754
PKMVDISCO.PC 752
PKMVINIACT.PC 752
PKMVPRSMSG1~4.PC 756
PKMVPRSMSG 失败 762
PKMVRCVMSG1~4.PC 754
PKMVRCVMSG 中发生超时 762
PKMVSCLCALPM.PC 758
PKMVSCLPUTP1.PC 756
PKMVSCLPUTP2.PC 756
PKMVSCLPUTP3.PC 757
PKMVSCLSETP1.PC 757
PKMVSCLSETP2PC 757
PKMVSENDMSG1~4.PC 754
PKMVSETPRSPM.PC 755
PKMVSETRCVPM.PC 754
PKMVSETSNDPM.PC 753
PKMVSETVR.PC 759
PKMVWAITTASK.PC 759
PKPOSTERR.PC 358
PKPRGRGETEN.PC 357
PKRBGETBUFID.PC 503
PKRBGETCVCNT.PC 459
PKRBGETFSCNT.PC 459

PKRBGETID.PC 358
PKRBGETPRGRT.PC 460
PKRBGETSFLG.PC 359
PKRCACTIVE.PC 562
PKRCCREATE.PC 560
PKRCDELETE.PC 560
PKRCEXPORT.PC 561
PKRCIMPORT.PC 562
PKRCRESTORE.PC 561
PKRCSAVE.PC 561
PKRSTWC.PC 357
PKSETVAR.PC 356
PKSNENBSENS.PC 362
PKSNSETMTOFS.PC 365
PKSNSETVPNAM.PC 363
PKSNSETVTOFS.PC 364
PKSNSTART.PC 366
PKSNSTOP.PC 366
PKTRGETNUMCL.PC 381
PKTRSETCLPOS.PC 380
PKWCCHECK.PC 354
PKWCEND.PC 354
PKWCRSTPRDST.PC 354
PKWCSTART.PC 354
PLC 方案操作所需的设定 555
Plug & Play 参考 435
Plug & Play 的缓冲区功能 501
put_part_cst 729
put_queue 731
排出传送带传感器的设定（［DI］/［RI］＋［放入工件］时） 95,205
排出传送带传感器的设定（［HDI］＋［放入工件］时） 96,206
排出传送带的设定 85,194
排出基准位置的设定、机器人的位置示教 144,251

<Q>

其他程序 759
其他有用功能 544
启动步骤 737
启动多台机器人时的注意事项 407
前言 1
区域的详细设定 440
确认传感器的状态 565
确认传送带的状态 563
确认传送带工作站的状态 566
确认固定工作站的状态 569
确认机器人的扭矩 573
确认生产状况 562

<R>

R-30*i*B Compact Plus 时 679,696
R-30*i*B Mate Plus 时 678,694
R-30*i*B Mini Plus 时 680,697
R-30*i*B Plus 时 676,691
［RI］＋［程序的执行］时 323
［RI］＋［放入工件］的情况 321
［RI］＋［通过视觉检出］的情况 322
run_vis_cst 729
如果工件旋转，则补偿偏移 653
软件选项的构成 5

<S>

start_srv 730
设定 588,640
设定方法 682
设定连接到主板的脉冲编码器 275
设定碰触销的工具坐标系 45,167
设定前的其他准备工作 259
设定应用程序前的准备工作 28,152
设置安装位置 618
设置脉冲编码器 274
生产中无法变更设定 654
生产状况 617
拾取失败时发出通知 540
使用［DI］/［RI］ 162
使用［DI］/［RI］ 39
使用［HDI］ 40,163
使用 *i*RPickTool 的跳过动作实施范围时的安全逃避动作 513
使用方案变更工作单元 544
使用方案的工作单元的变更 586
使用托盘的情况 337
示教点的平顺指定 403
示教用电脑的设定 658
视觉程序 115,225
视觉程序（二维） 635
视觉程序（三维） 628
视觉设定 103,213,327,611
视觉追踪 7
适用探讨 21
输出信息接收相关程序 754
输出信息解析相关程序 755
输入信息发送相关程序 753
数值寄存器 450
锁存程序 747
锁存失败 763

<T>

特定信息的选择显示 585
停止/启动传送带不工作 653
通信电缆 658
通信设定 738
通信相关程序 752
图标的使用方法 584
图像读入的时机延迟 653
推荐事项 26
托盘设定 79,188,292,410,640
托盘位置设定 328

<U>

update_nfnd 730

<V>

V_RUN_FIND 732
VT_GET_FOUND 732

< W>

网络的构筑.....259
网络电缆的连接.....259
位置寄存器.....451
位置寄存器操作的设定.....751

< X>

系统变量文件.....657
系统构成例.....14
系统重启时处理停止前已检出的工件.....543
先执行指令.....406
线传送带的情况.....347
限制最终轴的轴可动范围.....570
相机的连接确认.....34,157
相机的设置和连接.....277
相机的准备.....610
相机校准.....103,213

< Y>

一次拍摄检测出多个工件.....760
应用程序的设定.....71,175
用户报警.....452
用户定义对象形状的使用.....587
用户坐标系.....452
用语.....9,410,734
有关标准 TP 程序的名称.....468
有关传感器的程序.....362,460
有关传送带工作站的程序.....366,460
有关工作单元的程序.....354,455
有关固定工作站的程序.....375,463
有关机器人的程序.....358,359,459,460
有关输入信号.....681
有关托盘的程序.....380,467
有关抓手的程序.....455
有关追踪程序.....382
与缓冲区中的拾取/放置相关的机器人的动作.....501
与机器人控制装置直接连接.....707
与脉冲分配器连接.....709
预整列设定.....408,493
圆弧传送带的情况.....305,350

< Z>

在 3rd Party 视觉中未执行拍摄和检出.....762
在电脑上设定 IP 地址.....658
在多台机器人上设定脉冲编码器.....276
在箱子中以各种方向放置.....532
直线距离指定.....405
直线最高速度功能.....406
指定防止负载错误的 *i*RPickTool 机器人最高速度..518
制约.....735
制约事项.....26,587,610,618,638
主程序.....384,388,392,742
主程序的创建例.....601
注意事项.....638
抓手的设定.....184
抓手的详细设定.....439
抓手和区域的设定.....436,437,493
抓住位置不正确.....764
追踪参数设定表的设定.....601
追踪动作的微调整.....150,258,401,433,453,493
追踪拾取 HDI 功能的有效.....682
追踪坐标系的设定误差的调整.....401,433,453
追踪坐标系的设定误差的调整（使用 ADJ_OFS 时）.....493
追踪坐标系的追加.....346
追踪坐标系设定.....86,195,304
字符串寄存器.....452
自定义传感器任务的步骤.....720
自定义的要点.....726
最初的微调整.....401
坐标转换参数计算.....751
坐标转换参数计算程序.....748
坐标转换参数计算相关程序.....756

说明书改版履历

版本	年月	变更内容
03	**2021 年 10 月**	• **支持 7DF5 系列 14 版(V9.40P/14)** • **支持 R-30*i*B Mini Plus 控制装置**
02	**2020 年 1 月**	• **对应 7DF1 系列 22 版(V9.10P/22)** • **对应 R-30*i*B Mate Plus/R-30*i*B Compact Plus 控制装置**
01	**2017 年 3 月**	

B-83924CM/03