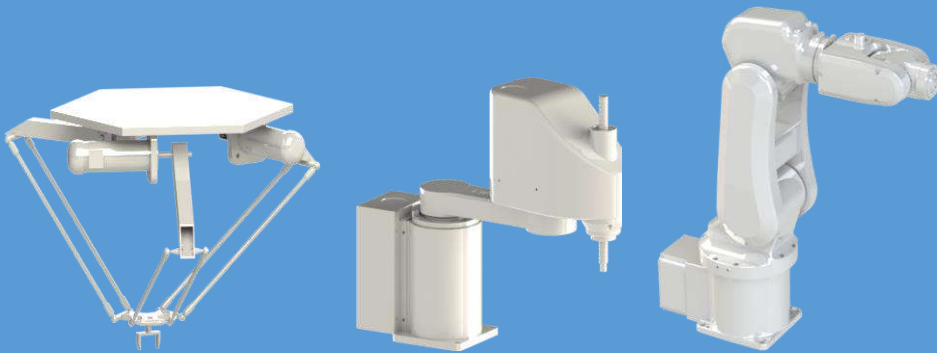


机器人控制器 用户参考手册

本手册为用户解释如何设置软件，配置机器人和轴，使用图形界面以及 CNC 指令。



-

1. 控制器初始化.....	1
1.1. 连接机器人控制器	1
1.2. 基本介绍	2
1.3. 设备更新	3
1.4. 配置网络设置	4
1.5. 将数据上传到控制器	5
1.6. 设置 CODESYS	6
1.7. 第一次运行程序	7
1.8. CODESYS 的变量跟踪	8
A) TRACE 的配置	8
B) 加载 TRACE	11
2. 图形化用户界面： GUI	12
2.1. 介绍	12
2.2. GUI 简介	12
2.3. SR 选项	12
2.4. 单个轴的配置 AXIS	12
2.5. 轴的点动 AXIS JOG	13
2.6. 轴回零选项 HOMING	14
2.7. 群组选项 GROUPS	14
2.8. 相机配置	16
A) 相机的初始化	16
B) 连接设备	18
C) 启动窗口 START WINDOW.....	18
D) 基本窗口 BASIC INTERFACES	19
E) 图案抓取 FETCH PATTERN.....	19
F) 移动传送带 MOVE BELT	20
G) MEASURE 测量	21
H) COMPUTE 计算	21
I) UPDATE CONFIGURATION 更新配置	21
2.9. CNC 文件	22
3. 控制器配置文件： ECAT.....	24
3.1. 软件需求	24
3.2. 单轴伺服驱动器 ENI 文件生成步骤	24
3.3. 四六轴伺服驱动器 ENI 文件生成步骤	26
4. 控制器配置文件： 运动控制	31
4.1. XML 文件结构	31

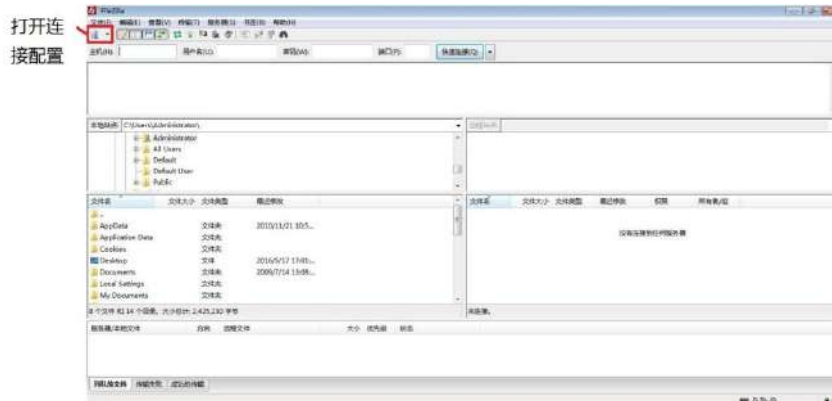
4.2. 驱动列表	31
4.3. 电机群组列表	33
4.4. 角度模式	37
A) 欧拉角	37
B) 偏向(ROLL)、俯仰 (PITCH)、横滚 (YAW)	37
C) 表示角度的例子.....	38
4.5. 耦合矩阵	38
4.6. 为电机群组配置运动模型	39
A) 六轴机器人	39
B) DELTA (并联) 机器人 (3 轴).....	41
C) 直角坐标系机器人 (3 轴)	43
D) SCARA 水平多关节机器人 (4 轴)	43
E) 码垛机器人 (4 轴).....	45
5. 控制器配置文件: 机器人视觉	47
5.1. 相机列表	48
5.2. 应用列表	49
6. 控制器配置文件: HANDS.....	49
6.1. 结构	50
6.2. MOTION_CONFIG 附加要求	51
7. 轨迹规划.....	53
7.1. 混合 Mcs 下的两种运动:线/圆到线/圆	53
7.2. 混合两个 Acs 运动: 直接运动到直接运动	54
7.3. 混合 Acs / Mcs 到 Mcs / Acs 运动: 多重混合运动	54
A) BLENDING 开始于 Acs(直线)至 Mcs(线性).....	54
B) BLENDING 开始于 Mcs(线性)至 Acs(直线).....	55
C) 转换模式: MC 转换模式.....	55
8. PLC 视觉系统的功能块 (FB)	55
A) 视觉构建	55
B) VISION_SETAPPLICATION 视觉集成应用.....	55
C) VISION_OBJECTDETECTION 视觉检测对象.....	56
D) MATCHTHRESHOLD COMPUTATION EXAMPLE 匹配阈值计算示例	57
E) VISION_READCAMERASTATUS 视觉读取相机状态.....	57
F) VISION_RESET 视觉复位	58
G) VISION_READOBJECT 视觉读取对象.....	58
H) VISION_READOBJECTQUEUE 视觉读取对象队列.....	58

1. 控制器初始化

1.1. 连接机器人控制器

需要软件: FileZilla(<https://filezilla-project.org/download.php?type=client>)或具有相同功能的 FTP 客户端。

为了能够正确使用设备, 电脑必须使用以太网线连接机器人控制器的 LAN 1 接口。连接完成后, 在电脑商启动 FileZilla 然后即刻在机器人控制器中上传或下载文件。当 FileZilla 打开后, 点击图 1 所显示的连接配置按钮:



图表 1: 检视 FileZilla: 打开连接配置向导

一个新的窗口将会出现。用户必须将下面展示的信息填写进图2所展示的区域中:

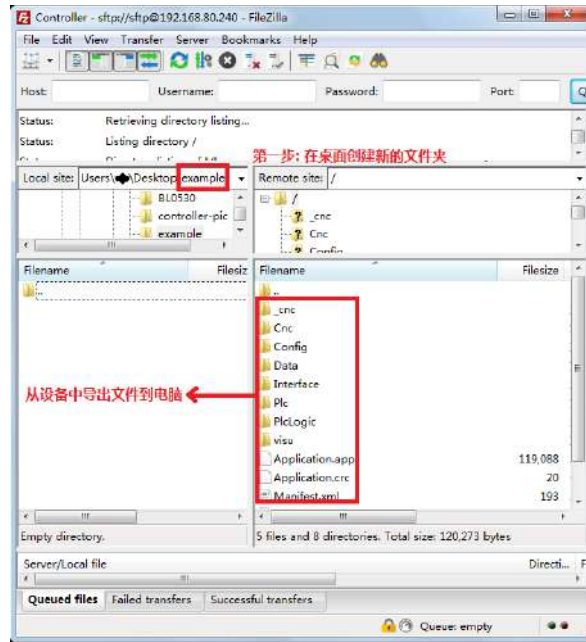
- 主机和端口: 输入设备地址 (例如192.168.80.240) 和端口号 “22”。
- 连接协议: 点击下拉列表右侧的箭头然后选择 “SFTP”。
- 用户: “sftp”
- 密码: “sftp”



图表 2: 连接设置

填写完全部信息后点击“连接”。现在用户应当能够在右侧的远程站点窗口中看见设备中的文件夹和文件。

使用 FileZilla 将图3中显示的文件全部下载到本地计算机中。例如在图示中展示的“example”文件夹便创立于桌面上(C:\Users\Desktop\example)并将其内容显示在左侧的本地站点窗口中。



图表 3: FileZilla 文件传输

从现在开始,用户可以访问所有设备的配置文件,本文档中显示。

请注意“Interface”文件夹中包含的图形化用户界面 (GUI), 如何使用此界面将在本节之后进行描述。

1.2. 基本介绍

机器人控制器是一个可同时处理驱动和控制单个运动轴和机器人组(机器人、自动化机械...)的通用平台。用户可以使用主机电脑(非本地嵌入式 PC)上的软件 GUI 或通过 CodeSys PLC 程序来控制机器人设备。

控制器子文件单元可以通过 CoDeSys 访问 (“Device(ARC Controller)” 文件夹下)、下载和上传“Files” 配置文件。亦可通过“FileZilla” 等软件连接进控制器的 IP 地址, 使用用户名: “ftp” 以及密码: “ftp” 登入。

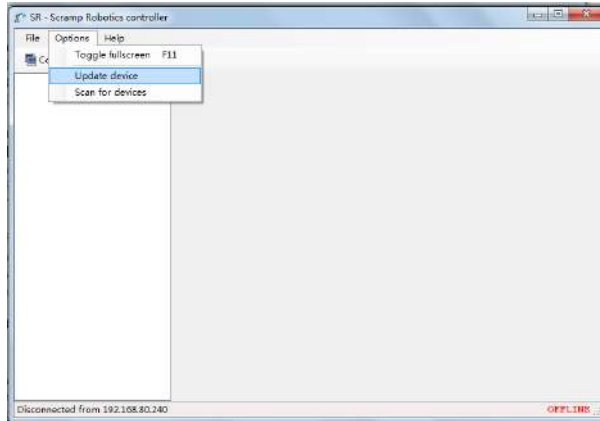
当用户访问控制器后, 以下文件夹含有使用设备的必要文件:

文件夹	描述
Cnc	这个文件夹下包含了组成 CNC 执行器的程序文件。
Data 数据	存放由文本文档 (*.txt) 组成的控制器日志 (控制器执行时输出的文本和消息, 这些文本和消息可以通过连接上 VGA 显示器看到)。
Interface 界面	包含 ARC_Gui.exe (可执行图形化用户界面) 执行以及使用 GUI 给予指令给控制器时所需文件。
Plc	包含了 ArcaLibrary.library 以及 ARC_controller.devdesc.xml 用于正确配置 CodeSys 资源库。
Config	包含了三个配置文件: 运动控制, ECAT 和机器人视觉。用户在导入自己的设置时必须对应三种配置文件分别命名为 “MOTION_Config.xml”、“ECAT_Config.xml”, “HAND_Config.xml” 和“VISION_Config.xml”。如果有任一文件丢失则控制器无法被正确配置, 因此当不需要某一功能 (例如机器人视觉) 时, 无论如何必须放

	入空的配置文件。
其它文件	Readme.txt 文件包含了以上的控制器架构信息。 Relase.txt 里可以查看控制器的运行环境和 GUI 版本号。

1.3. 设备更新

设备上的软件可以使用 GUI 以及新的软件包更新。如果设备必须更新,打开 GUI 没有连接到设备,点击选项→更新设备如图 4 显示。如果用户已经连接到设备,只需点击左上角的“断开”按钮的 GUI 并按照上面的指示。



图表 4: 设备的更新

点击这个选项后,窗口显示在图 1.3.2 就会出现。首先,检查目标设备通过单击第一行“Check target”按钮:出现目标 IP 地址。之后,选择要更新的软件包通过单击第二行的“Browse package”按钮,选择软件更新包。如果这两个操作都正确,则是两个绿色符号。如有异常,请检查所选设备和软件是否有误。

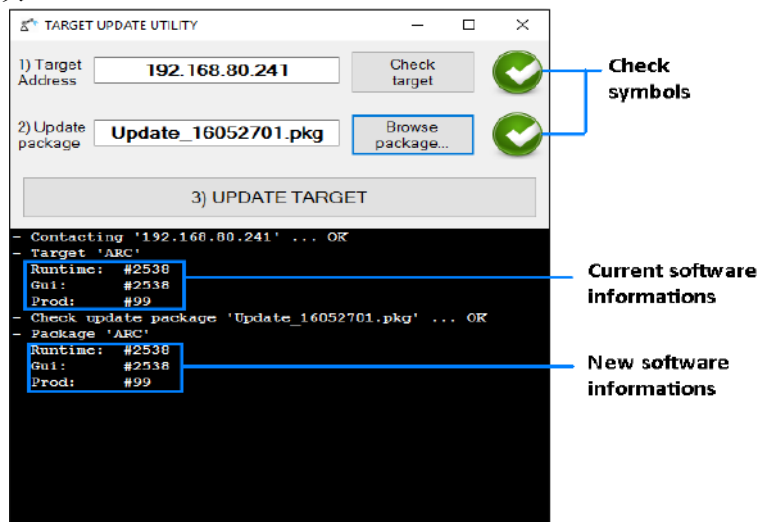


图 1.3.2: 更新窗口

在窗口的底部,显示软件版本信息。如果信息是正确的,点击“UPDATE_TARGET”按钮更新设备。一旦更新完成,这两个软件的信息应该相同。

1.4. 配置网络设置

正确的配置网络（IP 地址，Mask 地址和 EtherCAT 地址）后便可以操作 GUI，客户可依照需要在 SR 选项中的 Network 下更改参数值。

The screenshot shows a 'Network configuration' dialog box with the following fields and annotations:

- Device NAME:** ymlhw-2kg (Annotated: 机器人控制器在网络上的名称)
- Select the network configuration for the main Ethernet interface for the controller:** (Annotated: 控制器的IP地址、Mask地址)
- IP Address:** 192.168.80.111
- Mask:** 255.255.255.0 (Annotated: 这个地址必须配置才能使 GUI连接上控制器。)
- Internet connectivity:**
 - Gateway:** 192.168.80.253
 - 1st DNS:** 192.168.80.253
 - 2nd DNS:** 8.8.8.8
- EtherCAT virtual ethernet port:** (Annotated: EtherCAT连接信息)
 - IP Address:** 192.168.0.100
 - Mask:** 255.255.255.0

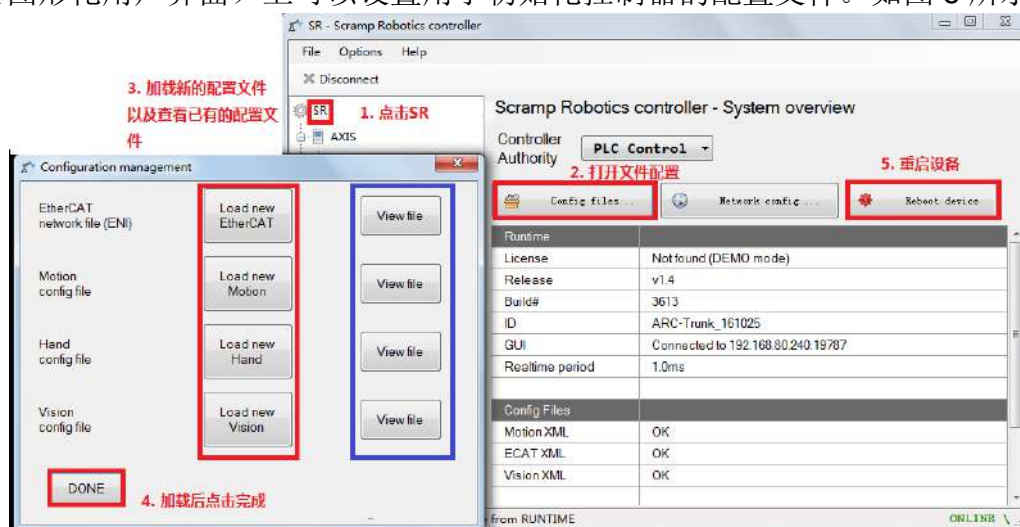
Buttons: OK, Cancel

图表 5: 网络配置表格

当所有的参数修改后，用户可以按下“OK”按钮然后开始使用机械手控制器。

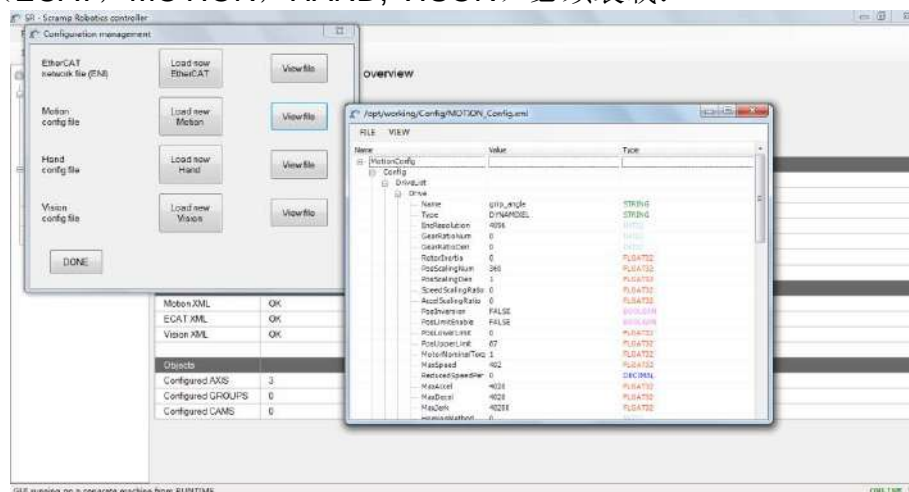
1.5. 将数据上传到控制器

在 GUI（图形化用户界面）上可以设置用于初始化控制器的配置文件。如图 6 所示，



图表 6: 加载新的配置文件

点击“Load New Configuration Files”按钮，客户可以选择打开观看或导入配置文件。每一个配置文件（ECAT, MOTION, HAND, VISON）必须装载：



图表 7: 查看已有的配置文件

当文件被选择后，GUI 向导会检查一切是否正常。完成后点击“Done”按钮文件将被保存到控制器上。当配置文件被正确导入后，控制器必须重启。点击“Reboot Device”然后选择是将设备重启。

1.6. 设置 CoDeSys

创建一个 PLC 程序然后使用 ArcaLibrary 中的块文件来创建机器人应用和运动控制需要安装 CoDeSys 运行环境并配置相应地网络连接，用于连接控制器。

首先用户需要安装 CoDeSys 在电脑上，请依照以下步骤操作：

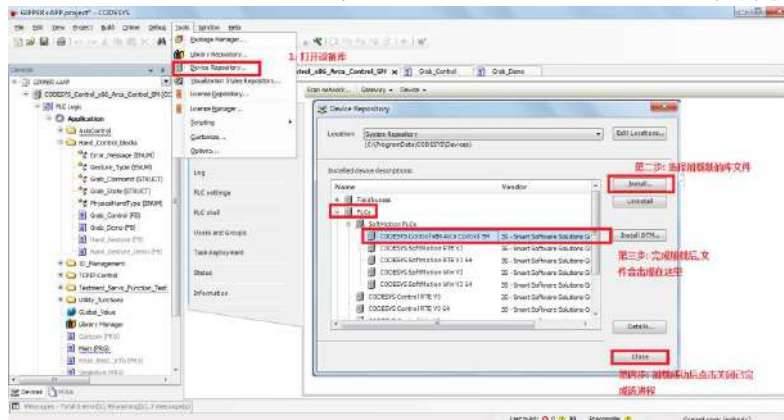
点击“Setup_CODESYSV35SP8.exe”以启动安装向导。

按照安装向导指示，使用默认设置安装。

完成程序安装后，用户必须将控制设备配置文件和 ArcaLibrary 装入设备资源库以便使用控制器和所有的块程序。安装步骤如下：

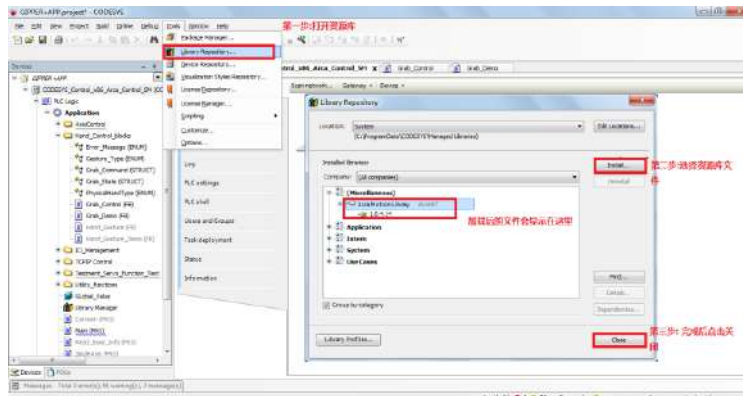
启动 CodeSys v3.5.8.1;

选择工具栏中的“Tools -> Device Repository”然后点击“INSTALL”，在计算机中找到并选择文件“ARC_Controller.devdesc.xml”。完成后 ARC Controller 对象将被加入资源库中。



图表 8: 加载新的设备库文件

点选工具栏中的“Tools -> Library Repository”然后点选“ArcaLibrary_compiled-library”将文件安装到资源库中。



图表 9: 加载新的资源库文件

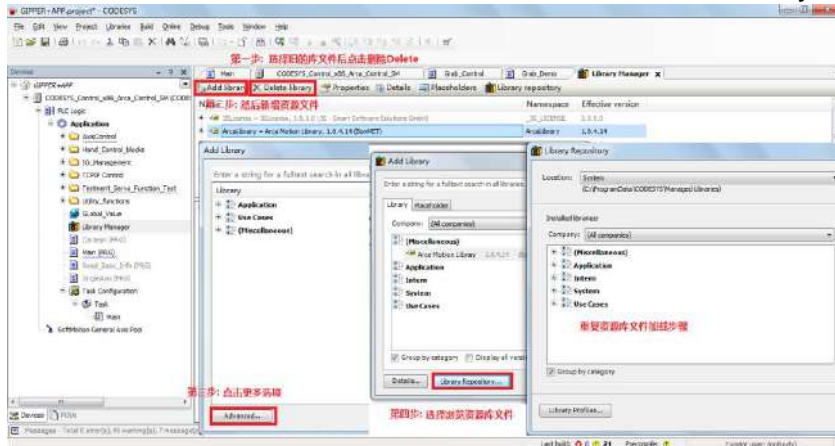
请注意导入库的步骤: 如果用户想要打开一个项目并升级 ArcaLibrary 的版本则需要在安装新库之前先卸载旧库。操作的步骤如下:

打开项目，寻找到“Library Manager”然后检查右侧窗口,选择 ArcaLibrary 然后 “Delete Library”

删除 ArcaLibrary 后，点击“Add Library”然后选择“Advance...”。

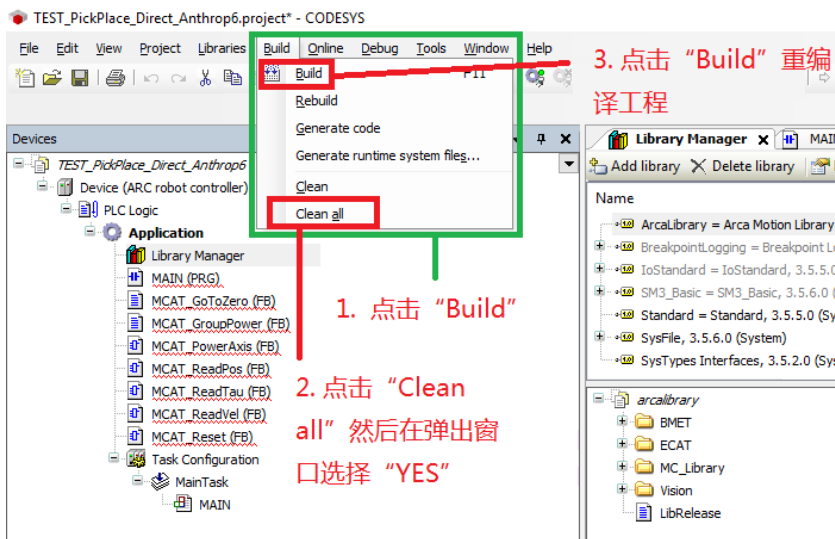
打开资源库然后在本地电脑中寻找到想要安装进去 ArcaLibrary.library 文件。

完成增加新库后，必须能在资源库中看到对象。然后关闭所有窗口再点击“Add Library”按钮，这一次选择“Miscellaneous”然后选择新增加的 Arca Motion Library（安装）。



图表 10: 删除旧文件,添加新文件

当新的库被装载后，在工具栏上点选“Build”。然后在新弹出的窗口点选“Clean All”然后点选“Yes”。完成后再次点击“Build”将使 CoDeSys 重编译整个项目。如果没有报错，用户可以在编译完成后运行工程。如何运行工程将在下一章节描述。



图表 11:编译程序

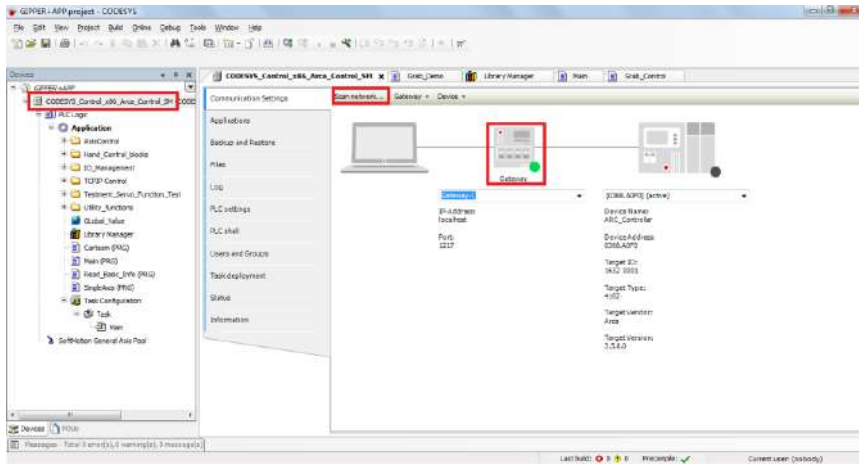
1.7. 第一次运行程序

在安装库后，即可开始启动运行第一个程序。在完成资源库所需对象的导入后，跟随以下步骤运行程序：

选择“文件 -> 打开项目”然后打开项目文件“GRIPPER+APP.project”

选择“新建 -> 重建所有”（如果报错则按照上一节的步骤删除并重新添加库文件）

选择节点“设备 (Codesys_Control_x86)” ->右键点击-> 选择“编辑对象”，设备连接窗口将会如下图一般出现：

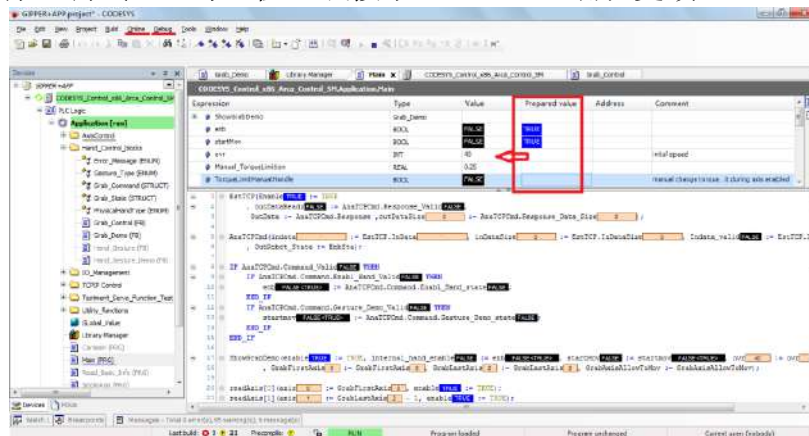


图表 12: 设置设备连接

将控制器的 IP 地址输入高亮框体中然后按下回车键。如果控制器被识别则绿灯将会出现，否则会显示“没有设备响应扫描请求”。如果出错请检查控制器设置以及连接线缆。

双击“MAIN (PRG)”打开主程序页面。选择菜单窗口“在线 -> 登录”：编译器会编译并下载程序到控制器中并改变编译环境为在线模式。

在变量列表中的“PREPARED VALUE”列，双击在“POWER”变量上使该变量成为蓝色的“TRUE”。选择“调试 -> 写入值”或按下 CTRL+F7 确认更改。



图表 13: 写入变量数值以启动程序

机器人将启动（请注意“Powered 供电”变量的改变）。

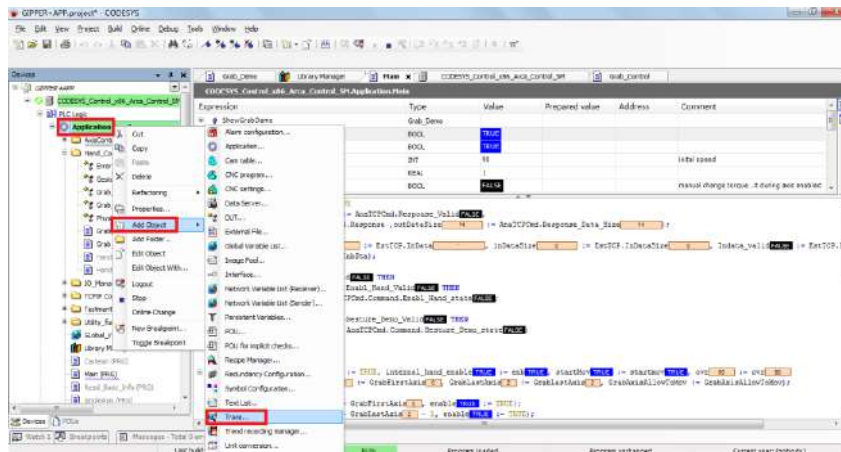
（如图所示 CoDeSys 在执行时是可以更改变量的数值的）

1.8. CodeSys 的变量跟踪

当 CodeSys 程序运行时，可以使用跟踪变量，以便更好的了解它们是如何演变的。跟踪可以很容易地配置为应用程序中的附加对象

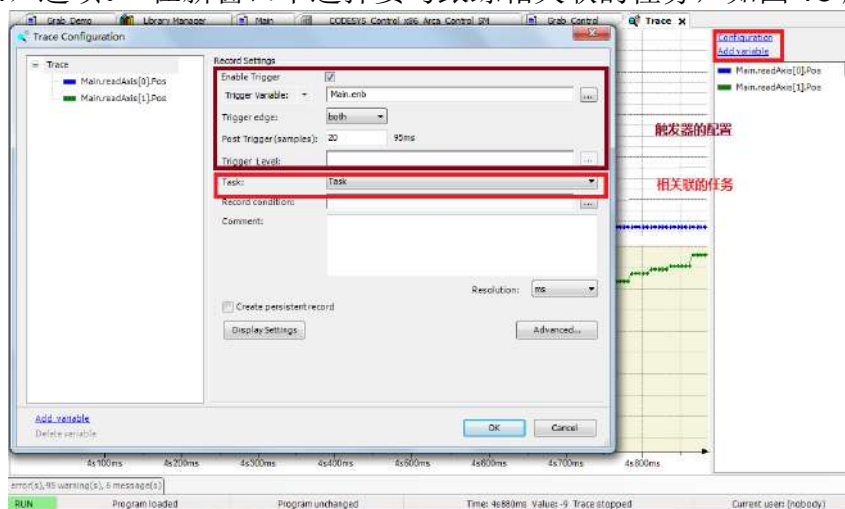
a) Trace 的配置

打开 CodeSys 应用程序后，右键单击“应用程序”上的左侧树菜单。选择“添加对象 add objekt” -> “观测曲线 trace”，如图 14 所示



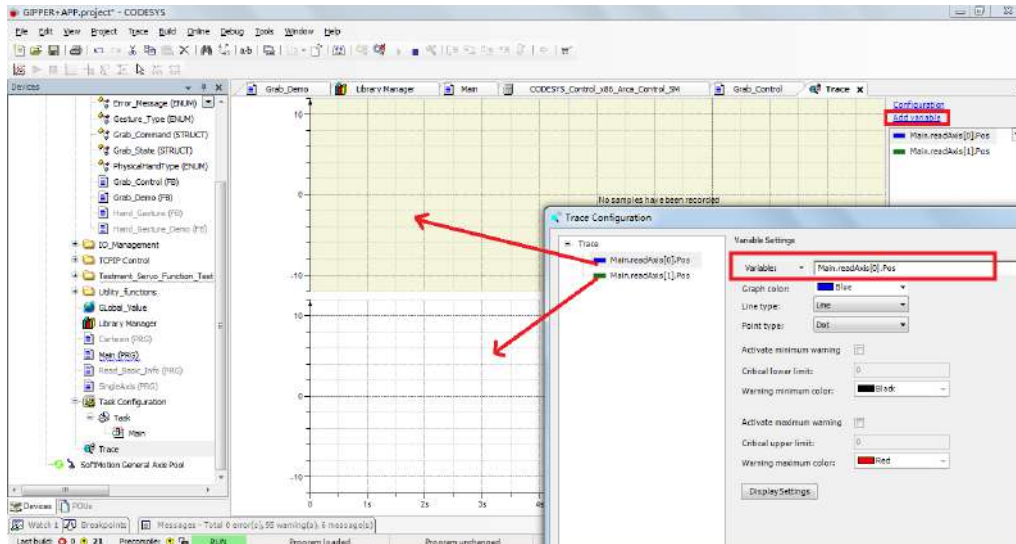
图表 14: 创建新的观测曲线(trace)

选择要跟踪的变量名称，然后单击“确定”。选择所创建的新的观测曲线并打开配置 (configuration) 选项。在新窗口中选择要与跟踪相关联的任务，如图 15 所示。



图表 15: 配置观测曲线(trace)

选择任务之后,还有很多的其他选项。触发器的配置可用于检测应用运行进程中的特定条件。可以选择要关联的触发器 (Trigger variable),检测信号 (Trigger edge) (正, 负或两者), 要记录的样本的数量 (samples) 以及触发值 (Trigger level)。在上面的例子中, 一个触发器被关联到主程序的布尔变量 `Main.enb`。当该变量发生变化的时候, 将检测到触发, 并在触发检测后记录 20 个采样。常规配置完成后, 用户可以继续添加要跟踪的变量。点击右边的“添加变量”。



图表 16: 添加新的观测曲线

首先选择要跟踪的变量。之后，可以选择许多图形选项，如颜色，线型，每个样品的点类型和临界极限。在上面的示例中,主程序的变量 **AxisPos[0]**和 **AxisPos[1]**将追踪，图表颜色为蓝色和绿色，每个采样点通过一条线显示连接到另一个采样点。客户还可以选择配置上下限警告，一旦选定变量假设值超出设定的数值范围,所采集的样本将被标为所选定的颜色。

请注意，并非所有 PLC 程序变量都可以被直接绘制在视窗中：例如，如果用户发现不能直接选择数组的元素进行观测。，请尝试在 PLC 程序中创建一个新的全局变量，将旧变量的值赋给新变量，然后尝试绘制新变量。

示例:假设一个机器人的位置被存储在 PLC 程序数组 **Robot_Positions = [Pos_X、Pos_y Pos_Z, Rot_A, Rot_B, Rot_C]**。这样机器人位置不能在轨迹中绘制...但是,如果用户指定位置之外的一个新的变量数组,可以跟踪新变量。

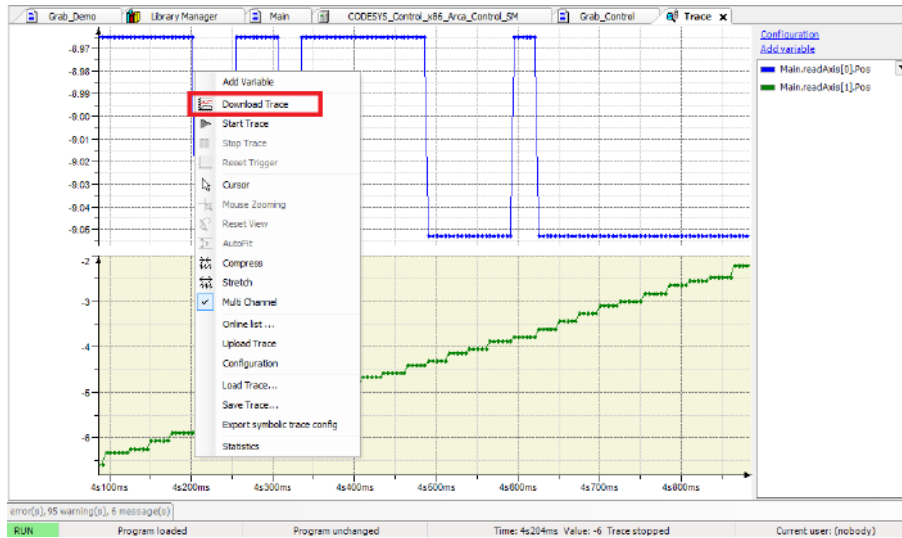
Position_X : REAL;

Position_X := Robot_Positions[0];

新的变量 **position_x** 可以绘制在跟踪！

b)加载 Trace

完成曲线的配置之后，启动运行 PLC 程序。打开创建的，右键单击图形并选择“下载轨迹 download trace”。这样，在程序运行时，所采集的变量数值将被绘制在图表中。在图 1.8.4 中显示了上面示例中配置的跟踪变量



图表 17: 加载要追踪的曲线

一旦跟踪完成，有很多选项可以使用。通过右键单击轨迹，可以添加光标，以便滚动每个跟踪变量的值。还可以以.trace和.txt格式保存 trace，或加载先前保存的 trace（右键单击 trace 并选择“保存 save trace”或“加载 load trace”）。

2.图形化用户界面： GUI

2.1.介绍

Scrapm 机器人同时支持图形化界面（GUI）和 PLC 程式两种操纵方式。GUI 适用于进行轴的运动测试和机器人群组的运动模拟，同时也可以检验配置文件是否正确。第二种方法则借助 Codesys 软件来编写和执行 PLC 程序。连接好控制器之后，打开 GUI 软件使用“Ctrl+C”或者点选“File->Connect” 进行联机。

2.2.GUI 简介

Scrapm 机器人控制器主界面如图6.2.1所示。在窗口的左面部分由三个选项标签容许用户访问设备不同的部分：BMC、AXIS、GROUP、CAMERAS 或 CNC。



图表 18: GUI 主界面视图

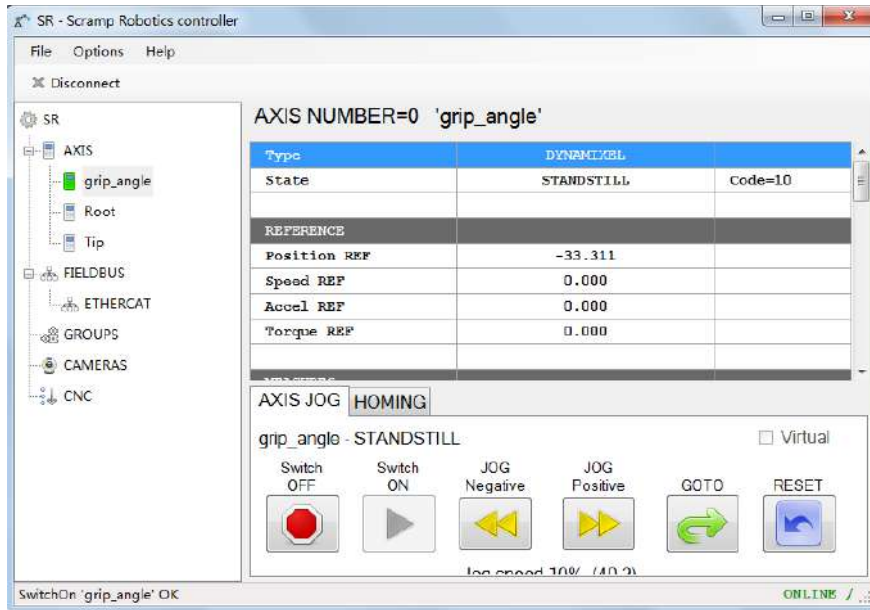
2.3.SR 选项

通过选择 SR 可以设置“GUI/CNC”和“PLC”之间的控制器操纵权限切换。选择“PLC”选项可以运行写在 Codesys 环境中的 PLC 程序。而“GUI”选项是使用作用于机器人的 GUI 指令。请注意这两个选项只能选其一：当选择了“GUI/CNC”选项控制权限将不容许运行 PLC 程序，而如果“PLC”选项被选中则 GUI 指令无效。每次切换两个权限时需要输入密码。在开发端，这两个密码是“GUI”（如果用户希望从 PLC 切换到 GUI）以及“PLC”（如果用户希望从 GUI 切换到 PLC）。

在控制器权限下面是““Config files”, “Network Config (更改网络配置)”和“Reboot device (重启设备)”按钮。其操作说明请参照1.4 和1.5章节。

2.4.单个轴的配置 AXIS

通过选择左边三个树形选项中的“AXIS”，客户可以激活并操作每一个运动轴。主视窗显示了每一个运动轴的当前位置和状态。根据轴的状态，每一个轴都会获取到很多运动信息，如轴的状态、位置、速度、加速度、位置比例以及运动限位等。

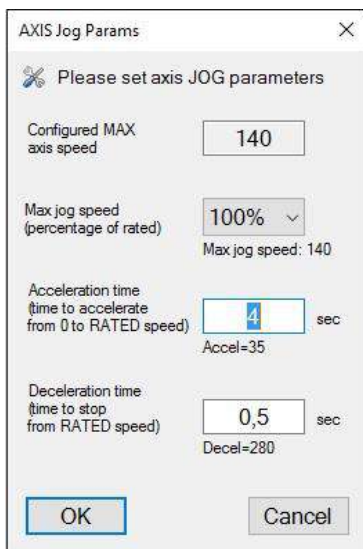


图表 19: 单个运动轴的具体信息

在主视窗下有两个文件夹可以选择：“AXIS”和“HOMING”。第一个可用于执行点动，第二个选项是用于机械轴的回零。为了能够执行以上两个动作，必须通过单击“SWITCH ON”按钮来对运动轴进行使能。如果该轴已正常打开，左侧树状列表上相应的图标会变成绿色，如果出错则会转换为红色，此时，重启设备再次使能，仍然出错的话，则请检查设备配置文件是否有误。

2.5. 轴的点动 Axis Jog

在每个轴所属的 axis 栏下，可以控制和设置单个轴的驱动，手动控制其正向或反向运动、如果出现错误警报还可以重设“STATE”状态然后查看设置参数。同时也可以作“HOMING”回零动作，并储存当前位置到驱动器中作为参考零点。



图表 20 Jog 参数

“SWITCH ON/OFF”按钮开启或关闭伺服使能，“JOG Negative”和“JOG Positive”按钮提供当前单个轴的点动功能（正方向或负方向由运动轴的方向确定）。“GOTO”按钮可以用于具体指定工作区间里机器人必须用直接路径到达的一点。当驱动器报错时使用“RESET”按钮进行重置。在这些按钮以下用户可以点击“Jog settings”按钮（图 20）用于修改点动设定：当按钮被按下后一个新的窗口将会出现并提供给用户实际的点动参数。修改参数只需要点击相应的参数值即可。

如果只有点动速度需要改变的话，用户可以直接点击侧边栏上的更改数值“Jog speed”“Jog setting”两个按钮。

图 6.5.3 展示了“GOTO”的工作原理：当按钮被按下后，一个新的窗口会出现。在这里用户可以设定轴要到达的位置。同时需

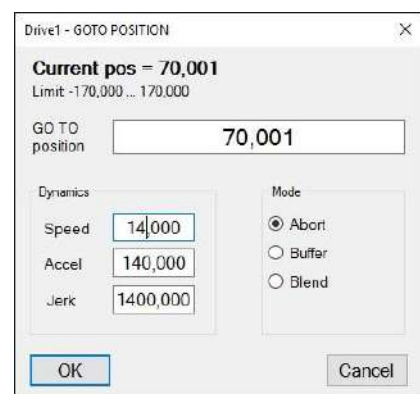


图6.5.3: “Go To”指令样例。

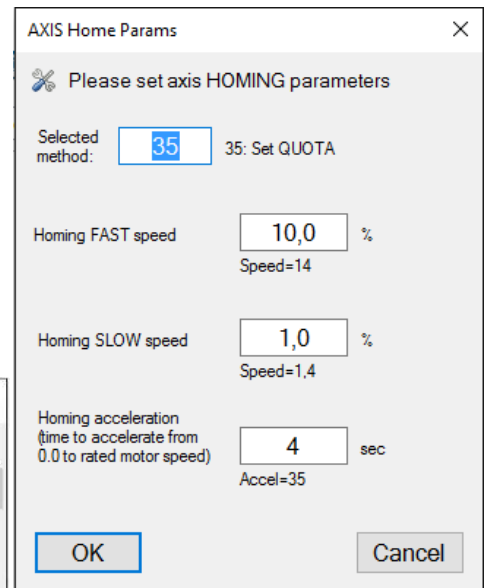
要设定值的也包括了执行运动所必要的最高速度，最高加速度和 **Jeark** 值。当设定完所有的参数后，点击“OK”按钮机器人就会开始执行计划好的动作。

2.6. 轴回零选项 HOMING

“HOMING”文件夹可以被用于设定电机轴的零点位置（图 6.6.1）。在电机轴运动到预定位置后（例如使用点动指令移动轴），点击“START HOMING 开始加载”按钮执行指令获取当前位置作为该轴的零点位置。



图表 21：轴回零概述界面



图表 22：设置回零参数

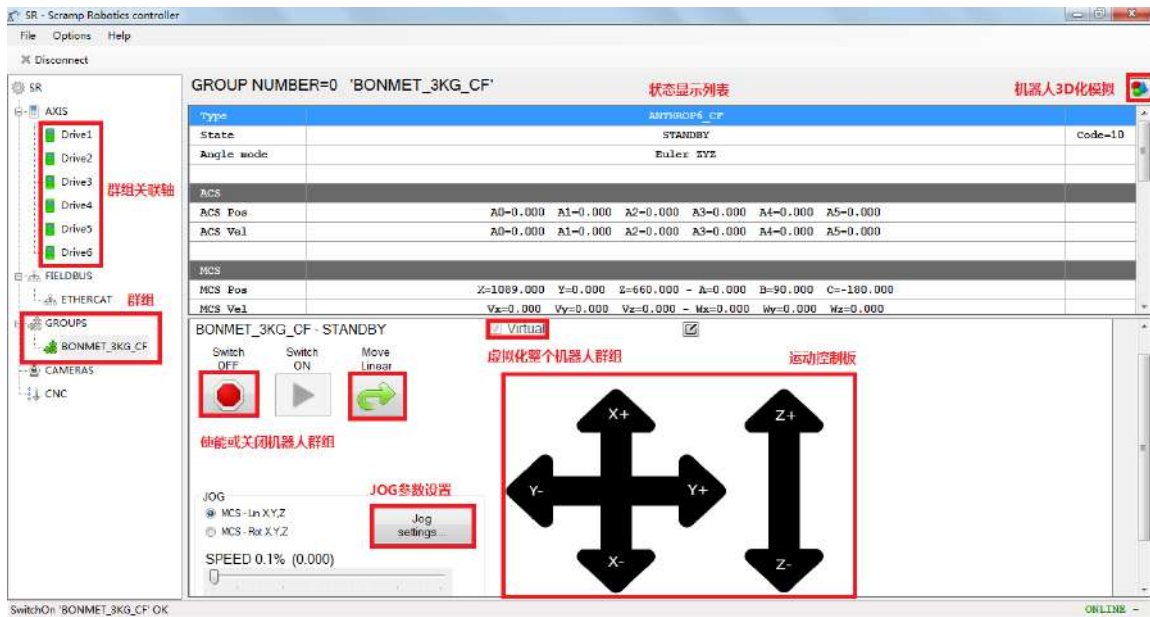
按钮“Homing settings”中的设置执行回零的相关参数。图 22 显示了可修改的参数。被选中的回零方式被用来执行回零步骤（默认值 35，对应于 Set QUOTA 方法）。

回零快速度“Homing FAST speed”和慢“Homing SLOW speed”是驱动器强制进入归零进程后所遵循的参考速度。两项数值是最高速度的百分比。

回零加速“Homing acceleration”是用于设定电机速度从 0 到额定转速的时间。

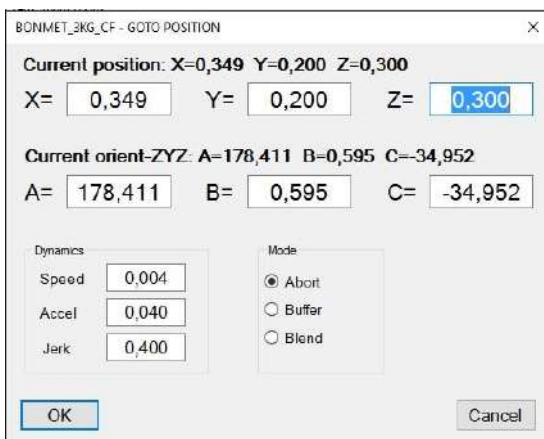
2.7. 群组选项 GROUPS

在左边的树状列表中选择“GROUPS”，用户可以检查电机群组的实时状况。这个跟“AXIS”选项非常相像。用户点击组名称后可供使用的指令和信息就会出现。



图表 23: 机器人群组窗口概述

无论对群组使能或是移动都与之前的单轴指令不同。在信息窗口中显示的是组状态。而角度模式（描述机器人末端的旋转姿态）为欧拉角 ZYZ 模式。而显示的位置信息则是在 ACS、MCS 坐标系下的位置参数和机器人动力学参数。



图表 24: 群组直线差补运动

Jerk 值，与单轴面板中的“Go To”是同一道理。一旦运动被计划好后，点击“OK”按钮让机械手执行动作。如果在运动过程中，有任一轴到达了限位则运动将停止且报错。报错以后为了能够再次使用群组指令，用户必须首先重设每个报错的轴然后再重新对机器人使能。

关于群组的点动运动，“Jog settings”按钮以及“Jog speed”侧边栏依然可用，他们的作用与单轴面板里时是一样的。对于电机群组运动来说，选择以直线点动或 MCS 坐标系中的角度姿态旋转都是可行的。如果做直线点动，点动按钮指令机器人运动时就会沿着 MCS 的 X、Y、Z 轴，而当使用姿态点动的时候则点动指令会沿着 X、Y、Z 或欧拉角 ZYZ 旋转机械手的末端。

即使命令名称和单轴中的命令相同，但功能并非一样。首先当“Switch ON”按钮被按下时，每一个被配置进电机群组的伺服都会使能（在图 23 中显示的机器人组“BONMET_3KG_CF”配置了全部六个轴，当群组使能时，每一个电机都使能了）。与之相同，点击“Switch OFF”时每一个配置进群组的电机都会关闭使能。“Move Linear”按钮取代了单轴面板中的“Go To”按钮：按下后窗口显示如图 24 中的机器人将执行一个直线差补运动指令（轨迹定义在工作区域中且使用反向动力学驱动动作，注意奇异点会导致机械手报错）。

在这个窗口下，用户可以同时设定机械手可以到达的位置和旋转角度。可以设定最高速度、加速和

2.8. 相机配置

相机校准是在 MCS（机器坐标系）中进行物体检测，对象位置精确获取和旋转之前的必要过程。相机校准的目的是在 MCS 中获得相机位置和姿态。然后，可以将对象位置从相机自身的坐标系表述变换为 MCS 表述。当相机被安装在机器人系统中，则首先要进行相机姿态校准。

相机校准可以以不同的方式进行处理，在我们的系统中，提供给客户基于棋盘（或基于图案）的过程以便于操作。以下是进行姿态校准之前的先决条件：

- 已校准相机的内部参数
- 5×8网格标准棋盘。
- 该棋盘（或其他图案）应放置在机器人的活动范围内，或可由传送带送至机器人活动范围以内。如果棋盘必须通过传送带传送，MCS 中的正向传送带应该是已知的。

相机校准进程可以通过使用我们的 GUI（图形用户界面）来完成。

a)相机的初始化

我们的机器人视觉系统支持 IFM O2D222型号的照相机。控制其与相机建通过 TCP/IP 协议进行通讯。IFM O2D222 相机支持物体远程探测功能，处理信息及抓取的图像可通过 TCP/IP 协议传送给我们的控制器。

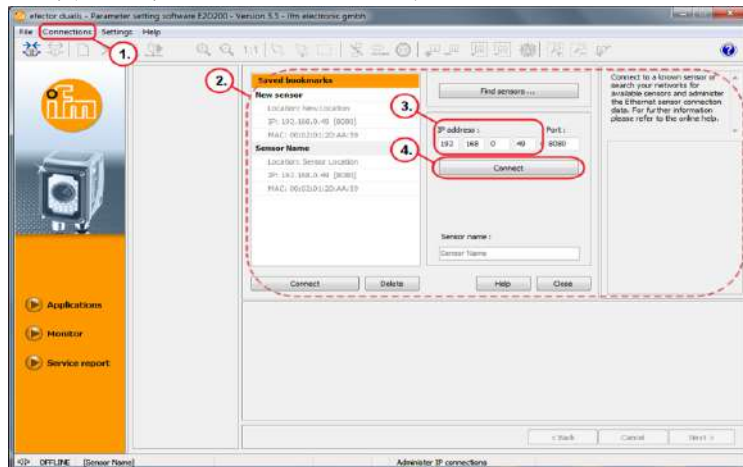
IFM 提供一些软件和操作文档来辅助客户使用其产品，其中比较有用的资料连接如下。

操作软件 (<http://www.ifm.com/ifmus/web/dualis-download.htm>)

安装向导 (<http://www.ifm.com/mounting/706370UK.pdf>)

应用手册 (<http://www.ifm.com/mounting/704420UK.pdf>)

相机的初始化可独立于机器人控制器进行，请遵循其使用向导。但在使用相机之前，应该根据任务要求来设定相机应用。该进程需借助 IFM 官方提供的应用软件来实现。其下载连接请参见其[官方网站](#)。下载并安装完成之后，开始相机初始化进程。



图表 25: 初始化相机

1. 选择菜单 连接/ IP 地址
2. 打开连接窗口后显示出 IP 地址输入窗
3. 输入 IP 地址
4. 点击连接

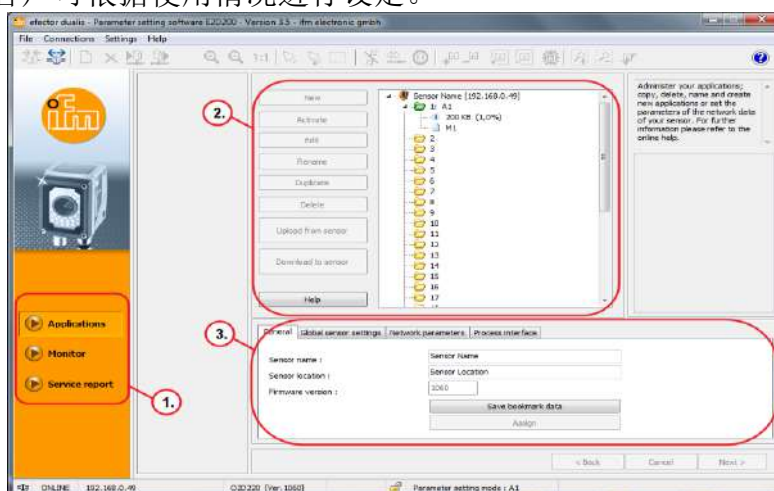
网络参数设定如下：

General	Global sensor settings	Network parameters	Process interface
IP address :	192 . 168 . 0 . 49	XML-RPC port :	8080
Subnet mask :	255 . 255 . 255 . 0	Video port :	50002
Gateway :	192 . 168 . 0 . 201	TCP/IP port :	50010
Speed and duplex mode :	Automatic recognition	MAC address :	00:02:01:20:AA:59
<input type="button" value="DHCP mode"/> <input type="button" value="Sensor reboot"/> <input type="button" value="Assign"/>			

图表 26: 网络参数设定

如果该参数已经被更改过，客户还可以依据章节 IFM 所提供的操作手册的 5.3.6 章节来找到设备参数并更改。

- 连接成功后，点击“应用 applications”
- 在区域2中可定义新的相机应用
- 在区域3中存储有全局参数，其子窗口“通用 General”和“网络参数 Network parameters”，客户可根据使用情况进行设定。



图表 27: 应用设定

在子窗口“全局传感器设置 Global sensor settings”中，参数如下

General	Global sensor settings	Network parameters	Process interface
Trigger input debouncing :	<input type="button" value="On"/> <input checked="" type="button" value="Off"/>		
External selection of the application :	<input type="button" value="Off"/> <input type="button" value="v"/>		
<input type="button" value="Assign"/>			

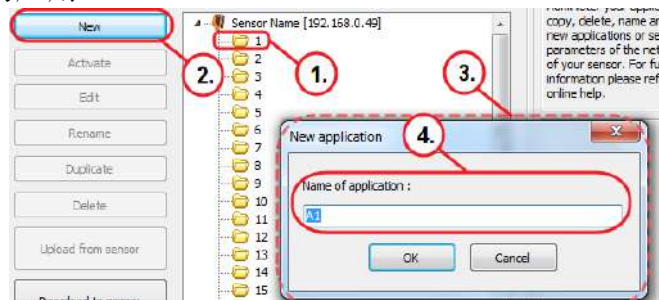
图表 28: 全局传感器设置

在子窗口“进程界面 Process interface”，参数如下

General	Global sensor settings	Network parameters	Process interface
Process interface :	<input type="button" value="TCP/IP"/> <input type="button" value="v"/>	<input type="button" value="Extended ..."/>	
Protocol version :	<input type="button" value="V1 <contents>CR LF"/> <input type="button" value="v"/>	<input type="button" value="Assign"/>	
Output format :	<input type="button" value="ASCII"/> <input type="button" value="v"/>	<input type="button" value="Sensor reboot"/>	

图表 29: 进程界面参数设定

设定完成后，创建新的应用。



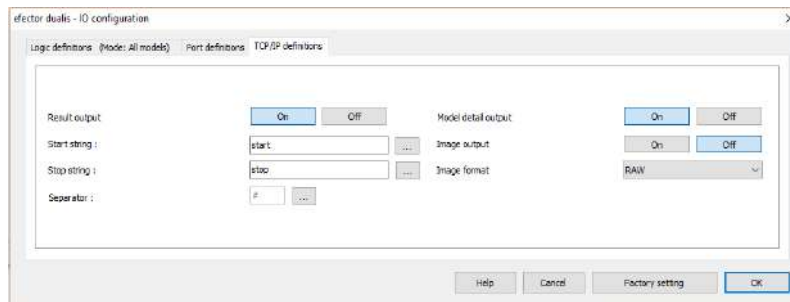
图表 30: 新应用的创建流程

1. 选择文件夹1
2. 点击“新建 New”
3. 弹出命名窗口，输入名称后
4. 点击“确认 OK”

完成设定的流程如下：

- 图像质量 Image quality
- 定义模型 Model definition
- 过程界面 Process Interface
- 触发器设定 Trigger configuration
- 功能检测 Function test

该流程也可参见安装向导的章节 8.7 – 8.13.为匹配相机与控制器的通讯，TCP/IP 协议设定如下。



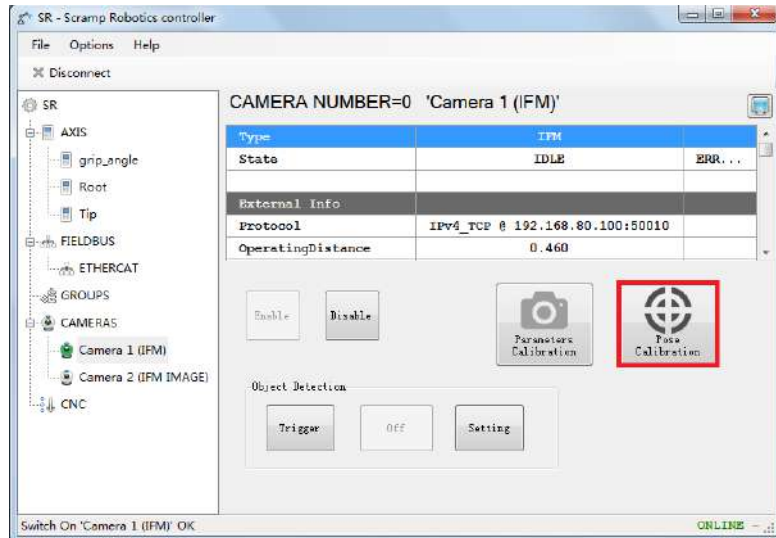
图表 31: TCP/IP 设定

b)连接设备

客户需使用一个网络中继器(路由器)来连接机器人控制器，相机及 PC。需使用一根 Ethernet 网线连接相机与网络中继器，另一根网线连接网络中继器与 PC，并设定有限连接的 IP 地址在 192.168.80.(000-255)范围内。

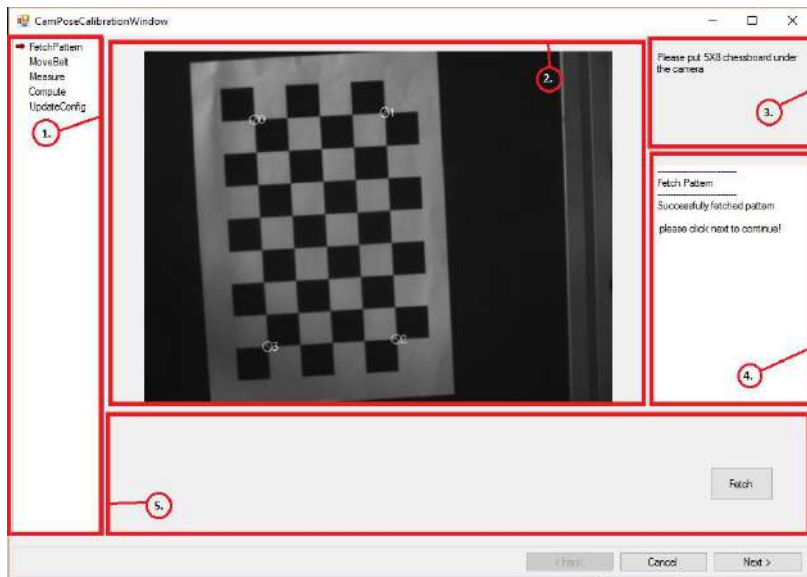
c)启动窗口 Start Window

在摄像头配置文件中，使能 Enable 后单击[Pose Calibration]（姿势校准）开始逐步程序。



图表 32: 使能开启相机

d)基本窗口 Basic Interfaces



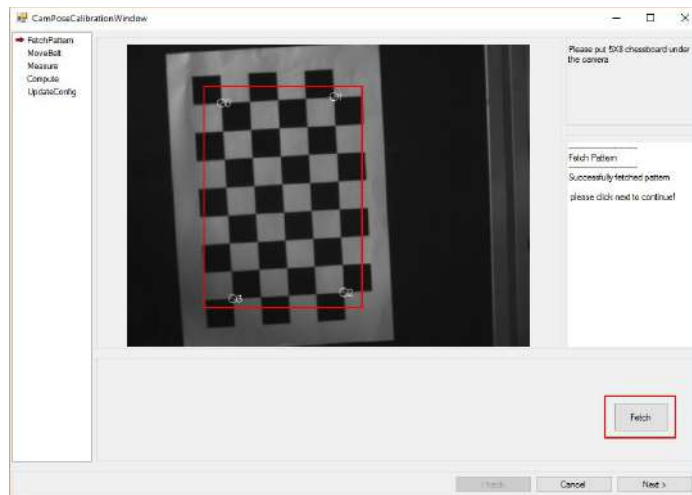
图表 33: 开始校正流程

如图33所示，1区域显示当前步骤，2区域显示相机捕获的图像，3区域显示内容提示，字段4显示进程状态信息，5区域为操作区域。

e)图案抓取 Fetch Pattern

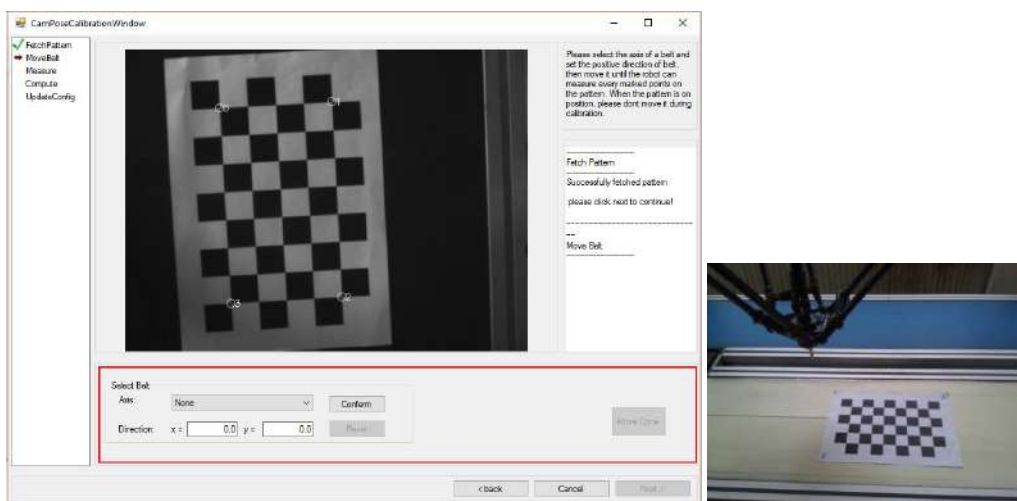
当棋盘处在相机视野内时，点击[Fetch 获取]按钮来触发相机扫描图像。如果抓取成功，相机将标记一些参考点作为锚点，用于之后步骤的测量。然后单击[下一步]继续。

如果失败，请检查图像，并检查棋盘是否处在相机视野内，然后确认棋盘是否为标准的 5x8 网格，照明条件是否足以清晰地识别它。



图表 34: 相机抓取图案

f) 移动传送带 Move Belt

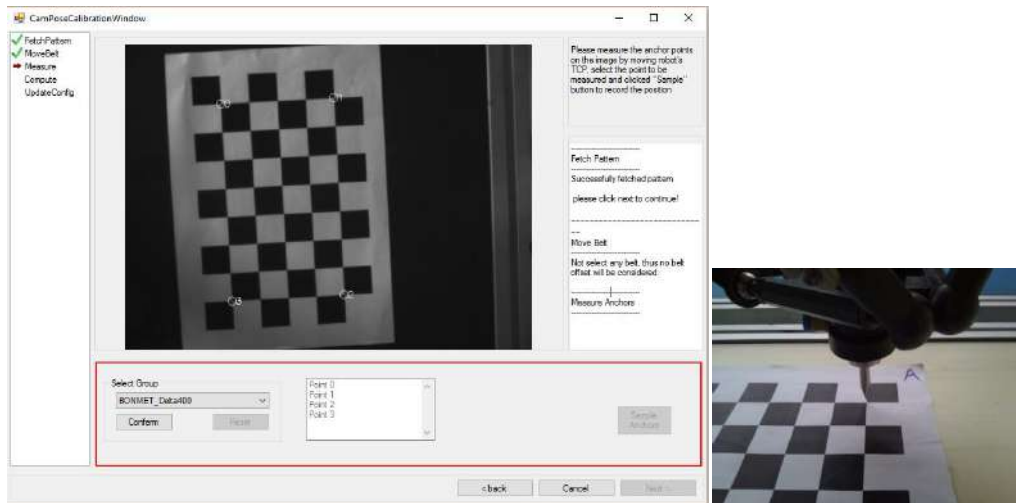


图表 35: 传送带设置栏

如果没有皮带，选择[None 无]，然后单击[Confirm 确认]，便可以跳过此步骤，不会考虑皮带运动。如果装配有传送带的话，选择传送带所在轴，并确定依据于机器人坐标系来确定皮带运动方向，单击[确认]。如皮带运动方向为 x 轴负方向，则 $x=-1$ ， $y=0$ 。如果选择有错误，请单击[重置]以删除选择，然后重新选择。

皮带设置成功后，请移动皮带将棋盘转移到适当的位置，机器人可以在该位置测量所有标记的定位点。然后单击[移动完成]记录皮带移动偏移。如果位置不够好，可以重新移动后并单击[移动完成]以更新皮带位置。

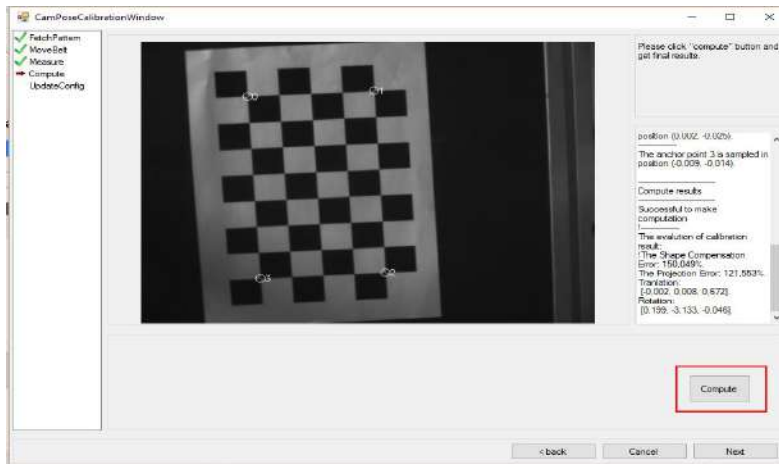
g) Measure 测量



图表 36: 测量参考点

移动机器人末端来依次测量所需的所有点（此例中有四个点）。一旦 TCP 移动到足够接近目标点。点击[Sample Anchors]来记录当前点的位置。可以点击 re-measured 用以更新数据。对所有点进行采样后，单击[下一步]继续。

h) Compute 计算

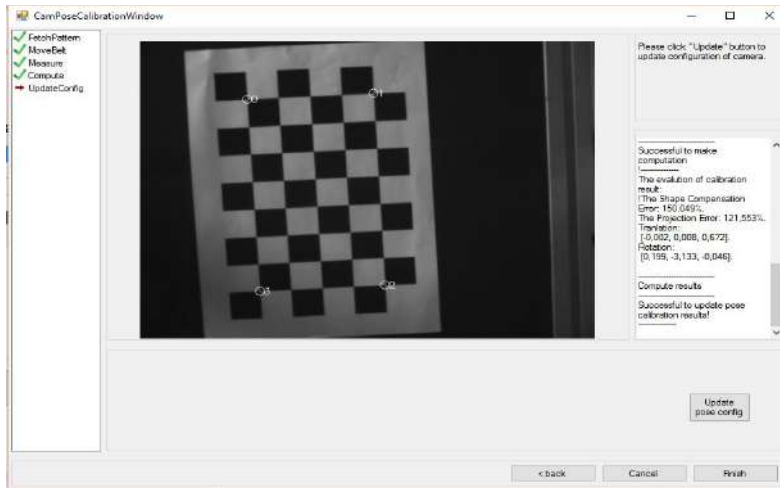


图表 37: 计算测试结果

在此步骤中，单击[计算]以获取校准结果。该设备提供两个评估值以评估结果：形状补偿误差以及投影误差。形状补偿误差是对点形进行整形时产生的值偏差，建议值不大于 10%。投影误差是投影点和实测点之间的平均偏差，建议值不大于 1%。

i) Update Configuration 更新配置

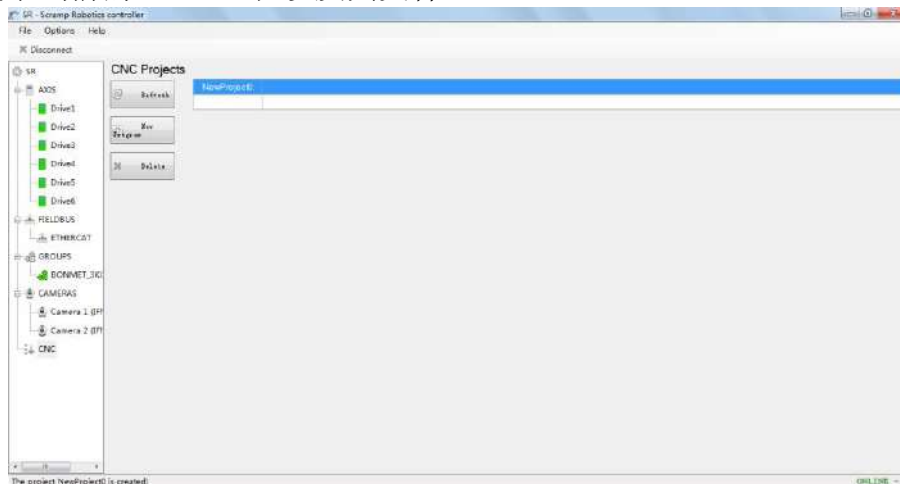
如果校准结果满意，最后一步便可更新配置文件。新的校准平移和摄像机姿态的旋转将被写入视觉配置文件并覆盖旧的值。该新文件可通过 FTP 从控制器中获取。



图表 38: 更新配置文件

2.9.CNC 文件

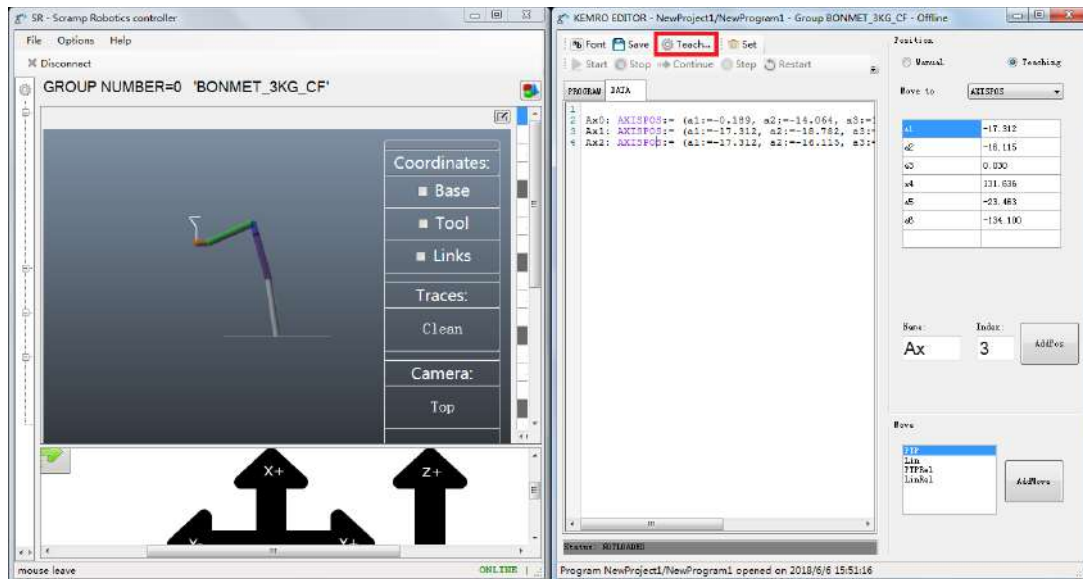
借助 CNC 文件，客户可以快速获取机器人运动空间参考点信息，并编写 CNC 工程来控制机器人的运动。如图所示，CNC 选项中有两个主要的按钮：“Refresh 刷新”按钮容许客户刷新主界面，“New project 新建”按钮可以被用于创建新的 CNC 工程。在这两个按钮之下时一个表格描述了当前的 CNC 工程以及其文件。



图表 39: CNC 主概略面板

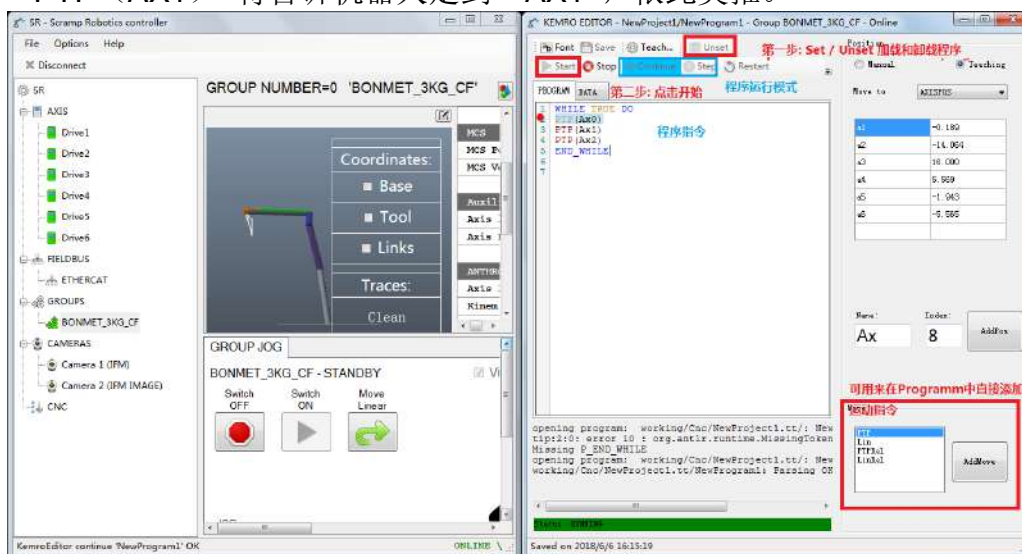
有两种方式可以添加 CNC 程序文件（右键点击再表格上以及选择“Add file”），以及编辑已创建的文件（双击文件名称）。在打开文档后，一个新的窗口将会弹出且在工具栏下有两个主要区域：“PROGRAM”和“DATA”。

如图所示，借助于六轴机器人的 3D 虚拟影像来编写运动程序。移动机器人到工作区空间中的某一点，点击“示教 Teach”，该数值将被储存在“DATA”中。



图表 40: CNC 示教功能

在上图中,有三个点被记录下来:AX0,AX1,AX2 和 AX3。记录完这些点以后,在“PROGRAM”区域中可以编写机器人运动的程序文本文件。在这个样例中,每一个在“DATA”中记录的点都被“PROGRAM”中的命令关联起来:“PTP(AX0)”会命令机械手移动工作点到记录的点“AX0”,“PTP (AX1)”将告诉机器人走到“AX1”,依此类推。



图表 41: CNC 程序文本

在“PROGRAM”区域,用户可以实施许多 C 语言的指令。例如 For 循环、while 循环、if、case 等。“Font”提供了需要选项改变字体和格式。完成程序的编写后,可以通过工具栏上的指令进行测试。首先点击“Set”按钮去测试写好的程序(每次只能运行一个程序)。状态显示在窗口的底端,“READY”状态显示后“Continue”和“Step”指令将可以使用。点击“Continue”按钮持续运行写好的程序。点击“Step”按钮每次只运行一行写好的指令。点击“Restart”按钮重启程序。点击“Unset”,当前程序将变为不可用。该状态下,用户可以修改程序然后再进行测试。

3.控制器配置文件：ECAT

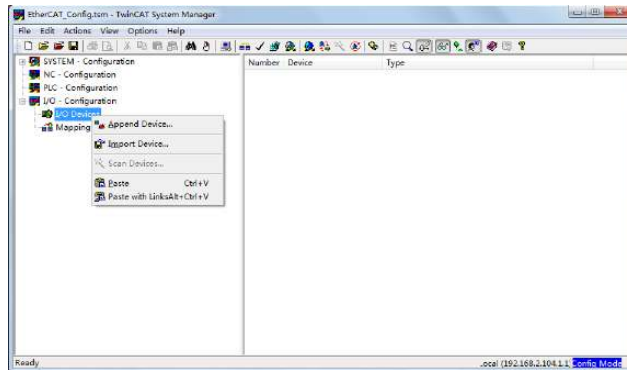
“ECAT_Config”文件包含了用于保证系统运行以及初始化控制器与子机（驱动器）之间的通讯的基本信息。在本章节会详细描述 ECAT_Config 的设置。用户可以参照制作驱动器 EtherCAT 配置的步骤给他们自己的驱动器生成 ECAT（ENI）文件。

3.1. 软件需求

- 在 PC 上安装 TwinCAT 2 NC
- 单轴伺服驱动器：Bonmet SSx4 系列 XML
- 4/6 轴伺服驱动器：Scramp 6 轴系列 XML
- 将上述文件安装在 TwinCAT folder (C:\TwinCAT\Io\EtherCAT)

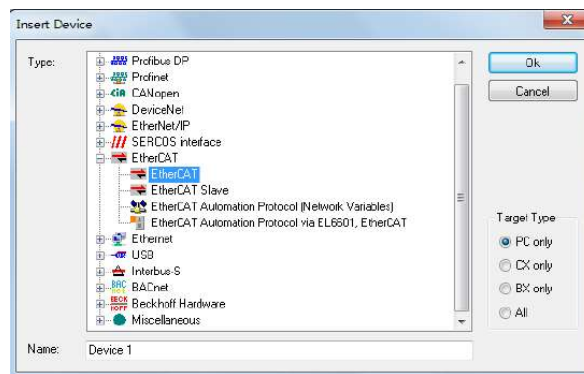
3.2. 单轴伺服驱动器 ENI 文件生成步骤

下面的步骤展示了如何为机器人控制器制作 ENI 文件。打开 TwinCAT system manager 后，点击创建 Edit > 空白文件 New



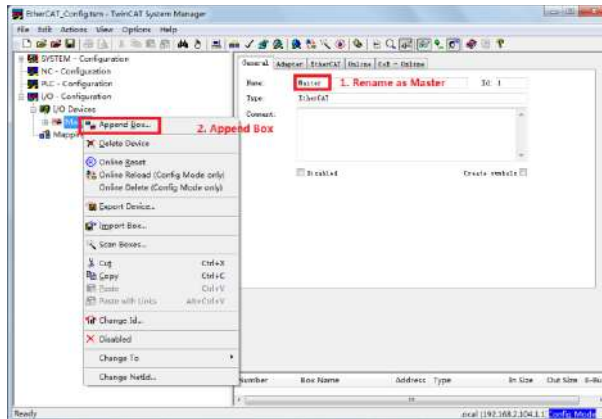
图表 42：添加新的 EtherCAT 文件

右键单击 I/O devices 并选择“添加设备 Append Device”。选择 EtherCAT > EtherCAT 并点击确认。



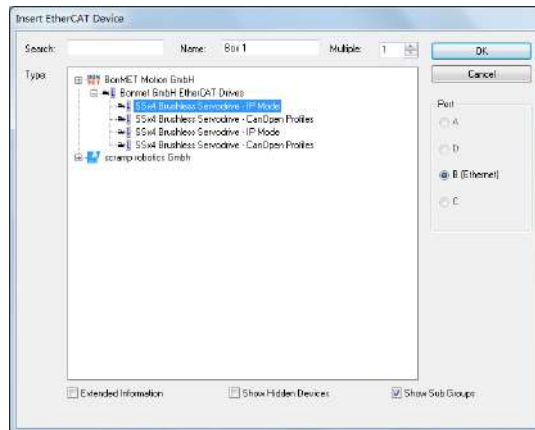
图表 43：选择 EtherCAT 设备

重命名新设备为主机“Master”。

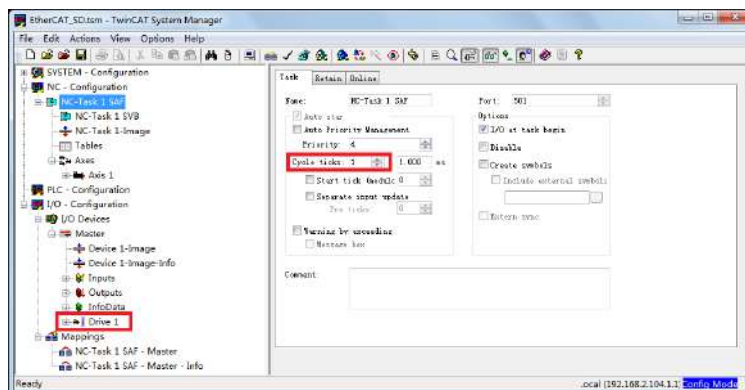


图表 44: 添加 box 到 EtherCAT 设备中

右键选择主机“Master”并选择“Append Box”。选择 BonMET SSx4 文件并单击确认。当弹出窗口提示是否附加 NC 配置时，选择“是 Yes”。

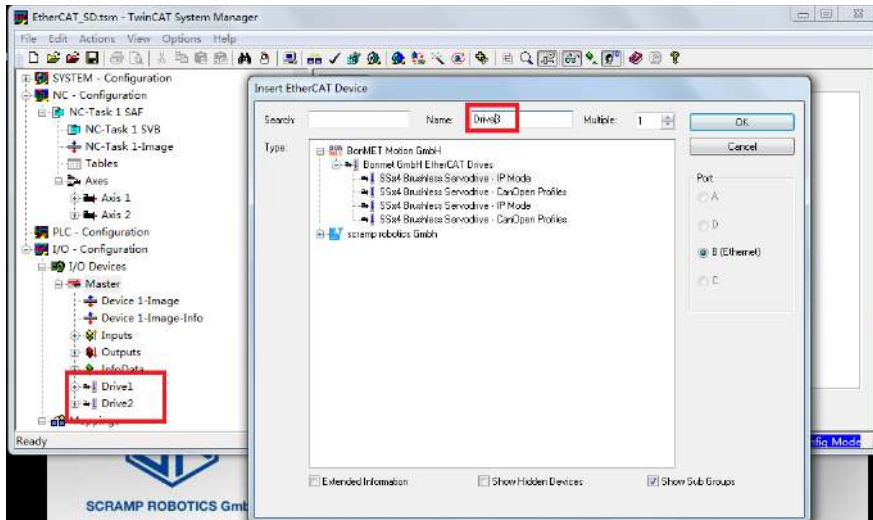


图表 45: 添加 XML 文件



图表 46: 设定周期时间为 1 毫秒

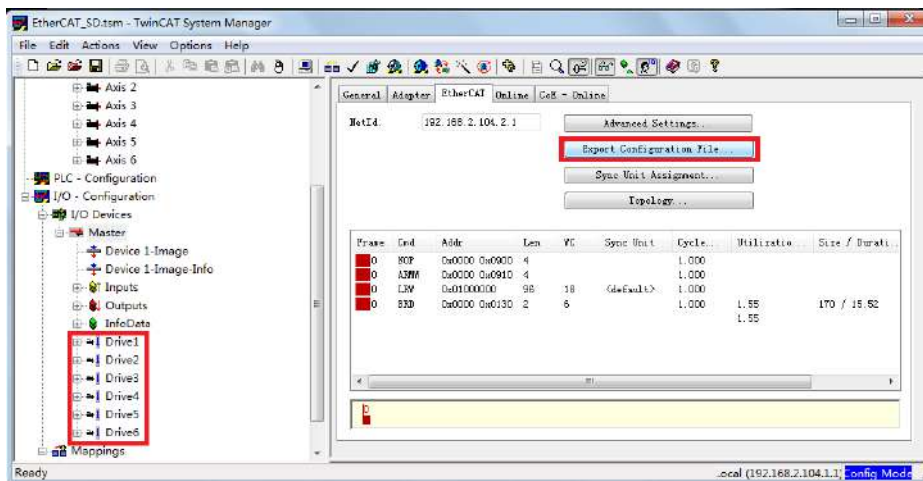
更改 box 名称为“Drive1”以匹配 Motion 文件中的命名。重复添加 box 到包含足够的运动轴。（在该样例中配置了 6 个驱动轴）。重命名每个轴以反映该轴的功能。（注意该名称要与 Motion 文件中每个轴的名称相匹配）



图表 47: 添加足够多的运动轴

选择 NC-Configuration > NC-Task 1 SAF 并设定插补时间(1ms)。

最后, 保存 Twincat 配置文件 (文件 File > 存储为 Save As), 选择 EtherCAT 主机 “Master” 下的 EtherCAT 选项。

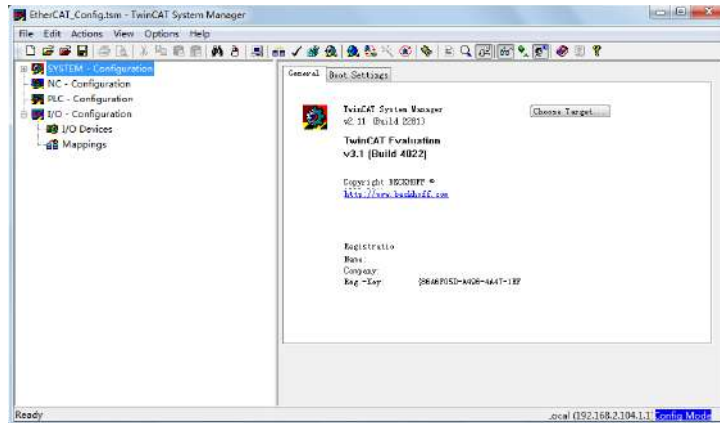


图表 48: 输出 EtherCAT_Config 文件

点击输出配置文件 “Export configuration file” 并存储为 XML 格式; 重命名为 ECAT_DefaultConfig.xml 并加载至机器人控制器。

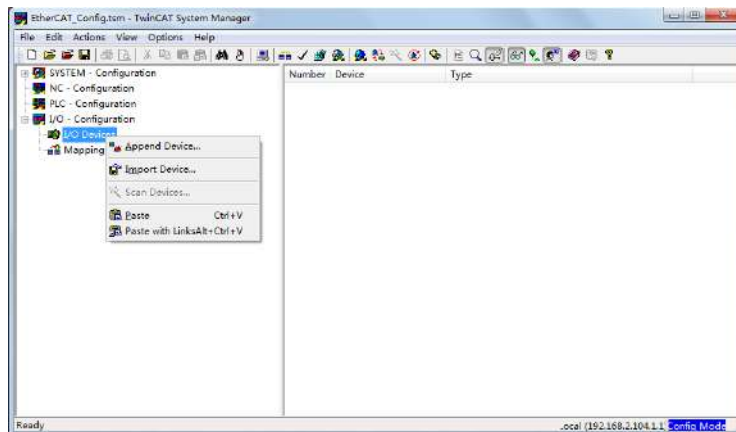
3.3. 四六轴伺服驱动器 ENI 文件生成步骤

打开 TwinCAT system manager 并点击新建 Edit > 空白文件 New

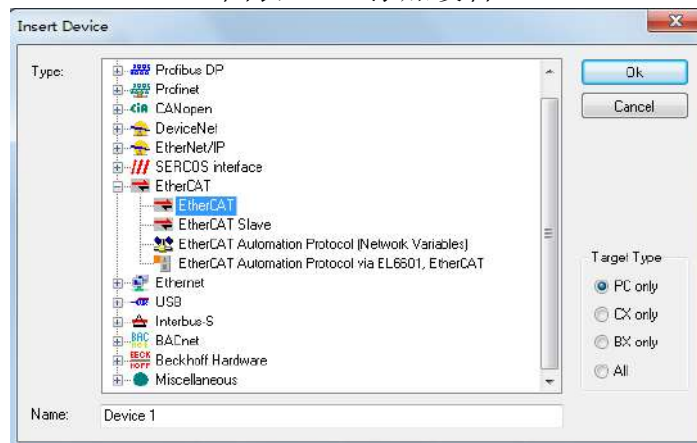


图表 49 打开 TwinCAT 文档

右键单击 I/O devices 并选择“添加设备 Append Device”。选择 EtherCAT > EtherCAT 并点击确认。

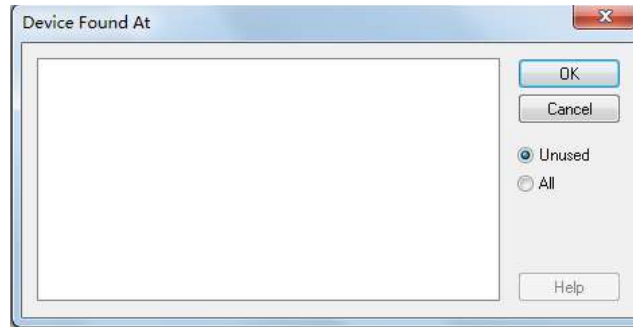


图表 50: 添加设备



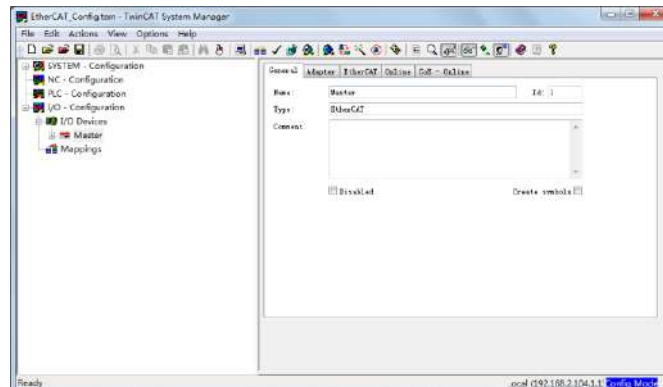
图表 51: 选择 EtherCAT/EtherCAT

如果配置过程中为连接设备，则弹出提示框，点击删除 cancel 跳过该步骤即可。



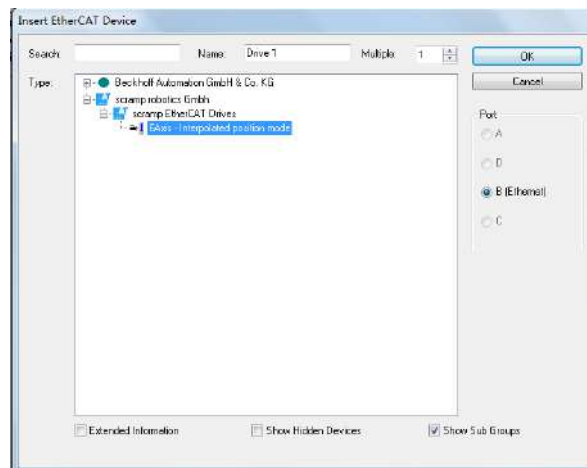
图表 52: 点击 *cancel* 跳过该步骤

重命名新设备为主机“Master”。



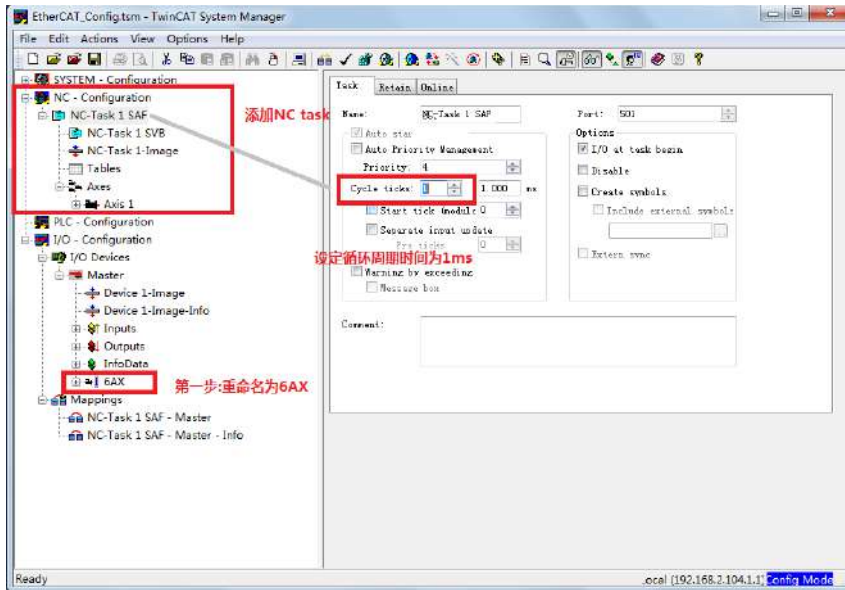
图表 53: 重命名新设备为 “Master”

右键点击主机 “Master” 并选择添加 Box“Append Box”。选择 Scrap robotics GmbH/6Axis – interpolated position mode 并点击确认,当弹出窗口提示附加 NC 配置文件时点击是“*Yes*”。



图表 54: 添加 *box* 到主机 “Master”

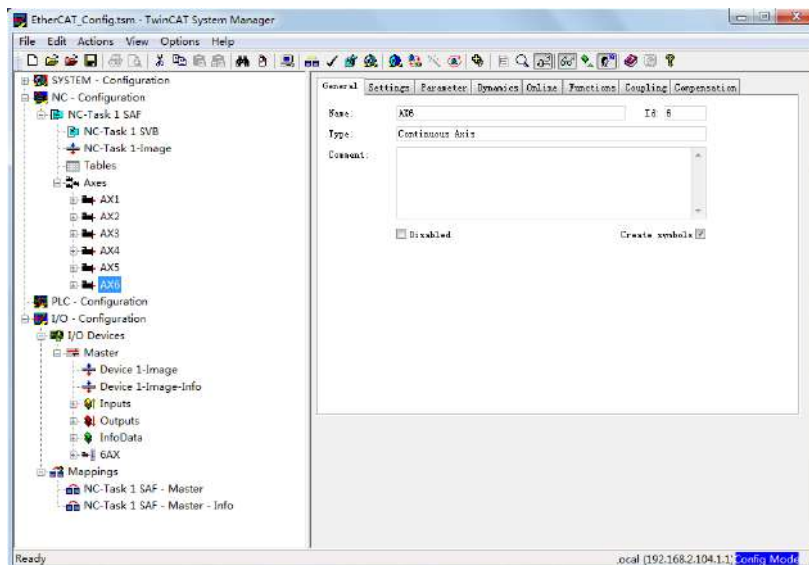
重命名所添加的 *box* 为 6AX, 保持该名称与 Mition_Config.xml 中的主名称相一致(Motion 文件将在下一章节介绍).设置 NC task 的插补时间为 1ms。



图表 55: 重命名所添加的 box 并设置 NC 任务

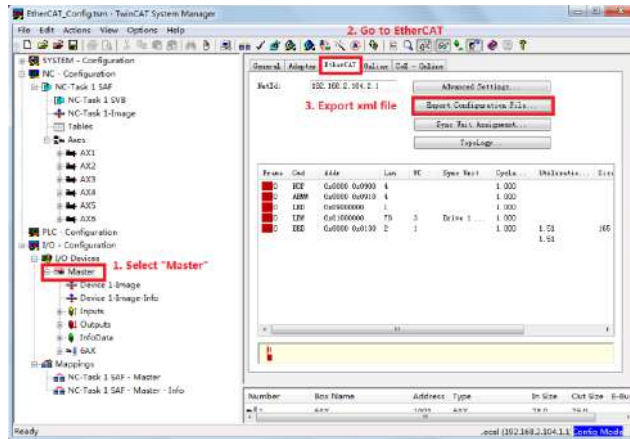
重复添加足够多的 box 来匹配对应数量的伺服驱动器(在该样例中仅使用了一个 6 轴伺服驱动器)。确保 box 名称与 Motion 文件相一致。

点击 NC task 并添加 6 个运动轴，命名为 AX1-6。



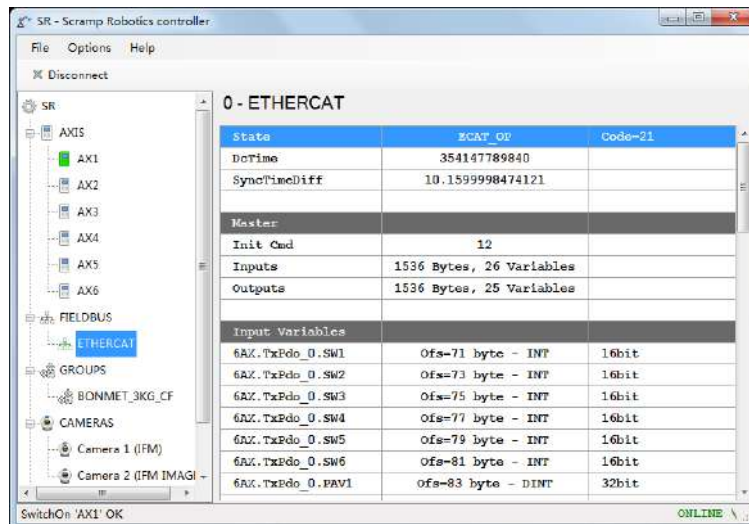
图表 56: 对“6AX”添加 6 个运动轴并命名

选择 EtherCAT 主机“Master”的右侧窗口中 EtherCAT 选项，点击导出配置文件 Export configuration file 并存储为 XML 格式；命名为 ECAT_DefaultConfig.xml 并上传至机器人控制器中。



图表 57: 导出配置文件

当客户完成了上述步骤，并将文件加载至控制器中后，打开 GUI / Ethercat,状态显示为 ECAT_OP. 请注意 EtheCAT 文件名称一定要与 Motion 文件名称保持一致，否则的话即使显示该状态，仍不能使能运动轴。



图表 58: 检查连接状态

4. 控制器配置文件：运动控制

4.1. XML 文件结构

运动配置文件使用可扩展标记语言 (XML) 的结构进行编写。XML 是一种可以同时被人类和机器可读的用于编译格式化文档的标志性语言。在机器人控制器内部中，“MOTION_DefaultConfig.xml”文件代表了用户的机器人控制设置：其中包含了轴配置信息、机器人的组成部分（动力模型和动态模型的描述）以及一些其他信息，例如工作区间速度、加速度和加速度限制、抵消以及机器人结构的偏移和平移等。

接下来，“MOTION_DefaultConfig.xml”文件将分成两个部分：运动轴和群组来进行介绍。

4.2. 驱动列表

这里是文件的第一个部分，包含了所有的驱动器配置信息。接下来的表格展示了在驱动器列表中一个的名为“AX1”对象的所有参数的结构样例：

```
<Drive>
<Name>AX1</Name>
<Type>COE402_EL</Type>
<Links>
<SW>6AX. TxPdo_0. SW1</SW>
<PAV>6AX. TxPdo_0. PAV1</PAV>
<VAV>6AX. TxPdo_0. VAV1</VAV>
<TAV>6AX. TxPdo_0. TAV1</TAV>
<CW>6AX. RxPdo_0. CW1</CW>
<TPOS>6AX. RxPdo_0. TPOS1</TPOS>
<VOFS>6AX. RxPdo_0. VOFS1</VOFS>
<TOFS>6AX. RxPdo_0. TOFS1</TOFS>
<OD_Offset>#x000</OD_Offset>
</Links>
<EncResolution>131072</EncResolution>
<GearRatioNum>121</GearRatioNum>
<GearRatioDen>1</GearRatioDen>
<PosScalingNum>360. 0</PosScalingNum>
<PosScalingDen>1. 0</PosScalingDen>
<SpeedScalingRatio>1. 0</SpeedScalingRatio>
<AccelScalingRatio>1. 0</AccelScalingRatio>
<PosInversion>FALSE</PosInversion>
<PosLimitEnable>TRUE</PosLimitEnable>
<PosLowerLimit>-170. 0</PosLowerLimit>
<PosUpperLimit>+170. 0</PosUpperLimit>
<MotorNominalTorque>0. 64</MotorNominalTorque>
<MaxSpeed>140. 0</MaxSpeed>
<ReducedSpeedPerc>100. 0</ReducedSpeedPerc>
<MaxAccel>1400. 0</MaxAccel>
<MaxDecel>1400. 0</MaxDecel>
<MaxJerk>14000. 0</MaxJerk>
```

```

<HomingMethod>35</HomingMethod>
<HomeOffset>0.0</HomeOffset>
<HomingSwitchSpeedPerc>10.0</HomingSwitchSpeedPerc>
<HomingIndexSpeedPerc>5.0</HomingIndexSpeedPerc>
<HomingAccelPerc>10.0</HomingAccelPerc>
<HomingTimeoutMs>30000</HomingTimeoutMs>
<UseAbsValid_Sw15>1</UseAbsValid_Sw15>
</Drive>

```

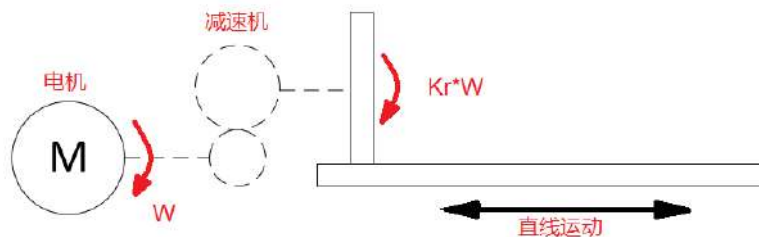
所有包含在指令<Drive> 和</Drive>之间元素都是用于引用到一个轴上，下列表格中解释了参数是如何使用的：

参数	描述	类型	数值实例
Name	运动轴的名字：必须与 ECAT 文件中定义的其中一个一致。	Char 字符	AX1
Type	告诉控制器轴是一个真实轴（COE402_EL）或只是一个内部虚拟轴（VIRTUAL）。一个虚拟轴并不能与驱动器进行通讯。指令 COE 仅适用于单轴伺服驱动器	AxisType	COE COE402_EL VIRTUAL
Links	声明四六轴驱动器每个激活轴的 TxPDO 和 RxPDO 地址		
EncResolutionEnc	该运动轴中编码器的分辨率	int32	131072
GearRatioNum	减速机参数比：电机转速	in32	121
GearRatioDen	减速机参数比：负载转速	int32	1
RotorInertia	电机转子转动惯量（不含减速器）	double	0.0035e-4
AxisStructure	告诉电机是否要转换成线性单位（例如置换为米），或转数（默认值）。	AxisStruct (PRISM, REV)	PRISM
PosScalingNum	应用减速机后的位置比例单位。	double	360.0
PosScalingDen	位置比例单位的负载端转数	int32	1.0
SpeedScalingRatio	增量/秒乘以驱动速度	double	1.0
AccelScalingRatio	增量/秒平方乘以驱动加速度	double	1.0
PosInversion	致反控制器要给予驱动器的位置值，反之亦然。	uint8	FALSE
PosLimitEnable	开启轴位限制的使用。	uint8	TRUE
PosUpperLimit	软限位上限。（变速箱接头的上限）	double	+170.0
PosLowerLimit	软限位下限。（变速箱接头的下限）	double	-170.0
MotorNominalTorque	电机驱动关节的额定转矩。	double	0.318
MaxSpeed	应用减速机后电机能达到的最高速度，单位/每秒。（电机转速*360/60/减速机）	double	140.0
ReducedSpeedPerc	以百分比为单位限制最高速度。	double	100.0
MaxAccel	应用减速机后轴能达到的最高加速度，单位/秒平方。（速度/加速时间）	double	1400.0

MaxDecel	应用减速机后轴能达到的最高减速度，单位/秒平方。（速度/减速时间）	double	1400.0
MaxJerk	应用减速机后轴能达到的最高加加速度值，单位/秒立方。（加速度/加速时间）	double	14000.0
HomingMethod	回零方式（CANOpen 规格）为轴设置新的零点位置所需的归位方法（CANopen 规范）	uint8	35
HomeOffset home	回零补偿，用于设置一个回零后的参考位置。可以添加到设置的回原点位置的单位偏移	double	0.0
HomingSwitchSpeedPerc	回零速度，最高速度的百分比。在回原点过程中使用的最大速度的百分比	double	10.0
HomingIndexSpeedPerc	回零速度，最高速度的百分比。在回原点过程中使用的最大速度的百分比	double	5.0
HomingAccelPerc	回零加速，最高加速的百分比。在回原点过程中使用的最大加速度的百分比	double	10.0
HomingTimeoutMs	回零过程超时设置，单位毫秒。	uint32	30000
UseAbsValid_Sw15	-	-	1

必须要重申一次，驱动器的名字与“ECAT_DefaultConfig.xml”文件中的某一驱动器名称一致是非常重要的。若名称没有被定义则通讯不能建立。

为了将轴虚拟化而不从控制器输出任何指令，TYPE（轴类型）必须被给予“VIRTUAL”指令而不是指定为真实轴的“COE/COE402_EL”指令。



图表 59: 轴、减速机和直线动作转换

轴反转参数必须依照设备动力模型所需求的方向进行正确设定，错误的设置可能导致警报或非预期的运动方式。

4.3. 电机群组列表

用户可以将驱动列表中所有的或一部分完成初始化的轴定义成群组。

```
<AxesGroupList>
  <AxesGroup>
    <Name>BONMET_3KG_CF</Name>
```

```
<Kinematic>
  <KinType>Anthrop6_Cf</KinType>
  <DriveNameList>
    <DriveName>AX1</DriveName>
    <DriveName>AX2</DriveName>
    <DriveName>AX3</DriveName>
    <DriveName>AX4</DriveName>
    <DriveName>AX5</DriveName>
    <DriveName>AX6</DriveName>
  </DriveNameList>
  <Params>
    <L1>0.32</L1>
    <L2>0.270</L2>
    <L2b>0.00</L2b>
    <L3>0.07</L3>
    <L4>0.299</L4>
    <L5>0.0785</L5>
    <L6>0.050</L6>
    <L7>0.00</L7>

    <b1>0</b1>
    <b2>0</b2>
    <b3>0</b3>
    <b4>0</b4>
    <b5>0</b5>
    <b6>0</b6>
    <f1>0</f1>
    <f2>0</f2>
    <f3>0</f3>
    <f4>0</f4>
    <f5>0</f5>
    <f6>0</f6>
    <f_w0_1>0.002</f_w0_1>
    <f_w0_2>0.002</f_w0_2>
    <f_w0_3>0.002</f_w0_3>
    <f_w0_4>0.002</f_w0_4>
    <f_w0_5>0.002</f_w0_5>
    <f_w0_6>0.002</f_w0_6>

    <m2>0</m2>
    <m3>0</m3>
    <m4>0</m4>
    <m5>0</m5>
    <m6>0</m6>

    <r2x>0</r2x>
    <r2y>0</r2y>
```

```
<r3x>0</r3x>
<r4x>0</r4x>
<r4z>0</r4z>
<r5x>0</r5x>
<r5y>0</r5y>
<r6y>0</r6y>
<r6z>0</r6z>

<I1yy>0</I1yy>
<I2zz>0</I2zz>
<I2xy>0</I2xy>
<I2yz>0</I2yz>
<I3xy>0</I3xy>
<I3xz>0</I3xz>
<I3yz>0</I3yz>
<I4yy>0</I4yy>
<I4xy>0</I4xy>
<I4xz>0</I4xz>
<I4yz>0</I4yz>
<I5xx>0</I5xx>
<I5yy>0</I5yy>
<I5xy>0</I5xy>
<I5xz>0</I5xz>
<I5yz>0</I5yz>
<I6xx>0</I6xx>
<I6yy>0</I6yy>
<I6zz>0</I6zz>
<I6xy>0</I6xy>
<I6xz>0</I6xz>
<I6yz>0</I6yz>
</Params>
</Kinematic>
<BaseOffset>
  <X>0.0</X>
  <Y>0.0</Y>
  <Z>0.0</Z>
  <RotA>0.0</RotA>
  <RotB>0.0</RotB>
  <RotC>0.0</RotC>
</BaseOffset>
<ToolOffset>
  <X>0.0</X>
  <Y>0.0</Y>
  <Z>0.0</Z>
  <RotA>0.0</RotA>
  <RotB>0.0</RotB>
  <RotC>0.0</RotC>
```

```

    </ToolOffset>
    <AnglesMode>ZYZ</AnglesMode>
    <MaxSpeed>0.4</MaxSpeed>
    <MaxAccel>4</MaxAccel>
    <MaxJerk>40</MaxJerk>
  </AxesGroup>
</AxesGroupList>
</Config>
</MotionConfig>

```

本部分是由<AxesGroup> 和 </AxesGroup>指令进行编译。动力结构包含了不同种类的参数被用于机械手结构的基础，在接下来的章节将讲述提供支持的动力结构以及提出参数设置样例。

参数	描述	类型	数值实例
Name	电机群组的名称。	char	BONMET_3KG_CF
Kinematic	包含了所有用户定义的机器人参数的结构。例如构成机器人制动部分的驱动器列表、几何参数和动态基数。	structure	**
BaseOffset	应用在机器人基座的坐标系转换(生效在动力结构之前)，由以下组成：直线位移 (X、Y、Z)，沿着主轴线旋转 (A、B、C)	structure	*
ToolOffset	应用在机器人终端的坐标系转换 (描述工具/终点的维度效应，生效在动力结构之后)，由以下组成：线性位移 (X、Y、Z)，沿着主轴线旋转 (A、B、C)	structure	*
MaxSpeed	工作区域工具/终点最高速度效应[m/s]，工具的最大工作空间速度/末端执行器[m / s]	double	1.0
MaxAccel	工作区域工具/终点最高加速度效应[m/s^2]；工具的最大工作空间加速度/末端执行器[m / s ^ 2]	double	10.0
MaxJerk	工作区域工具/终点最高加加速度效应[m/s^3]；工具的最大工作空间抖动/末端执行器[m / s ^ 3]	double	100.0
AnglesMode 角度模式	角度姿态 ABC 使用何种方式表示：可以是欧拉角 (ZYZ) 或是偏向、俯仰、横滚 (RPY)。	char	ZYZ
Coupling 耦合	耦合矩阵容许软件分离轴运动以及处理动力模型。是一个由轴数量决定维度的方形矩阵。	matrix	*

* are depicted in Figure 54

** see chapter 4.6

<pre> <BaseOffset> <X>0.0</X> <Y>0.0</Y> <Z>0.0</Z> <RotA>0.0</RotA> <RotB>0.0</RotB> <RotC>0.0</RotC> </BaseOffset> </pre>	<pre> <ToolOffset> <X>0.0</X> <Y>0.0</Y> <Z>0.0</Z> <RotA>0.0</RotA> <RotB>0.0</RotB> <RotC>0.0</RotC> </ToolOffset> </pre>	<pre> <Coupling>␣ <Row1>1.0;0.0;0.0;0.0</Row1>␣ <Row2>0.0;1.0;0.0;0.0</Row2>␣ <Row3>0.0;0.0;1.0;-0.4</Row3>␣ <Row4>0.0;0.0;0.0;1.0</Row4>␣ </Coupling>␣ </pre>
BaseOffset	ToolOffset	Coupling matrix for scara

图表 60: BaseOffset, ToolOffset 以及耦合矩阵

4.4. 角度模式

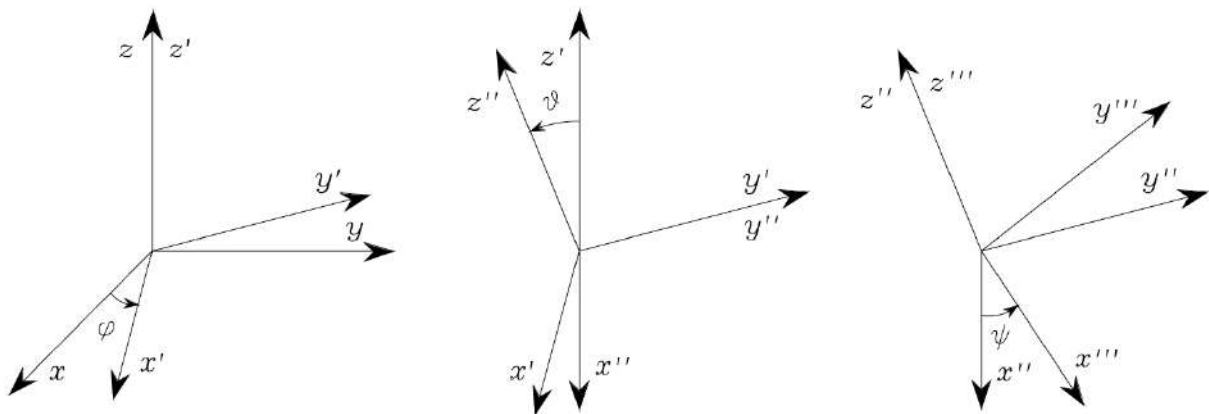
旋转矩阵给出了笛卡尔坐标系下工作区中由 9 个元素所组成的姿态，计算起始于两个可能的角度模式：欧拉角与偏向、俯仰、横滚。

a) 欧拉角

由三个独立的参数代表一个姿态构成的最小代表，可以得到一组三个角度的数据 $\Phi=[\varphi, \vartheta, \psi]$ 。ZYZ 型旋转由以下基础的旋转所构成：

- 围绕 **Z** 轴将坐标系旋转 φ 角。
- 接着，围绕坐标系的 **Y** 轴旋转 ϑ 角。
- 然后，围绕坐标系的 **Z** 轴旋转 ψ 角。

旋转过程如下图 60 所示：



图表 61: 欧拉角 ZYZ 旋转角度。

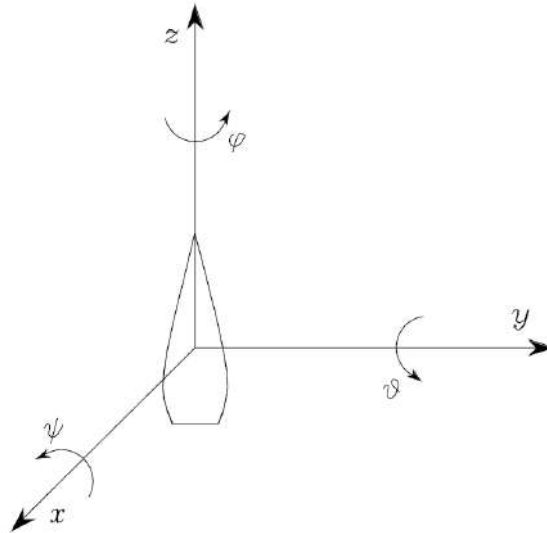
b) 偏向(Roll)、俯仰 (Pitch)、横滚 (Yaw)

欧拉角还可以借用航空学中的偏向、俯仰和横滚来描述。这种名为 ZYX 的角度通常用于描述沿着固定在目标物上的坐标系框架轴的旋转。

偏向、俯仰和横滚角的旋转结果可以从以下基本步骤中组合得出：

- 围绕 **Z** 轴旋转参考系 φ 角（偏向）。
- 围绕 **Y** 轴旋转参考系 ϑ 角（俯仰）。
- 围绕 **X** 轴旋转参考系 ψ 角（横滚）。

图 61 显示了 RPY（偏向，俯仰和横滚）角组成概念：



图表 62：偏向，俯仰和横滚角的组成概念。

c)表示角度的例子

以下两个例子都表示使用带球形手腕的六轴机器人所标定的零度位置，其代表相同的旋转姿态并使所有电机轴处于零位：

$$T = [X, Y, Z, 0.0^\circ, -90.0^\circ, -180.0^\circ]$$

如果用户希望旋转机器人的末端以平行于 XY 平面（例如在坐标向下的码垛环境时）则 ZYZ 型的姿态表述成：

$$T = [X, Y, Z, 0.0^\circ, -180.0^\circ, -180.0^\circ]$$

这种情况下腕关节的旋转控制可使用 ZYZ 或 RPY 角，请注意工作区域的角度并非由机械手的轴坐标表示。

4.5.耦合矩阵

它允许运动配置文件设置耦合矩阵，使得关于同一关节的多轴和旋转运动的匹配和不匹配成为可能：即，如果手腕关节旋转的同时发生平移，使用该矩阵可借助另一关节位置的运动来解耦其中一种运动形式。在此情形下，控制器总是看到彼此解耦的轴仅作出旋转或平移运动。耦合矩阵的数学结构展示如下：

	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6
Axis 1 =	a1*Axis1 +	a2*Axis2 +	a3*Axis3 +	a4*Axis4 +	a5*Axis5 +	a6*Axis6
Axis 2 =	b1*Axis1 +	b2*Axis2 +	b3*Axis3 +	b4*Axis4 +	b5*Axis5 +	b6*Axis6

Axis 3 =	c1*Axis1 +	c2*Axis2 +	c3*Axis3 +	c4*Axis4 +	c5*Axis5 +	c6*Axis6
Axis 4 =	d1*Axis1 +	d2*Axis2 +	d3*Axis3 +	d4*Axis4 +	d5*Axis5 +	d6*Axis6
Axis 5 =	e1*Axis1 +	e2*Axis2 +	e3*Axis3 +	e4*Axis4 +	e5*Axis5 +	e6*Axis6
Axis 6 =	f1*Axis1 +	f2*Axis2 +	f3*Axis3 +	f4*Axis4 +	f5*Axis5 +	f6*Axis6

从此表中可以看到耦合矩阵是由多个系数组成，这使得解耦矩阵可以表示为：

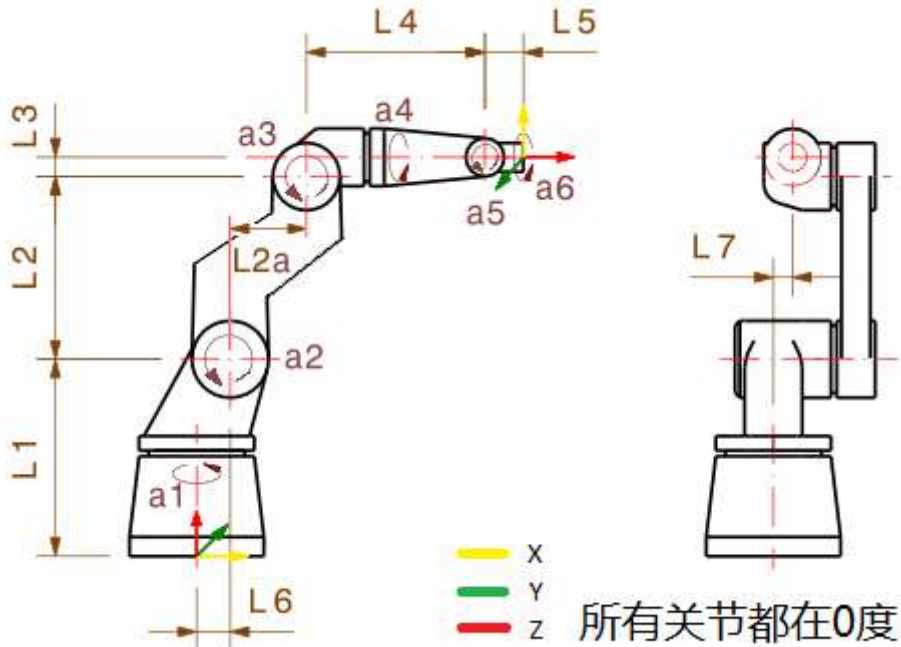
$$Coupling = \begin{bmatrix} a1 & a2 & a3 & a4 & a5 & a6 \\ b1 & b2 & b3 & b4 & b5 & b6 \\ c1 & c2 & c3 & c4 & c5 & c6 \\ d1 & d2 & d3 & d4 & d5 & d6 \\ e1 & e2 & e3 & e4 & e5 & e6 \\ f1 & f2 & f3 & f4 & f5 & f6 \end{bmatrix}$$

矩阵必须被设置在运动配置文件中电机群组的位置，矩阵中的每一行对应一个电机轴而且系数之间用“;”和“,”做分隔。如果配置中一行里的轴数量少于电机群组中的设置时将会产生报错且动力模型无法完成初始化。如果耦合矩阵并没有被定义在配置文件中，则控制器会生成一个参数为0的初始表格。这表示动力学计算时没有任何耦合参与。

4.6. 为电机群组配置运动模型

a) 六轴机器人

这种形态的机器人是最著名且广泛使用的结构。是由三轴机械臂沿着笛卡尔坐标系 (X,Y,Z) 进行位移，加上三轴球形腕关节可以沿着基坐标主轴线进行各种姿态的旋转 (A,B,C)。动力学模型如下：



图表 63: 六轴机器人的参数标示

动力模型中各个臂之间所需要计算的几何长度标为 L1-L6, 单位为米[m]。电机关节名称 a1-a6 有固定的旋转方向，图表中机械手的姿态被默认为所有电机轴位置为零。机器人控制器中提

供两种运动学求解方式，一种是递归的，另一种是闭合形式的，这给予了用户在机器人应用中的灵活性。

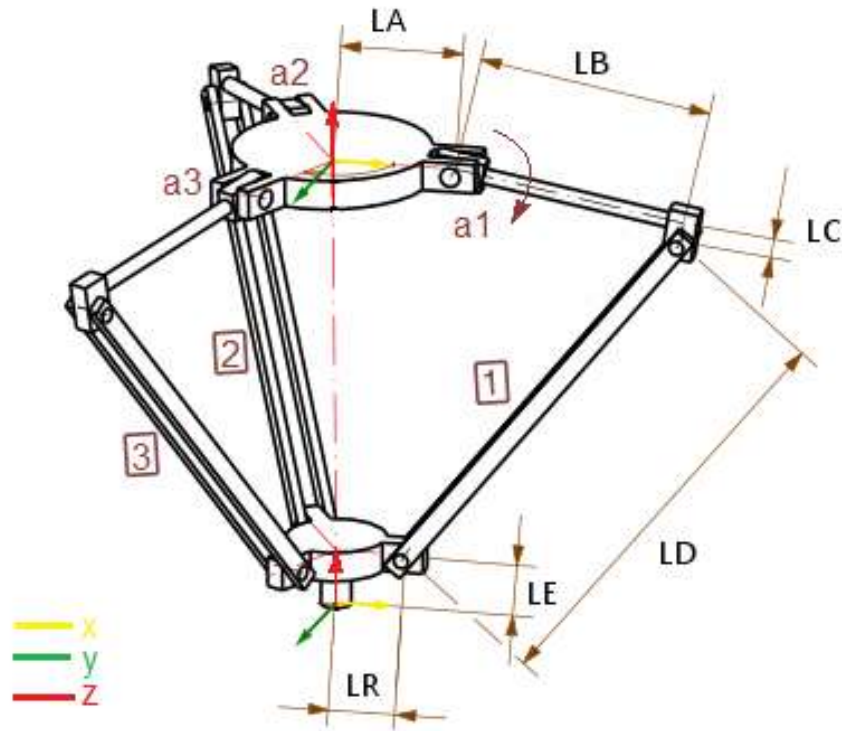
参数	描述	类型 (数值样例)
KinType	描述使用何种的动力学模型。	递归运算 Anthrop6_Rec 封闭式运算 Anthrop6_Cf
DriveNameList	机械手的驱动器列表，必须是已经在驱动列表以及 ECAT 配置文件中已有的轴。	<DriveNameList> <DriveName>Drive1</DriveName> <DriveName>Drive2</DriveName> <DriveName>Drive3</DriveName> <DriveName>Drive4</DriveName> <DriveName>Drive5</DriveName> <DriveName>Drive6</DriveName> </DriveNameList>
Params 参数	用于构造动力模型和动态模型的参数值。	
Geometric parameters 几何参数	臂长参数，单位米[m]。	<L1>0.320</L1> <L2>0.270</L2> <L2b>0.00</L2b> <L3>0.070</L3> <L4>0.299</L4> <L5>0.0785</L5> <L6>0.05</L6> <L7>0.00</L7>
Dynamic parameters 动态参数	用于构造机械手的动力模型所需的动态参数： Viscous friction: 轴的粘滞摩擦[与维度无关] Static friction: 轴的静态摩擦[与维度无关] Dynamic friction: 轴的动态摩擦[与维度无关] Masses: 组成机器人每段手臂的质量[Kg] Center of mass's position: 每一段手臂（包含电机）的质量中心在笛卡尔坐标系上以 X、Y、Z 表示，单位米[m]。 Inertia tensor's elements: 每一段手臂质量中心的主要和偏差惯性。（Inertia tensor's elements: 每个手臂质心位置相对于轴 X, Y, Z 的主惯性和偏离惯性项。）	Viscous friction <b1>0.0</b1>到<b6>0.0</b6> Static friction <f1>0.0</f1>到<f6>0.0</f6> Dynamic friction <f_w0_1>0.0</f_w0_1>到 <f_w0_6>0.0</f_w0_6> Masses <m1>0.0</m1>到<m6>0.0</m6> Center of mass's position（二轴为例） <r2x>0.0</r2x> <r2y>0.0</r2y> <r2z>0.0</r2z> Inertia tensor's elements（二轴为例） <l2xx>0.0</l2xx> <l2yy>0.0</l2yy> <l2zz>0.0</l2zz> <l2xy>0.0</l2xy> <l2xz>0.0</l2xz> <l2yz>0.0</l2yz>

须注意电机旋转方向的选择必须按照用户选择的电机群组动力模型的方向来设置。例如斯卡普 3 公斤六轴拟人机械手的反转表格如下：

Drive1	Drive2	Drive3	Drive4	Drive5	Drive6
FALSE	FALSE	FALSE	FALSE	TRUE	TRUE

b)Delta (并联) 机器人 (3 轴)

Delta 并联机器人结构由三个臂、三个轴关节以及每个手臂由两段臂长连接组成的运动平台。



图表 64: Delta 机器人几何参数以及关节旋转角度

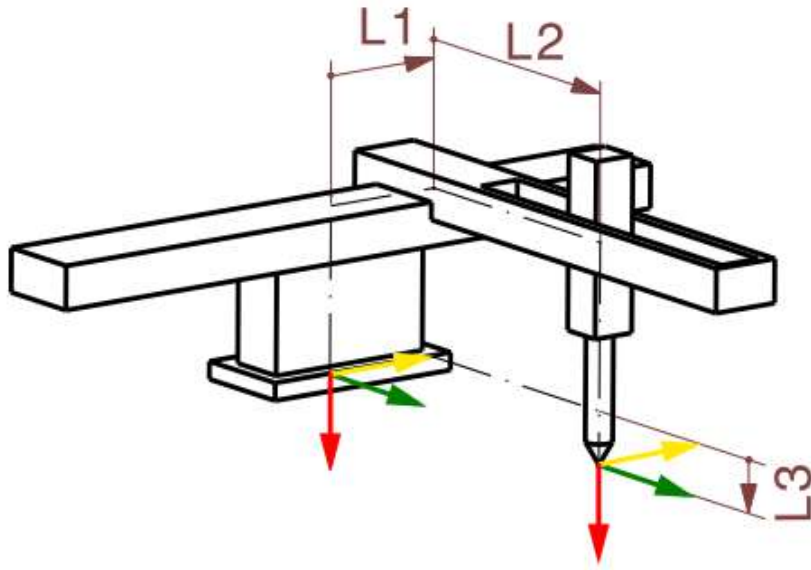
参数	描述	类型 (数值样例)
KinType	描述使用何种的动力学模型。必须选择一个默认的值才能被软件正确读取。	Closed-form solver: 封闭解算器 Delta3_Cf
DriveNameList 驱动器名称列表	机械手的驱动器列表，必须是已经在驱动列表以及 ECAT 配置文件中已有的轴。	<DriveNameList> <DriveName>Drive1</DriveName> <DriveName>Drive2</DriveName> <DriveName>Drive3</DriveName> </DriveNameList>
Params 参数	用于构造动力模型和动态模型的参数值。	
Geometric parameters 几何参数	臂长参数，单位米[m]。	<LA>0.127</LA> 基地半径 <LB>0.180</LB> 短臂臂长 <LC>0.000</LC> 肘部位移 <LD>0.450</LD> 长臂臂长 <LE>0.000</LE> 工具位移 <LR>0.0625</LR> 工具半径
Dynamic parameters 动态参数	动态参数用于构造机械手的动力模型： Viscous friction: 轴的粘滞摩擦[与维度无关] Static friction: 轴的静态摩擦[与维度无关] Dynamic friction: 轴的动态摩擦[与维度无关] Masses: 组成机器人每段手臂的质量[Kg]	Viscous friction 0.0 Static friction <F>1.36</F> Dynamic friction <F_W0>0.0</F_W0> Masses <MT>0.220</MT> <MR>0.287</MR> <ME>0.250</ME> <MC>0.020</MC>

Delta 机器人的轴运动反转须按照如下设定以避免不在预期的动作:

Drive1	Drive2	Drive4
FALSE	FALSE	FALSE

c) 直角坐标系机器人 (3 轴)

直角坐标系机器人：使用三个关节沿着 X、Y、Z 方向运动，如下所示：



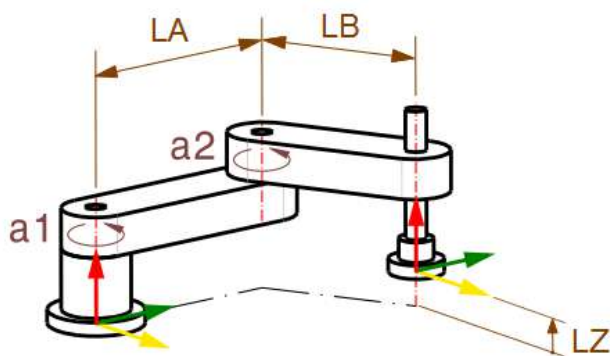
图表 65: 直角坐标系机器人

笛卡尔坐标机器人需要被设置的参数如下：

参数	描述	类型 (数值样例)
KinType	描述使用何种的动力学模型。必须选择一个默认的值才能被软件正确读取。	Closed-form solver: 封闭解算器 Cartesian (Recursive only) 笛卡尔(递归)
DriveNameList	机械手的驱动器列表，必须是已经在驱动列表以及 ECAT 配置文件中已有的轴。	<DriveNameList> <DriveName>Drive1</DriveName> <DriveName>Drive2</DriveName> <DriveName>Drive3</DriveName> </DriveNameList>
Params	用于构造动力模型和动态模型的参数值。	
Geometric parameters	臂长参数，单位米[m]。	<LA>0..0</LA> <LB>0..0</LB> <LC>0.0</LC>

d) SCARA 水平多关节机器人 (4 轴)

带有平面平移手臂和可旋转平移腕部的串联机器人，其沿着 Z 轴上下运动并且再次控制刀具围绕 Z 轴的方向。SCARA 机器人有三种类型的模型可供选择：判别类型的方式是看上下的柱体关节是在一、三或第四个位置。



图表 66: SCARA 机器人参数

Prismatic 轴在第一位置时机械手的所有运动都沿着 Z 轴进行。在第三、第四位置时，旋转轴被转换为绕着 Z 但是保持第一和第二轴为平行手臂的方式执行。动力模型自动从配置文件中辨认控制柱轴（“PRISM”）的驱动器的基坐标。在一个模型中只可配置一个 **Prismatic** 轴，否则动力模型会报错。下表展示了设置笛卡尔坐标系电机群组所需的参数：

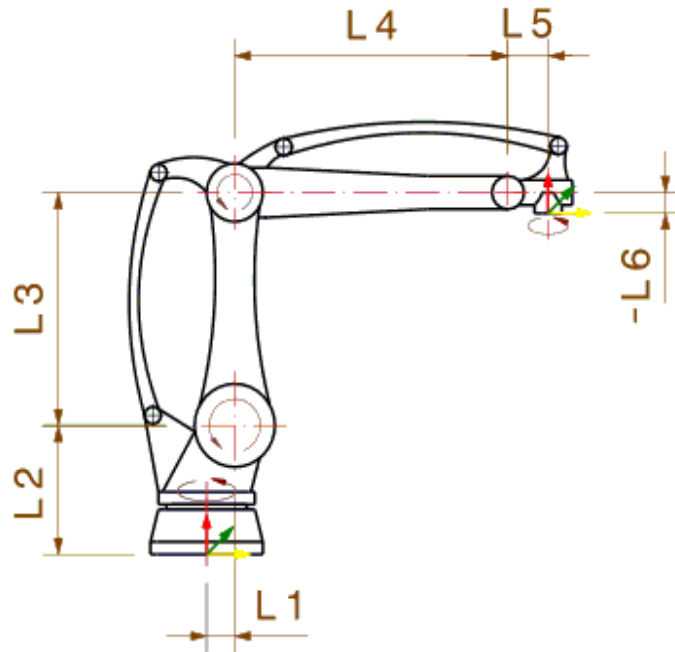
参数	描述	类型 (数值样例)
KinType	描述使用何种的动力学模型。必须选择一个默认的值才能被软件正确读取。	Closed-form solver: 封闭解算器 Scara4_Cf Scara4
DriveNameList	机械手的驱动器列表，必须是已经在驱动列表以及 ECAT 配置文件中已有的轴。	<DriveNameList> <DriveName>Drive1</DriveName> <DriveName>Drive2</DriveName> <DriveName>Drive3</DriveName> <DriveName>Drive4</DriveName> </DriveNameList>
Params	用于构造动力模型和动态模型的参数值。	
Geometric parameters	臂长参数，单位米[m]。	<LA>0.300</LA> <LB>0.300</LB> <LZ>0.220</LZ>
Dynamic parameters	动态参数用于构造机械手的动力模型： Static friction: 轴的静态摩擦 [与维度无关] Dynamic friction: 轴的动态摩擦 [与维度无关]	Static friction <f1>0.0</f1> 至 <f6>0.0</f6> Dynamic friction <f_w0_1>0.0</f_w0_1> 至 <f_w0_6>0.0</f_w0_6>

耦合矩阵必须被纳入计算，因为旋转轴在一些状态下（取决于机械手的机械结构）会同时影响到旋转。在这种情况下旋转轴需要根据柱轴作出补偿。**Delta** 机械手的驱动反转必须依照如下进行设定以避免不在预期的动作：

Drive1	Drive2	Drive3	Drive4
TRUE	TRUE	TRUE	TRUE

e)码垛机器人 (4 轴)

该机器人具有类似于六轴机器人的结构：其由沿着笛卡尔位移（ X ， Y ， Z ）的运动的三关节臂机构组成，但是腕部仅由一个关节构成，该关节给出额外的旋转 围绕 Z 轴。由于其特殊的机械结构，末端执行器端子总是保持平行于 X - Y 平面，所以手腕只可以围绕 Z 轴旋转。该机器人的动力模型如下：



图表 67: 码垛机器人臂长参数

在机器人动力模型的计算中需要的手臂的几何长度中用 LX 表示，单位为米[m]。关节旋转方向用“ aX ”标记。图中的姿态假定机器人所有关节处于零位置。与六轴机器人不同，码垛机器人不能假设手臂的不同配置以到达同一点：仅有两种可用配置由第四关节（腕关节）限定，并且彼此围绕沿 Z 的 180° 旋转而彼此不同轴。

参数	描述	类型 (数值示例)
DriveNameList 驱动器列表	构成机器人驱动部分的驱动器列表必须使用与 DriveList 部分和 ECAT 配置文件中已有的轴	<pre><DriveNameList> <DriveName>Drive1</DriveName> <DriveName>Drive2</DriveName> <DriveName>Drive3</DriveName> <DriveName>Drive4</DriveName> </DriveNameList></pre>
Params 参数	用于构造动力模型和动态模型参数值的	
Geometric parameters 几何参数	臂长参数，单位米[m]。	<pre><L1>0.420</L1> <L2>0.90</L2> <L3>1.300</L3> <L4>1.300</L4> <L5>0.300</L5> <L6>0.300</L6></pre>

要注意的是，驱动器的反转必须根据用户想要使用的电机群组动力来设置。

Drive1	Drive2	Drive3	Drive4
FALSE	FALSE	FALSE	FALSE

5. 控制器配置文件：机器人视觉

作为运动配置, 视觉参数也须在 XML 文件中配置。在 XML 文件中的根目录标签<VisionCon>包含了相机列表。这个列表中的每个元素都是相机<camera>标签中的相机节点。

```
<Camera>
  <Name>Camera 1 (IFM)</Name>
  <Type>IFM</Type>
  <Protocol>IPv4_TCP</Protocol>
  <IP>192.168.80.100</IP>
  <Port>50010</Port>
  <TimeoutMs>1000</TimeoutMs>
  <OperatingDistance>0.46</OperatingDistance>
  <FocalLengthX>1050.233</FocalLengthX>
  <FocalLengthY>1052.020</FocalLengthY>
  <CenterX>348.488</CenterX>
  <CenterY>210.551</CenterY>
  <RadialDistortCo1>0.039</RadialDistortCo1>
  <RadialDistortCo2>-0.790</RadialDistortCo2>
  <ImageWidth>640</ImageWidth>
  <ImageHeight>480</ImageHeight>
  <ImageDepth>8</ImageDepth>
  <ImageChannel>1</ImageChannel>
  <PoseTranslation>-1.405414 -0.2401229 -0.02921262</PoseTranslation>
  <PoseRotation>0.07078827 -0.01720034 0.02894594</PoseRotation>
  <DefaultApplication>0</DefaultApplication>
  <Application>
    <Name>IFM_Application</Name>
    <Algorithm>REMOTE</Algorithm>
    <AlgorithmData>
      <Width>5</Width>
      <Height>8</Height>
    </AlgorithmData>
  </Application>
  <Application>
    <Name>DetectChessboard</Name>
    <Algorithm>CHESSBOARD</Algorithm>
    <AlgorithmData>
      <Width>5</Width>
      <Height>8</Height>
    </AlgorithmData>
  </Application>
  <Application>
    <Name>DetectCoins</Name>
    <Algorithm>GHTBALL</Algorithm>
    <AlgorithmData>
      <TemplImageFile>Vision_Source/templ.jpg</TemplImageFile>
      <TemplDataFile>Vision_Source/templData.xml</TemplDataFile>
    </AlgorithmData>
  </Application>
</Camera>
```

```

    <CannyLowThresh>50</CannyLowThresh>
    <CannyHighThresh>100</CannyHighThresh>
    <MinDist>100</MinDist>
    <Levels>360</Levels>
    <Depths>2</Depths>
    <MaxBufferSize>1000</MaxBufferSize>
    <VotesThresh>20</VotesThresh>
  </AlgorithmData>
</Application>
</Camera>

```

5.1. 相机列表

在每一个相机节点中，参数包含了相机的基本信息、通讯、模型参数以及应用信息。

参数	描述	类型 (数值样例)
Name 名称	驱动的名称：必须与“ECAT_DefaultConfig.xml”文件中声明的其中一个名称一致。	Camera 1
Type 类型	选择控制器支持的相机类型，现在支持的相机列在表 4.1.2 中。	IFM
Protocol (optional) 协议 (可选)	使用何种协议与远端相机通讯。 使用远程摄像机时的通讯协议类型	IPv4_TCP
IP (optional)	使用 TCP/IP 协议时的 IP 地址。	192,168,80,100
Port (optional) 端口 (可选)	使用 TCP/IP 协议时的端口号。	50010
TimeoutMs	通讯超时设定	1000
OperatingDistance 操作距离	镜头到工作区间的距离，单位毫米[mm]。	10,0
FocalLength X	沿 X 方向上的焦距，单位像素。	1050.233
FocalLength Y	沿 Y 方向上的焦距，单位像素。	1052.020
Center X 中心	拍摄区域的中心点，单位像素。	348.488
Center Y 中心	拍摄区域的中心点，单位像素。	210.551
RadiaDistortCo[1,2]	弧形畸变系数/弧度失真的系数	0.039, /-0.79
ImageWidth 图像宽度	相机图片宽度，单位像素。	640
ImageHeight 图像高度	相机图片高度，单位像素。	480
ImageDepth 图像深度	像素位深，一个像素中包含的一个通道里有多少位[bit]。位数包含一个通道的像素值	8
ImageChannel 图像频道	一个像素中有几个通道，一般来说灰度值只有一个通道而色彩有三个通道。	1
PoseTranslation 翻转	将相机转换为世界坐标系的向量。 相机在世界坐标系中的翻转矢量	-1.4 -0.24 -0.029
PoseRotation 旋转	世界坐标系旋转角度的向量。 相机在世界坐标系中的旋转矢量	0.07 -0.017 0.029

DefaultApplication 默认应用程序	选择默认应用。 默认应用程序编号	1
Application List 应用程序列表	应用程序数据结构。	请看章节 5.2

控制器所支持的相机类型如下所列。

相机类型	通讯方式	智能相机	数字 I/O	彩色
IFMO2D222	TCP/IP	YES	YES	NO

5.2.应用列表

每个相机节点包含了一个或多个应用节点（使用<application>标签）描述每个应用的参数。

参数	描述	样例
Name	应用名称	IFM_APP
Algorithm 算法	算法类型。如果相机是智能型且可以自己处理图像则算法可以设置成“REMOTE”。	CHESSBOARD
AlgorithmData 算法数据	相机在世界坐标系旋转角度的向量。 相机在世界坐标系中的旋转矢量	<AlgorithmData> <Width>5</Width> <Height>8</Height> </AlgorithmData>

应用程序的算法可以由用户从以下列表中选择，不同的算法类型具有其对应的算法数据结构。

算法类型	描述	数据结构
REMOTE 远程	触发智能相机（例如 IFM）处理图像和检索结果。	应用设置一般是由第三方软件完成，因此算法数据可以被忽略。
CHESSBOARD 棋盘	检测黑白棋盘 定期检测棋盘	参数值中包含棋盘的宽、高和其中方块的数量。

6.控制器配置文件: HANDS

只有在使用应用程序中的辅助轴时，才需要附加配置文件。在 HANDS 配置文件中定义辅助轴的所有参数，如图 67 所示。

```

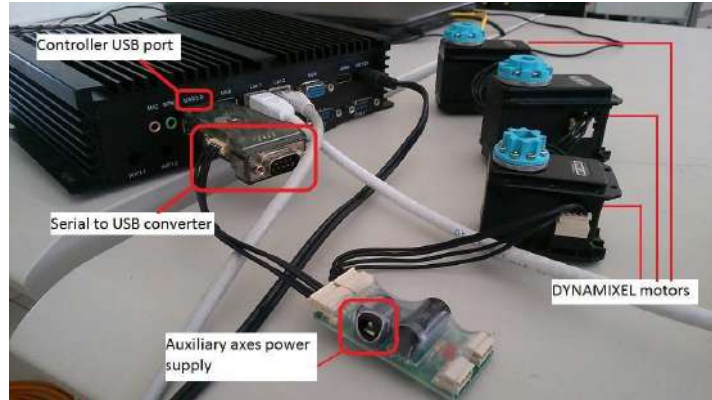
<?xml version="1.0" encoding="ISO-8859-1"?>
- <HandsConfig xsi:noNamespaceSchemaLocation="HANDS_Config.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Chain>
    <Proto>Dynamixel</Proto>
    <NameChain>G1_Chain</NameChain>
    - <Com>
      <Type>Serial</Type>
      <IfName>Rs485</IfName>
      <Device>/dev/ttyUSB0</Device>
      <Baudrate>1000000</Baudrate>
    </Com>
    - <Motors>
      - <Motor>
        <Name>grip_angle</Name>
        <Id>1</Id>
        <CalibPos>2067</CalibPos>
      </Motor>
      - <Motor>
        <Name>Root</Name>
        <Id>2</Id>
        <CalibPos>2888</CalibPos>
      </Motor>
      - <Motor>
        <Name>Tip</Name>
        <Id>3</Id>
        <CalibPos>1717</CalibPos>
      </Motor>
    </Motors>
  </Chain>
</HandsConfig>

```

图表 68: HANDS 配置文件示例

6.1. 结构

类型	描述	样值
Chain	每个电机群组都必须以此来声明	<Chain>...</Chain>
Proto	包含所使用电机的类型	Dynamixel
NameChain	电机群组的名称 (用在 Motion_Config.xml)	G1_Chain
Com	用以定义通讯方式	<Com>...</Com>
Type	通讯的类型	Serial
IfName	通讯协议名称	RS485
Device	连接的设备	/dev/ttyUSB0
Baudrate	波特率	1000000
Motors	包含所有使用的电机	<Motors>...</Motors>
Motor	声明电机	<Motor>...</Motor>
Name	电机的名称	grip_angle
Id	电机编号	1
CalibPos	零点位置	2067



图表 69: 三个 DYNAMIXEL 辅助轴与机器人控制器的连接示例

6.2. MOTION_Config 附加要求

为了使用机器人应用程序中的辅助轴，必须在上述 MOTION 配置文件中包含一些附加字段。首先，在 MOTION 配置文件的 DriveList（驱动列表）中，用户必须为每个电机辅助轴链内添加新的参数，在上面的例子中，三个辅助轴被配置在 HANDS 配置文件，所以三个驱动必须添加内部运动配置文件的 drivelist。

```

- <DriveList>
  - <Drive>
    <Name>grip_angle</Name>
    <Type>DYNAMIXEL</Type>
    <DynChain>G1_Chain</DynChain>
    <DynMotor>grip_angle</DynMotor>
    <EncResolution>4096</EncResolution>
    <PosScalingNum>360.0</PosScalingNum>
    <PosScalingDen>1.0</PosScalingDen>
    <PosInversion>FALSE</PosInversion>
    <PosLimitEnable>FALSE</PosLimitEnable>
    <PosLowerLimit>0.0</PosLowerLimit>
    <PosUpperLimit>87.0</PosUpperLimit>
    <MotorNominalTorque>1.0</MotorNominalTorque>
    <MaxSpeed>402.0</MaxSpeed>
    <MaxAccel>4020.0</MaxAccel>
    <MaxDecel>4020.0</MaxDecel>
    <MaxJerk>40200.0</MaxJerk>
    <MaxTorque>1</MaxTorque>
  </Drive>

```

图表 70: MOTION 配置文件中包含附加驱动器的示例

在新的驱动器条目中必须指定驱动器名称，电机类型，其所属的链（DynChain）和电机名称（DynMotor）。请注意，这些信息必须与 HANDS 配置文件中描述的相同！

之后，辅助轴列表必须添加到 MOTION 配置文件的 AxesGroupList 字段内。在图 5.2.2 所示的示例中，列表已添加到 Kinematic 和 BaseOffset 字段之间。

```

        <IOyz>u</IOyz>
    </Params>
</Kinematic>
- <AuxAxesList>
    <DriveName>grip_angle</DriveName>
    <DriveName>Root</DriveName>
    <DriveName>Tip</DriveName>
</AuxAxesList>
- <BaseOffset>
    <X>0.0</X>

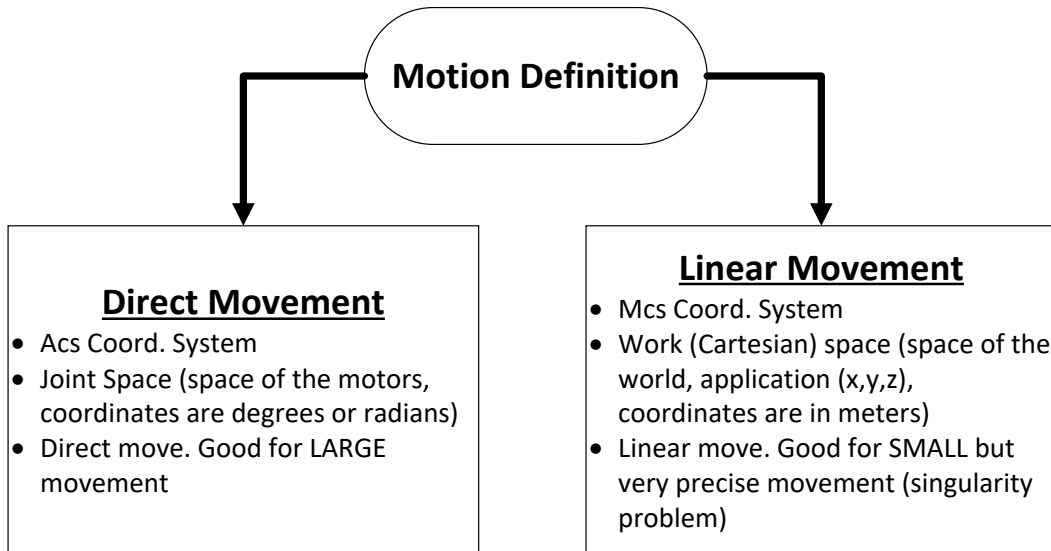
```

图表 71: 辅助轴列表

在 AuxiliaryAxesList (AuxAxesList) 中，必须插入驱动器名称 (DriveName)。驱动器名称必须与上述“Drive (驱动器) -> Name (名称)”字段中指定的相同。关于与辅助轴相关的 PLC 块的使用，请参见本手册的后续章节。

7. 轨迹规划

在接下来的一章中，将解释在控制器上可用的轨迹规划和路径生成的主要原理。该机器人的运动可以分为两个主要类型：电机空间下规划的 **DIRECT** 运动，（接合空间 -> **ACS** 坐标系），并且不考虑所规划运动的工作空间可行性，或在笛卡尔空间（真实世界，（X, Y, Z）空间 -> **MCS** 坐标系中）规划的线性运动，。



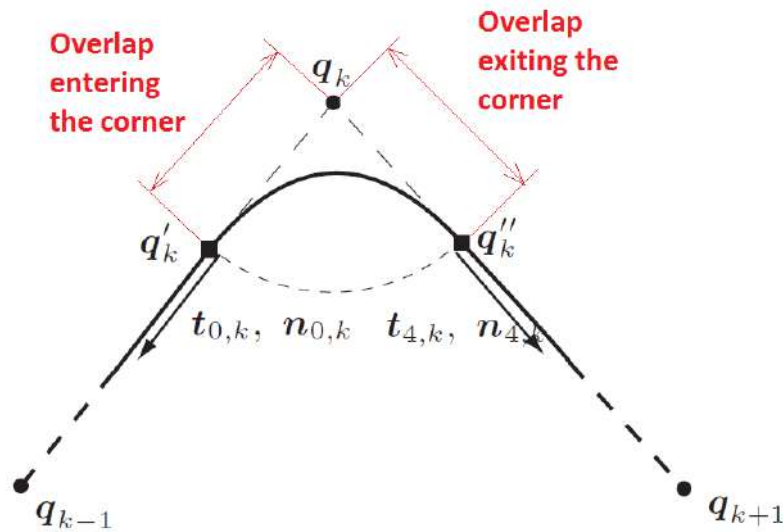
图表 72: 运动类型

对于长距离的运动，应优选使用 **DIRECT** 运动，因为它允许避免典型的线性运动的奇点问题。运动规划基于电机空间（度/弧度）。为了规划生成应用空间（工作空间 ->笛卡尔空间）中的精确运动，则必须使用线性运动，同时考虑奇点的问题（在运动学模型中处理）。为了避免在每个运动段的结束处的速度停止，可以以 **blending** 方式进行混合，使得可以借助微小地路径变动重新安排路径，以防止机器人的停止并且保持有下一个运动的所必需地速度。依据于运动的类型，有几种混合运动的可能性（直接，线性和它们之间的组合）。

7.1. 混合 Mcs 下的两种运动:线/圆到线/圆

为了在工作空间（Mcs 坐标系）中连接两个运动，根据上一个和下一个运动方式产生四种可能的情况:

- Line --> Line 线->线
- Line --> Circle 线->圆
- Circle --> Circle 圆->圆
- Circle --> Line 圆->线



图表 73: Mcs 模式下的两条直线运动的混合

定义 **blending** 进程的关键参数是 **OVERLAP**: 代指当前运动长度和下一运动长度的相 **blending** 的起始和结束点的百分比。**OVERLAP** 的值可以从 0 变化到 2, 其中 0 表示没有混合, 2 表示完全混合 (从当前运动的一半开始并且以下一个运动的一半结束)。

OVERLAP 的百分比计算如下, (ovl_par 为系数的值):

当前运动: $\text{overlap_actual_move} := 1.0 - ((\text{ovl_par} * 0.5) / 2.0)$;

下个运动: $\text{overlap_next_move} := (\text{ovl_par} * 0.5) / 2.0$;

这意味着, 使用混合运动参数为 1 时的实际运动的连接点是实际长度的 75%, 而结束点是下一个长度的 25%。参数 2.0 (所允许的最大值) 意味着混合实际和下一个长度的 50%。

Blending 进程也可以发生在圆形运动和线性或圆形之间, 其概念相同。

7.2. 混合两个 Acs 运动: 直接运动到直接运动

借助于连接两个直接移动 (在 Acs 坐标系中) 的方式, 可以避免在完成一个运动后机器人的停止。为此, 重新安排电机的轨迹, 依据于较慢的那个路径来百分化所有其他路径, 并以这种方式使各个电机周处于同步, 开始和结束全部同步。重新计划的目的是在考虑所定义的速度参数, 如加速度, 减速度, jerk 等值的情况下, 保证大部分阶段都能维持在一个恒定速度。

7.3. 混合 Acs / Mcs 到 Mcs / Acs 运动: 多重混合运动

运动与轨迹规划系统也可以对多种运动进行混合: Acs / Mcs 和 Mcs / Acs.

a) Blending 开始于 Acs(直线)至 Mcs(线性)

两个运动之间的连接基于关闭 Acs 运动配置以及一个能够开启与下一个线性运动的连接的多项式曲线而完成。**Blending** 开始于 Mcs 系统下的目标运动进程, 并使用当前路径与下一路径的长度的百分比来适配连接点。请注意, **Blending** 是在 Mcs 坐标系中进行的, 因此可以绕过机器人运动的奇异点问题。

b) Blending 开始于 Mcs(线性)至 Acs(直线)

前面提到的对称条件是线性运动到直接运动之间连接的规划。在这种情况下下的运动轨迹是基于电机的参考系统下所规划适配的，并关联于 **blending** 进程的百分比以确定线性路径的特征点，既此以过渡到 **Acs** 坐标。两个参考系统下的 **blending** 机制都可保障机器人顺利通过连接点（两个路间的连接位置），并确保速度上的连续性以避免损坏系统地可能。根据 **PLCopen** 标准的方法，以下组合的转换参数和混合类型实际上是是可以实现和允许的：

c) 转换模式：MC 转换模式

转换模式	参数	状态
tmNone	None 无	可用
tmStartVelocity tm 启动速度	保持第一次运动结束时使用的速度	不可用
tmConstantVelocity tm 恒速	保持恒定速度	不可用
tmCornerDistance tm 角距	路径上的拐角处的距离	不可用
tmMaxCornerDeviation tm 最大拐角偏差	计算到拐角的曲线的距离	不可用
tmRelativeCornerDistance tm 相对角距离	占有的路径总长度的重叠百分比 混合	可用

8. PLC 视觉系统的功能块（FB）

a) 视觉构建

此功能块启用配置的摄像机。

视觉构建		
输入	Enable	在上升沿启用摄像机或在下降沿禁用摄像机
	Camera ID	相机识别
输出	Status	相机的有效状态
	Error	错误信号
	Error ID	错误识别

b) Vision_SetApplication 视觉集成应用

此功能块将配置的应用程序设置为摄像机

Vision_SetApplication 视觉集成应用		
输入	Execute	在上升沿开始设置
	Camera ID	相机识别
	Application	申请编号
输出	Done	设置完成
	Busy	正在设置
	Error	错误信号
	Error ID	错误识别

c) Vision_ObjectDetection 视觉检测对象

此功能块触发相机进行对象检测。检测可以在周期性的等待期内完成。时间戳和检测到的对象号将返回。

Vision_ObjectDetection		
输入	Execute	在上升沿开始设置
	Camera ID	相机识别
	Periodic	定期检测或不检测
	DelayMs	两次检测之间的毫秒延迟
	MatchThreshold	两个连续帧中的对象可能被标识为相同的。
输出	Done	检测完成
	Busy	正在检测
	TimeStamp	触发时间实例以微秒为单位
	ObjectNumber	检测到的对象编号
	Error	错误信号
	Error ID	错误识别

如果视觉对象检测功能块的“周期性”输入设置为 TRUE，则将根据“DelayMs”输入的值周期性地设置检测触发。此输入的建议值为 250 ms（最小时间），因为这样，两次触发之间的总时间将会减少。

为了计算要设置为“MatchThreshold”输入的正确值，用户必须考虑两个触发事件之间的总时间以及要检测的物体是否已在传送带或类似装置上移动。MatchThreshold 值用于考虑在两个连续帧中检测到的对象之间的距离，并且特别地用于区分检测到的对象是否是相同的对象（根据传送带或类似装置移动）或两个不同的对象。

两个触发事件之间的总时间计算为相机评估时间和两次检测之间的延迟（“DelayMs”输入）之和。相机评估时间应由相机数据表或用户手册提供：IFM 相机用户手册提供评估时间典型值 50-800 ms。然而，该评估时间取决于许多因素，例如对象大小和取向，检测到的对象的数量，平滑度等。

为了评估两个帧之间的距离，需要以米计量的相机视场与以像素计量的相同视场之间的对应关系。传感器参考手册应提供计算所需的所有数据：例如，IFM 摄像机的用户手册提供了这些表格，其中摄像机视野与不同类型摄像机的摄像机操作距离相关：

Types O2D220 / O2D227 (normal lens)							
Operating distance L [mm]	50	75	100	200	400	1000	2000
Field of view size WxH [mm]	16x12	24x18	32x24	64x48	128x96	320x240	640x480
Resolution [mm]	0.1	0.2	0.3	0.4	0.8	2.0	4.0

Types O2D222 / O2D229 (wide-angle lens)							
Operating distance L [mm]	50	75	100	200	400	1000	2000
Field of view size WxH [mm]	33x24	50x36	66x47	132x94	264x189	660x472	1320x945
Resolution [mm]	0.3	0.4	0.5	0.9	1.7	4.0	8.0

Types O2D224 / O2D225 (telephoto lens)							
Operating distance L [mm]	50	75	100	200	400	1000	2000
Field of view size WxH [mm]	-	15x11	20x15	40x30	80x60	200x150	400x300
Resolution [mm]	-	0.08	0.12	0.25	0.52	1.25	2.52

d) MatchThreshold computation example 匹配阈值计算示例

假设，有一个传送带以 0.05m / s 的恒定速度（BeltSpeed = BS = 0.05m / s）传送待检测的物体。在距输送带 300 mm 处安装了一个 O2D222 摄像机。从上表中，摄像机视野为 198x141 mm（0.198x0.141 m）。可视窗口大小约为 640×480 像素，我们可以获得像素计比例（PxToM）：

$$\text{PxToM} = 640 / 0.198 = 3232 \text{ 像素/米}$$

两个帧之间的总时间（TotalTime）可以计算为

$$\text{总时间} = \text{延迟时间} + \text{最大建议评估时间} = 0.250 + 0.800 = 1.05 \text{ s}$$

对于计算，我们已经考虑了评估时间的指示的典型值之间的最大值。如果“Periodic”输入设置为 FALSE，则必须通过考虑用户设置的两个触发事件之间的时间间隔来计算总时间。最后，匹配阈值（MatchThreshold）可以计算为：

$$\text{MatchThreshold} = \text{BS} * \text{TotalTime} * \text{PxToM} = 0.05 * 1.05 * 3232 = 170 \text{ pixel}$$

由于该值的特征是有许多测量不确定性（如上所述的评估类型），所以使用作为视觉对象检测 FB 的“匹配阈值”输入更大的值（例如 200 像素）可能是有用的，

e) Vision_ReadCameraStatus 视觉读取相机状态

此功能块返回相机的状态。

Vision_ReadCameraStatus		
输入	Enable	启用时连续获取数据
	Camera ID	相机识别
输出	Valid	如果有效输出可用，则为真
	Disconnect	在断开状态为真
	Idle	在空闲状态为真
	ReleaseThread	在释放线程状态为真
	Busy	在忙碌状态为真
	ErrorState	在错误状态为真
	Error	错误信号
	ErrorID	错误识别

f) Vision_Reset 视觉复位

当出现错误时，此功能块将重置相机的状态。如果相机无法连接，功能块将设置相机状态以断开连接。如果在操作期间发生任何错误。相机将被重置为空闲状态。

Vision_Reset		
输入	Execute	在上升沿开始复位
	Camera ID	相机识别
输出	Done	重置完成
	Busy	正在重置
	Error	错误信号
	Error ID	错误识别

g) Vision_ReadObject 视觉读取对象

功能块返回当前对象队列中对象的信息。

Vision_ReadObject		
Inputs 输入	Enable	在启用时连续获取对象数据
	Camera ID	相机识别
	Object ID	对象标识
Outputs 输出	Valid	
	TimeStampUs	检测到被视为对象的实例
	Position	对象在 MCS 中的位置
	Velocity	MCS 中对象的速度
	Error	Error signal 错误信号
	Error ID	错误识别

h) Vision_ReadObjectQueue 视觉读取对象队列

此功能块更新对象 id 列表队列并返回当前检测到的对象的数量。

Vision_ReadObjectQueue		
Inputs 输入	Enable	启用时，在对象队列中连续获取数据
	CameraID	相机识别
	ObjectIDList	指向对象列表数组
	ListMaxSize	对象列表数组的最大大小
Outputs 输出	ObjectNumber	队列中当前对象的数量