快速使用 IDE 软件开发一个运动程序

Zhang, Shikun
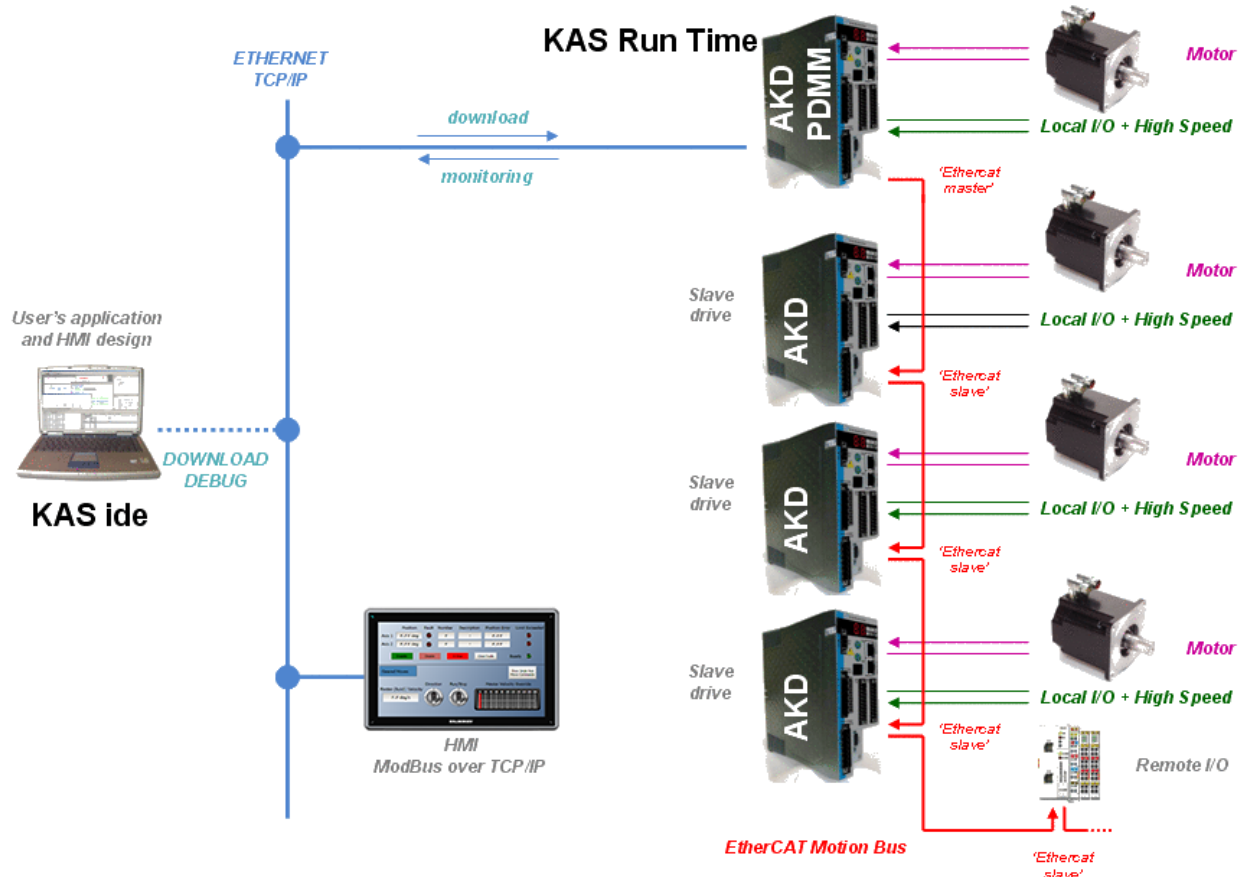
Contents

1. Kollmorgen Automation Suite 科尔摩根自动化组件

控制器：驱动器控制器一体机 PDMM ，独立控制器 PCMM

PDMM 有 800MHz 和 1.2GHz 两款 CPU 选择

PCMM 有 1.2GHz 和 双核 1.2GHz 两款 CPU 选择

伺服驱动器：AKD、AKD2G、S300、S700

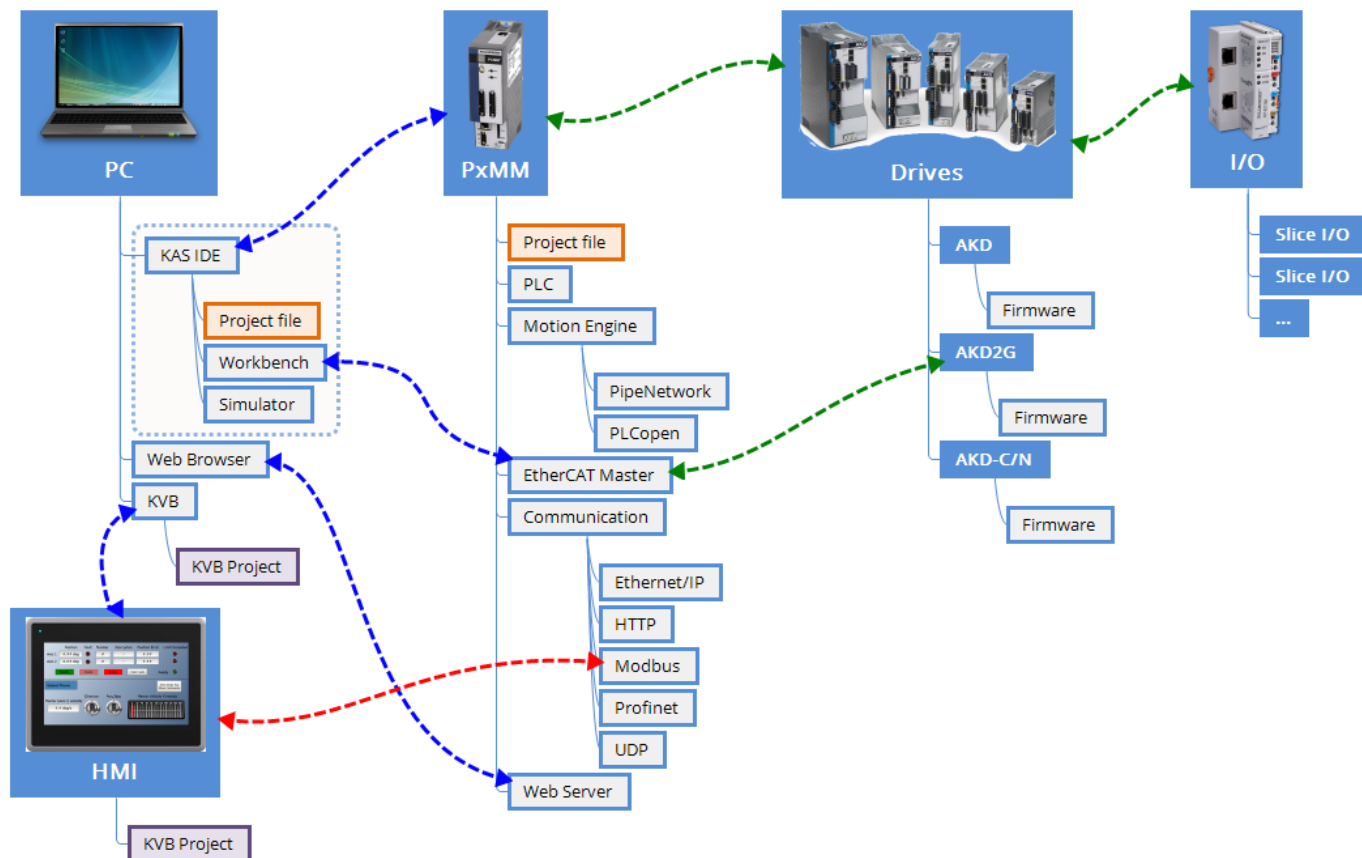| Figure 2-15: AKD | Figure 2-16: AKD2G | Figure 2-17: S300 | Figure 2-18: S700 |

伺服电机：AKM，CDDR，DDR，CDDR，DDL，KBM

2. KAS 构架示意图

KAS Architecture
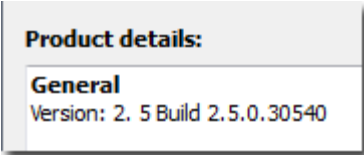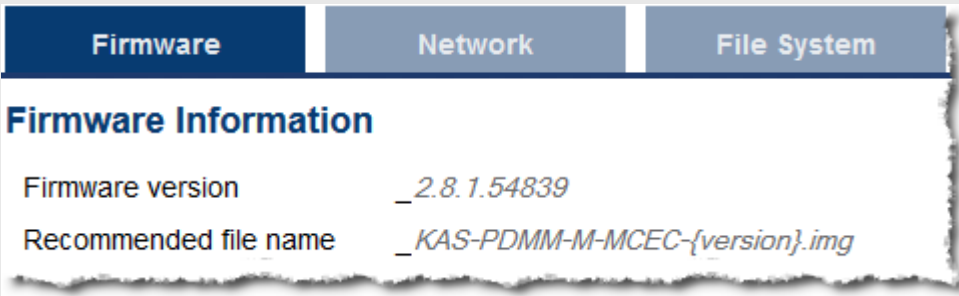
通讯关键字	
颜色	**通讯方式**
蓝色	Ethernet 以太网
绿色	EtherCAT 运动总线
红色	Modbus TCP/IP

KAS 安装软件包包含以下内容:

KAS-Setup-3.04.0.60001.exe

KAS-PCMM-M-M1EC-3.04.0.60001.img

KAS-PCMM-M-MCEC-3.04.0.60001.img

KAS-AKD PDMM-M-M1EC-3.04.0.60001.img

KAS-AKD PDMM-M-MCEC-3.04.0.60001.img

要注意使用的 IDE 版本和控制器里运行版本兼容性

Version Source	Found From	Example
IDE, Controller or Simulator	Help menu > About	**Product details:** **General** Version: 2. 5 Build 2.5.0.30540
PDMM	Web server Settings Firmware tab	**Firmware** **Network** **File System** **Firmware Information** Firmware version _2.8.1.54839 Recommended file name _KAS-PDMM-M-MCEC-{version}.img

3. 配置步骤

通过旋钮设置控制器的 IP 地址，通过 IE 游览器可以进入 Web Server 画面

IP Address Selection

- 10 position rotary switch
  - 0: for DHCP (if no DHCP server, AutoIP is used)
  - 1: for manual IP Address
    - default: 192.168.0.101
    - Setup in Web Server
  - 2-9: for static IP address (192.168.0.10x)

Manual IP Address **

IP Address				
Subnet Mask				
Default Gateway				

登入 LOGIN，密码为用户名 administrator

KAS Application 画面里

- 显示当前加载在控制器上的项目的一般信息
- 启动和停止运动程序
- 从 Axis 选项卡显示控制器运行的轴一些数据

Settings 画面里

- 显示和更新 KAS 运行时的固件

- 显示网络设置，并修改 IP 地址

- 将控制器固件重置为出厂设置

Diagnostic 画面里

此页面显示有关硬件状态(存储空间、内存和 CPU 温度)、错误和警报的信息。

运行 IDE 开发软件， 建立一个新的项目，加入一个新的控制器
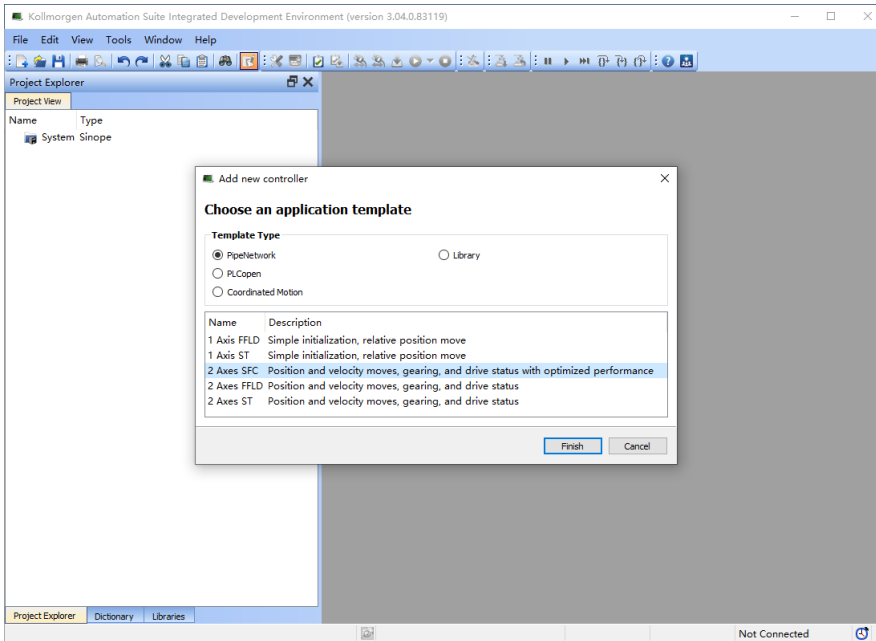
选择运动引擎选项（Pipe Network 或则 PLCopen），然后选择合适的例程模板。

Template Type	Template name	Description
PipeNetwork	1 Axis FFLD	Simple initialization, relative position move

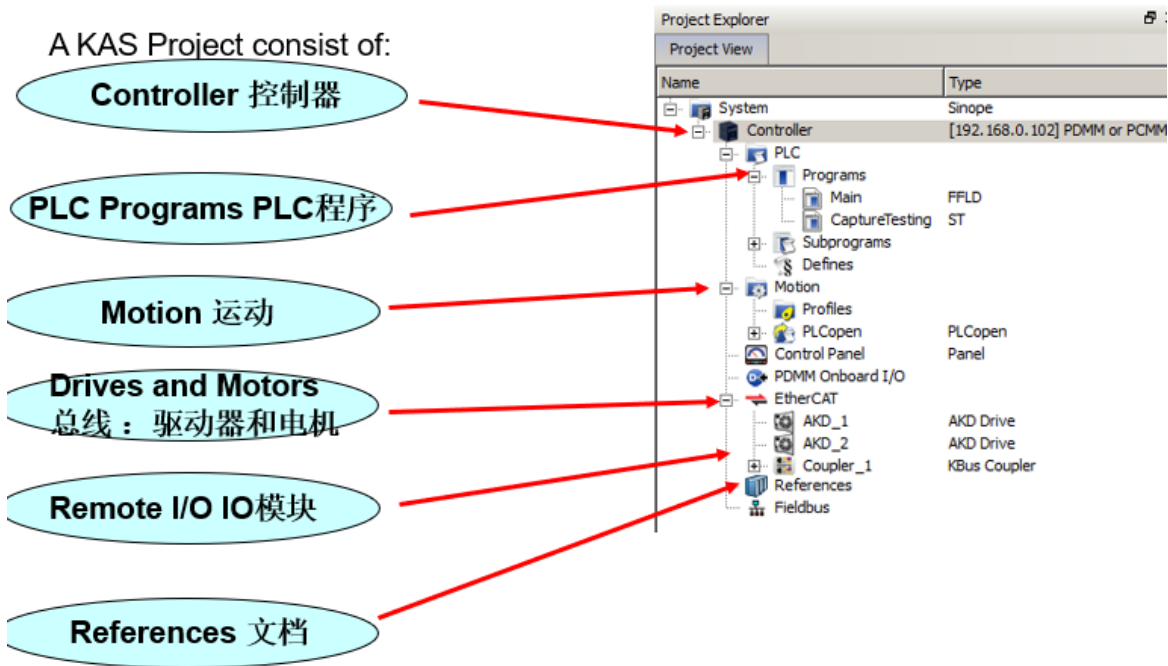| Template Type	Template name	Description
	1 Axis ST	Simple initialization, relative position move
	2 Axes FFLD	Position and velocity moves, gearing, camming, and drive status
	2 Axes ST	Position and velocity moves, gearing, camming, and drive status
	2 Axes SFC	Position and velocity moves, gearing, camming, and drive status with optimized performance.
PLCopen	1 Axis FFLD	Simple initialization, relative position move
	1 Axis ST	Simple initialization, relative position move
	2 Axes FFLD	Position and velocity moves, gearing, camming, and drive status
	2 Axes ST	Position and velocity moves, gearing, camming, and drive status
	2 Axes SFC	Position and velocity moves, gearing, camming, and drive status with optimized performance.
Coordinated Motion	2 Axes - Linear / Circular	Raster Scan Motion Path, 2 PLCopen axes
	3 Axes - Linear / Circular	Raster Scan Motion Path, 2 PLCopen axes and 1 PipeNetwork axis
	3 Axes - Linear (3D)	Diamond/Square Motion Path, 3 PLCopen axes
KAS Runtime	Library	Allows you to create a custom library (Create and Use Custom Libraries)

选择好后，点击 Finish。(默认选项：PipeNetWork / 2Axse SFC)
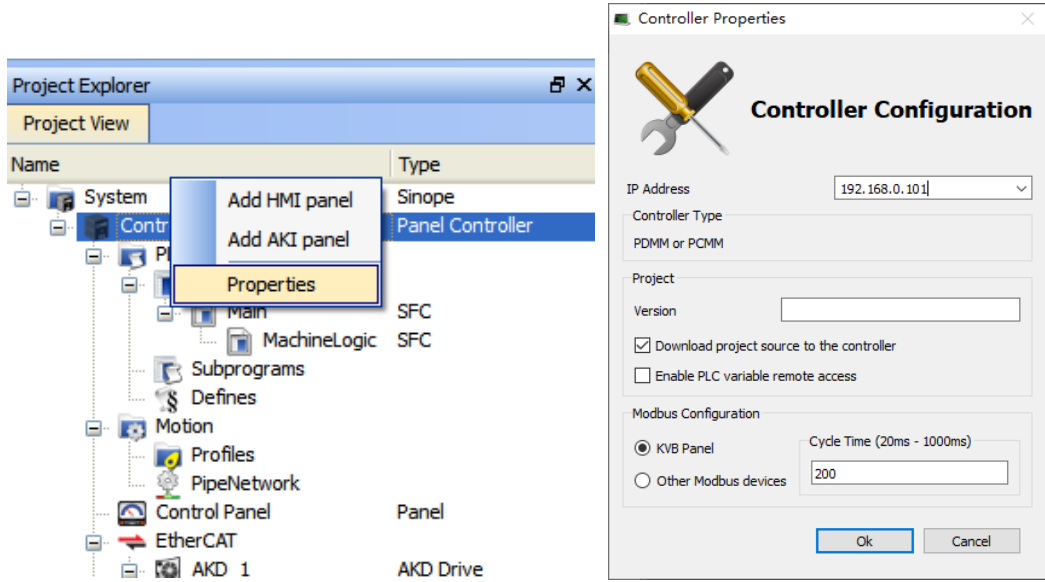
PLC 编程语言符合 IEC-6113，支持 SFC、FBD、IL、ST、FFLD 的五种编程语言。

进入 IDE 开发界面

A KAS Project consist of:

- **Controller 控制器**
- **PLC Programs PLC程序**
- **Motion 运动**
- **Drives and Motors 总线：驱动器和电机**
- **Remote I/O IO模块**
- **References 文档**

选择控制器 Controller 右键属性 Properties，修改 IP 地址
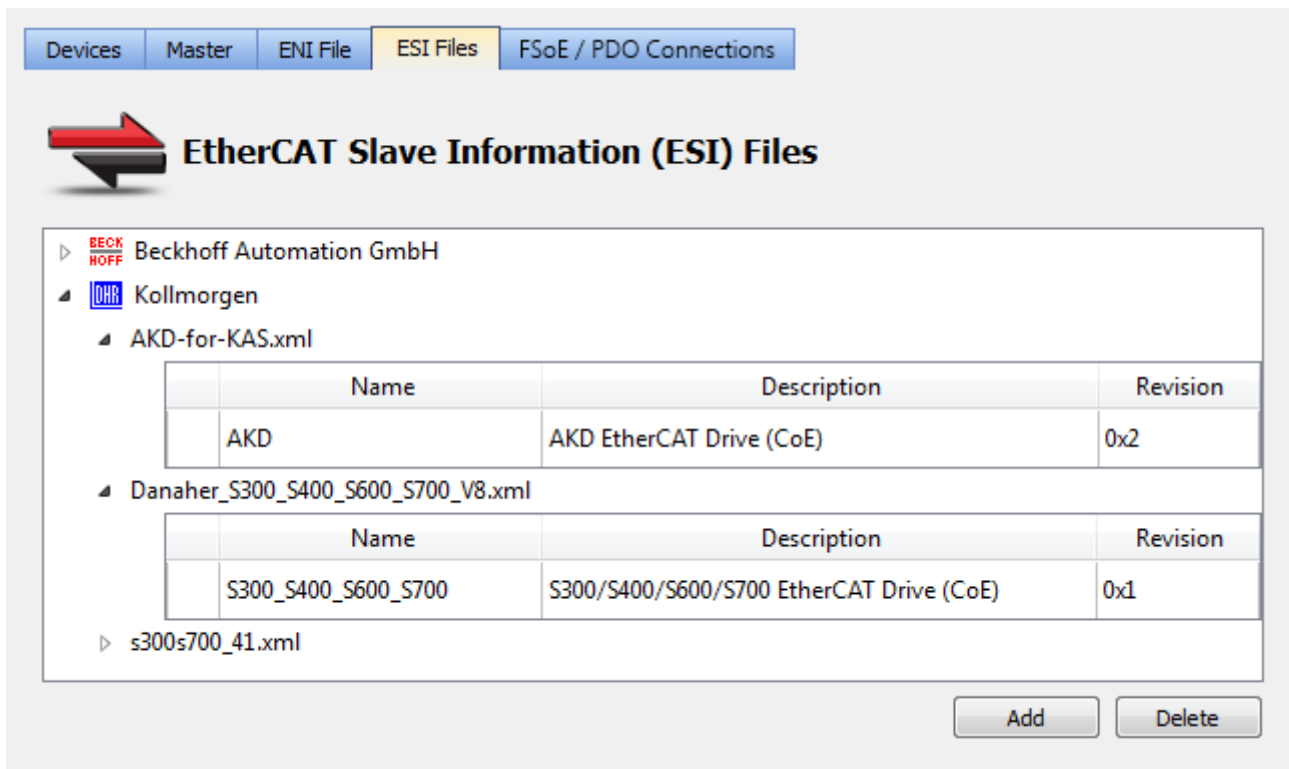

配置 EtherCAT 运动总线，添加离线设备 Add Device 或者 扫描在线设备 Scan Devices
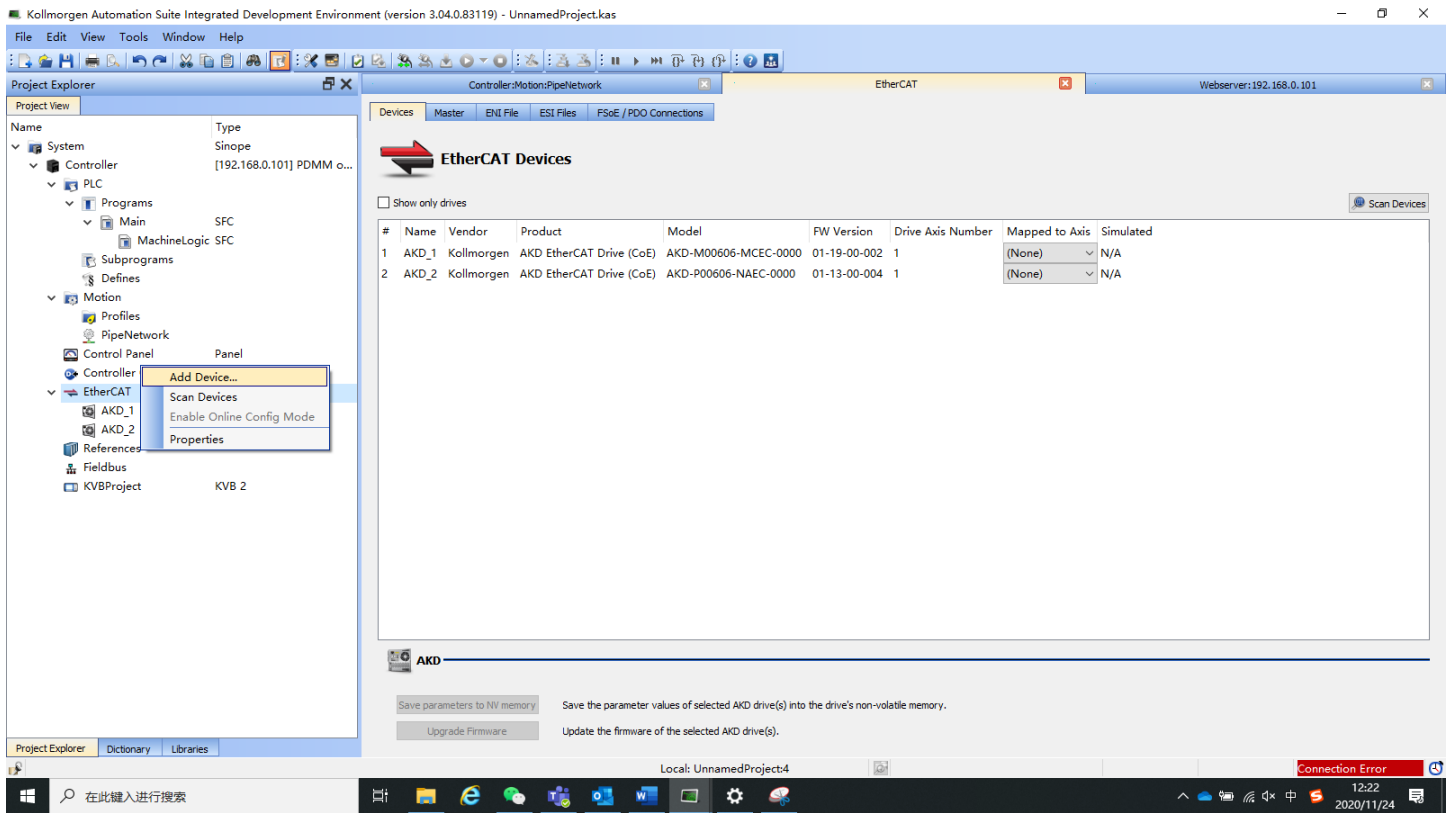
第三方需添加 ESI 文件（XML 文件）

在 Master 画面里设置 EtherCAT 总线扫描周期（也是底层 Motion 周期）
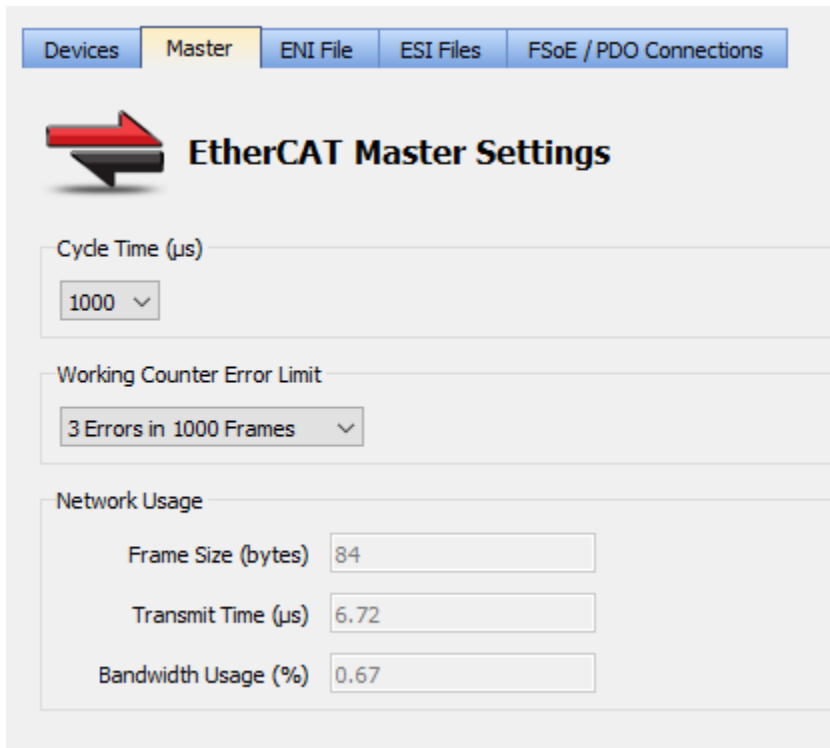
选择 EtherCAT 下面的设备(例如 AKD_1), 右键属性 Properties, 可以 Configuration 配置 AKD (AKD2G) 驱动器, PDO Selection 选择 PDO, 修改 PDO Editor, CoE Init 初始化。

默认的 AKD PDO 数据

轴的运动关系的建立，默认模板例程 在 PipeNetWork 里 Axis1 和 Axis2 与 Master 主轴是齿轮比关系，通过右键添加新的模块
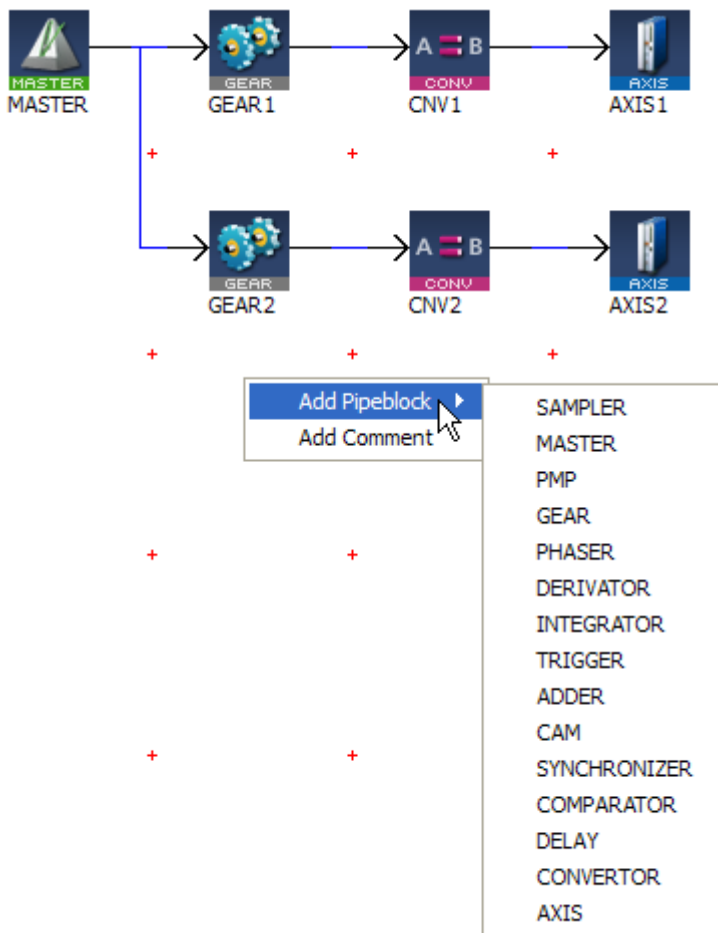
主要有以下模块可供选择:

数据源：主轴 MASTER，主轴 PMP，采样轴 SAMP
运动关系：求导 DERIV，积分 INTEGR，叠加 ADD，离合 SYNCH，延迟 DELAY，比较 COMPAR，捕捉 TRIGGER，齿轮比 GEAR，凸轮 CAM，相位偏移 PHASER
耦合：CONV
伺服轴：AXIS

添加 齿轮比 GEAR3 、 耦合 CNV3 、 伺服轴 AXIS3
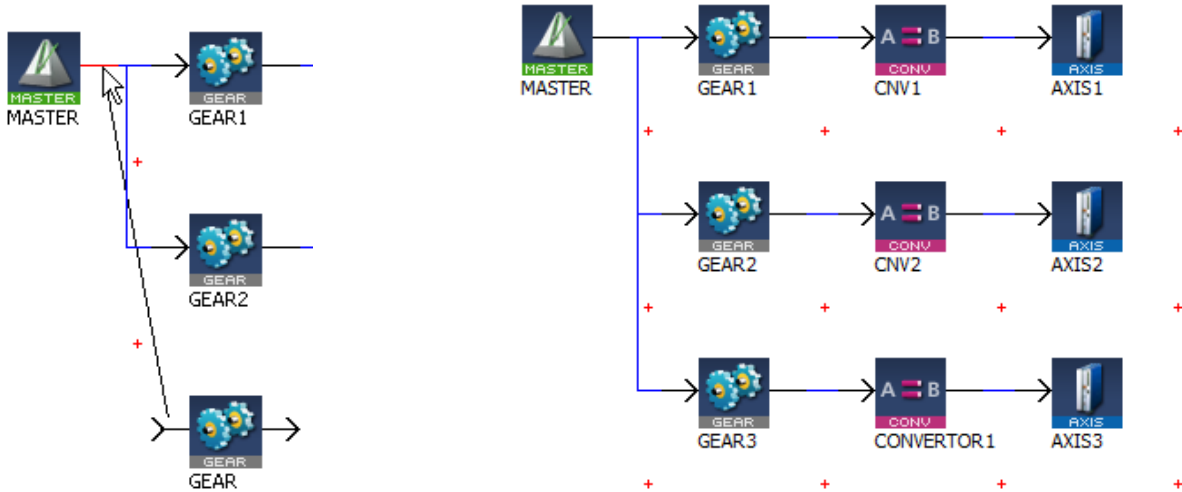
默认齿轮比是 1.0 ； 齿轮比变化 Ratio Slope Type 默认最大立刻 。
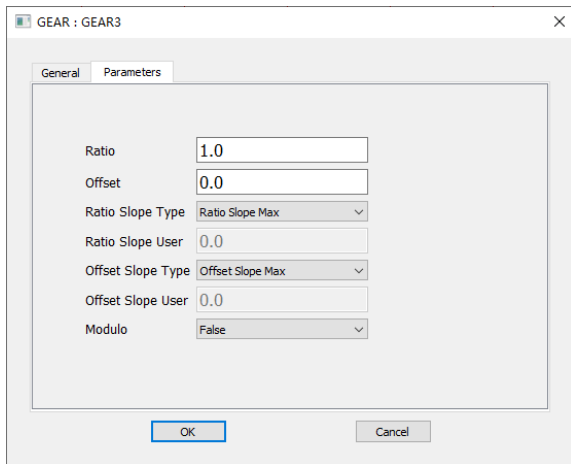
从 PipeNetWork 看三个轴都和主轴为 1:1 的齿轮比关系，也可以看成三个轴之间为 1:1 齿轮比关系。

在 EtherCAT 画面里，把 PipeNetWork 中定义的 AXIS 连接到对应的实际伺服驱动器上。
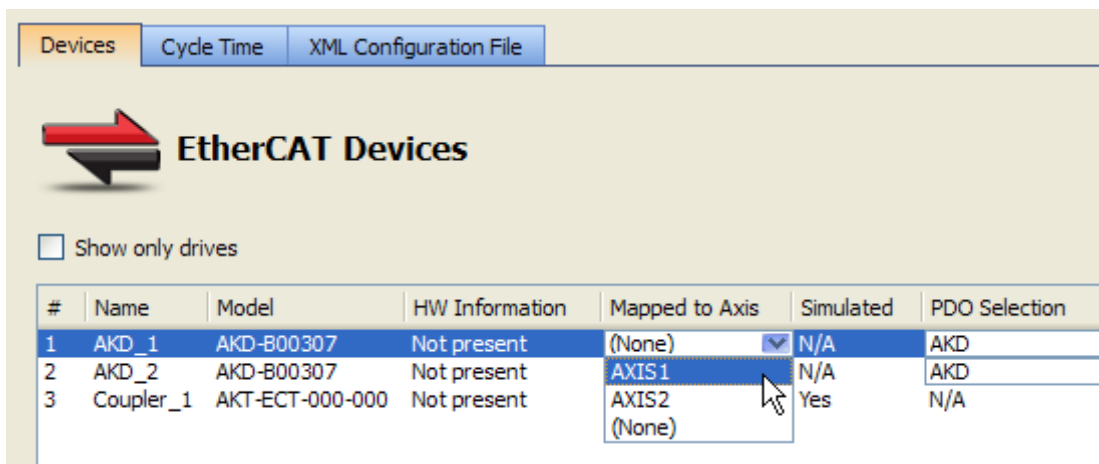
程序中对 AXIS 模块的操作就是对实际伺服电机操作。

PLC 程序 Programs 框架
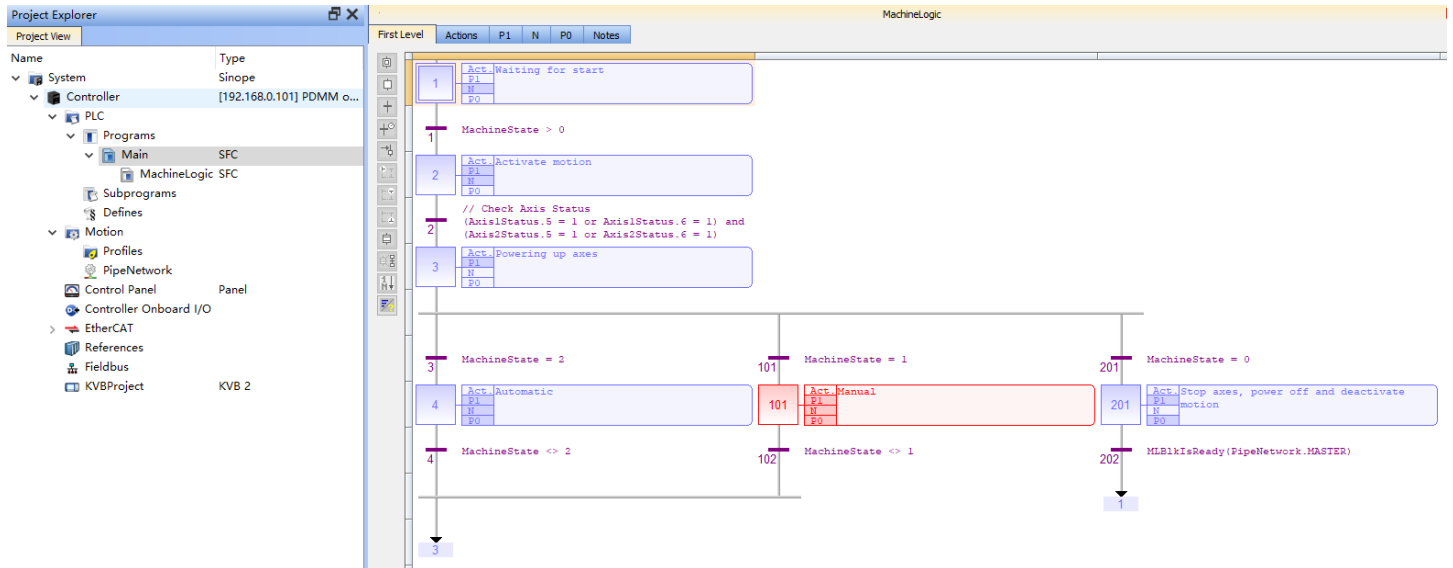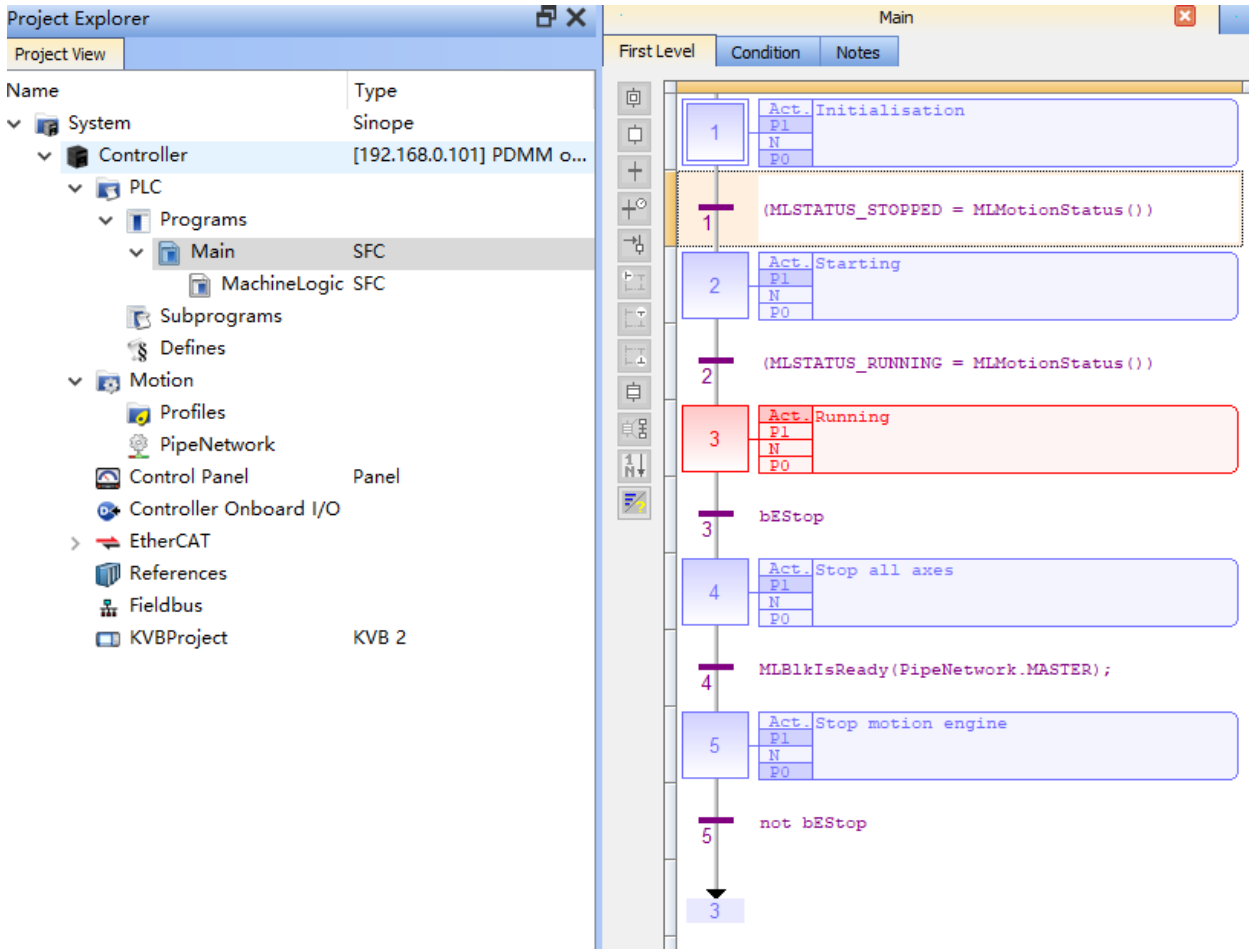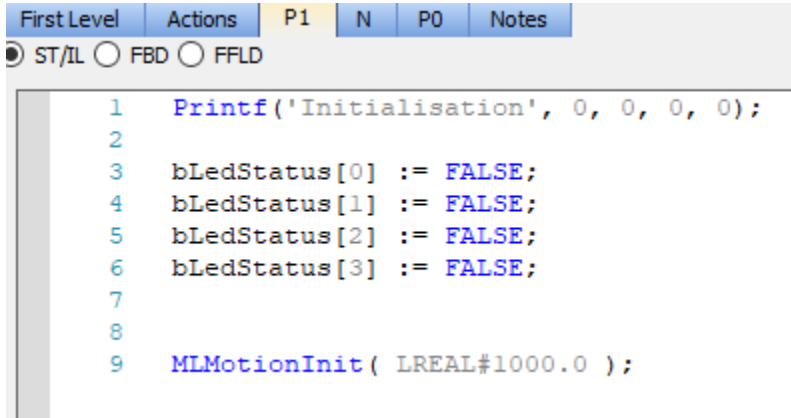
在 SFC 结构中，Main 是主程序，MachineLogic 是 Main 的子程序（在合适的位置被激活或关闭，注意不是调用关系）

每个 SFC 包含：P1 进入执行一次，N 重复执行，P0 退出执行一次，Actions 激活关闭它的子 SFC 程序。

4. 模板程序步骤

模板 Main 程序中包含 5 步:

第一步: Initialisation 初始化

在 P1 段里: 初始化运动库
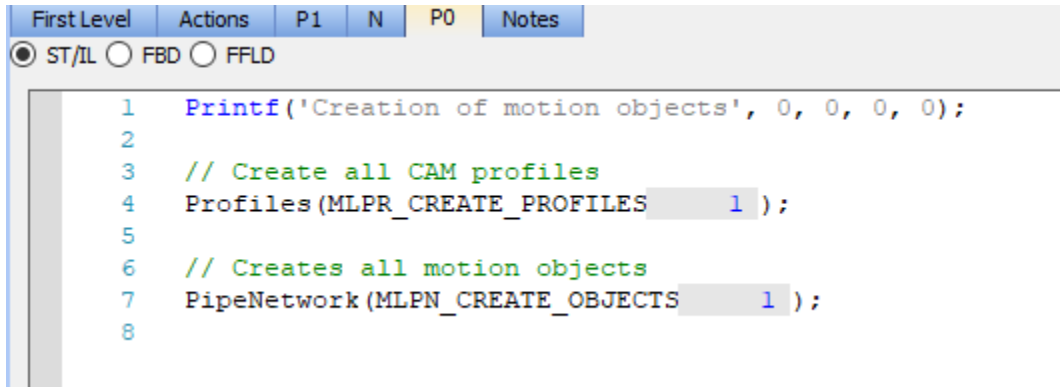
```
First Level    Actions    P1    N    P0    Notes
◉ ST/IL  ○ FBD  ○ FFLD

1    Printf('Initialisation', 0, 0, 0, 0);
2
3    bLedStatus[0] := FALSE;
4    bLedStatus[1] := FALSE;
5    bLedStatus[2] := FALSE;
6    bLedStatus[3] := FALSE;
7
8
9    MLMotionInit( LREAL#1000.0 );
```

在 N 段里: 没有程序
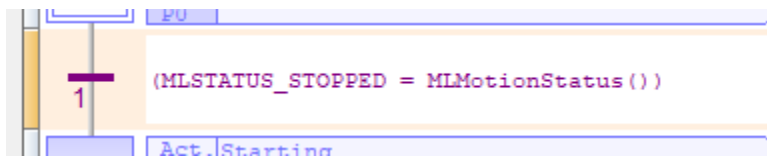
在 P0 段里: 建立凸轮数据, 建立 PipeNetWork 中的运动模块

```
First Level    Actions    P1    N    P0    Notes
◉ ST/IL  ○ FBD  ○ FFLD

1    Printf('Creation of motion objects', 0, 0, 0, 0);
2
3    // Create all CAM profiles
4    Profiles(MLPR_CREATE_PROFILES    1 );
5
6    // Creates all motion objects
7    PipeNetwork(MLPN_CREATE_OBJECTS    1 );
8
```

进入下一步条件为: 运动总线出于停止无故障状态

```
                    P0

  ┬      (MLSTATUS_STOPPED = MLMotionStatus())
  1

        Act.Starting
```

第二步: Starting 开始 EtherCAT 总线, 建立同步

在 P1 段里: 开始 EtherCAT 通讯建立

```
First Level    Actions    P1    N    P0    Notes
◉ ST/IL  ○ FBD  ○ FFLD

1    bLedStatus[0] := TRUE;
2
3    MLMotionStart();
```

在 N 段里: 没有程序

在 P0 段里: 没有程序

进入下一步条件为：运动总线处于运行状态

```
        P0

        (MLSTATUS_RUNNING = MLMotionStatus())
    2
        Act Running
```

说明：一般会在这一步的 N 增加程序，通过判断超过多长时间（例如 20S）没有进入下一步，则 EtherCAT 通讯建立有问题，进而切除通讯故障，重启通讯建立，例如下面程序：

```
First Level   Actions   P1   N   P0   Notes
● ST/IL ○ FBD ○ FFLD

    1    //激活EtherCAT总线
    2    MLMotionStart();
    3    MotionBusActivate:=TRUE;
    4    bMotionEngineStarted := FALSE;
```

```
First Level   Actions   P1   N   P0   Notes
● ST/IL ○ FBD ○ FFLD

    1    //激活运动总线，计时器，20秒内完成总线同步
    2    MotionBusTime(MotionBusActivate,T#20s);
    3
    4    //监控总线状态
    5    MotionEngineStatus:=MLMotionStatus();
    6    //MLSTATUS_NOT_INITIALISED 0 (*Motion not initialised*)
    7    //MLSTATUS_RUNNING 1 (*Motion is running*)
    8    //MLSTATUS_STOPPED 2 (*Motion is stopped*)
    9    //MLSTATUS_ERROR 3 (*Motion is in error*)
    10
    11    //20秒内没有完成总线同步
    12    IF MotionBusTime.Q THEN
    13        IF MotionEngineStatus = 2 THEN
    14            MLMotionStart();              //再次激活总线
    15            MotionBusActivate:=FALSE;
    16        ELSE
    17            MLMotionRstErr();             //清除总线故障
    18        END_IF;
    19    ELSE
    20        MotionBusActivate:=TRUE;
    21    END_IF;
    22
    23    //监控主站状态
    24    Inst_ECATMasterStatus( TRUE );
    25    EtherCATSynStatus:=Inst_ECATMasterStatus.State;
    26    //EC_STATE_NO_COMMUNICATION   0   (* 0x00 = No Communication to device *)
    27    //EC_STATE_INIT               1   (* 0x01 = Device in Init state *)
    28    //EC_STATE_PREOP              2   (* 0x02 = Device in Pre-operational state *)
    29    //EC_STATE_BOOTSTRAP          3   (* 0x03 = Device in Bootstrap state *)
    30    //EC_STATE_SAFEOP             4   (* 0x04 = Device in Safe-Operational state *)
    31    //EC_STATE_OP                 8   (* 0x08 = Device in Operational state *)
    32
    33    //闪烁1秒
    34    MotionBusActivate:=Pluse1S( TRUE, T#1S )  ;
```
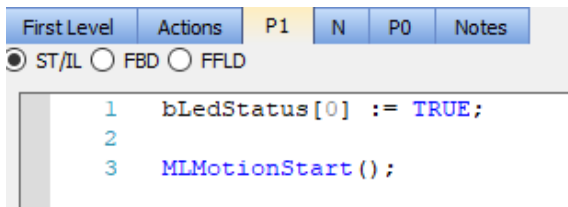
```
First Level   Actions   P1   N   P0   Notes
● ST/IL ○ FBD ○ FFLD

    1    MotionBusActivate := TRUE;
    2    bMotionEngineStarted := TRUE;
```

在 字典 Dictionary 中 Main 里建立对应的变量

```
□ 📄 Main
    MotionBusRun            BOOL
    MotionBusTime           TON
    Pluse1S                 blink
    Inst_ECATMasterStatus   ECATMasterStatus
    MotionEngineStatus      DINT
    MotionBusActivate       BOOL
    MotionBusReset          BOOL
⊞ 📊 MCFB AKDFaultLookup1
```

第三步：Running　客户可以在此编写自己的控制程序

在 P1 段里：没有程序

在 N 段里：　没有程序

在 P0 段里：　没有程序

在 Action 里：　激活 SFC 子程序 MachineLogic

First Level	Actions	P1	N	P0	Notes

```
1    MachineLogic(N);
```

退出条件为：急停变量 Estop 为 TRUE

```
          P0
3 ┤├  bEStop

   Act.Stop all axes
```

第四步：Stop all axes 因为是急停进入的，所以停止所有轴的运动

在 P1 段里：主轴速度降为 0，设备状态 MachineState 赋值 0（0 是急停，1 是手动，2 是自动）

First Level	Actions	P1	N	P0	Notes
● ST/IL ○ FBD ○ FFLD					

```
1    bLedStatus[3] := TRUE;
2    bLedStatus[2] := FALSE;
3
4    Printf('Stop axes', 0, 0, 0, 0);
5    MLMstRun(PipeNetwork.MASTER    14 , LREAL#0.0);
6
7    MachineState    1  := 0;
8
```

在 N 段里：　没有程序

在 P0 段里：　没有程序

退出条件为：判断主轴减速到 0 了（Ready OK）

```
          P0
4 ┤├  MLBlkIsReady(PipeNetwork.MASTER);

   Act.Stop motion engine
```
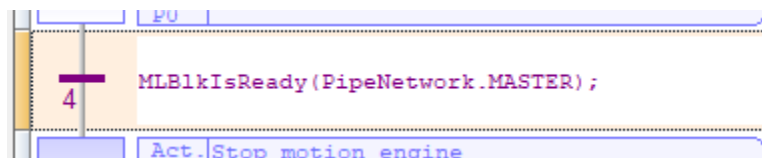
第五步：Stop motion engine 停止运动库（主要解开 PipeNetWork 中的运动关系）

在 P1 段里：伺服轴去使能，解除 PipeNetWork 中的运动关系

```
1    Printf('PowerOff', 0, 0, 0, 0);
2
3    // Power off all axes
4    PipeNetwork(MLPN_POWER_OFF        5 );
5    bLedStatus[1] := FALSE;
6
7    // desactivate Pipes
8    PipeNetwork(MLPN_DEACTIVATE        6 );
9
```

在 N 段里： 没有程序

在 P0 段里：没有程序

退出条件为：急停变量 Estop 为 FALSE，返回第三步
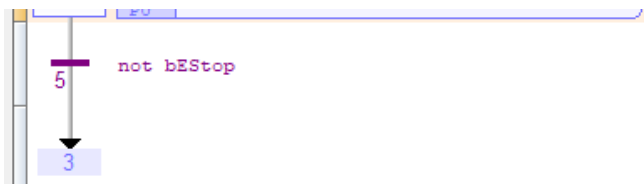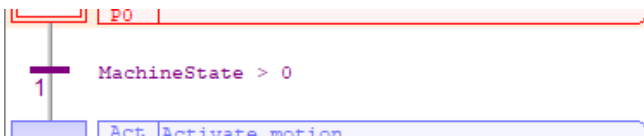

模板 MachineLogic 程序中根据设备状态分三部分程序段：自动、手动、急停。

第一步：Waiting for start

在 P1 段里： 没有程序

在 N 段里： 没有程序

在 P0 段里： 没有程序

进入下一步条件为：MachineState>0 设备状态进入手动或自动

第二步：Activate motion 激活 PipeNetWork 内的运动模块

在 P1 段里：激活 Pipes，伺服轴连接到 Pipes 上，伺服轴上使能（客户也可以在后面单独对每个轴使能操作）


```
1    Printf('Activation Pipes', 0, 0, 0, 0);
2
3    // Activates the pipes
4    PipeNetwork(MLPN_ACTIVATE        2 );
5
6    // Connect the axes to the pipes
7    PipeNetwork(MLPN_CONNECT        3 );
8
9    Printf('PowerON', 0, 0, 0, 0);
10
11   // Power on all axes
12   PipeNetwork(MLPN_POWER_ON        4 );
```

在 N 段里：读取伺服轴的状态

First Level	Actions	P1	N	P0	Notes

◉ ST/IL ○ FBD ○ FFLD

```
1    AxislStatus    29  := MLAxisStatus(PipeNetwork.AXIS1    18 );
2    Axis2Status   153  := MLAxisStatus(PipeNetwork.AXIS2    22 );
```

| Bit	Description
0	Initialized (1 if initialized)
1	Power (1 if power is on) Is linked to bit 1 (Switched on) of the Status Word For more information on the status machine see "CANopen Status Machine"
2	Enabled (1 if enabled) Is linked to bit 0 (Ready to switch on) of the Status Word
3	Found (1 if found on the network). EtherCAT ▶ state is Pre-Operational, see State Machine.
4	Configured (1 if configured) EtherCAT state is Safe-Operational, see State Machine.
5	Running (1 if running) EtherCAT state is Operational, see State Machine.
6	Error (1 if in error)
7	Simulated (1 if working with a simulated axis)
8	Connected (1 if a pipe is connected)
9	Warning (1 if the drive signals a warning)
10	Stopping (1 if the drive is performing a Stop)
11	Stopped (1 if the drive has finished the Stop)
12 to 31	Reserved

在 P0 段里： 没有程序

进入下一步条件为：判断伺服轴上使能，无故障


```
// Check Axis Status
(Axis1Status.5 = 1 or Axis1Status.6 = 1) and
(Axis2Status.5 = 1 or Axis2Status.6 = 1)
```
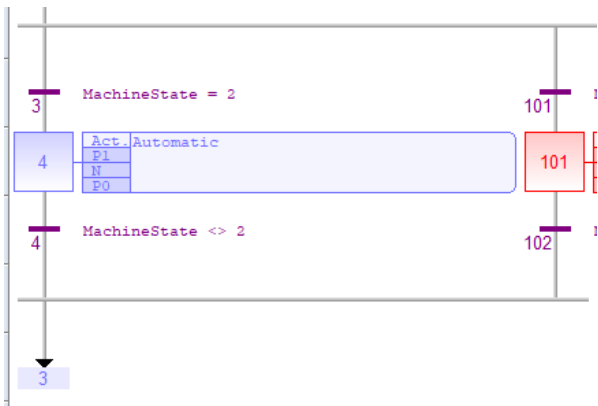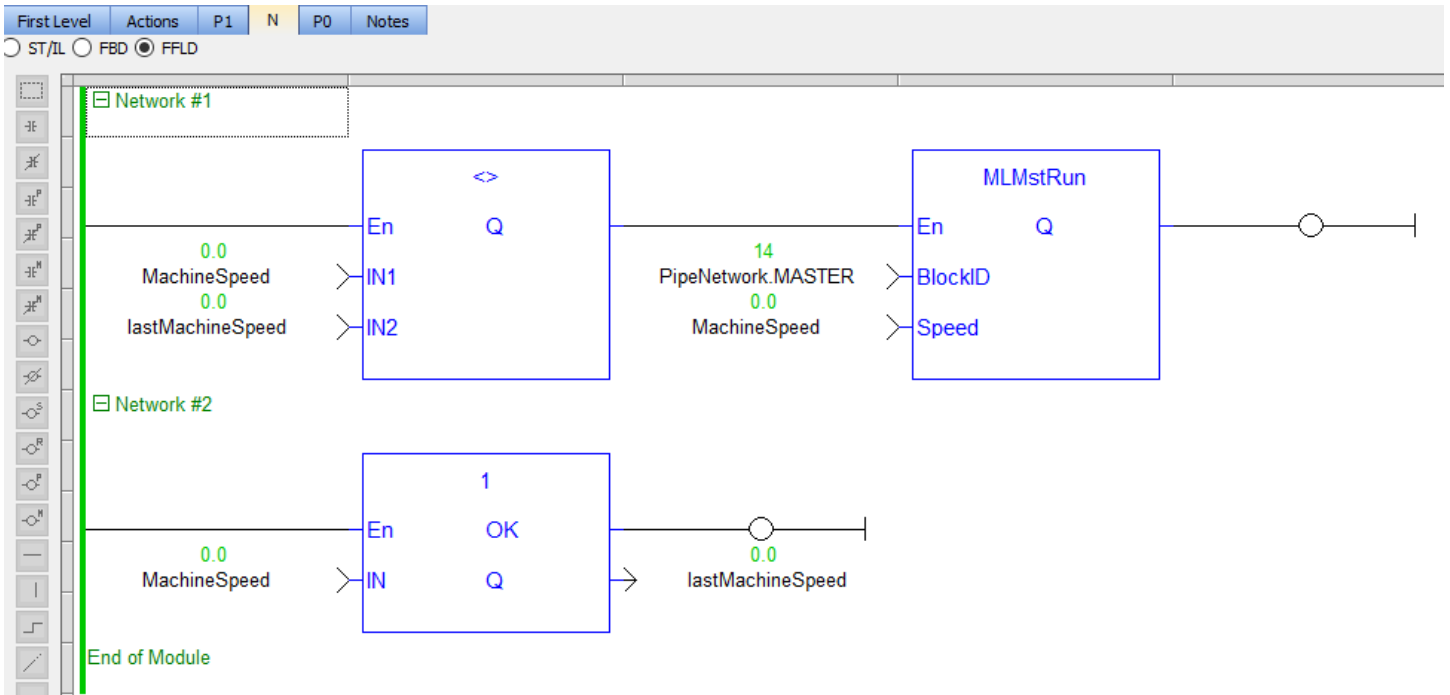
第三步：Powering up axes　没有程序 作用分支
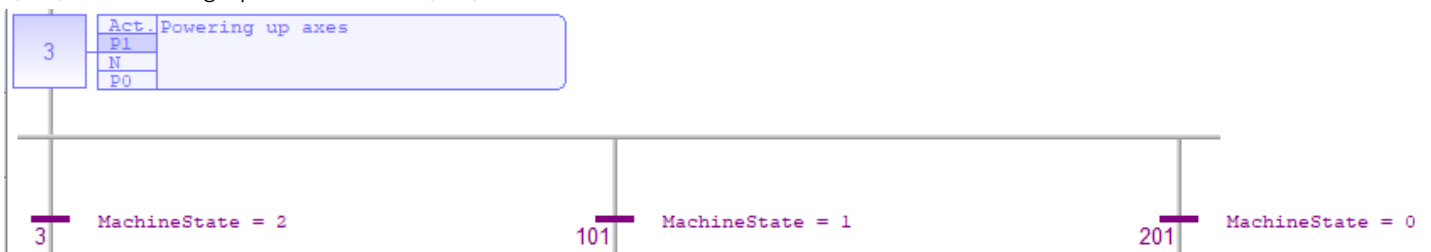
第四步：　Automatic　自动程序

进入条件：MachineState=2　　退出条件：MachineState<>2 跳转到第三步

在 P1 段里：主轴按给定速度运行，一般会用 0.0 代替 MachineSpped


```
1    Printf('Auto Mode', 0, 0, 0, 0);
2
3    // Start motion
4    MLMstRun(PipeNetwork.Master    14 , MachineSpeed    0.0 );
5
6
```
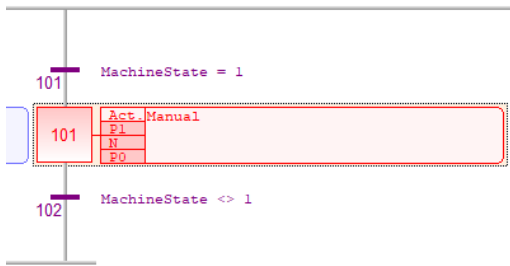
在 N 段里：当设定速度和当前速度不一致时，主轴按设定速度运行，保存设定速度。

在 P0 段里： 没有程序

第 101 步：Manual   手动程序
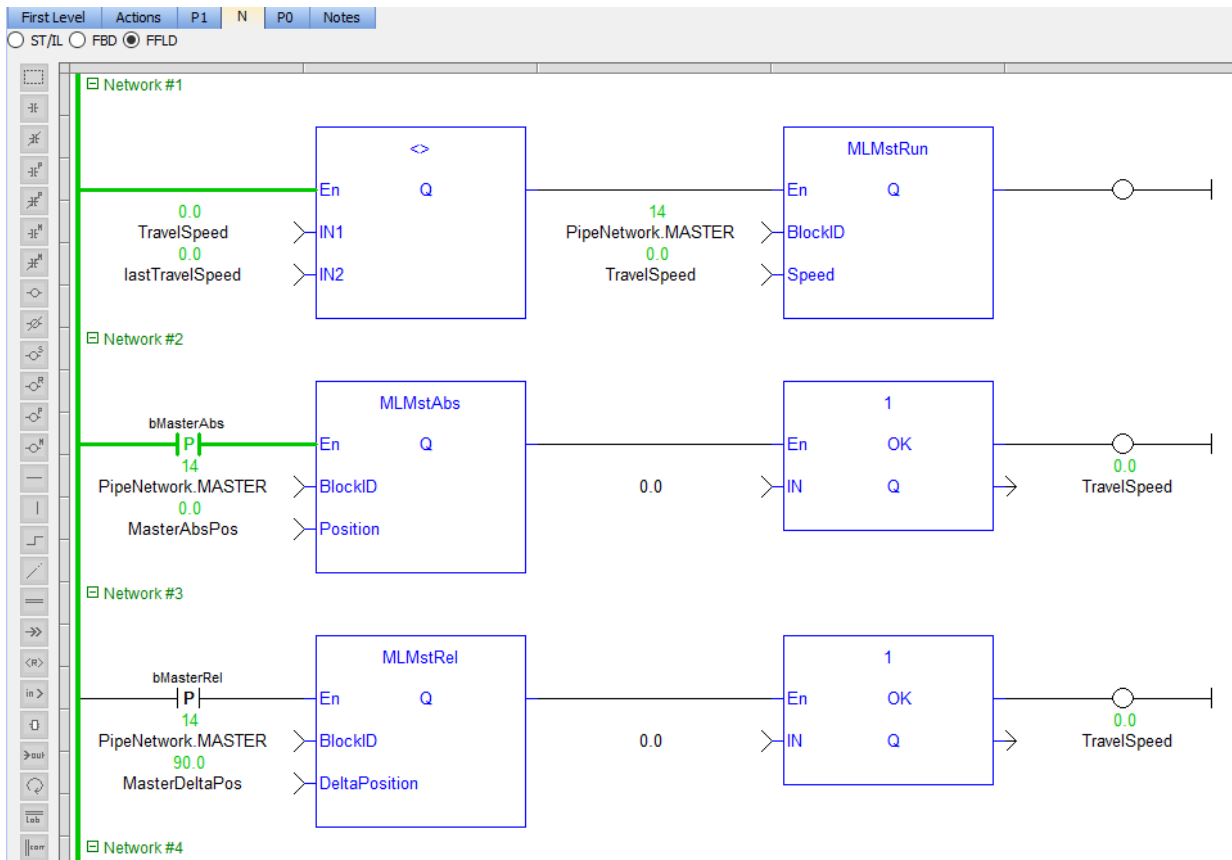进入条件：MachineState=1   退出条件：MachineState<>1 跳转到第三步

在 P1 段里：主轴按给定速度运行，一般会用 0.0 代替 TravelSpeed
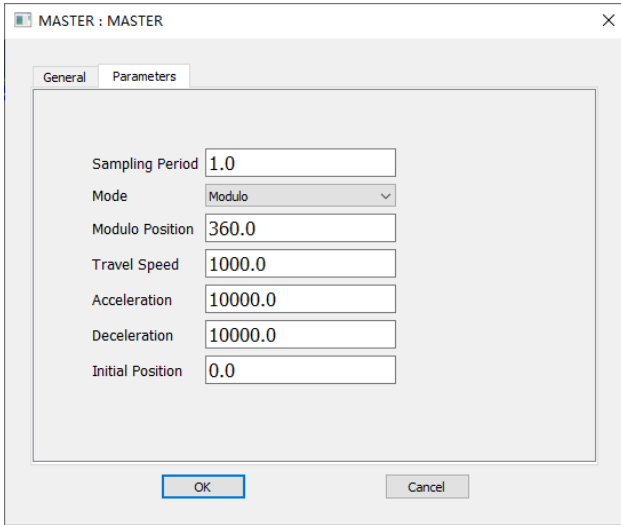

在 N 段里： 1、当设定速度和当前速度不一致时，主轴按设定速度运行，保存设定速度。

2、绝对位置定位指令上升沿触发，主轴移动到 MasterAbsPos 位置上。

3、相位位置定位指令上升沿触发，主轴移动 MasterDeltaPos 距离。

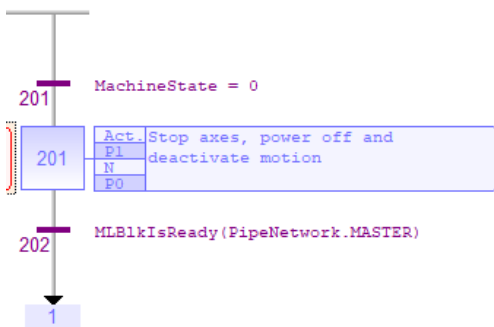移动的速度和加减速在那设置呢？在 PipeNetWork 中的 MASTER 模块里设置默认值，当然你可以通过相关函数实时修改这些设置。

在 P0 段里： 没有程序

第 201 步：Stop axes， power off and deactivate motion 运行过程中急停 停止轴运动，去使能，解耦运动关系 此程序段和 Main 的第四和五步一样。

进入条件：MachineState=0 退出条件：主轴停止 Ready 跳转到第一步

在 P1 段里： 主轴速度减速到 0


```
1    Printf('Stop axes', 0, 0, 0, 0);
2
3    MLMstRun(PipeNetwork.MASTER     14 , LREAL#0.0);
4    bLedStatus[2] := FALSE;
```

在 N 段里： 没有程序

在 P0 段里：驱动器去使能，解耦运动关系


```
1    Printf('PowerOff', 0, 0, 0, 0);
2
3    // Power off all axes
4    PipeNetwork(MLPN_POWER_OFF     5 );
5    bLedStatus[1] := FALSE;
6
7    // desactivate Pipes
8    PipeNetwork(MLPN_DEACTIVATE     6 );
9
```

单独操作伺服轴的 PipeNetWork 函数有哪些?

Power Stage	Motion Control	Inquiry Functions	Position setting
MLAxisPower	MLAxisAbs	MLAxisGenPos	MLAxisWritePos
MLAxisPowerDOff	MLAxisAdd	MLAxisPipePos	MLAxisReAlign
	MLAxisMoveVel	MLAxisCmdPos	
	MLAxisRel	MLAxisReadActPos	
	MLAxisStop	MLAxisFBackPos	
		MLAxisStatus	
		MLAxisReadGenStatus	
		MLAxisGenIsRdy	
		MLAxisTimeStamp	
		MLAxisDriveNumber	

还有一些对模块参数的修改函数。

具体详解 F1 帮助