Standard SFC Template

Kollmorgen Automation Suite - KAS

Writed by: Souza, Leonardo

09/2018 – rev1.0

Place: Barueri. São Paulo.

Contacts:
Phone / Fax: (+55 11) 2110-4219
Mobile: (+ 55 11) 98899-4712
Email: leonardo.souza@kollmorgen.com

Revision: rev1.0

Table of Contents

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

1.0 Introduction

If we look at PLC projects, we see that every machine deals with the same functionalities. Generally, machines powered by servo drives, need to follow a sequence of steps during its development:

I. Start a Network Protocol (if available);
II. Power on all drives;
III. Elect a reference method to home each axis;
IV. Develop a Manual Mode, allowing an operator to Jog an axis and change setup positions;
V. Finally development of an Automatic Mode, which runs desired machine functionalities;

All the steps described above are the core project parts and contained within the Main program, they establish a pathway for the machine code to be developed. In addition to these steps, the program requires some Auxiliary Programs to deal with Status, Faults, IOs, Recipe and/or any special routine.

This Standard SFC Template proposes a division of known functionalities into modules, organizing them using the high expressive power provided by KAS using SFC language. The objective is to provide a structure capable to control machines of different types and complexity levels, using the same SFC Structure and Auxiliary Programs.

Sequential Function Chart (SFC)

The SFC language is a state diagram. Graphical steps are used to represent stable states, and transitions describe the conditions and events that lead to a change of state. Using SFC highly simplifies the programming of sequential operations as it saves a lot of variables and tests just for maintaining the program context.

TIP: It's is important to know SFC Fundamentals to understand Standard SFC Template. A document explaining how it works can be found at KAS Help, section:
Technical References > Programming Languages > Sequential Function Chart (Link -> Click Here)

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

2.0 Specification

This template uses KAS (Kollmorgen Automation Suite) to build a complete automation solution, providing support to Motion Control, PLC Logic, and HMI. The program utilizes two IEC-61131 languages: SFC as a top-level program, building the state machine and ST for the coding inside of step qualifiers and on Auxiliary Programs.

PLCOpen has been chosen as motion engine for the application, composed by 2 Axes as type Servo and 1 Axis as a Virtual Master. All of them was configured with a ratio 1:1, with 360 degrees per revolution and without a rollover position.

Using KAS 2 Axis Demo as base hardware of development, this template is fully adapted to run with Demo's setup or using KAS Simulator.

2.1 EtherCAT Network

This template uses KAS (Kollmorgen Automation Suite) to build a complete automation solution, providing

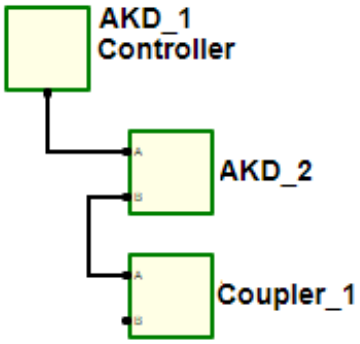


Figure 1. Template's EtherCAT Network.

2.2 Inputs and Outputs

2.2.1 Digital Inputs

AKD01 X7:10 – DIN1 – Limit Switch 1
AKD01 X7:09 – DIN2 – Free
AKD01 X7:04 – DIN3 – Free
AKD01 X7:03 – DIN4 – Free
AKD01 X8:06 – DIN5 – Free
AKD01 X7:05 – DIN6 – Free
AKD01 X7:02 – DIN7 – Free
AKD01 X35:02 – DIN21 – Free
AKD01 X35:03 – DIN22 – Free
AKD01 X35:04 – DIN23 – Free
AKD01 X36:02 – DIN24 – Free
AKD01 X36:03 – DIN25 – Free
AKD01 X36:04 – *DIN26* – Free

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

AKD02 X7:10 – DIN1 – Limit Switch 2
AKD02 X7:09 – DIN2 – Free
AKD02 X7:04 – DIN3 – Free
AKD02 X7:03 – DIN4 – Free
AKD02 X8:06 – DIN5 – Free
AKD02 X7:05 – DIN6 – Free
AKD02 X7:02 – DIN7 – Free

2.2.2 Digital Outputs

AKD01 X7:7 – DOUT1 – Free
AKD01 X7:5 – DOUT2 – Free
AKD01 X35:7 – DOUT21 – Free
AKD01 X36:7 – DOUT22 – Free

AKD02 X7:7 – DOUT1 – Free
AKD02 X7:5 – DOUT2 – Free

2.2.3 Analog Input

AKD01 X8:10 – AIN – Free
Auxilia
AKD02 X8:10 – AIN – Free

2.2.4 Analog Output

AKD01 X8:08 – AOUT – Free

AKD02 X8:08 – AOUT – Free

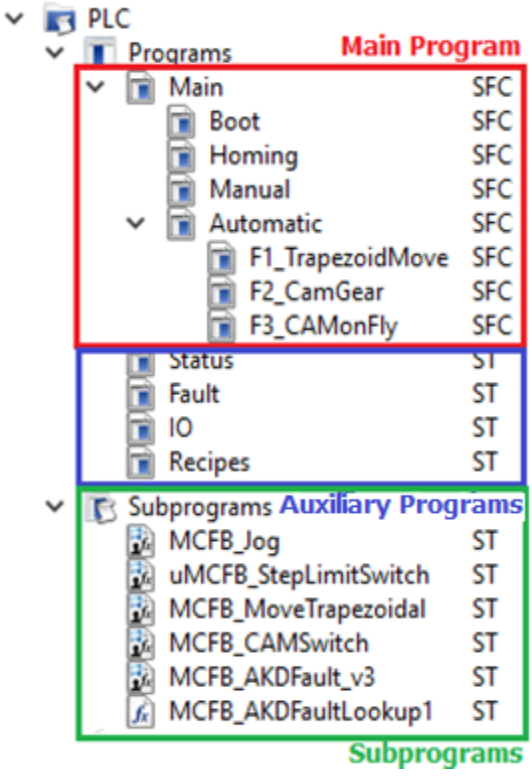
Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

3.0 How does Standard SFC Template works?

Standard SFC Template divide KAS Project into "Functionalities Groups", routines related to machine process are located inside of Main Program and Auxiliary Programs are developed using Standalone Programs.

The biggest advantage of using SFC structure is to translate the machine process in a state diagram. SFC is a good choice because:

- Each state of the process can be clearly mapped to a SFC step;
- The transition between steps is made using conditions, these are easily identified in the code;
- The relation between steps and activity flow from one state to another is visualized/programmed in a graphical manner, is easy to comprehend due conditions and interlockings which relate the evolution of step active states.
- SFC structure is a mirror of the controlled process and allows broke software into small parts. With this feature is easy to control what is being executed on each scan cycle of the controller.
- Machines mostly have similar state diagrams (Homing, Manual Mode, Automatic Mode…), this characteristic allows scalability of Standard SFC Template into more complex machines.



The **Main Program** is responsible to translate machine state diagram to PLC software. Is responsible to initialize the network, address, power and reference every axis, besides run all functionalities inside Manual and Automatic Modes.

Auxiliary Programs have as objective provide information to Main Program. They've an established run cycles, without any interlocking conditions, in other words, keep always running. The Template uses this programs to read machine status, deal with fault and reset routines and manage IOs and recipes.

Template's **Subprograms** are particles of code developed to perform a specific function. They're used inside of programs called as instances declared on project's dictionary.

Besides all PLC Programs structure described above, this project has a section to setup Motion and PLCOpen Axis and to manage EtherCAT Network devices. We've two different user interfaces, first using Control Panels inside KAS and second with a HMI project, developed using KVB (Kollmorgen Visualization Builder).

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

3.1 Dictionary

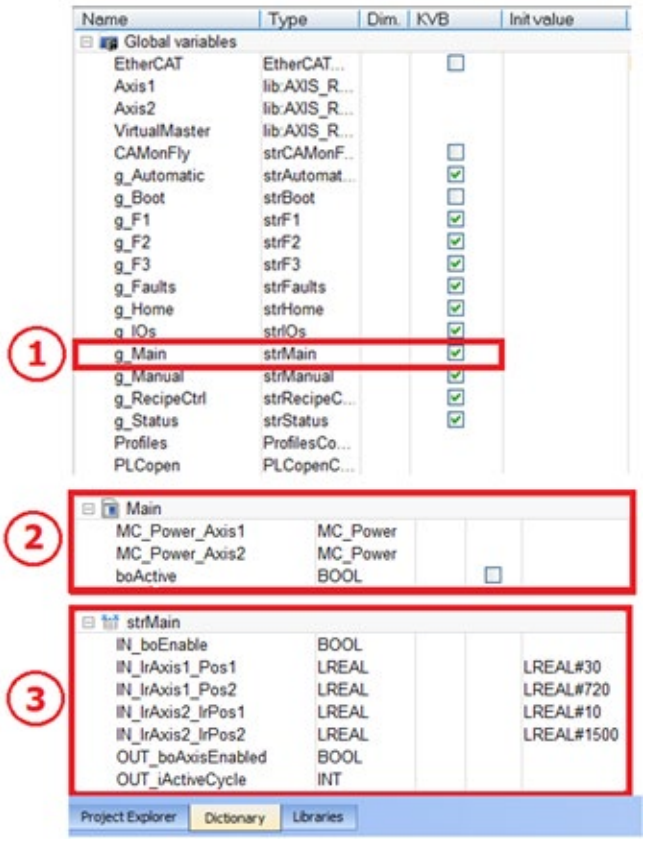


Figure 3. Project Dictionary. 1. Main Structures declares as Global Variable. 2. Main Local Variables. 3. Main Structure.

Each Program has a group of dedicated variables, related to its functionality. These variables are divided into two groups:

Local Variables

They can be accessed only inside their program. In Figure 3, the number 2 shows Local Variables that belongs to the Main Program.

Global Variables and Structures

These variables are special, so can be accessed on the entire project. During Template development we establish that each program has a Structure of Variable with his name: "str + Program Name". Adopting this definition is easy to identify from which program the Global Variable came from.

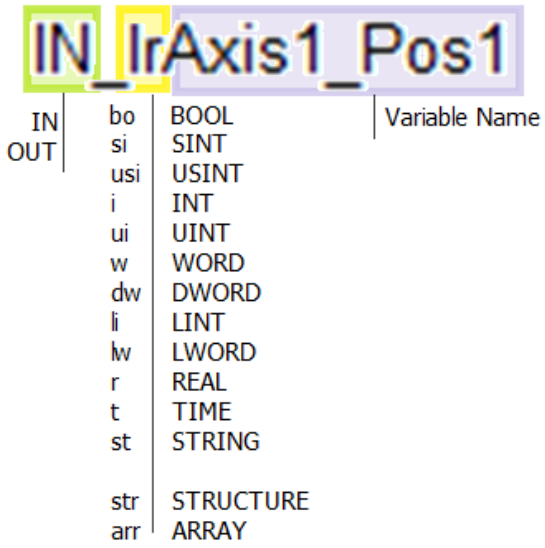
In Figure 3, the number 3 shows the Structure of Variables that belongs to Main Program and number 1 shows it being declared as a Global Variable.

TIP: During development is recommended to create all program variables as Local and change it to Global only if necessary.

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

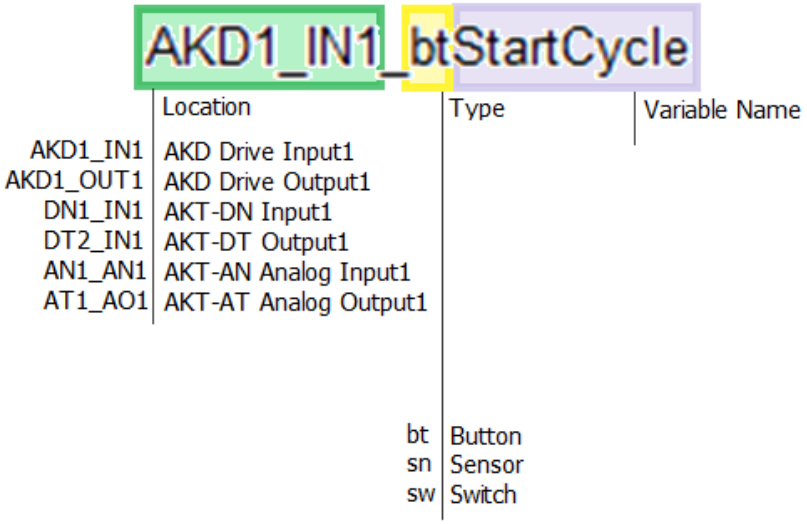
3.1.1 Naming Variables

To allow identification if the variable represent an Input or Output and its Type, the name of the variable is divided into 3 parts:



´ - Indicate if the variable is an INput or OUTput;
(**Only for global variables.** Local Variables don't have this part);
´ - Indicate variable data type;
´ - Variable Name;

3.1.2 Physical Inputs and Outputs



Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

3.1.3 Naming Subprograms

Some part of codes may be reused multiple time or for different jobs and are therefore programmed as subprograms. This minimizes the amount of duplicated code pieces. Subprogram use the Location (UDFB) underscore and then the individual name

e.g. UDFB_AxisHND, UDFB_AxisSDO

Those structure can be instantiated on any level in the programming. Starting with g if global otherwise start with INST underscore individual name of the UDFB and numbering if used multiple time.

e.g. Inst_AxisHND1, g_Inst_AxisSDO2

3.1.4 Usage of structure in connection with subprograms

Structure naming uses str as type following by the subprogram which is using it. If multiple subprograms are using the same structure make sure the subprograms have a common name start and use this as the structure name.

e.g. strAxisHND, strAxis

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

3.2 Main - SFC Program – State Machine Routine

The state machine and all functionalities related to machinery process are developed inside of Main Program, interlocking between each state is made using SFC Structure. Child

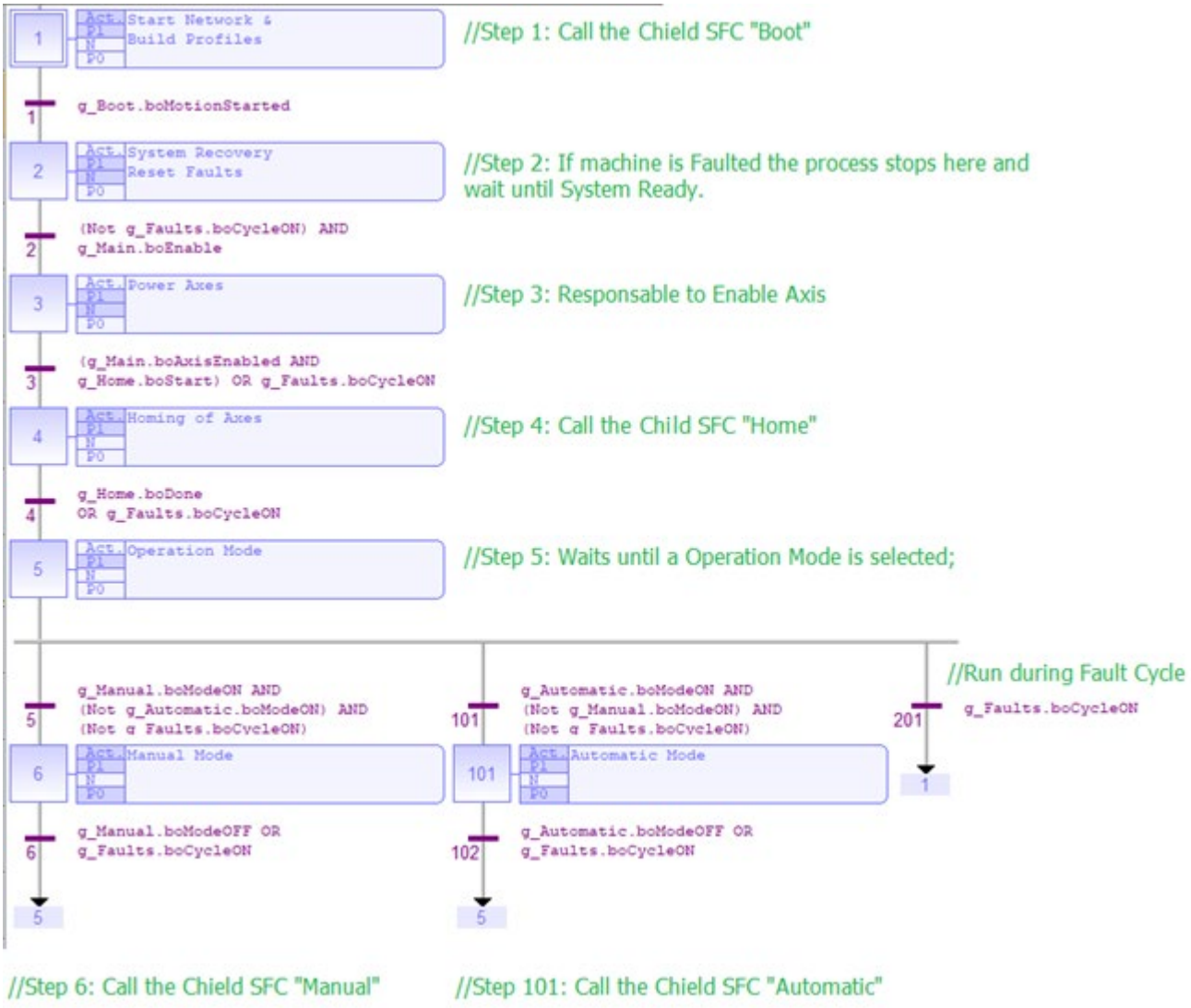


Figure 4. Main Program: SFC structure is used to build the state machine.

As long PDMM Controller is running, the Main Program is active. It indicates which state of operation is active. This feature facilitates the debugging process, while you're connected to the controller is clear to figure which part of the code is running and where it stops.

The program organization concept into a Main and Children SFC Programs also helps to organize the development process, by making clear to programmer all steps he needs to take to complete de project.

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

This division method can be equal to a large range of equipment, being easy to build a code for more complex machines using this same structure.

Global Variables – Main – SFC Program

```
g_Main          : strMain ;      // Call Variable Structure from Main as a Global Variable;
```

Local Variables - Main - SFC Program

```
MC_Power_Axis1      : MC_Power ;  //MC_Power to Enable Axis 1;
MC_Power_Axis2      : MC_Power ;  //MC_Power to Enable Axis 2;
```

Variables Structure – Main - SFC Program

```
boEnable            : BOOL ;          //Call routine to Enable Axes.
boAxisEnabled       : BOOL ;          //Indicates that all Axes are Enabled;
iActiveCycle        : INT ;           //Indicates active cycle of Main Program;
Axis1_lrPos1        : LREAL := LREAL#30 ;        //Axis 1: Setpoint Position 1;
Axis1_lrPos2        : LREAL := LREAL #720 ;      //Axis 1: Setpoint Position 2
Axis2_lrPos1        : LREAL := LREAL #10 ;       //Axis 2: Setpoint Position 1;
Axis2_lrPos2        : LREAL := LREAL #1500 ;     //Axis 2: Setpoint Position 2;
```

3.2.1 Boot – Child SFC - Network Startup and Motion Profiles

The Child SFC program Boot is called by the 1st Step of Main Program. It's responsible to Initialize EtherCAT Network and Motion Engine, set PLCOpen Addresses and create Motion Profiles. It sets a variable inside Step 3, noticing that Initialization is finished.

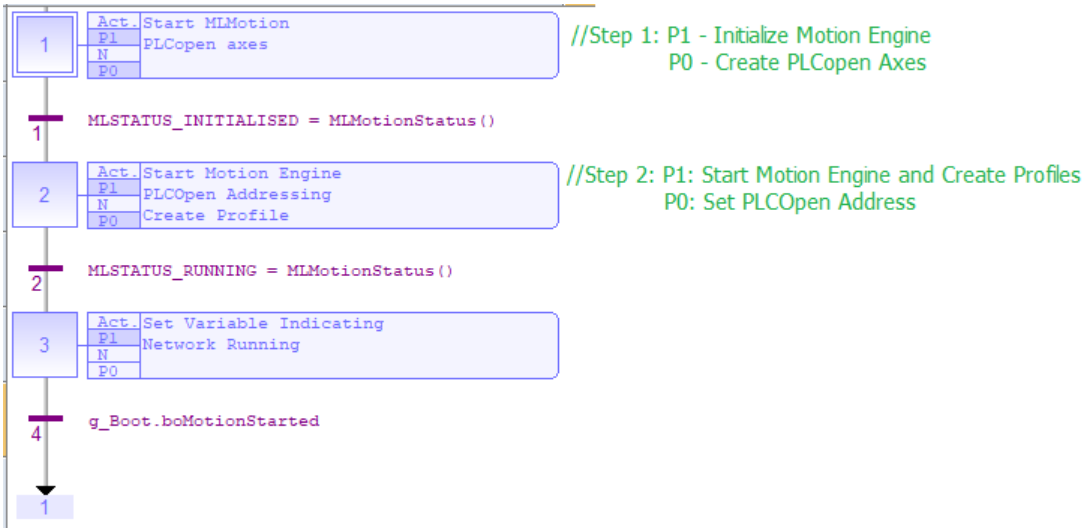


Figure 5. Boot: Child SFC program that starts machine network up.

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

Global Variables – Boot - Child SFC Program

g_Boot : strBoot ; //Call Variable Structure from Boot as a Global Variable;

Variables Structure – Boot - Child SFC Program

boMotionStarted : BOOL ; //Indicates that Boot routine is finished ;

3.2.2 Home – Child SFC – Axes Reference Routine

All servo driven shafts need to go through a reference process during its initialization. In Template all Home Routines are centered within the Child SFC Home, which performs reference to Axis 1 and Axis 2, using the function block MCFB_StepLimitSwitch.

The two axes move until the digital input mapped on function block is triggered, so feedback position of the axes is zeroed and then they moto to the respective Setpoint Position 1 of each axis.

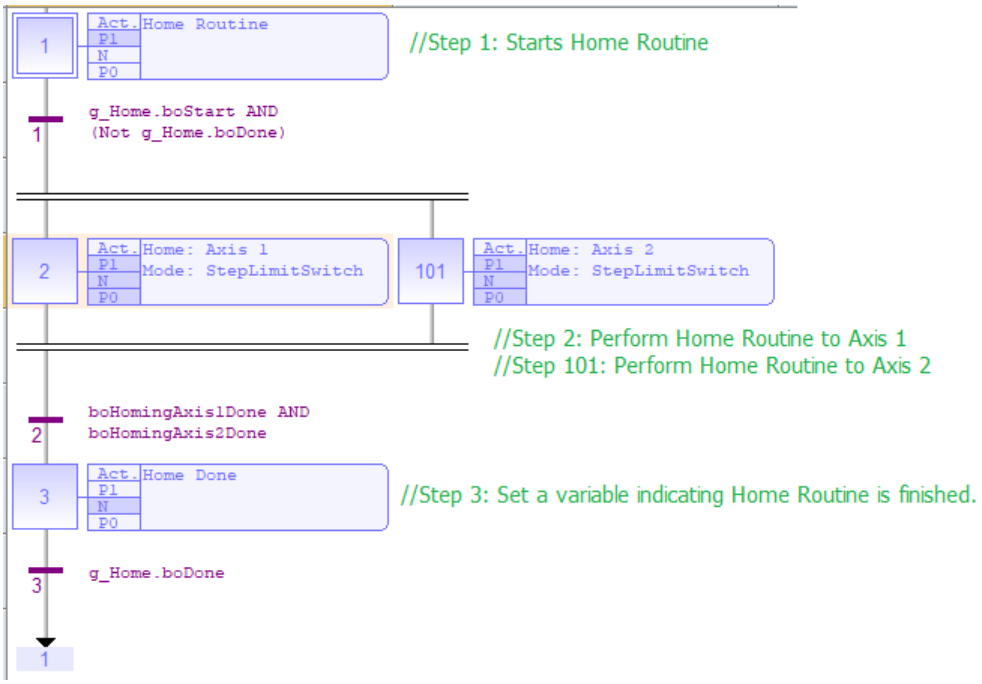


Figure 6. Home: Child SFC that performs Home Routine on axes;

Global Variables – Home – Child SFC Program

g_Home : strHome ; //Call Variable Structure from Home as a Global Variable;

Local Variables - Home - Child SFC Program

boActive : BOOL ; //Indicates that Home Routine is Active;
boHomingAxis1Done : BOOL ; //Indicates that Home of Axis 1 is Done;
boHomingAxis2Done : BOOL ; //Indicates that Home of Axis 2 is Done;
MC_Halt_Axis1 : MC_Halt ; //Function Block to decelerate Axis 1 to zero on beginning;
MC_Halt_Axis2 : MC_Halt ; //Function Block to decelerate Axis 2 to zero on beginning;

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

```
MCFB_LimitSwitchRef_Axis1  : uMCFB_StepLimitSwitch;   //Home Axis1 to a limit switch – AKD1_IN1;
MCFB_LimitSwitchRef_Axis2  : uMCFB_StepLimitSwitch;   //Home Axis1 to a limit switch – AKD2_IN1;
MC_MoveAbsolute_Axis1      : MC_MoveAbsolute ;        //Runs an Absolute Move on Axis 1 to Pos1;
MC_MoveAbsolute_Axis2      : MC_MoveAbsolute ;        //Runs an Absolute Move on Axis 2 to Pos1;
```

Variables Structure – Home - Child SFC Program

```
boStart            : BOOL ;   //Call Routine to Home Axes;
boDone             : BOOL ;   //Indicates that Home Routine is Done to all axes;
boAxis1Recovery    : BOOL ;   //Variable runs Home Routine for a 2nd time to Axis1;
boAxis2Recovery    : BOOL ;   //Variable runs Home Routine for a 2nd time to Axis2;
```

3.2.3 Manual – Child SFC – Manual Mode

Manual Mode is available after Home Process is finished, being on an agreement to Main Program state machine. This mode allows the axes to be positioned in their respective setpoints (Pos1 and Pos2) and the adjustment of these positions using Jog commands. Subsequently to adjustment, the operator can save the new position, updating setpoints and actual recipe.

On Manual Mode exits command, both axes are positioned in their respective setpoints (Pos1).

TIP: It is important to reset all function blocks used in the routines, programmed inside of SFC Strutures. This condition ensures that we can exit Manual Mode and enter it again, running all the routines.

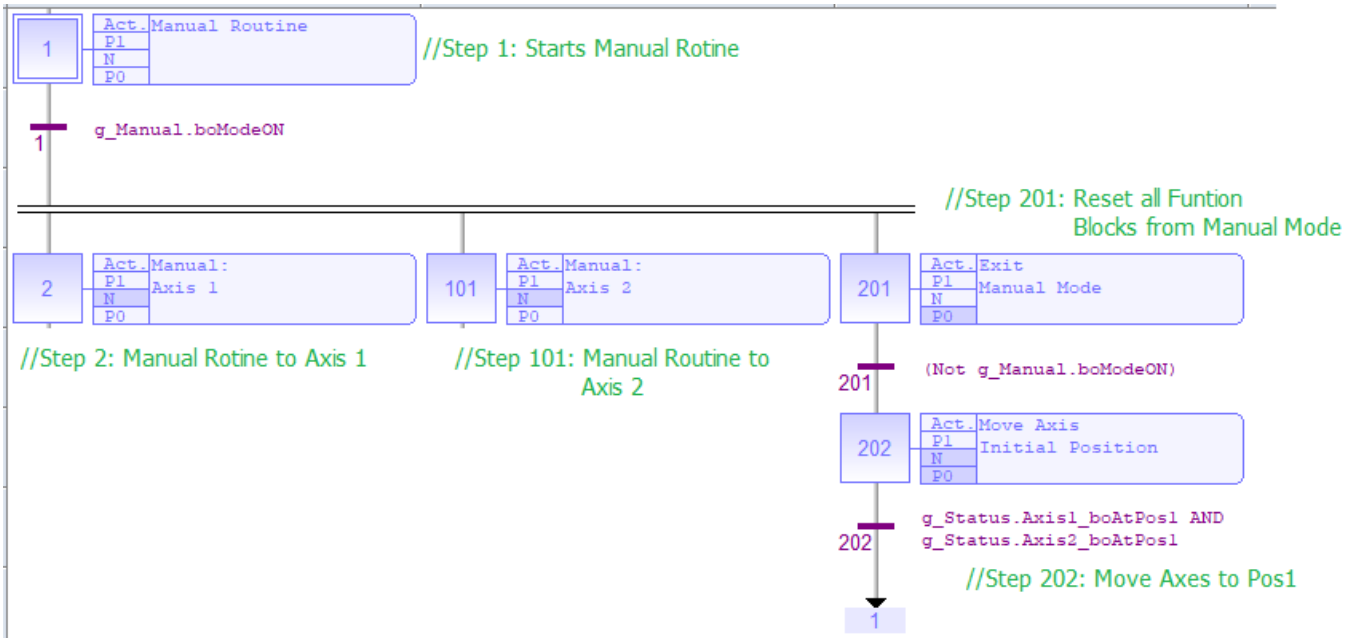


Figure 7. Manual: Child SFC that performs Manual Routine on axes;

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

Global Variables – Home – Child SFC Program

```
g_Manual        : strManual ;   //Call Variable Structure from Manual as a Global Variable;
```

Local Variables - Home - Child SFC Program

```
MC_MoveAbsolute_Axis1_1    : MC_MoveAbsolute ;      //Runs an Absolute Move on Axis 1 to Pos1;
MC_MoveAbsolute_Axis1_2    : MC_MoveAbsolute ;      //Runs an Absolute Move on Axis 1 to Pos2;
MC_MoveAbsolute_Axis2_1    : MC_MoveAbsolute ;      //Runs an Absolute Move on Axis 2 to Pos1;
MC_MoveAbsolute_Axis2_2    : MC_MoveAbsolute ;      //Runs an Absolute Move on Axis 2 to Pos2;
MCFB_Jog_Axis1             : MCFB_Jog ;             //Runs a Jog movement on Axis1;
MCFB_Jog_Axis2             : MCFB_Jog ;             //Runs a Jog movement on Axis2;
```

Variables Structure – Home - Child SFC Program

```
boModeON           : BOOL ;          //Variable used to command/indicate that Manual Mode is ON;
boModeOFF          : BOOL ;          //Variable used to command/indicate that Manual Mode is OFF;
Axis1_boJogMinus   : BOOL ;          //Enables a Jog in the Plus direction on Axis1;
Axis1_boJogPlus    : BOOL ;          //Enables a Jog in the Minus direction on Axis1;
Axis1_boSavePos    : BOOL ;          //Save current adjustment to Pos1 or Pos2 and to actual recipe;
Axis1_boSelPos1Pos2: BOOL ;          //Select and position Axis1 at Pos1 or Pos2;
Axis2_boJogMinus   : BOOL ;          //Enables a Jog in the Plus direction on Axis2;
Axis2_boJogPlus    : BOOL ;          //Enables a Jog in the Minus direction on Axis2;
Axis2_boSavePos    : BOOL ;          //Save current adjustment to Pos1 or Pos2 and to actual recipe;
Axis2_boSelPos1Pos2: BOOL ;          //Select and position Axis2 at Pos1 or Pos2;
Axis1_lrJogSpeed   : LREAL := LREAL#10 ;      //Jog rate speed to Axis1;
Axis1_lrJogAccDecel: LREAL := LREAL#1000 ;    //Jog Linear Acc/Dec rate to Axis1;
Axis2_lrJogSpeed   : LREAL := LREAL#10 ;      //Jog rate speed to Axis2;
Axis2_lrJogAccDecel: LREAL := LREAL#1000 ;    //Jog Linear Acc/Dec rate to Axis2;
```

3.2.4 Automatic – Child SFC – Automatic Mode

Automatic Mode is also available after Home Routine is finished. Like other parts of the program, it's also oriented to group the code into functionalities.

Three different functions are available in automatic mode (F1, F2, and F3). This program isn't responsible for performing the functionalities, it only interlocks them and calls a Child SFC to run them.

Global Variables – Automatic – Child SFC Program

```
g_Automatic        : strAutomatic ;    //Call Variable Structure from Automatic as a Global Variable;
```

Variables Structure – Automatic - Child SFC Program

```
boModeON           : BOOL ;          //Variable used to command/indicate that Automatic Mode is ON;
boModeOFF          : BOOL ;          //Variable used to command/indicate that Automatic Mode is OFF;
```


Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

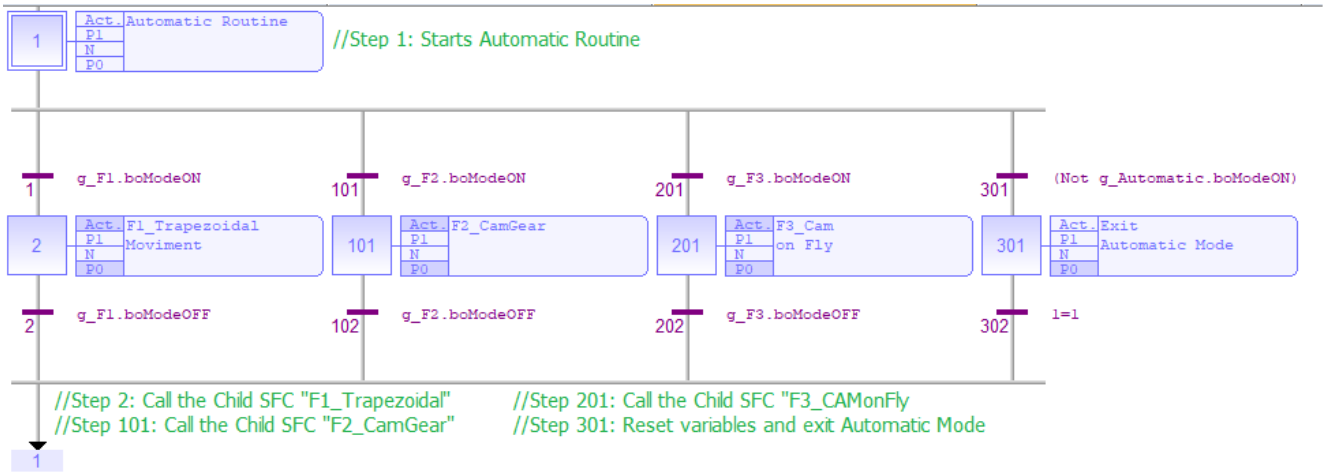


Figure 8. Automatic: Child SFC that performs Automatic Routine;

3.2.4.1 F1_TrapzoidMove – Child SFC – Trapezoidal Point-to-Point Movement

The first feature performed by Automatic Mode is responsible for moving the axes using an absolute position command traveled distance is set by positions P1 and P2.

The trapezoidal movement calculation is made by the auxiliary routine “MCFB_MoveTrapezoidal”, desired time for the movement is used as input and the block provide command speed and acceleration to perform a trapezoidal movement, which 1/3 of the time is used for accelerating, 1/3 is used for constant velocity, and 1/3 is used for decelerating.

Global Variables – F1_TrapzoidMove – Child SFC Program

```
g_F1            : strF1;        //Call Variable Structure from F1 as a Global Variable;
```

Local Variables – F1_TrapzoidMove – Child SFC Program

```
boAxis1_P1P2            : BOOL;         //Runs movement on Axis1 from P1 to P2;
boAxis1_P2P1            : BOOL;         //Runs movement on Axis1 from P2 to P1;
boAxis2_P1P2            : BOOL;         //Runs movement on Axis2 from P1 to P2;
boAxis2_P2P1            : BOOL;         //Runs movement on Axis2 from P2 to P1;
lrAxis1Distance         : LREAL;        //Calculate Axis1 Absolute distance between P1-P2;
lrAxis2Distance         : LREAL;        //Calculate Axis2 Absolute distance between P1-P2;
MC_Halt_Axis1           : MC_Halt;      //Function Block to decelerate Axis1 to zero;
MC_Halt_Axis2           : MC_Halt;      //Function Block to decelerate Axis2 to zero;
MC_MoveAbsolute_Axis1   : MC_MoveAbsolute;          //Runs an Absolute Move on Axis1
MC_MoveAbsolute_Axis2   : MC_MoveAbsolute;          //Runs an Absolute Move on Axis2
MCFB_MoveTrapezoidal_Axis1_P1P2 : MCFB_MoveTrapezoidal;     //Calculate Trap. Move to Axis1
MCFB_MoveTrapezoidal_Axis2_P1P2 : MCFB_MoveTrapezoidal;     //Calculate Trap. Move to Axis2
MyBlink                 : blink ;       //Trigger Move Axis 1 from P1 to P2
MyBlink1                : blink ;       //Trigger Move Axis 2 from P1 to P2
```


Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

Variables Structure – F1_TrapzoidMove - Child SFC Program

```
boModeON            : BOOL ;        //Variable used to command/indicate that F1 is ON;
boModeOFF           : BOOL ;        //Variable used to command/indicate that F1 is OFF;
Axis1_tMoveTime     : TIME := T#300ms ;          //Setpoint of Time to Axis1 move from P1-P2;
Axis1_lrVelMax      : LREAL := LREAL#1000 ;      //Setpoint of Velocity to Axis1 move from P1-P2;
Axis1_lrAccMax      : LREAL := LREAL#10000 ;     //Setpoint of Accel. to Axis1 move from P1-P2;
Axis1_lrRatio       : LREAL := LREAL#1 ;         //Axis1 Ratio between Physical and PLC Open;
Axis1_lrVelCmd      : LREAL ;       //Calculated Cmd. Velocity to perform Axis1 move from P1-P2;
Axis1_lrAccDecelCmd : LREAL ;       //Calculated Cmd. Accel. to perform Axis1 move from P1-P2;
Axis1_tErrMove      : TIME ;        //Error between set and calculated time to Move Axis 1 from P1-P2;
Axis2_tMoveTime     : TIME := T#1s ;             //Setpoint of Time to Axis2 move from P1-P2;
Axis2_lrVelMax      : LREAL := LREAL#500 ;       //Setpoint of Velocity to Axis2 move from P1-P2;
Axis2_lrAccMax      : LREAL := LREAL#10000 ;     //Setpoint of Accel. to Axis2 move from P1-P2;
Axis2_lrRatio       : LREAL := LREAL#1 ;         //Axis2 Ratio between Physical and PLC Open;
Axis2_lrVelCmd      : LREAL ;       //Calculated Cmd. Velocity to perform Axis2 move from P1-P2;
Axis2_lrAccDecelCmd : LREAL ;       //Calculated Cmd. Accel. to perform Axis2 move from P1-P2;
Axis2_tErrMove      : TIME ;        //Error between set and calculated time to Move Axis2 from P1-P2;
```

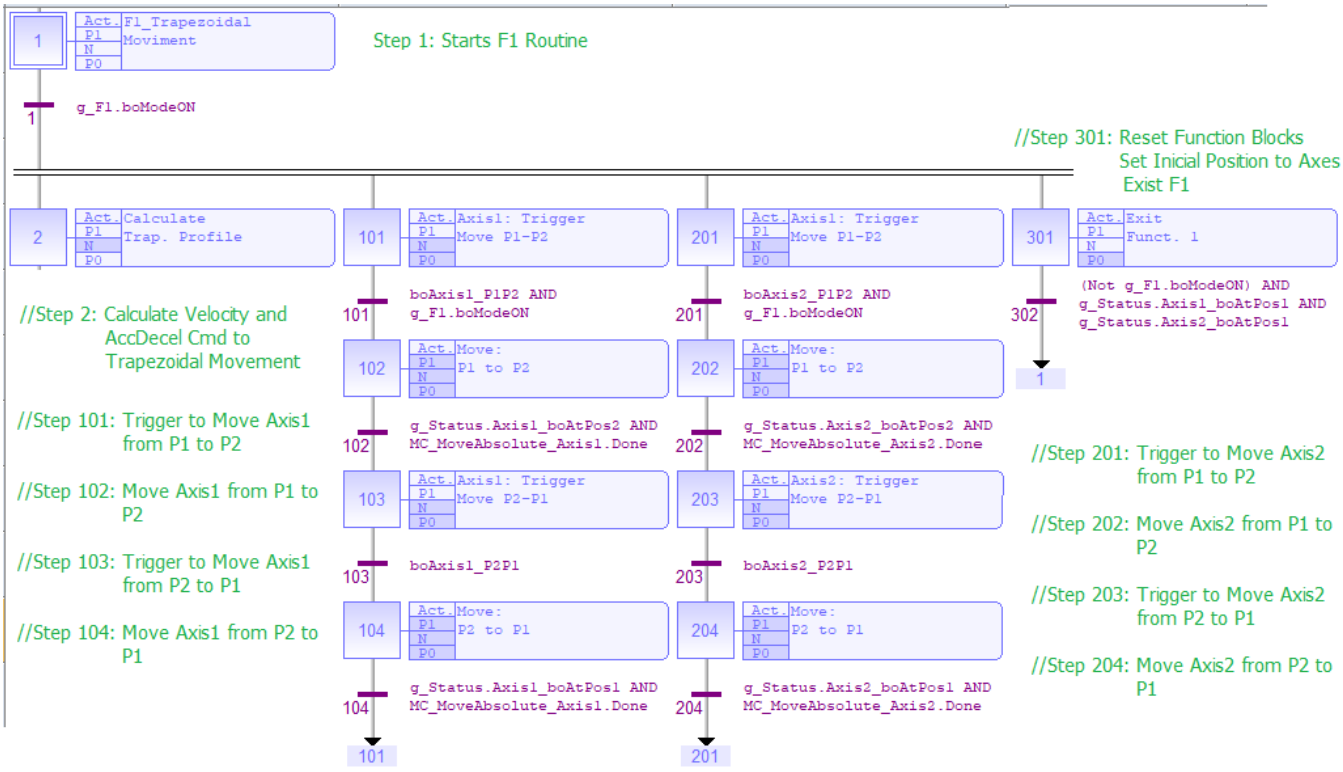


Figure 9. F1_TrapzoidMove: Child SFC that performs F1 Routine;

Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

3.2.4.2 F2_ CamGear – Child SFC – Camming ang Gearing Movement

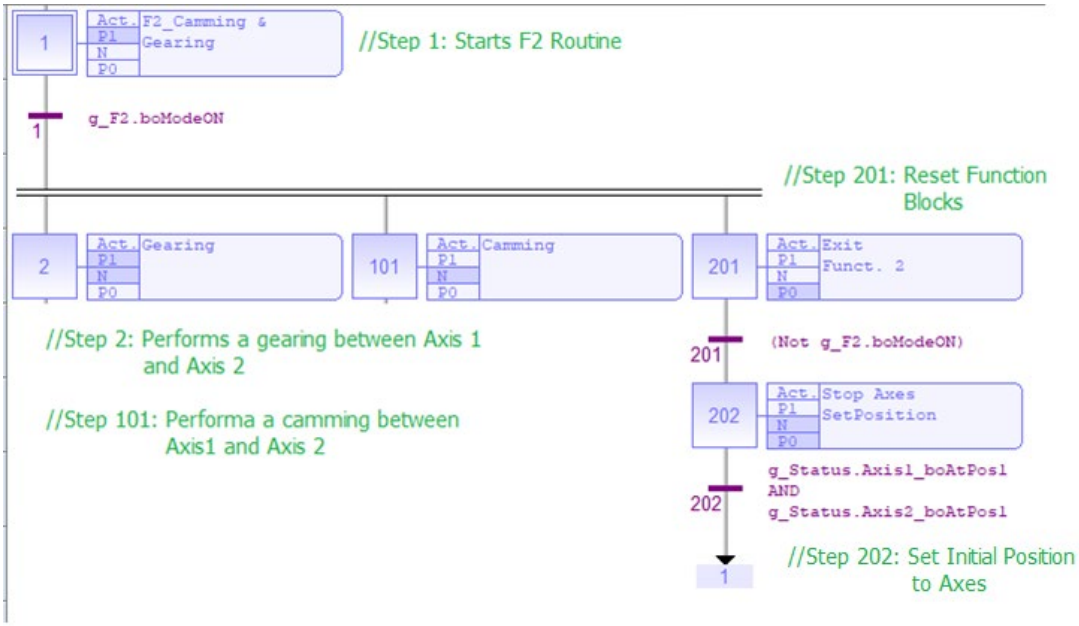


Figure 10. F2_CamGear: Child SFC that performs F2 Routine;

Global Variables – Automatic – Child SFC Program

```
g_F2            : strF2;        //Call Variable Structure from F2 as a GloSbal Variable;
```

Local Variables – Automatic – Child SFC Program

```
Inst_MC_SetPos_Axis1        : MC_SetPos ;          //Sets Initial Position (P1) to Axis1;
Inst_MC_SetPos_Axis2        : MC_SetPos ;          //Sets Initial Position (P1) to Axis2;
MC_MoveContVel_Axis1        : MC_MoveContVel ;     //Move Velocity to Axis1 Master;
MC_CamTblSelect_Profile1    : MC_CamTblSelect ;    //Read and initialize the specified profile;
MC_GearIn_Axis2             : MC_GearIn            //Axis2 follows Axis1 based on a ratio;
MC_CamIn_Profile1           : MC_CamIn ;           //Axis2 follows Axis1 based on the Cam Table;
MC_Halt_Axis1               : MC_Halt ;            //Function Block to decelerate Axis1 to zero;
MC_Halt_Axis1_1             : MC_Halt ;           //Function Block to decelerate Axis1 to zero;
MC_Halt_Axis2               : MC_Halt ;           //Function Block to decelerate Axis2 to zero;
MC_Halt_Axis2_1             : MC_Halt ;            //Function Block to decelerate Axis2 to zero;
```

Variables Structure – Automatic - Child SFC Program

```
boModeON            : BOOL ;    //Variable used to command/indicate that F2 is ON;
boModeOFF           : BOOL ;    //Variable used to command/indicate that F2 is OFF;
boStartMove         : BOOL ;    //Perform Axis1 move at a specified velocity;
boEngage_Gearing    : BOOL ;    //Enable gearing between Axis1 and Axis2, using gearing;
boGearingEngaged    : BOOL ;    //Indicate that Axis2 is engaged to Axis1;
boEngage_Camming    : BOOL ;    //Enable camming between Axis1 and Axis2;
```


Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

```
boCammingEngaged  : BOOL ;        //Indicate that Axis2 is engaged to Axis1, using camming;
boStopEngagement  : BOOL ;        //Disengage that Axis2 from Axis1;
lrMasterVel : LREAL   := LREAL#1000 ;      //Set Axis1 (Master) command velocity;
diRatioNumerator      : DINT := DINT#2 ;   //Numerator of Axis1/Axis2 ratio
diRatioDenominator    : DINT := DINT#1 ;   //Denominator of Axis1/Axis2 ratio
```

3.2.4.3 F3_CAMonFly – Child SFC - Camming on the Fly

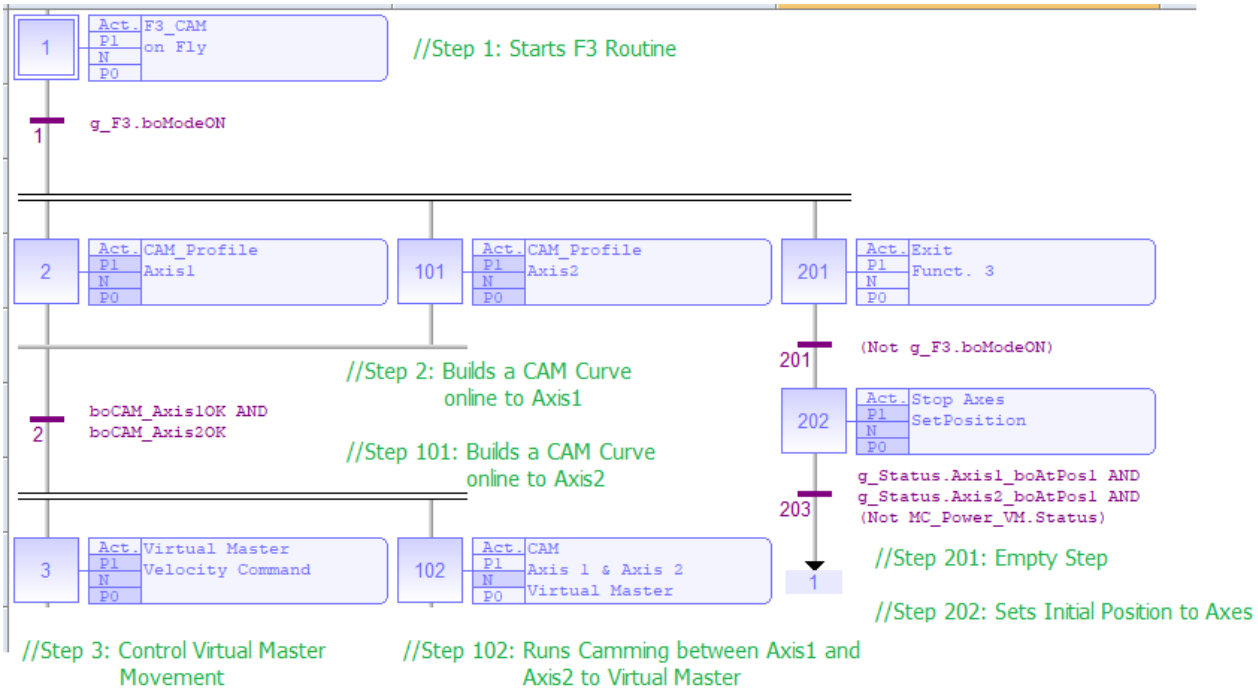


Figure 11. F3_CAMonFly: Child SFC that performs F3 Routine;

Global Variables – F3_CAMonFly – Child SFC Program

```
g_F3          : strF3;      //Call Variable Structure from F3 as a Global Variable;
```

Local Variables – F3_CAMonFly – Child SFC Program

```
boCAM_Axis1OK     : BOOL ;        //Indicate that CAM Profile Table to Axis1 is calculated;
boCAM_Axis2OK     : BOOL ;        //Indicate that CAM Profile Table to Axis2 is calculated;
boChangeVel       : BOOL ;        //Auxiliary flag that indicates changes on VM command velocity;
Inst_MC_SetPos_Axis1    : MC_SetPos ;          //Sets Initial Position (P1) to Axis1;
Inst_MC_SetPos_Axis2    : MC_SetPos ;          //Sets Initial Position (P1) to Axis2;
lrVM_Velocity1          : LREAL ;              //Command velocity to Virtual Master;
MC_CamIn_Axis1          : MC_CamIn ;           //Axis1 follows VM based on the Cam Table;
MC_CamIn_Axis2          : MC_CamIn ;           //Axis2 follows VM based on the Cam Table;
MC_Halt_VM              : MC_Halt ;            //Function Block to decelerate VM to zero;
```


Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

```
MC_Halt_Axis1_1          : MC_Halt ;              //Function Block to decelerate Axis1 to zero;
MC_Halt_Axis2_1          : MC_Halt ;              //Function Block to decelerate Axis2 to zero;
MC_MoveVelocity_VM    : MC_MoveVelocity ;      //Move velocity to Virtual Master;
MC_Power_VM           : MC_Power ;             //MC_Power to Enable Virtual Master;
MCFB_CAMSwitch_Axis1  : MCFB_CAMSwitch ;       //Builds a CAM curve online to Axis1;
MCFB_CAMSwitch_Axis2  : MCFB_CAMSwitch ;       //Builds a CAM curve online to Axis2;
TON_StartF3           : TON ;                  //Timer to delays F3 startup;
VelTrigger            : r_trig ;       //Auxiliary flag that indicates changes on VM command velocity;
```

Variables Structure – F3_CAMonFly - Child SFC Program

```
boModeON           : BOOL ;       //Variable used to command/indicate that F3 is ON;
boModeOFF          : BOOL ;       //Variable used to command/indicate that F3 is OFF;
boEngage_Camming   : BOOL ;       //Enable camming between Axis1 and Axis2 to VM;
boCammingEngaged   : BOOL ;       //Indicate that Axis1 and Axis2 is engaged to VM;
boStopEngagement   : BOOL ;       //Disengage that Axis1 and Axis2 from VM;
lrVM_Velocity      : LREAL := LREAL#10 ;          //Commanded velocity to VM;
```


Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

4.0 How scale Standard SFC Template to a 7 Axes Machine?

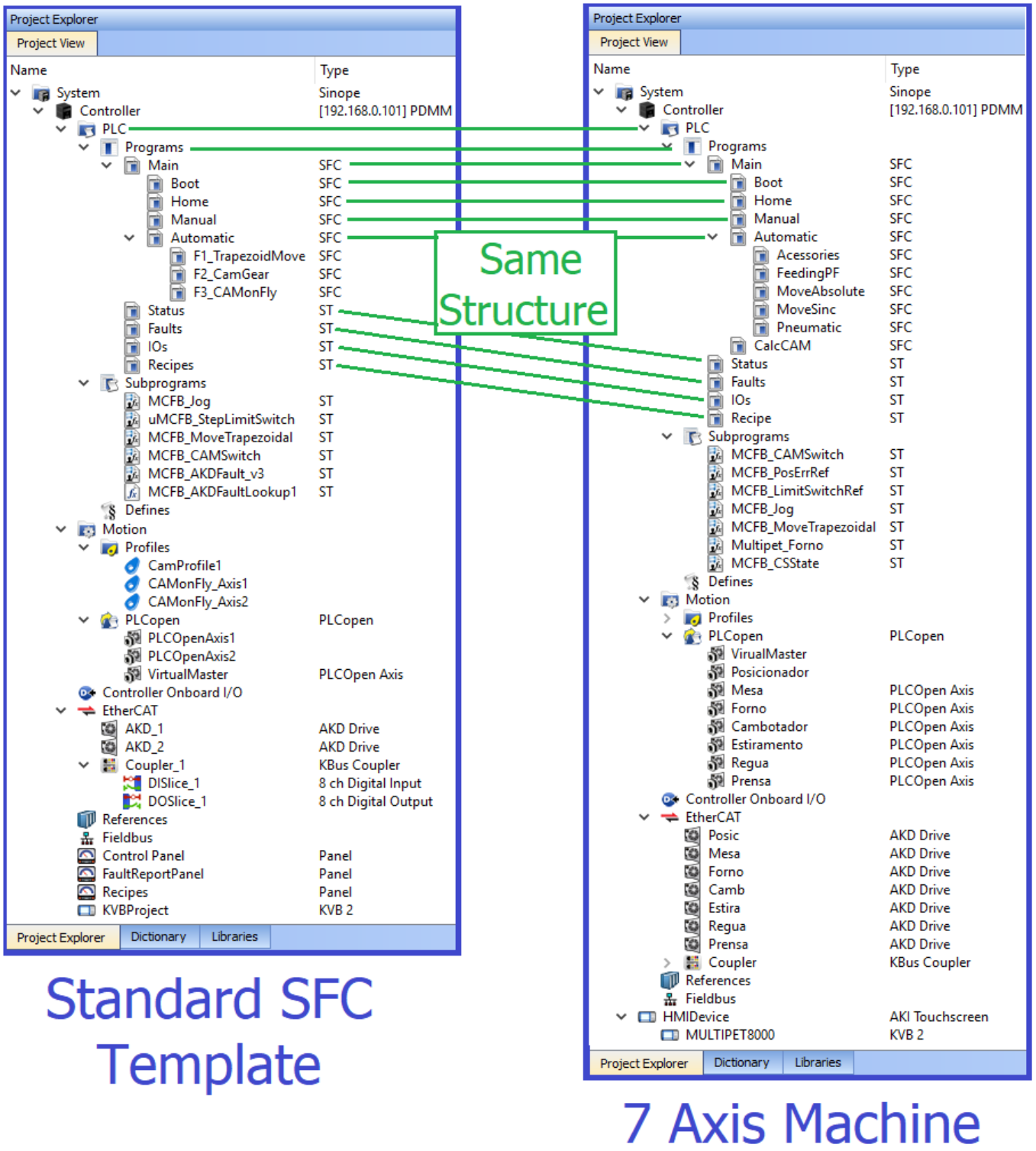
The template is designed to allow scalability for more complex projects. Making use of the structure, the programmer has a ponderous platform to organize and structure of reasoning, during the development of the project.

The division of the machine into functionalities and the organization of development in a state machine, allows the whole project to be structured before the development of logic. The SFC Structure helps the programmer to identify all the needed steps to start and finish the project. It builds all project parts using a graphical and intuitive environment.

In the following topics we see the Standard SFC Template scaled to a 7-axis machine.

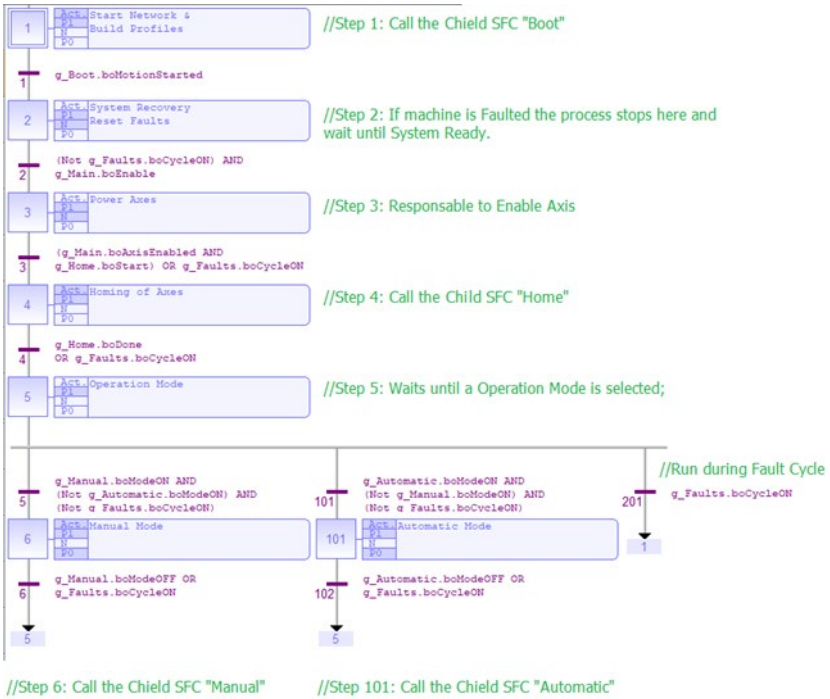
Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

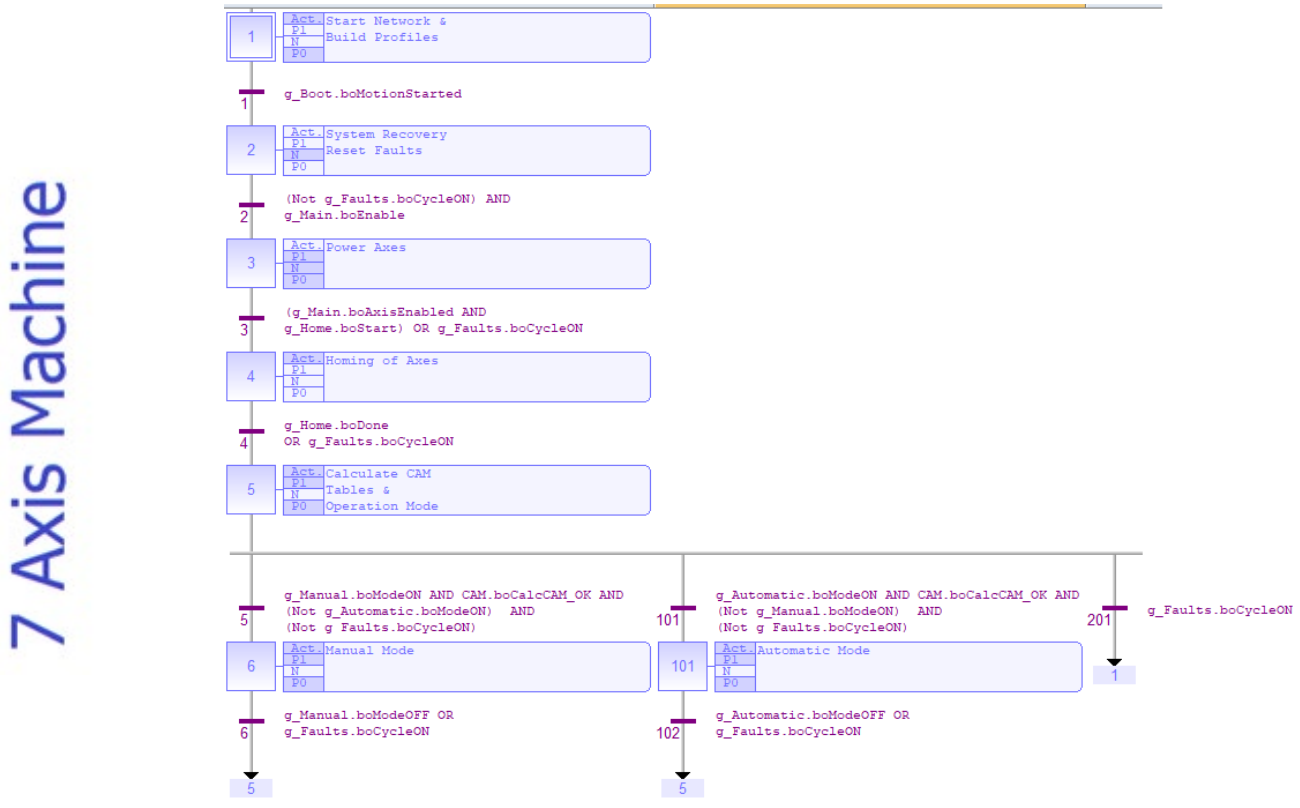
4.1 Project Breakdown



Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

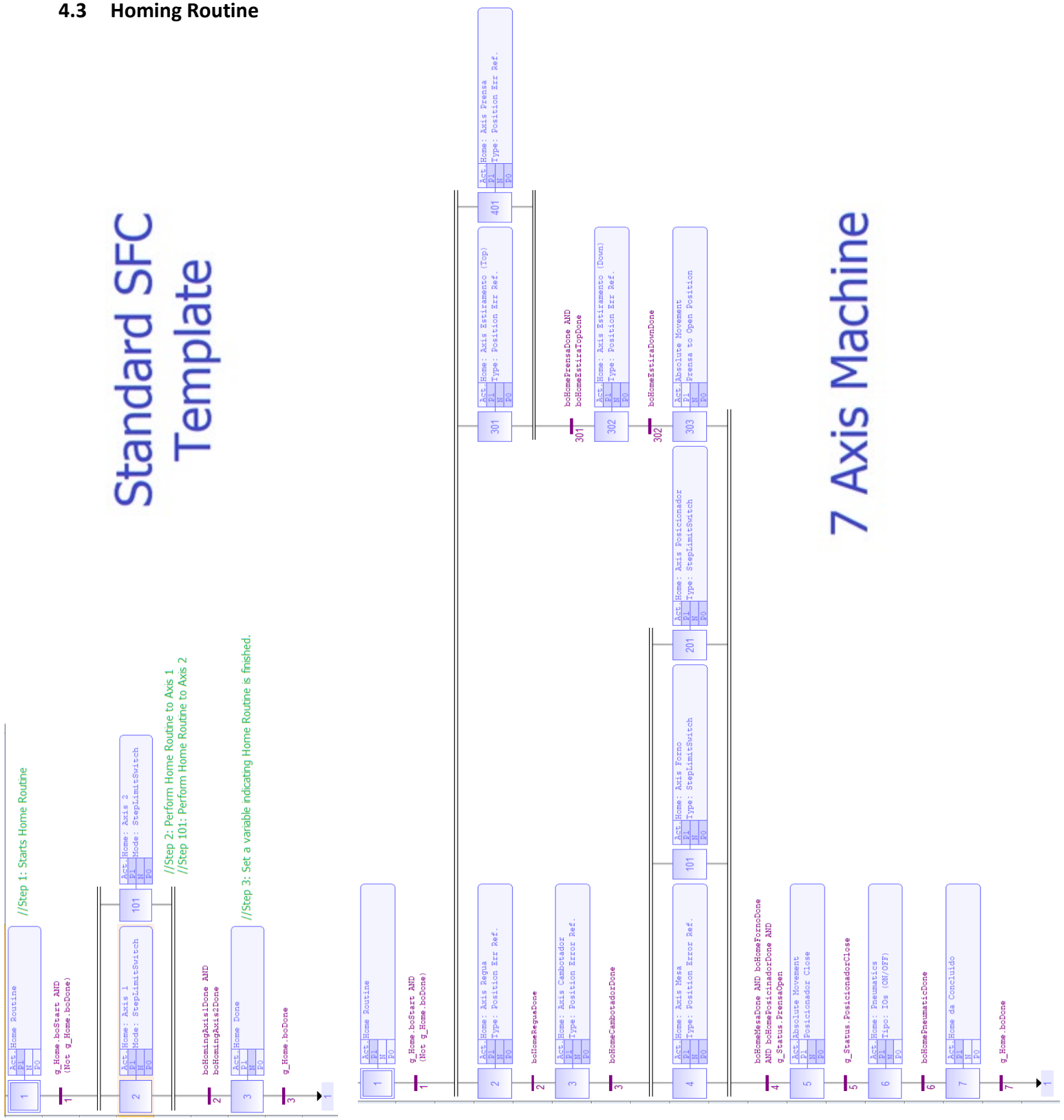
4.2 Main Routine





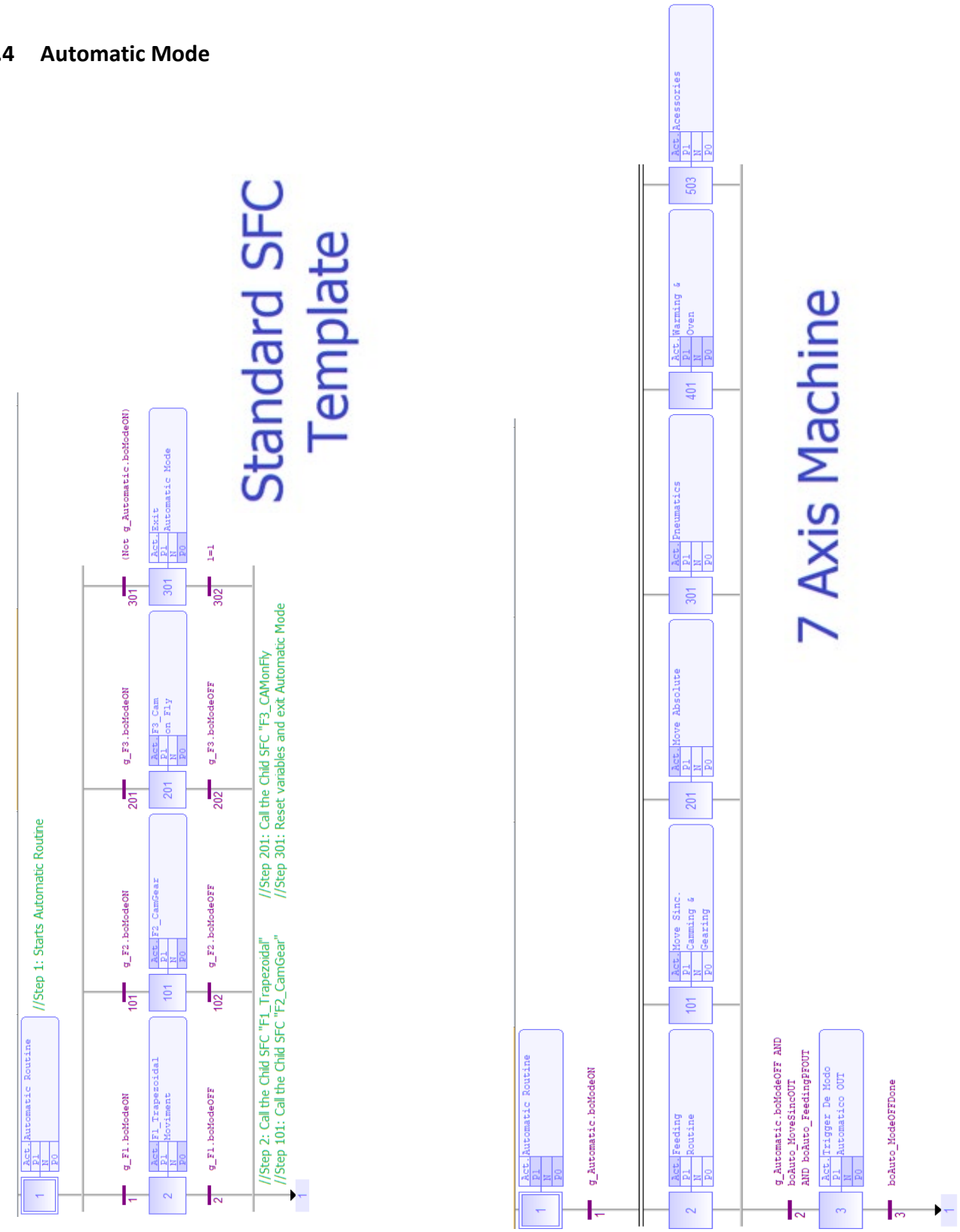
Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

4.3 Homing Routine



Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com

4.4 Automatic Mode



Avenida Tamboré – 1077 – Tamboré, Barueri - SP Brasil 06460-000 – Tel: (11) 4191-4771
www.kollmorgen.com.br / www.kdn.kollmorgen.com