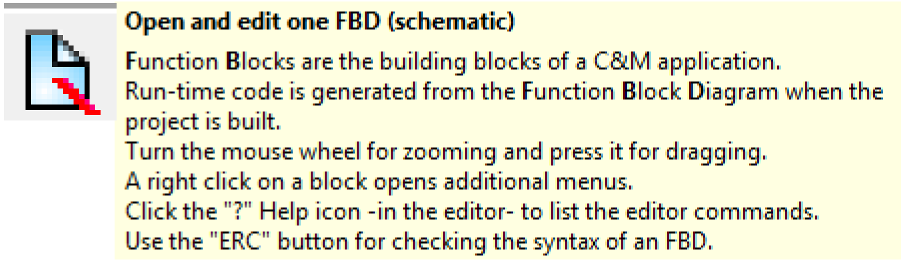
Click & Move Tutorial

Create a Project from Scratch

Description

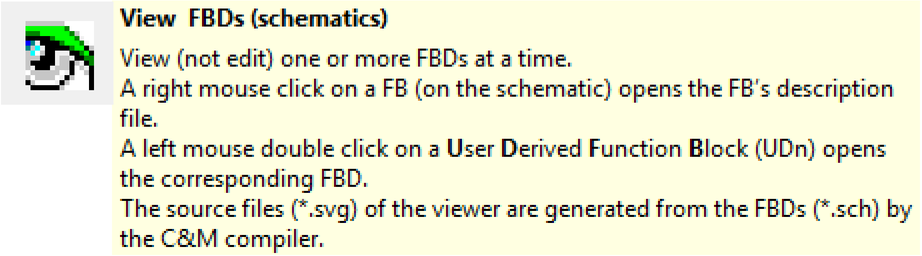
- This project will control the position of a Virtual Axis representing a CANopen servo-drive.

Tool Bar Descriptions



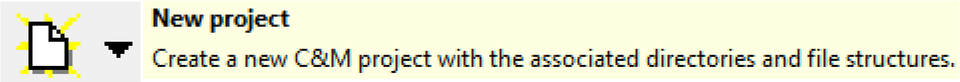
Open and edit one FBD (schematic)

Function Blocks are the building blocks of a C&M application.
Run-time code is generated from the Function Block Diagram when the project is built.
Turn the mouse wheel for zooming and press it for dragging.
A right click on a block opens additional menus.
Click the "?" Help icon -in the editor- to list the editor commands.
Use the "ERC" button for checking the syntax of an FBD.



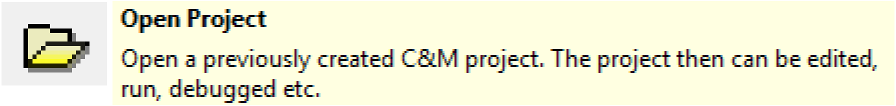
View FBDs (schematics)

View (not edit) one or more FBDs at a time.
A right mouse click on a FB (on the schematic) opens the FB's description file.
A left mouse double click on a User Derived Function Block (UDn) opens the corresponding FBD.
The source files (*.svg) of the viewer are generated from the FBDs (*.sch) by the C&M compiler.



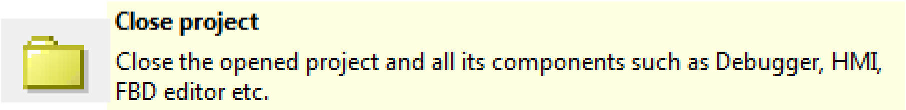
New project

Create a new C&M project with the associated directories and file structures.



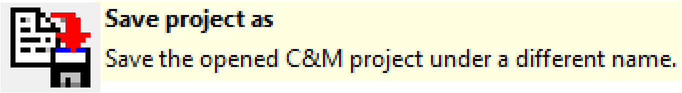
Open Project

Open a previously created C&M project. The project then can be edited, run, debugged etc.



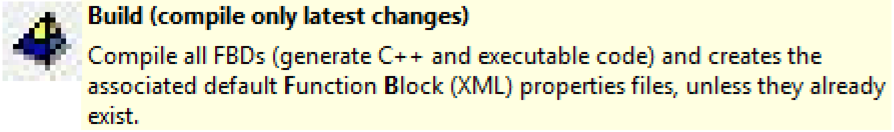
Close project

Close the opened project and all its components such as Debugger, HMI, FBD editor etc.



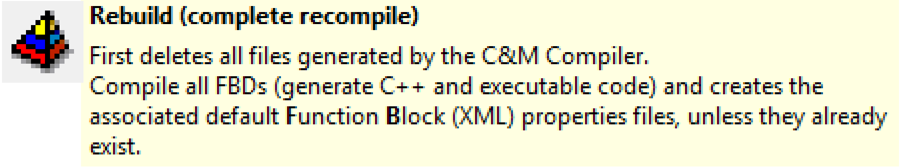
Save project as

Save the opened C&M project under a different name.



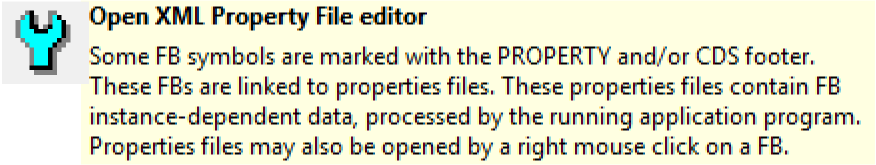
Build (compile only latest changes)

Compile all FBDs (generate C++ and executable code) and creates the associated default Function Block (XML) properties files, unless they already exist.



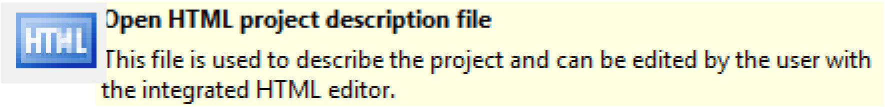
Rebuild (complete recompile)

First deletes all files generated by the C&M Compiler.
Compile all FBDs (generate C++ and executable code) and creates the associated default Function Block (XML) properties files, unless they already exist.



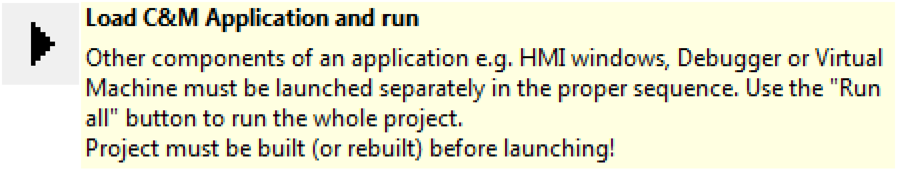
Open XML Property File editor

Some FB symbols are marked with the PROPERTY and/or CDS footer.
These FBs are linked to properties files. These properties files contain FB instance-dependent data, processed by the running application program.
Properties files may also be opened by a right mouse click on a FB.



Open HTML project description file

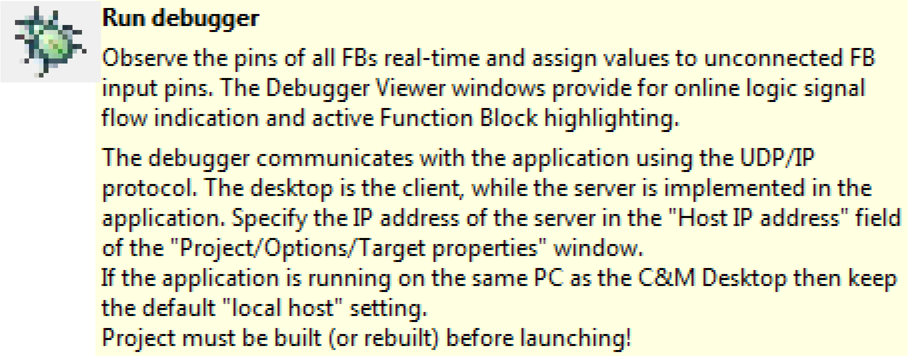
This file is used to describe the project and can be edited by the user with the integrated HTML editor.



Load C&M Application and run

Other components of an application e.g. HMI windows, Debugger or Virtual Machine must be launched separately in the proper sequence. Use the "Run all" button to run the whole project.
Project must be built (or rebuilt) before launching!

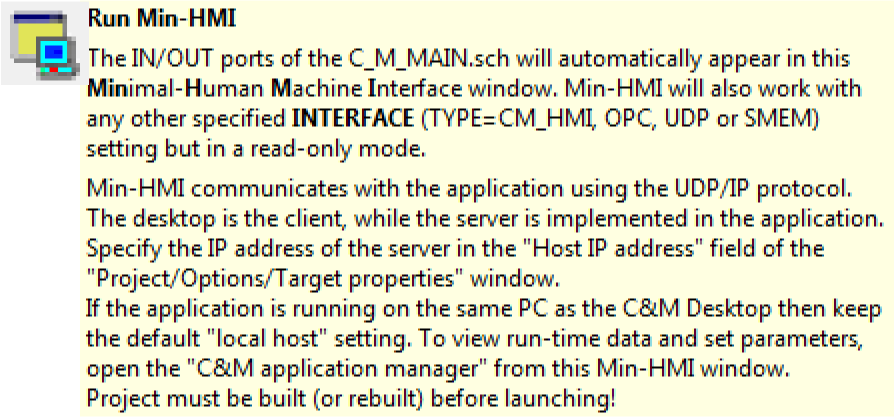
Tool Bar Descriptions (Cont.)



Run debugger

Observe the pins of all FBs real-time and assign values to unconnected FB input pins. The Debugger Viewer windows provide for online logic signal flow indication and active Function Block highlighting.

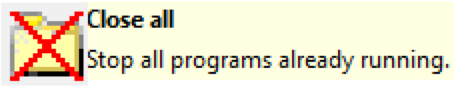
The debugger communicates with the application using the UDP/IP protocol. The desktop is the client, while the server is implemented in the application. Specify the IP address of the server in the "Host IP address" field of the "Project/Options/Target properties" window.
If the application is running on the same PC as the C&M Desktop then keep the default "local host" setting.
Project must be built (or rebuilt) before launching!



Run Min-HMI

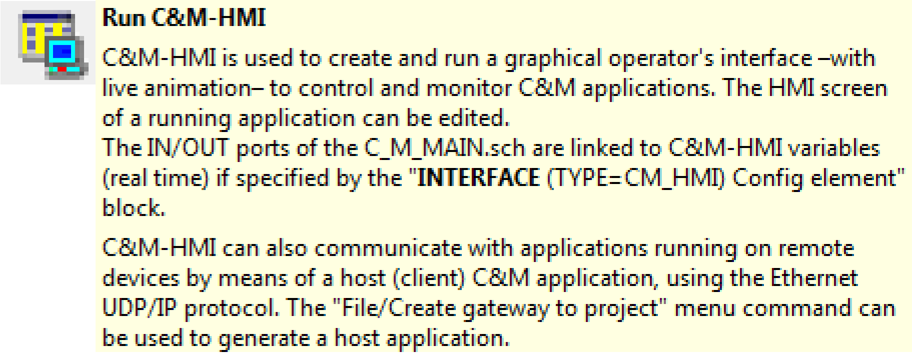
The IN/OUT ports of the C_M_MAIN.sch will automatically appear in this **Min**imal-**H**uman **M**achine **I**nterface window. Min-HMI will also work with any other specified **INTERFACE** (TYPE=CM_HMI, OPC, UDP or SMEM) setting but in a read-only mode.

Min-HMI communicates with the application using the UDP/IP protocol. The desktop is the client, while the server is implemented in the application. Specify the IP address of the server in the "Host IP address" field of the "Project/Options/Target properties" window.
If the application is running on the same PC as the C&M Desktop then keep the default "local host" setting. To view run-time data and set parameters, open the "C&M application manager" from this Min-HMI window.
Project must be built (or rebuilt) before launching!



Close all

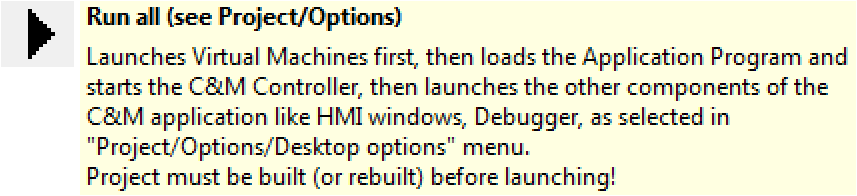
Stop all programs already running.



Run C&M-HMI

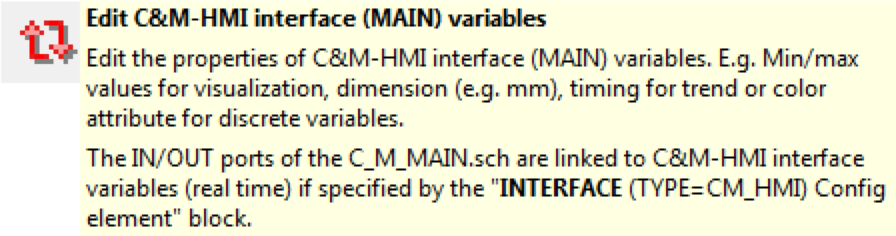
C&M-HMI is used to create and run a graphical operator's interface –with live animation– to control and monitor C&M applications. The HMI screen of a running application can be edited.
The IN/OUT ports of the C_M_MAIN.sch are linked to C&M-HMI variables (real time) if specified by the "**INTERFACE** (TYPE=CM_HMI) Config element" block.

C&M-HMI can also communicate with applications running on remote devices by means of a host (client) C&M application, using the Ethernet UDP/IP protocol. The "File/Create gateway to project" menu command can be used to generate a host application.



Run all (see Project/Options)

Launches Virtual Machines first, then loads the Application Program and starts the C&M Controller, then launches the other components of the C&M application like HMI windows, Debugger, as selected in "Project/Options/Desktop options" menu.
Project must be built (or rebuilt) before launching!

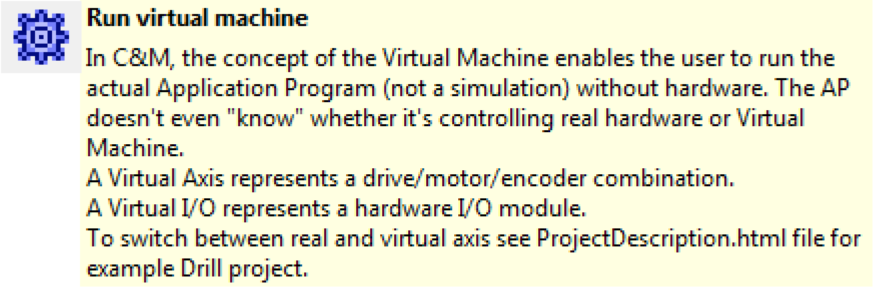


Edit C&M-HMI interface (MAIN) variables

Edit the properties of C&M-HMI interface (MAIN) variables. E.g. Min/max values for visualization, dimension (e.g. mm), timing for trend or color attribute for discrete variables.

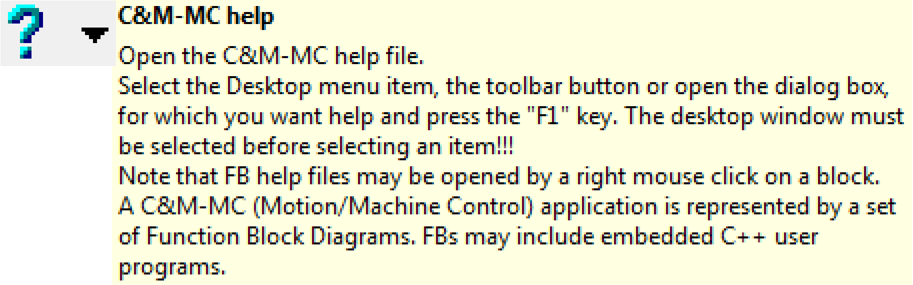
The IN/OUT ports of the C_M_MAIN.sch are linked to C&M-HMI interface variables (real time) if specified by the "**INTERFACE** (TYPE=CM_HMI) Config element" block.

Tool Bar Descriptions (Cont.)



Run virtual machine

In C&M, the concept of the Virtual Machine enables the user to run the actual Application Program (not a simulation) without hardware. The AP doesn't even "know" whether it's controlling real hardware or Virtual Machine.
A Virtual Axis represents a drive/motor/encoder combination.
A Virtual I/O represents a hardware I/O module.
To switch between real and virtual axis see ProjectDescription.html file for example Drill project.



C&M-MC help

Open the C&M-MC help file.
Select the Desktop menu item, the toolbar button or open the dialog box, for which you want help and press the "F1" key. The desktop window must be selected before selecting an item!!!
Note that FB help files may be opened by a right mouse click on a block.
A C&M-MC (Motion/Machine Control) application is represented by a set of Function Block Diagrams. FBs may include embedded C++ user programs.

Function Block Diagram Toolbar Descriptions

- Open a file
- Save a file
- Print
- CAM Processor
- Switch to board
- Sheet selection

1/1

- Use a library
- Execute a script file
- Run a User Language Program
- Zoom to fit
- Zoom in
- Zoom out
- Redraw
- Zoom select

Function Block Diagram Toolbar Descriptions (Cont.)

Undo previous operation

Redo actions that have been undone

Cancel

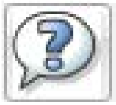
Execute command

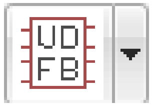
Get help for Eagle

Change the grid settings

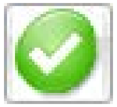
Select wire bend

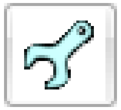
Function Block Diagram Toolbar Descriptions (Cont.)

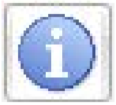

Help; list of detailed description of editor commands

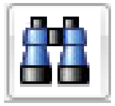


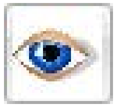
UDFB; create a User Derived Function Block from the active FBD (schematic). It will be placed in the project library


ERC; perform a syntactic checking of the connections in the FBD (schematic)

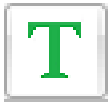

Change; change or preset properties that affect the appearance of objects


Info; display further details about an object on screen, e.g. name of source library

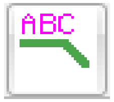

Find; locate blocks, pins or nets on the FBD (schematic)


Show; highlight objects, details are listed in the status bar. Also highlights pin and net, if connected

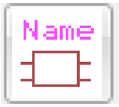
Function Block Diagram Toolbar Descriptions (Cont.)



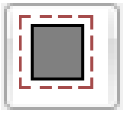
Text; add text to an FBD (schematic), the text command remains active after placing text with the mouse



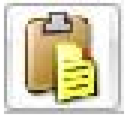
Label; display the instance name of a bus or net in any location, the second mouse click defines the location of the label



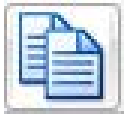
Name; display or edit the instance name of the selected object



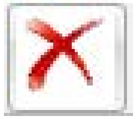
Group; define a group of objects for a successive command.
Objects are selected by click&dragging a rectangle or by drawing a polygon with the mouse.
A right click closes the polygon



Paste; using the commands GROUP, COPY, and PASTE, parts of an FBD can be copied to the same or different FBD



Copy; copy objects within the same FBD, the names of Buses and Nets are retained,
copies GROUPs to the system clipboard which can be retrieved by the PASTE command



Delete; delete the selected object, CTRL+ right mouse click deletes a previously defined GROUP

Function Block Diagram Toolbar Descriptions (Cont.)

Move; move objects, CTRL+ right mouse click moves a previously defined GROUP

Bus; bundle nets into a bus connection, Bus name is followed by a colon and then coma separated Net names

Junction; place a connection dot at the intersection of nets which are to be connected to each other

Net; draw individual connections (nets), two mouse clicks on the same point end the net.
The SHOW command highlights pin and net, if connected.

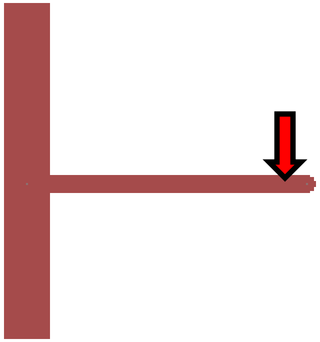
Add library elements to the FBD (schematic), a search function helps devices -in the library- to be found quickly

Function Block Diagram Mouse Controls

- Left click to place items onto the function block diagram.
- Scroll forward to zoom in.
- Scroll back to zoom out.
- Click and hold down scroll wheel to move the diagram.
- Right click to change FB/Connection orientation.

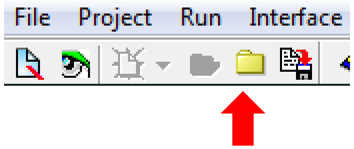
Tips

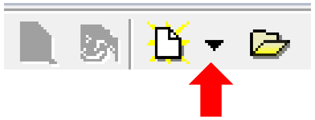
- When placing connections make sure to click directly on the Brown connector.

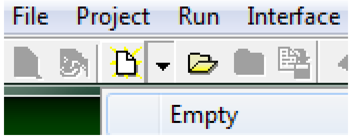


Opening a new project

- Double click C&M Icon.
- Click the ‘Close Project’ folder icon.
- Click the triangle next to the paper icon & select empty project.
- Name the project and click ‘OK’.

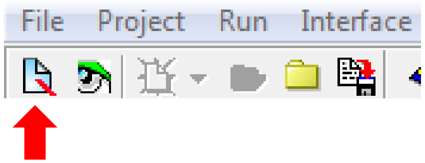


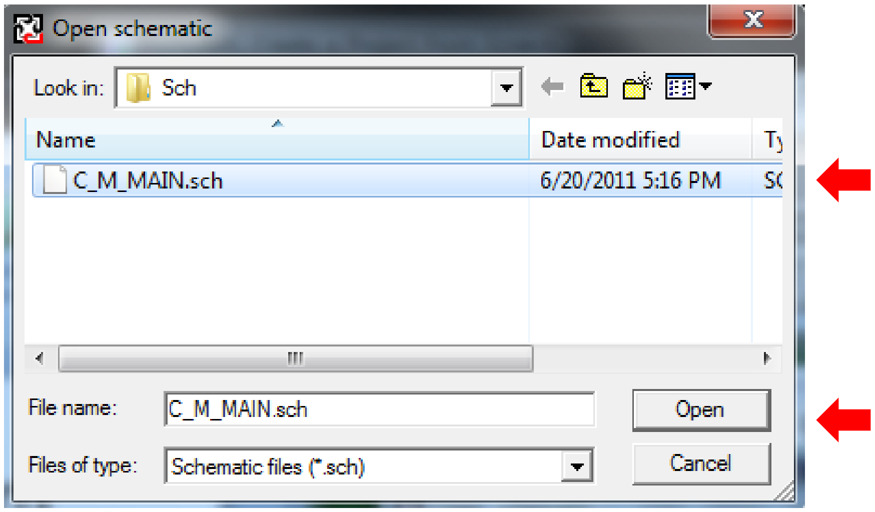




Open Function Block Diagram

- Click on the ‘Paper with red slash’ icon.
- Select ‘C_M_MAIN.sch’. & Click open.




Open schematic

Look in: Sch

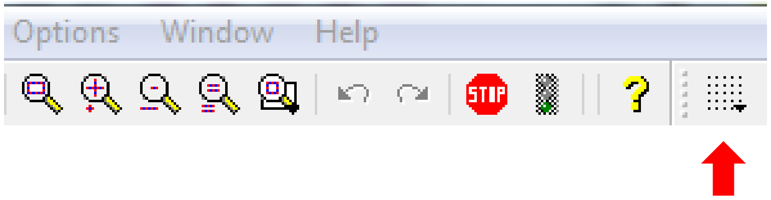
Name	Date modified	Ty
C_M_MAIN.sch	6/20/2011 5:16 PM	SC

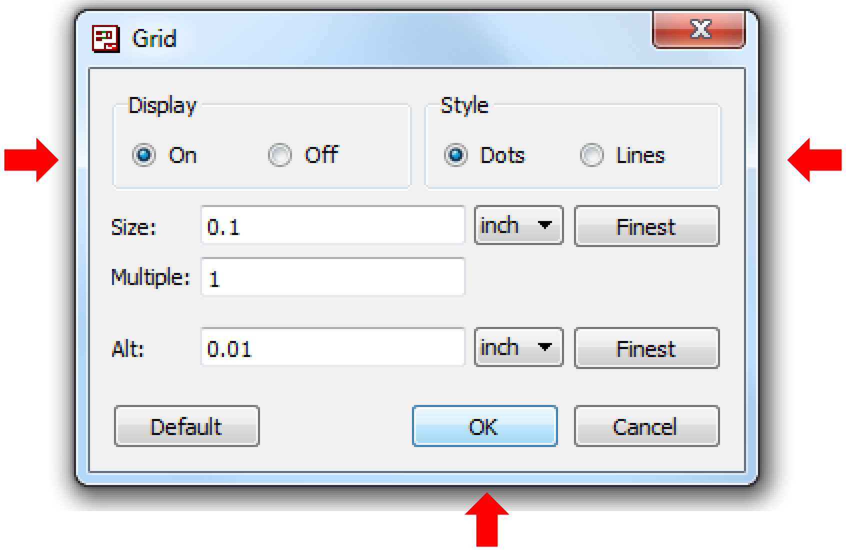
File name: C_M_MAIN.sch

Files of type: Schematic files (*.sch)

Set Grid to display

- Click on the ‘Grid’ icon in the upper right.
- Set Display to ‘On’.
- Set Style to either ‘Dots’ or ‘Lines’ & click OK.





Display: ● On ○ Off

Style: ● Dots ○ Lines

Size: 0.1 inch Finest

Multiple: 1

Alt: 0.01 inch Finest

CAN field-bus channel

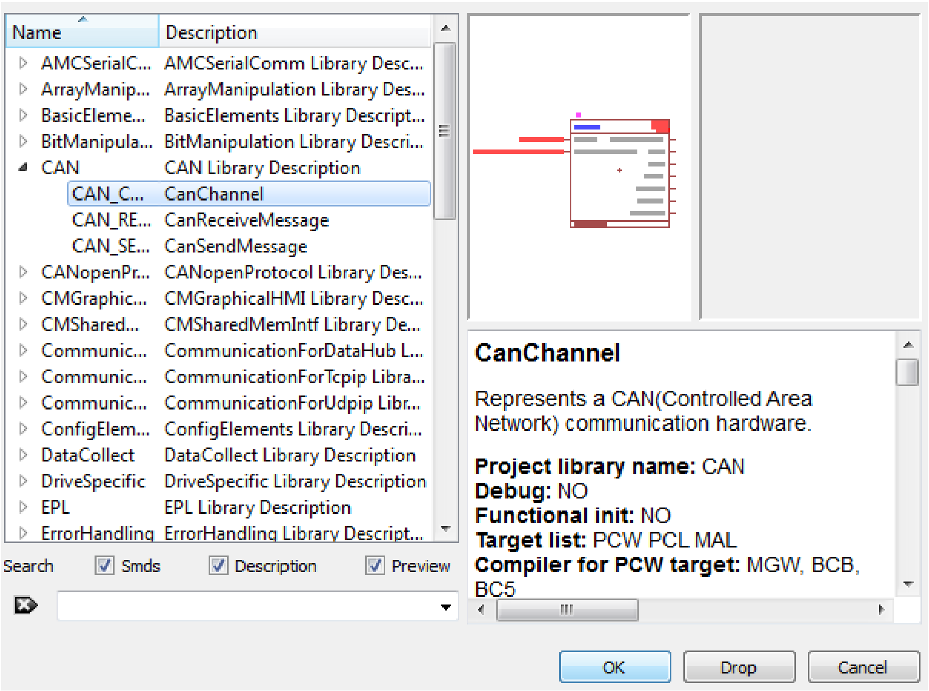
- **CanChannel** :

  Represents a CAN(Controlled Area Network) communication hardware.

  The CAN Channel FB represents the transfer line of data between different devices on the physical network. Its communication properties are configured in its property file. (e.g. card ID, boud-rate etc.) Using this FB the CandM project becomes a device on the CAN network, so it can communicate with other devices.

Place Function Block for a CAN field-bus channel

- Click the 'Function Block' Icon in the upper right.
- Select the 'CAN' pull-down on the left.
- Select 'CAN CHANNEL'
- Click 'OK'.
- Left click to place on diagram.



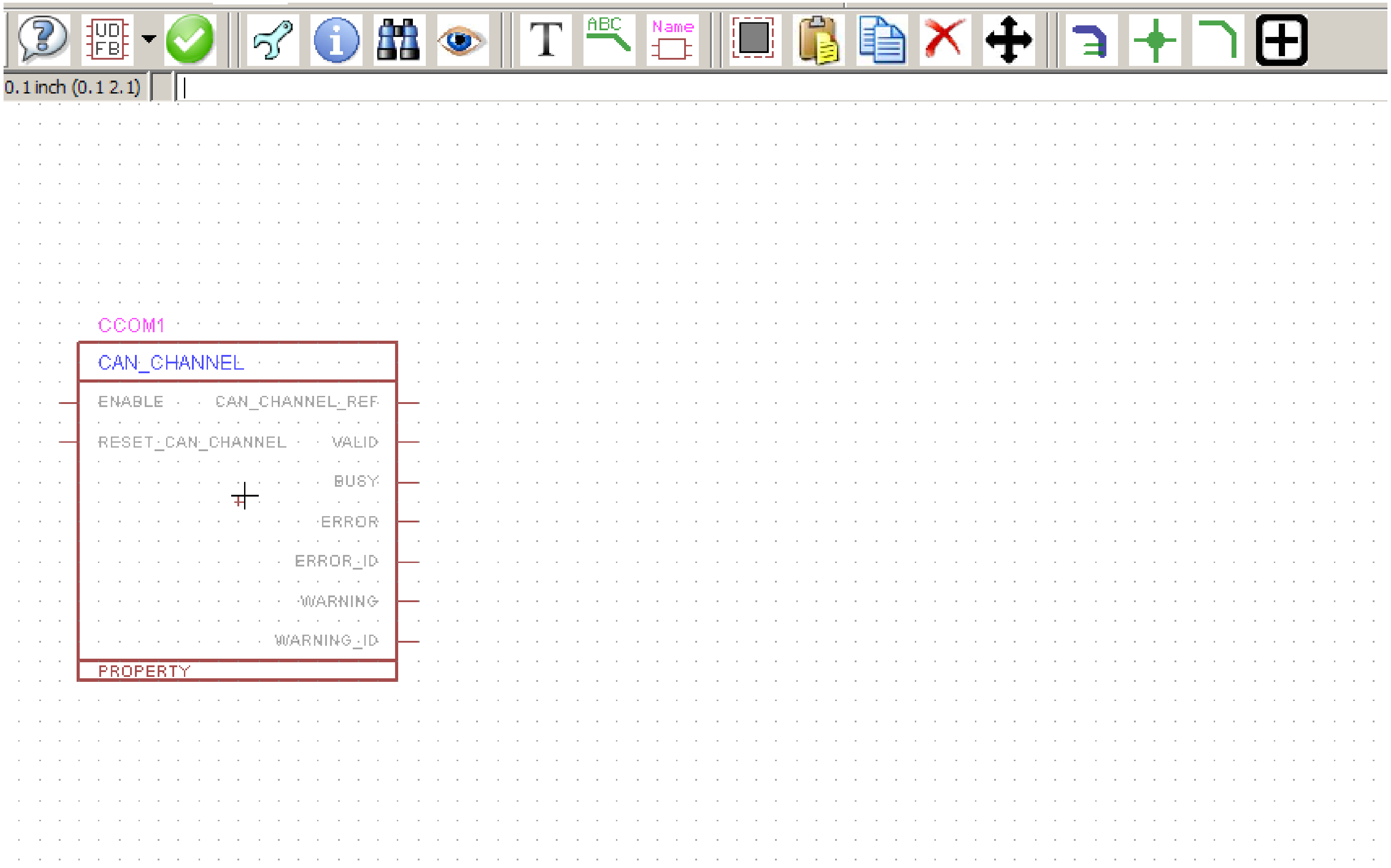
Name	Description
AMCSerialC...	AMCSerialComm Library Desc...
ArrayManip...	ArrayManipulation Library Des...
BasicEleme...	BasicElements Library Descript...
BitManipula...	BitManipulation Library Descri...
CAN	CAN Library Description
CAN_C...	CanChannel
CAN_RE...	CanReceiveMessage
CAN_SE...	CanSendMessage
CANopenPr...	CANopenProtocol Library Des...
CMGraphic...	CMGraphicalHMI Library Desc...
CMShared...	CMSharedMemIntf Library De...
Communic...	CommunicationForDataHub L...
Communic...	CommunicationForTcpip Libra...
Communic...	CommunicationForUdpip Libr...
ConfigElem...	ConfigElements Library Descri...
DataCollect	DataCollect Library Description
DriveSpecific	DriveSpecific Library Description
EPL	EPL Library Description
ErrorHandling	ErrorHandling Library Descript...

CanChannel

Represents a CAN(Controlled Area Network) communication hardware.

Project library name: CAN
Debug: NO
Functional init: NO
Target list: PCW PCL MAL
Compiler for PCW target: MGW, BCB, BC5

Diagram so far



CANopen Protocol

- **CO_Protocol** :

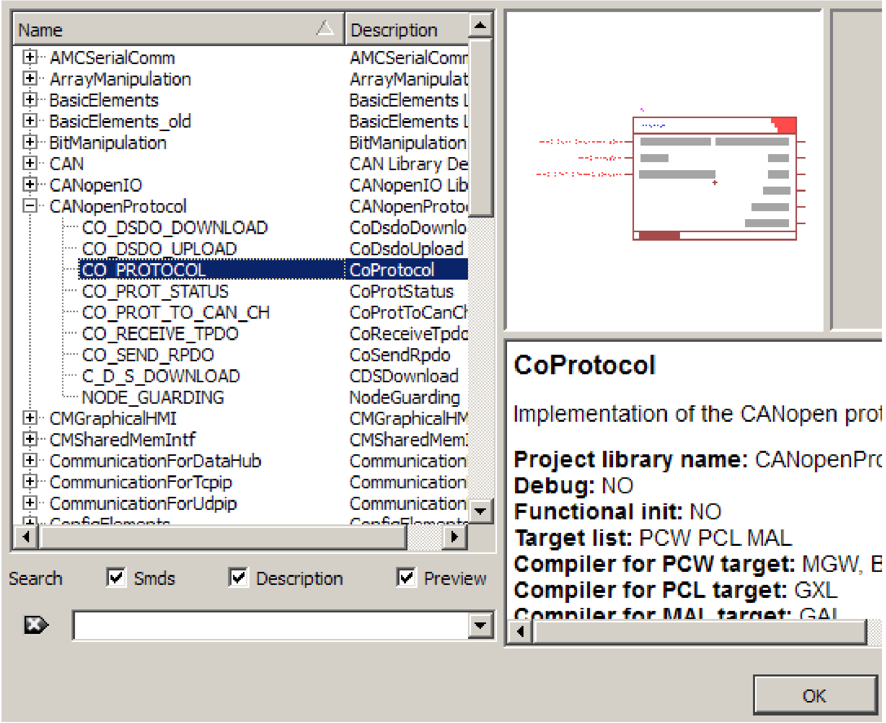
  Implementation of the CANopen protocols.

  This FB sends SYNC message to syncronize PDOs(process data object). This FB sends TimeStamp also to set the internal time of the CANopen nodes. The internal time is used for example in CANopen drives for interpolated motion.

Place Function Block for a CANopen Protocol

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘CANopenProtocol’ pull-down on the left.
- Select ‘CO PROTOCOL’
- Click ‘OK’.
- Left click to place on diagram to the right of the Channel FB.

⊞



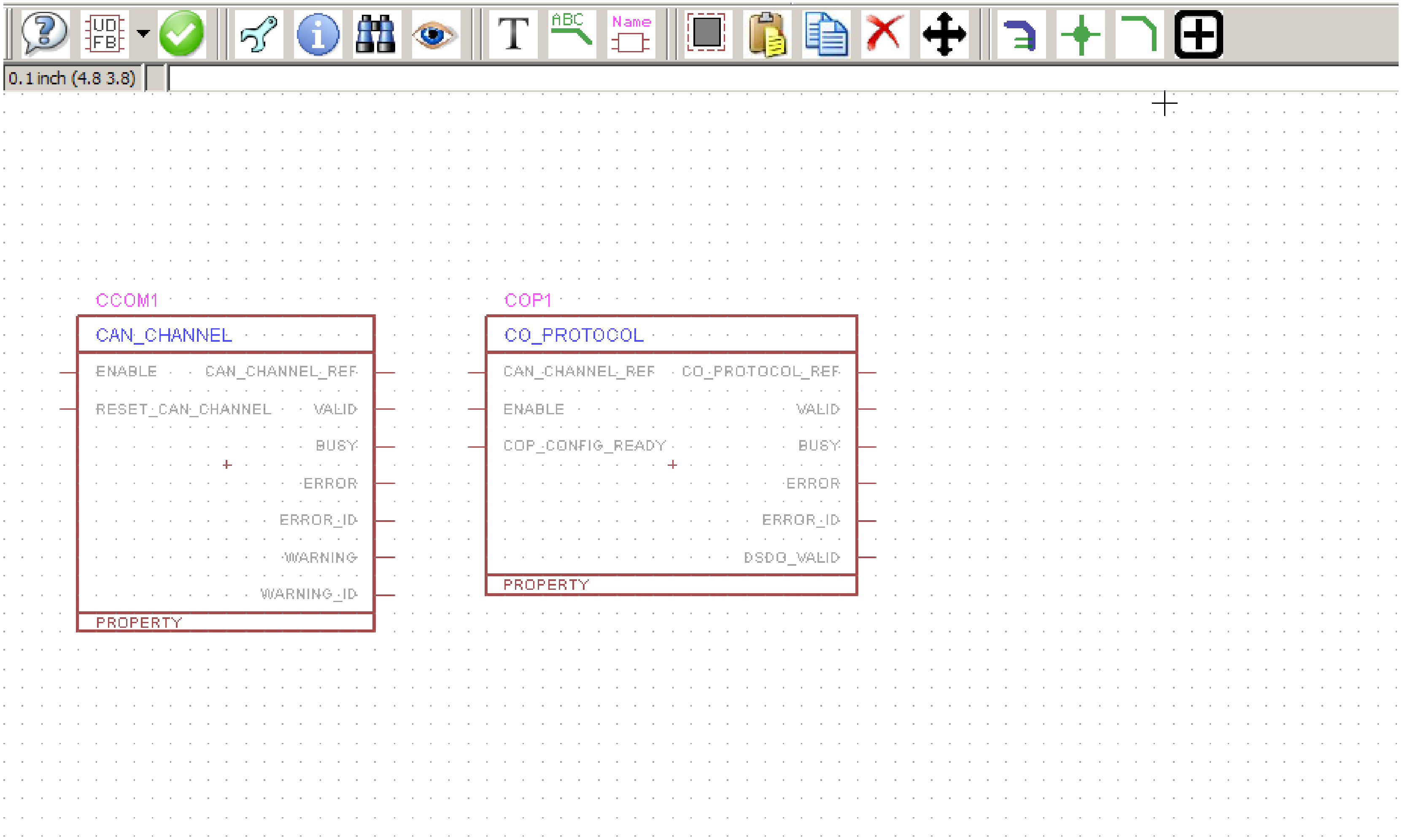
Name	Description
AMCSerialComm	AMCSerialComm
ArrayManipulation	ArrayManipulat
BasicElements	BasicElements L
BasicElements_old	BasicElements L
BitManipulation	BitManipulation
CAN	CAN Library De
CANopenIO	CANopenIO Lib
CANopenProtocol	CANopenProto
CO_DSDO_DOWNLOAD	CoDsdoDownlo
CO_DSDO_UPLOAD	CoDsdoUpload
CO_PROTOCOL	CoProtocol
CO_PROT_STATUS	CoProtStatus
CO_PROT_TO_CAN_CH	CoProtToCanCh
CO_RECEIVE_TPDO	CoReceiveTpdo
CO_SEND_RPDO	CoSendRpdo
C_D_S_DOWNLOAD	CDSDownload
NODE_GUARDING	NodeGuarding
CMGraphicalHMI	CMGraphicalHM
CMSharedMemIntf	CMSharedMem:
CommunicationForDataHub	Communication
CommunicationForTcpip	Communication
CommunicationForUdpip	Communication
ConfigElements	ConfigElements

CoProtocol

Implementation of the CANopen prot

Project library name: CANopenPrc
Debug: NO
Functional init: NO
Target list: PCW PCL MAL
Compiler for PCW target: MGW, B
Compiler for PCL target: GXL
Compiler for MAL target: GAL

Diagram so far



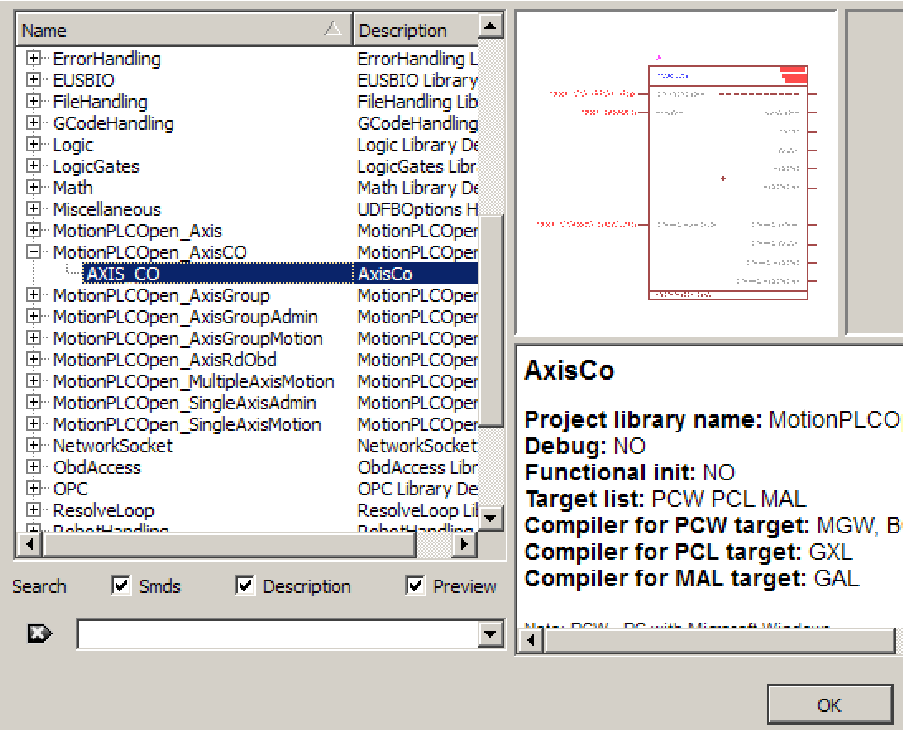
AxisCO

- AxisCO :

  Creates an AXIS_REF datastructure.
  The running application program (AP) is capable of downloading configuration data into hardware devices (e.g. servo-drives, I/O modules) via the field-bus. The configuration data is stored in a Configuration Download Sequence files (*.cds.data.txt, an AMC format).
  The CDS files must be located in .../projectname/Config directory.
  The name of the CDS file must be entered into the ConfigFile field of the property file of the hardware reference FB (e.g. AXIS_CO or IO_CO), corresponding to the hardware device.

Place Function Block for a AxisCO

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘MotionPLCOpenAxisCO ‘pull-down on the left.
- Select ‘AXIS CO’
- Click ‘OK’.
- Left click to place on diagram to the right of the Protocol FB.



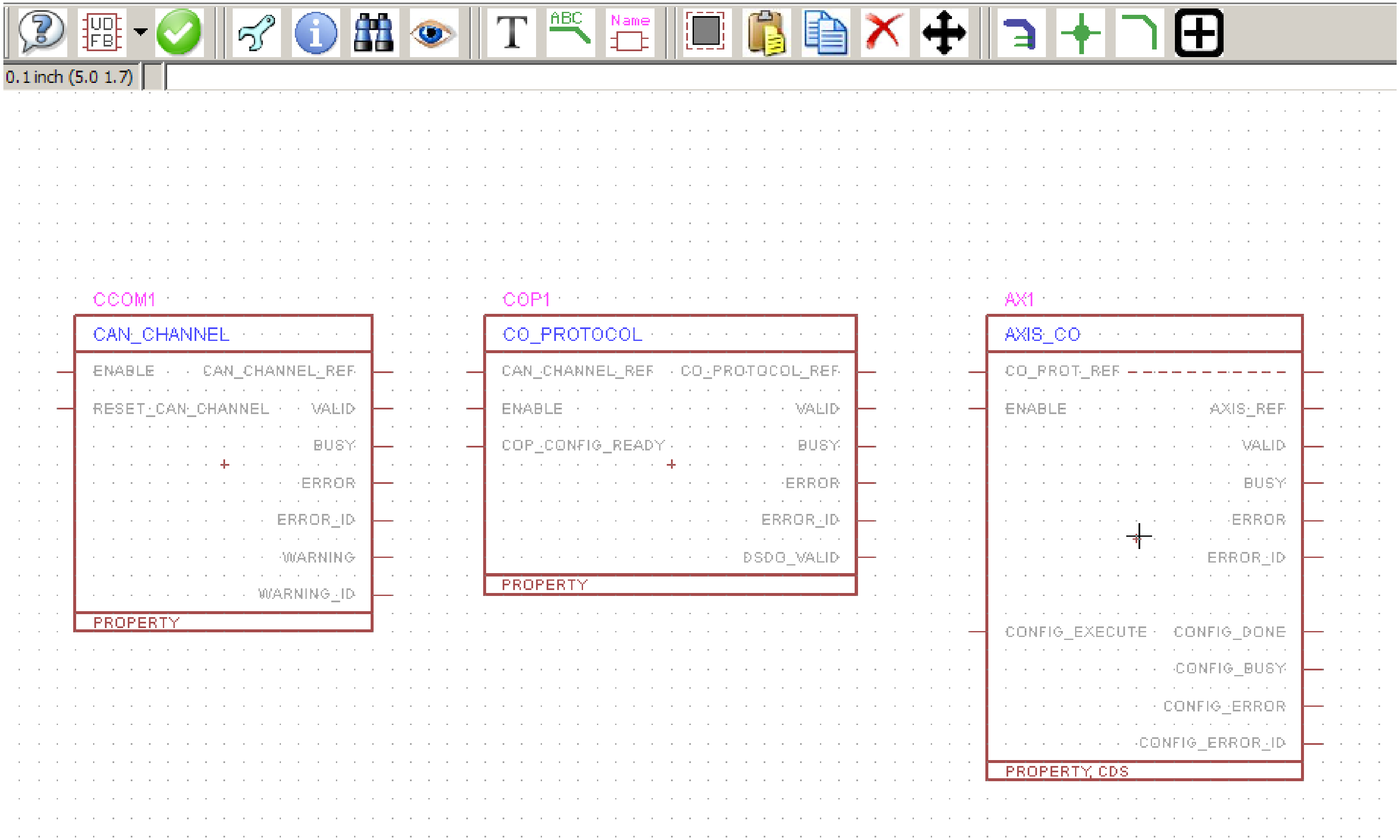
Name	Description
⊞ ErrorHandling	ErrorHandling L
⊞ EUSBIO	EUSBIO Library
⊞ FileHandling	FileHandling Lib
⊞ GCodeHandling	GCodeHandling
⊞ Logic	Logic Library De
⊞ LogicGates	LogicGates Libr
⊞ Math	Math Library De
⊞ Miscellaneous	UDFBOptions H
⊞ MotionPLCOpen_Axis	MotionPLCOper
⊟ MotionPLCOpen_AxisCO	MotionPLCOper
AXIS_CO	AxisCo
⊞ MotionPLCOpen_AxisGroup	MotionPLCOper
⊞ MotionPLCOpen_AxisGroupAdmin	MotionPLCOper
⊞ MotionPLCOpen_AxisGroupMotion	MotionPLCOper
⊞ MotionPLCOpen_AxisRdObd	MotionPLCOper
⊞ MotionPLCOpen_MultipleAxisMotion	MotionPLCOper
⊞ MotionPLCOpen_SingleAxisAdmin	MotionPLCOper
⊞ MotionPLCOpen_SingleAxisMotion	MotionPLCOper
⊞ NetworkSocket	NetworkSocket
⊞ ObdAccess	ObdAccess Libr
⊞ OPC	OPC Library De
⊞ ResolveLoop	ResolveLoop Lil
⊞ RobotHandling	RobotHandling

Search ☑ Smds ☑ Description ☑ Preview

AxisCo

Project library name: MotionPLCO
Debug: NO
Functional init: NO
Target list: PCW PCL MAL
Compiler for PCW target: MGW, B
Compiler for PCL target: GXL
Compiler for MAL target: GAL

Diagram so far



MC_Power

- **MC_POWER :**

  This Function Block controls the power stage (on or off).

  - If the MC_Power FB is called with the Enable true while being in Disabled, this either leads to Standstill if there is no error in the axis, or to ErrorStop if an Error exists.

  - It is possible to set an error variable when the Command is TRUE for a while and the Status remains false with a Timer FB and an AND Function (with inverted Status input). It indicates that there is a hardware problem with the power stage.

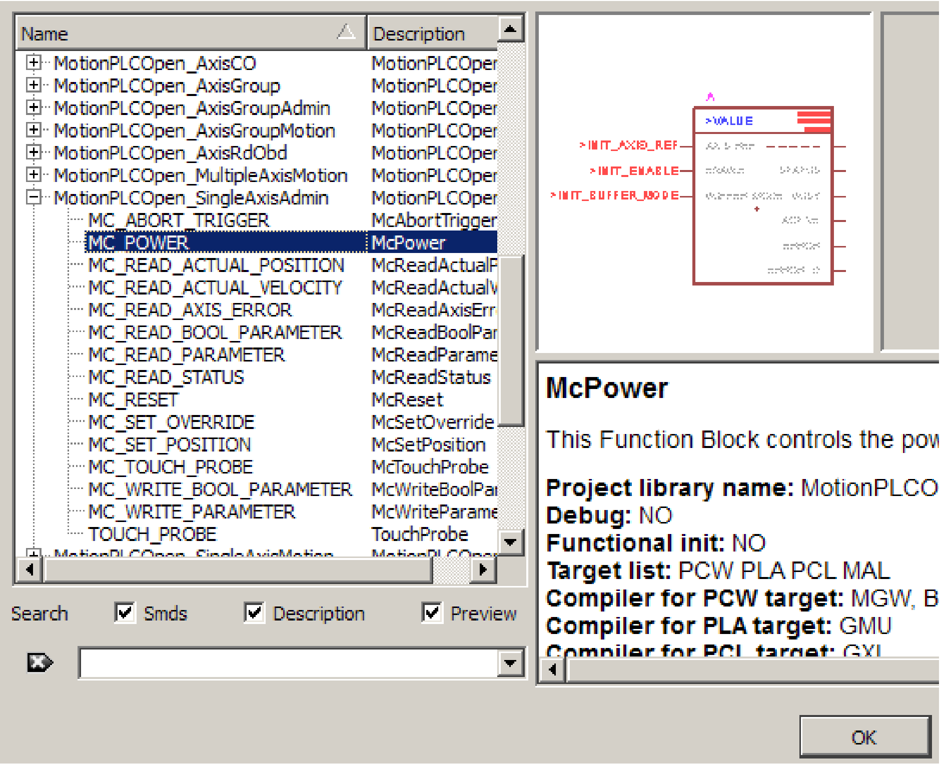
  - If power fails (also during operation) it will generate a transition to the ErrorStop state

  - When MC_Power is called with Enable false the axis goes to state Disabled for every state including ErrorStop.

  - The MC_Power and CM_Power_On/Off FBs can not be used on the same axis.

Set the function block for Motion Control Power

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘MotionPLCOpen_SingleAxisAdmin‘ pull-down on the left.
- Select ‘MC_POWER’
- Click ‘OK’.
- Left click to place on diagram to the right of the Axis_CO FB.



Name	Description
MotionPLCOpen_AxisCO	MotionPLCOper
MotionPLCOpen_AxisGroup	MotionPLCOper
MotionPLCOpen_AxisGroupAdmin	MotionPLCOper
MotionPLCOpen_AxisGroupMotion	MotionPLCOper
MotionPLCOpen_AxisRdObd	MotionPLCOper
MotionPLCOpen_MultipleAxisMotion	MotionPLCOper
MotionPLCOpen_SingleAxisAdmin	MotionPLCOper
MC_ABORT_TRIGGER	McAbortTrigger
MC_POWER	McPower
MC_READ_ACTUAL_POSITION	McReadActualP
MC_READ_ACTUAL_VELOCITY	McReadActualV
MC_READ_AXIS_ERROR	McReadAxisErr
MC_READ_BOOL_PARAMETER	McReadBoolPar
MC_READ_PARAMETER	McReadParame
MC_READ_STATUS	McReadStatus
MC_RESET	McReset
MC_SET_OVERRIDE	McSetOverride
MC_SET_POSITION	McSetPosition
MC_TOUCH_PROBE	McTouchProbe
MC_WRITE_BOOL_PARAMETER	McWriteBoolPar
MC_WRITE_PARAMETER	McWriteParame
TOUCH_PROBE	TouchProbe
MotionPLCOpen_SingleAxisMotion	MotionPLCOper

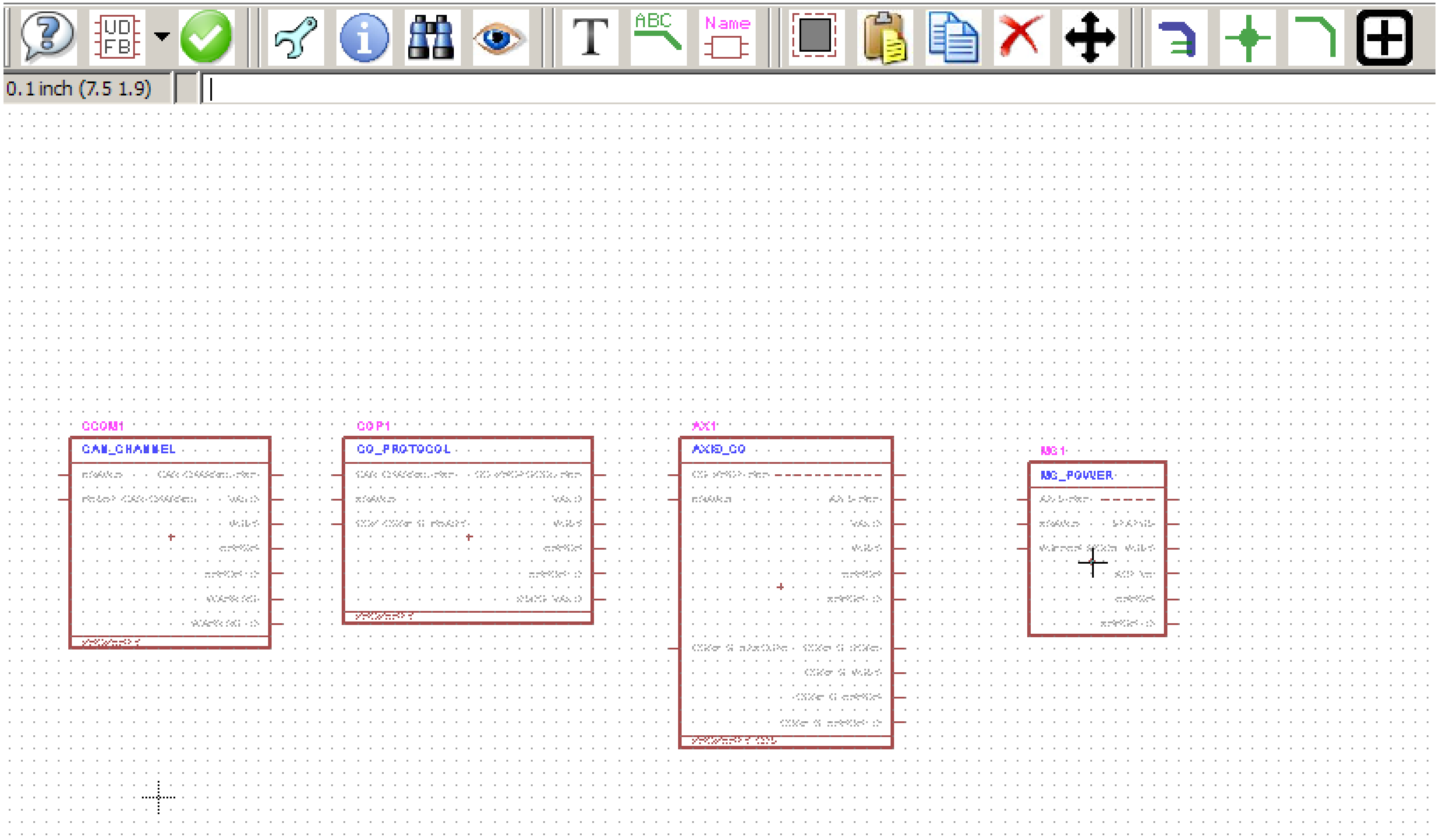
Search ☑ Smds ☑ Description ☑ Preview

McPower

This Function Block controls the pow

Project library name: MotionPLCO
Debug: NO
Functional init: NO
Target list: PCW PLA PCL MAL
Compiler for PCW target: MGW, B
Compiler for PLA target: GMU
Compiler for PCL target: GXL

Diagram so far



MC_MOVE _ABSOLUTE

- **MC_MOVE_ABSOLUTE:**

  This Function Block commands a controlled motion to a specified absolute position.

- For enumerated values to be applied at the BUFFER_MODE input, see: *C&M-MC help/PLCOpen Overview/Aborting Versus Buffered Mode*.

- For enumerated values to be applied at the DIRECTION input:

  DIR_SHORTEST = 0;
  DIR_POSITIVE = 1;
  DIR_NEGATIVE = 2;
  DIR_CURRENT = 3;

- This action completes with velocity zero if no further action are pending

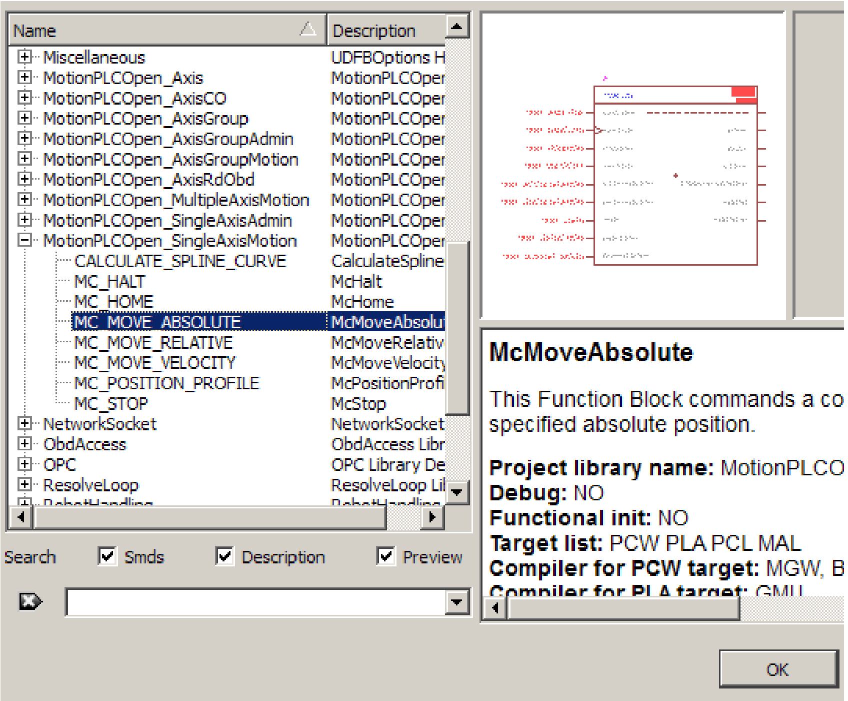
- If there is only one mathematical solution to reach the commanded position (like in linear systems), the value of the input Direction is ignored

- For modulo axis - valid absolute position values are in the range of [0, 360] (360 is excluded), or corresponding range. The application however may shift the commanded position of MC_MoveAbsolute into the corresponding modulo range. For relative positions, modulo 360 is applicable

- The Enum type 'shortest_way' is focused to a trajectory which will go through the shortest route. The decision which direction to go is based on the current position where the command is issued.

Set the function block for MC_MOVE

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘MotionPLCOpen_SingleAxisMotion‘ pull-down on the left.
- Select ‘MC_MOVE_ABSOLUTE’
- Click ‘OK’.
- Left click to place on diagram to the right of the MC_Power FB.



Name	Description
⊞ Miscellaneous	UDFBOptions H
⊞ MotionPLCOpen_Axis	MotionPLCOper
⊞ MotionPLCOpen_AxisCO	MotionPLCOper
⊞ MotionPLCOpen_AxisGroup	MotionPLCOper
⊞ MotionPLCOpen_AxisGroupAdmin	MotionPLCOper
⊞ MotionPLCOpen_AxisGroupMotion	MotionPLCOper
⊞ MotionPLCOpen_AxisRdObd	MotionPLCOper
⊞ MotionPLCOpen_MultipleAxisMotion	MotionPLCOper
⊞ MotionPLCOpen_SingleAxisAdmin	MotionPLCOper
⊟ MotionPLCOpen_SingleAxisMotion	MotionPLCOper
CALCULATE_SPLINE_CURVE	CalculateSpline
MC_HALT	McHalt
MC_HOME	McHome
MC_MOVE_ABSOLUTE	McMoveAbsolu
MC_MOVE_RELATIVE	McMoveRelativ
MC_MOVE_VELOCITY	McMoveVelocity
MC_POSITION_PROFILE	McPositionProfi
MC_STOP	McStop
⊞ NetworkSocket	NetworkSocket
⊞ ObdAccess	ObdAccess Libr
⊞ OPC	OPC Library De
⊞ ResolveLoop	ResolveLoop Lil

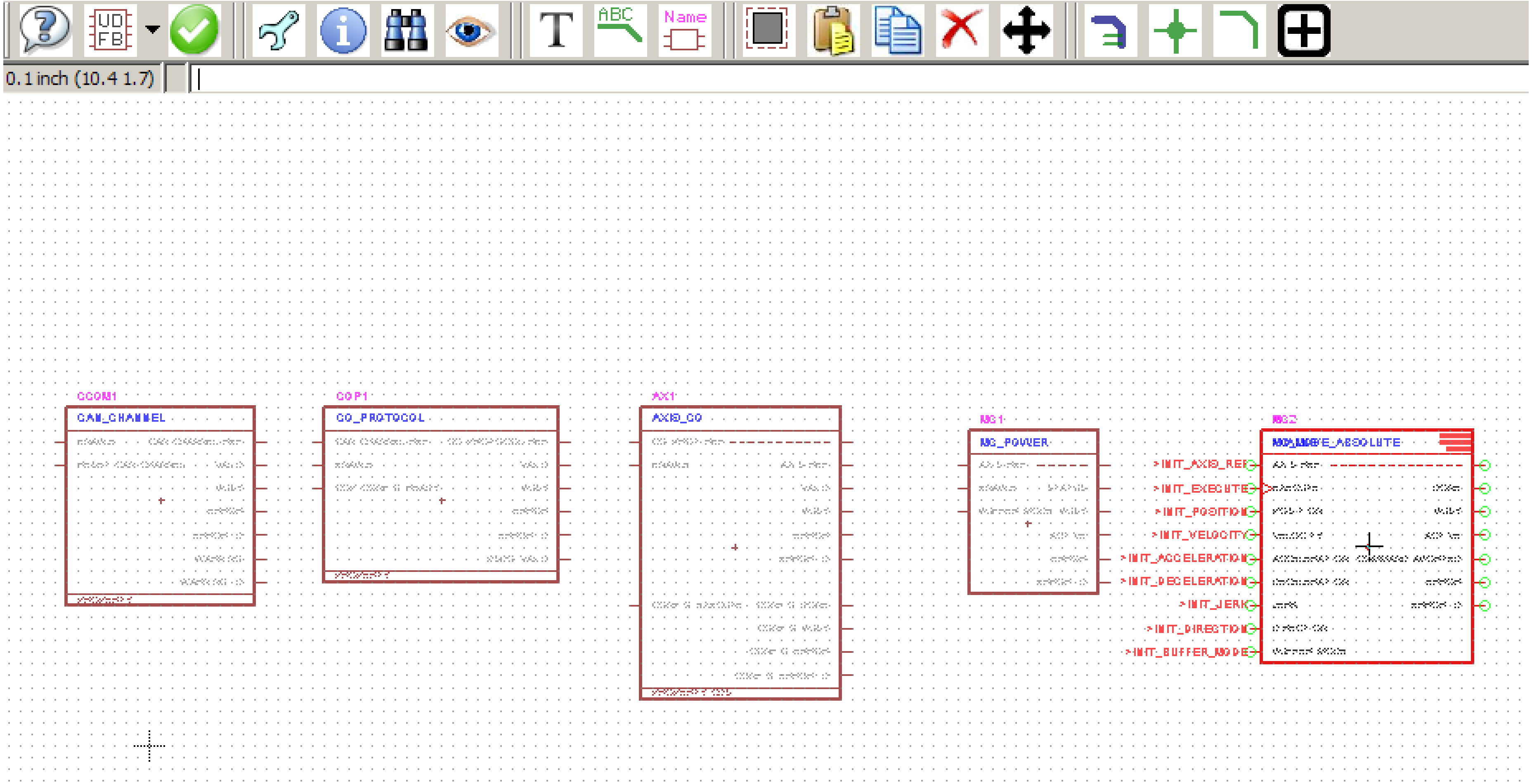
Search ☑ Smds ☑ Description ☑ Preview

McMoveAbsolute

This Function Block commands a co specified absolute position.

Project library name: MotionPLCO
Debug: NO
Functional init: NO
Target list: PCW PLA PCL MAL
Compiler for PCW target: MGW, B
Compiler for PLA target: GMU

Diagram so far



Rising falling edge detector

- **R_F_Trig:**

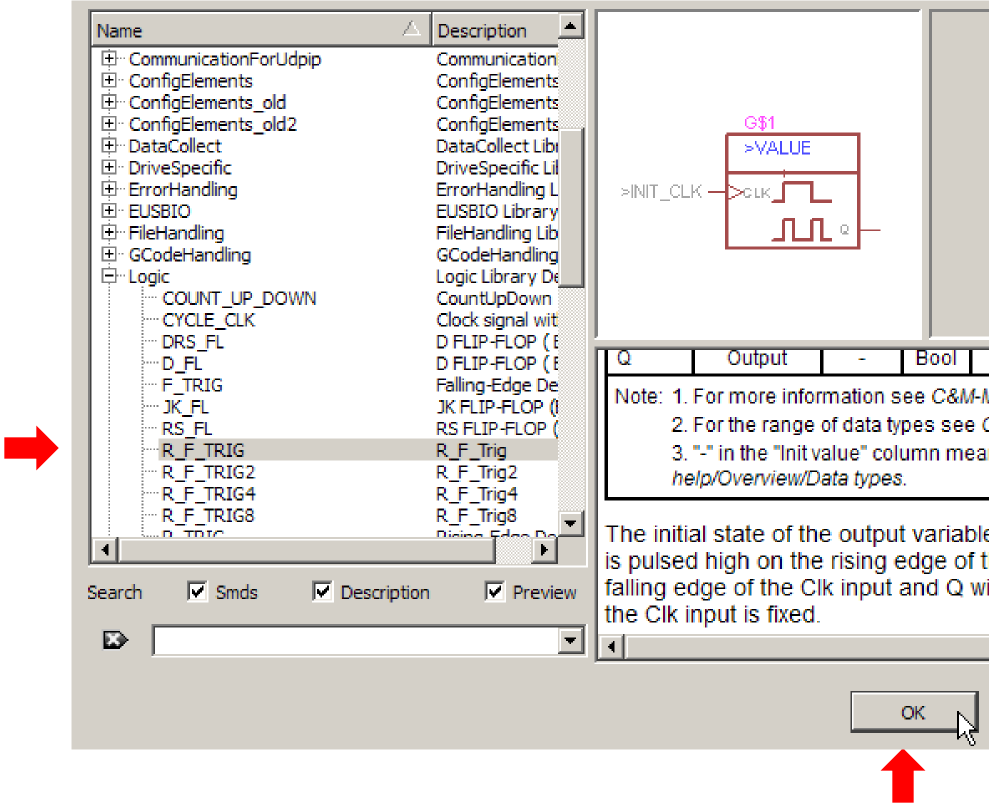
  Rising/Falling-Edge Detector (monostable; one program-cycle pulse length)

The initial state of the output variable Q is FALSE, and Q is pulsed high on the rising edge of the Clk input or on the falling edge of the Clk input and Q will be returned low if the Clk input is fixed.

Set the function block for R_F_Trig

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘Logic‘ pull-down on the left.
- Select ‘R_F_Trig’
- Click ‘OK’.
- Left click to place on diagram above the MC_Power FB.

⊞



Name	Description
CommunicationForUdpip	Communication
ConfigElements	ConfigElements
ConfigElements_old	ConfigElements
ConfigElements_old2	ConfigElements
DataCollect	DataCollect Libr
DriveSpecific	DriveSpecific Li
ErrorHandling	ErrorHandling L
EUSBIO	EUSBIO Library
FileHandling	FileHandling Lib
GCodeHandling	GCodeHandling
Logic	Logic Library De
COUNT_UP_DOWN	CountUpDown
CYCLE_CLK	Clock signal wit
DRS_FL	D FLIP-FLOP (
D_FL	D FLIP-FLOP (
F_TRIG	Falling-Edge De
JK_FL	JK FLIP-FLOP (
RS_FL	RS FLIP-FLOP (
R_F_TRIG	R_F_Trig
R_F_TRIG2	R_F_Trig2
R_F_TRIG4	R_F_Trig4
R_F_TRIG8	R_F_Trig8

Search ☑ Smds ☑ Description ☑ Preview

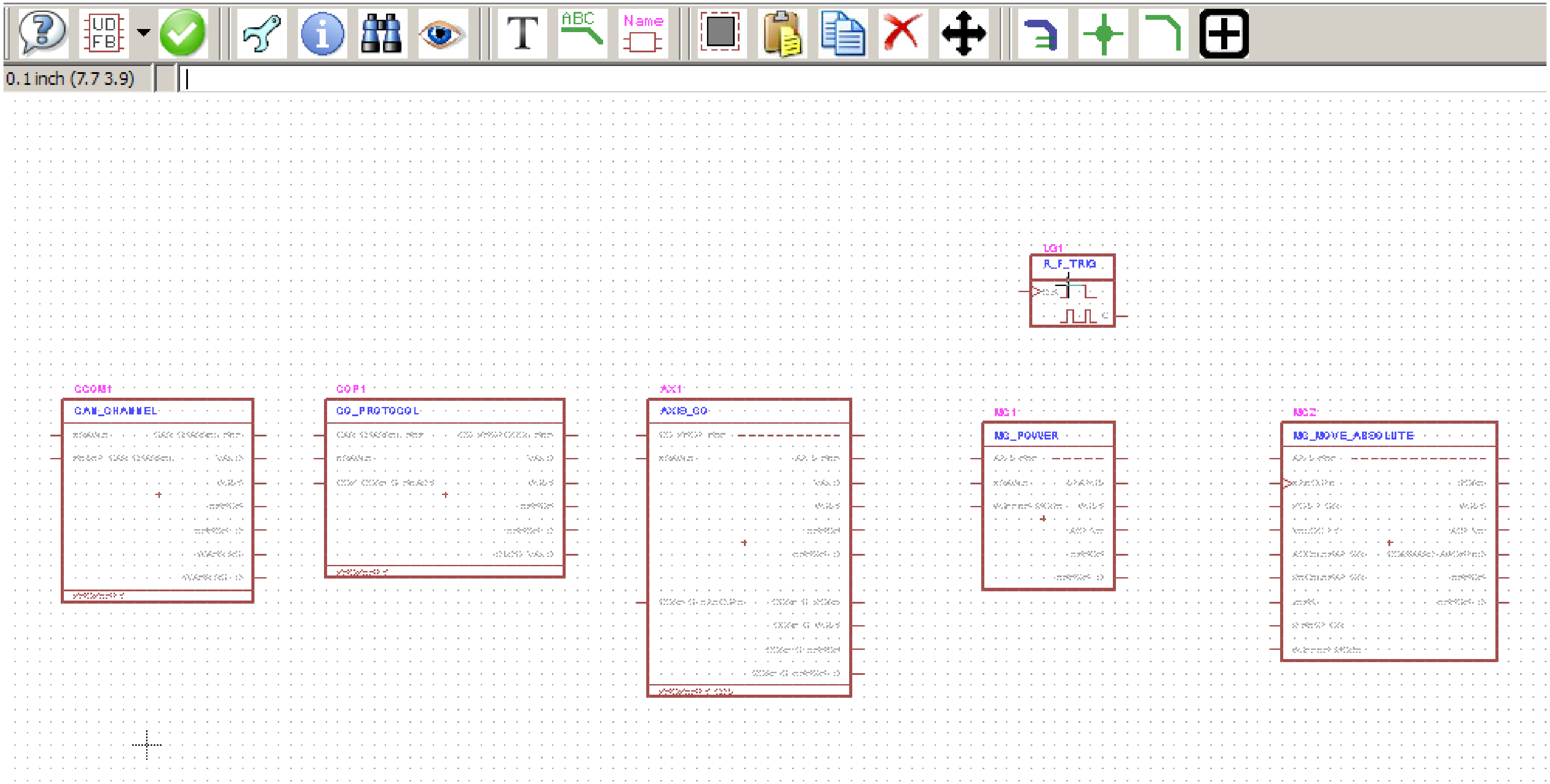
Q	Output	-	Bool

Note: 1. For more information see C&M-
2. For the range of data types see
3. "-" in the "Init value" column mea
help/Overview/Data types.

The initial state of the output variable is pulsed high on the rising edge of t falling edge of the Clk input and Q wi the Clk input is fixed.

OK

Diagram so far



2 Input AND Gate

- **AND2:**

  A **logic gate** performs a logical operation on one or more logic inputs and produces a single logic output. Because the output is also a logic-level value, an output of one logic gate can connect to the input of one or more other logic gates.

  An **AND gate** generates a true at its output if all inputs are true.

  A **truth table** is a table that describes the behavior of a logic Function Block (FB). It lists the value of the output for every possible combination of the inputs.

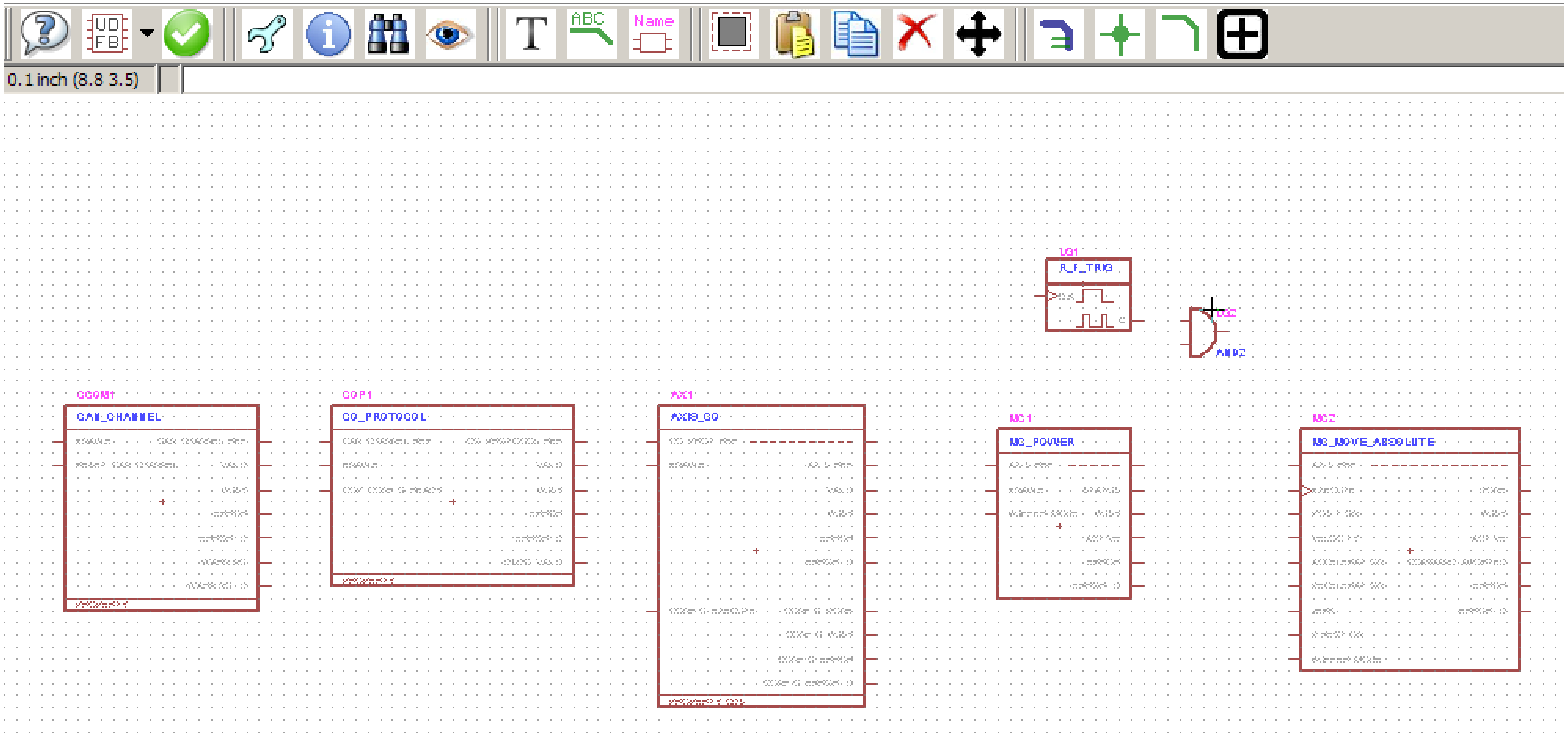
Set the function block for INPUT

- Click the ‘Function Block’ Icon in the upper right.

⊞

- Select the ‘Logic Gates‘ ➡ pull-down on the left.
- Select ‘AND2’
- Click ‘OK’.
- Left click to place on diagram to the right and below the R_F_Trig FB.

Diagram so far



Input

- **INPUT:**

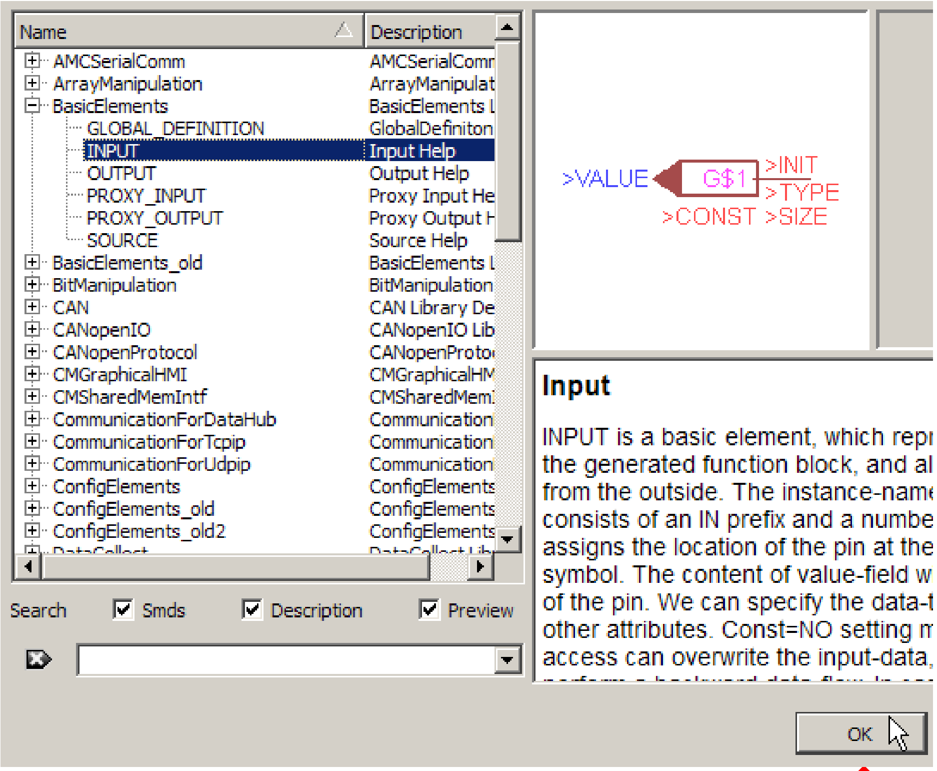
  Rising/Falling-Edge Detector (monostable; one program-cycle pulse length)

  INPUT is a basic element, which represents an input pin of the generated function block, and allows receiving data from the outside. The instance-name of the box always consists of an IN prefix and a number. This number assigns the location of the pin at the left-side of the box-symbol. The content of value-field will appear as the name of the pin. We can specify the data-type, init-value and other attributes. Const=NO setting means, that an internal access can overwrite the input-data, therefore we can perform a backward data-flow. In case of Const=YES we cannot overwrite the input-data inside the FB, so only forward data-flow is allowed.

Set the function block for INPUT

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘BasicElements‘ pull-down on the left.
- Select ‘INPUT’
- Click ‘OK’.
- Left click to place on diagram.
- Connect to ‘Position’ on the MC_Move FB.
- Connect to ‘CLK’ on R_F_Trig FB.

⊞

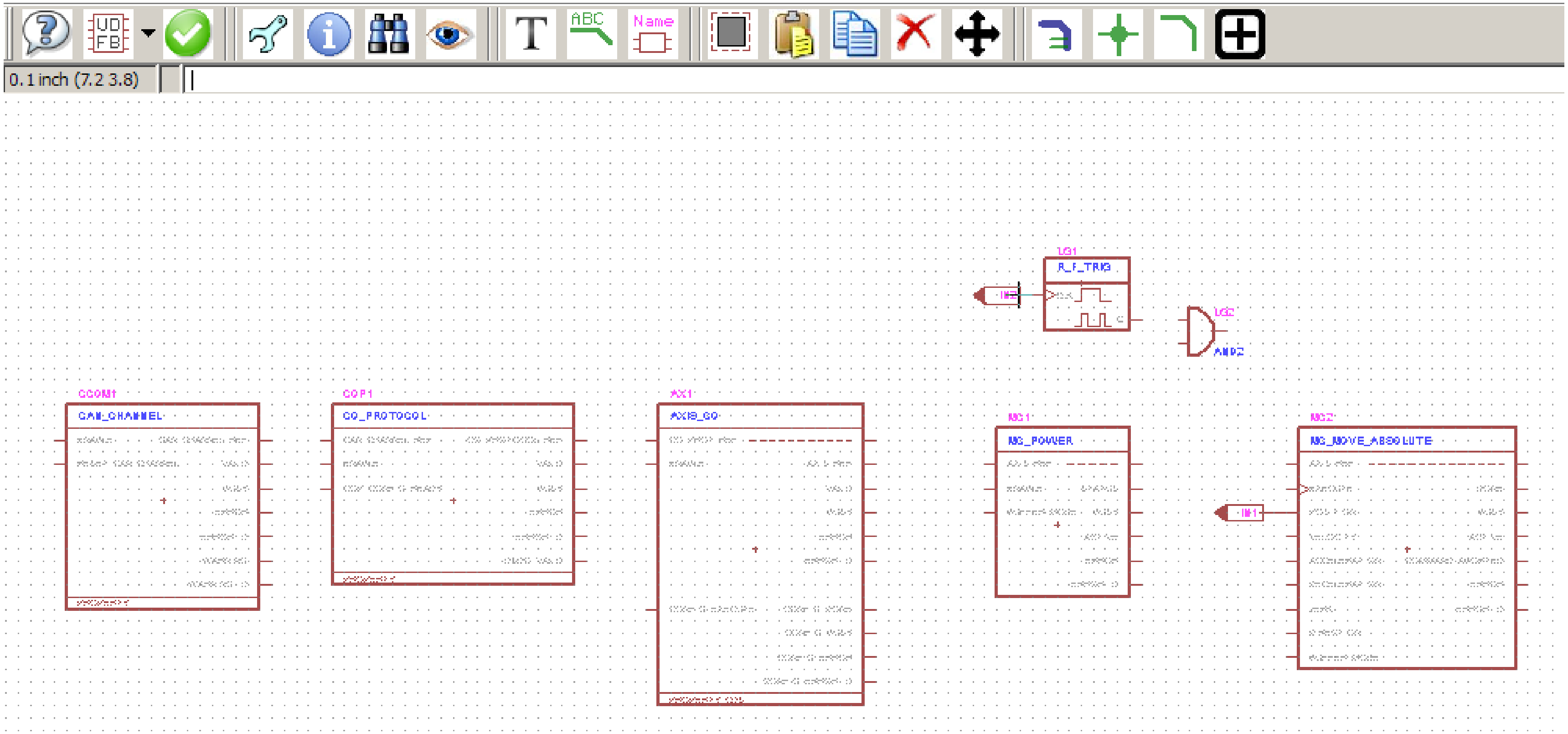


Name	Description
⊞ AMCSerialComm	AMCSerialComm
⊞ ArrayManipulation	ArrayManipulat
⊟ BasicElements	BasicElements L
GLOBAL_DEFINITION	GlobalDefiniton
INPUT	Input Help
OUTPUT	Output Help
PROXY_INPUT	Proxy Input He
PROXY_OUTPUT	Proxy Output H
SOURCE	Source Help
⊞ BasicElements_old	BasicElements L
⊞ BitManipulation	BitManipulation
⊞ CAN	CAN Library De
⊞ CANopenIO	CANopenIO Lib
⊞ CANopenProtocol	CANopenProto
⊞ CMGraphicalHMI	CMGraphicalHM
⊞ CMSharedMemIntf	CMSharedMem
⊞ CommunicationForDataHub	Communication
⊞ CommunicationForTcpip	Communication
⊞ CommunicationForUdpip	Communication
⊞ ConfigElements	ConfigElements
⊞ ConfigElements_old	ConfigElements
⊞ ConfigElements_old2	ConfigElements

Input

INPUT is a basic element, which repr
the generated function block, and al
from the outside. The instance-name
consists of an IN prefix and a numbe
assigns the location of the pin at the
symbol. The content of value-field w
of the pin. We can specify the data-t
other attributes. Const=NO setting m
access can overwrite the input-data,

Diagram so far



Resolve Loop

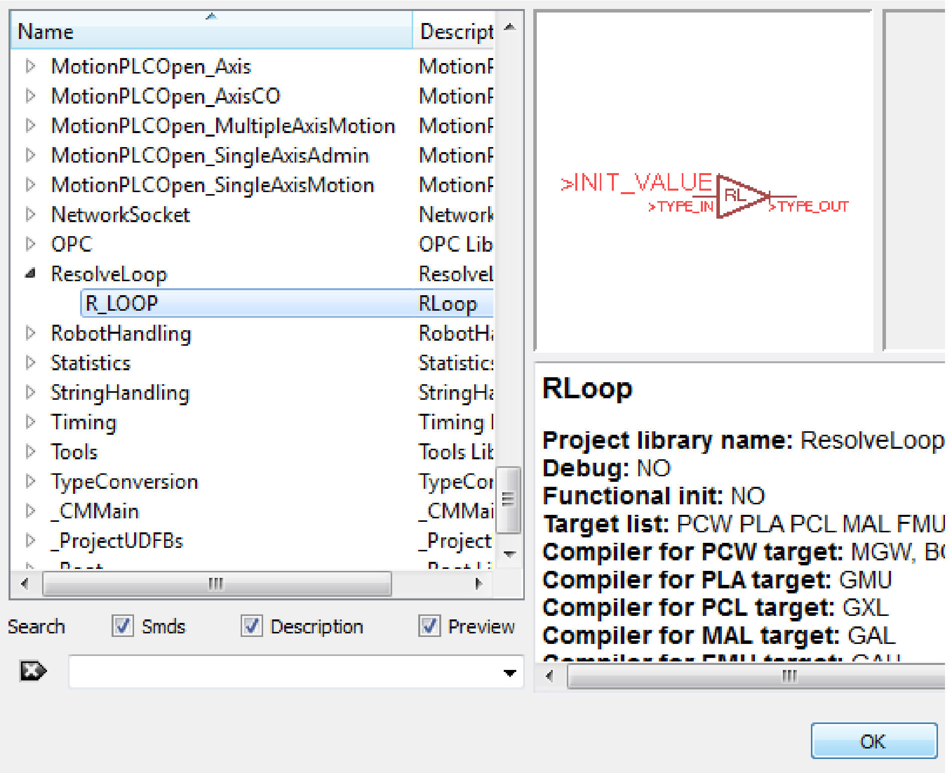
- **INPUT:**

  Care should be taken to avoid DF (Direct Feed through) loops.
  FBs whose current outputs depends on their current inputs are called Direct Feed-through FBs.
  A DF loop is e.g. when an inverter's output is connected to it's input. In this case the input/output can't be evaluated.
  Placing an R_LOOP (Resolve Loop) FB in a DF loop solves the problem by instructing the NetCompiler to store outputs first and then evaluate the loop.
  If the compiler detects a DF loop, it will stop and send an error message with a list of FBs in the loop.

Set the function block for RLoop

- Click the ‘Function Block’ Icon in the upper right.
- Select the ‘Resolve Loop‘ pull-down on the left.
- Select ‘RLoop’
- Click ‘OK’.
- Right click to flip the FB.
- Left click to place on diagram below AXIS_CO FB.

⊞



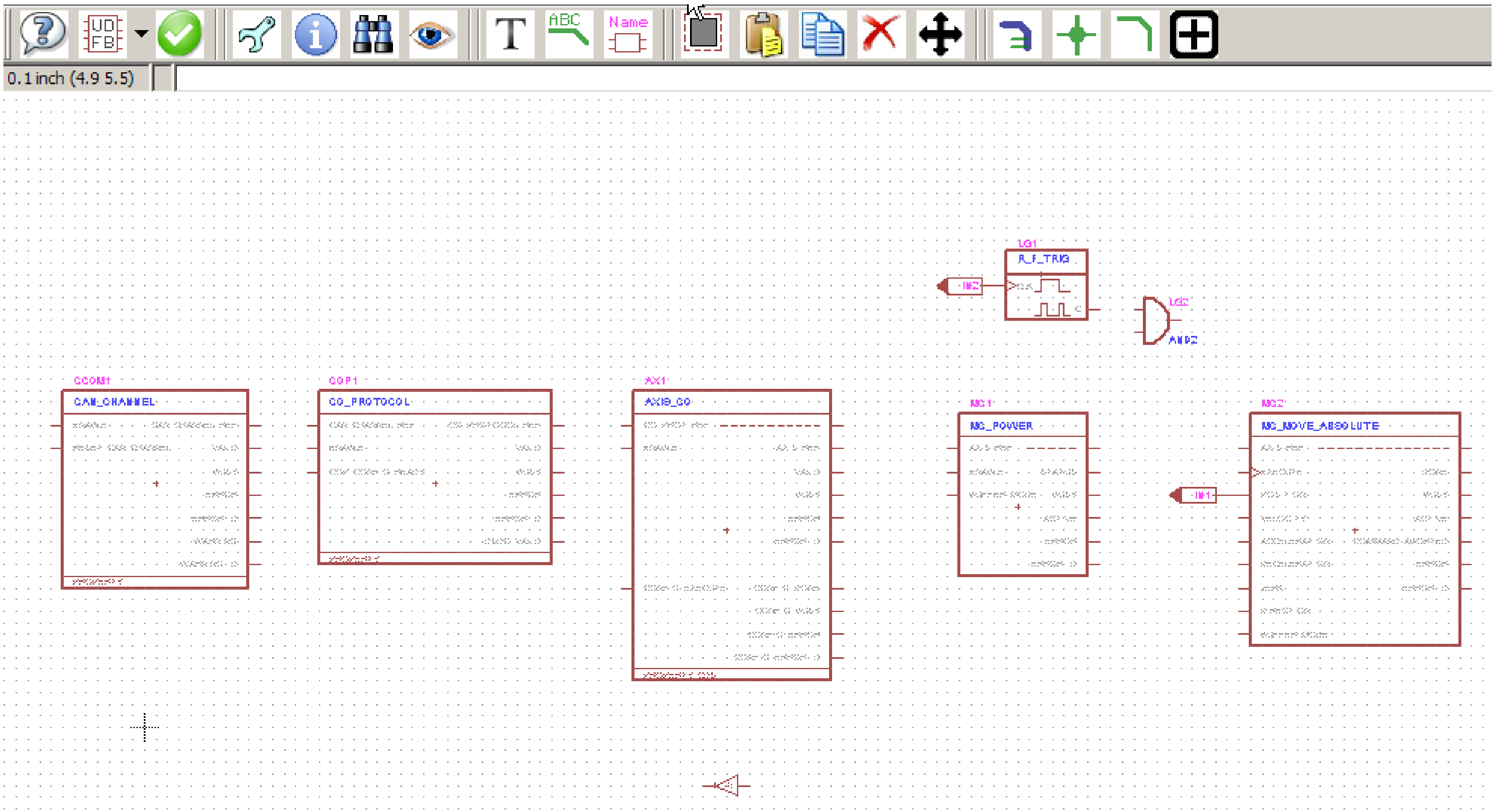
Name	Descript
MotionPLCOpen_Axis	MotionP
MotionPLCOpen_AxisCO	MotionP
MotionPLCOpen_MultipleAxisMotion	MotionP
MotionPLCOpen_SingleAxisAdmin	MotionP
MotionPLCOpen_SingleAxisMotion	MotionP
NetworkSocket	Network
OPC	OPC Lib
ResolveLoop	Resolvel
R_LOOP	RLoop
RobotHandling	RobotHa
Statistics	Statistic
StringHandling	StringHa
Timing	Timing l
Tools	Tools Lib
TypeConversion	TypeCor
_CMMain	_CMMai
_ProjectUDFBs	_Project

Search ☑ Smds ☑ Description ☑ Preview

RLoop

Project library name: ResolveLoop
Debug: NO
Functional init: NO
Target list: PCW PLA PCL MAL FMU
Compiler for PCW target: MGW, B
Compiler for PLA target: GMU
Compiler for PCL target: GXL
Compiler for MAL target: GAL

Diagram so far



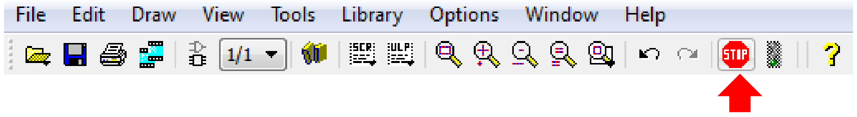
Extend Logic Connections

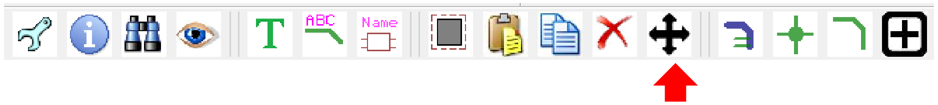
- Click the ‘Stop’ icon.

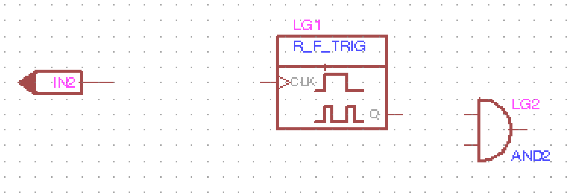
- Click the ‘Move’ icon.

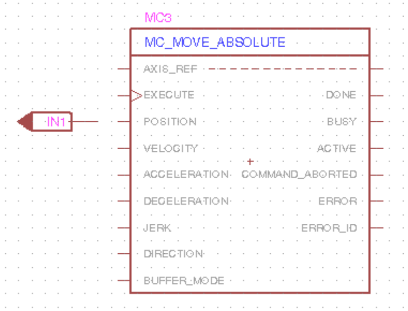
- Left click on the Logic FB connected to the R_F_Trig FB and drag to the left.

- Left click on the Logic FB connected to the MC_MOVE FB and drag to the left.



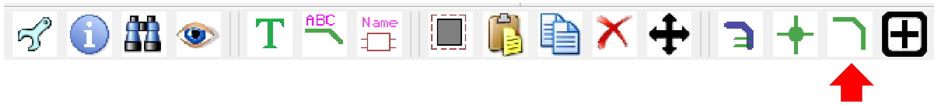




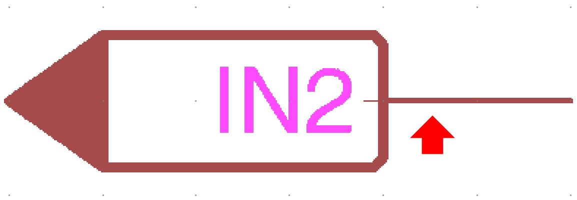


Drawing Connections

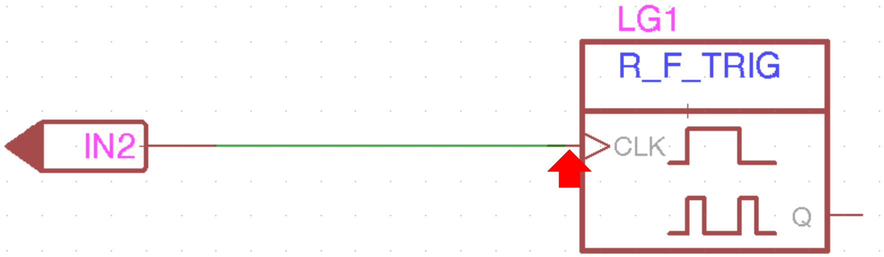
- Click the ‘Connections’ icon.



- Click towards the base of the FB connector.

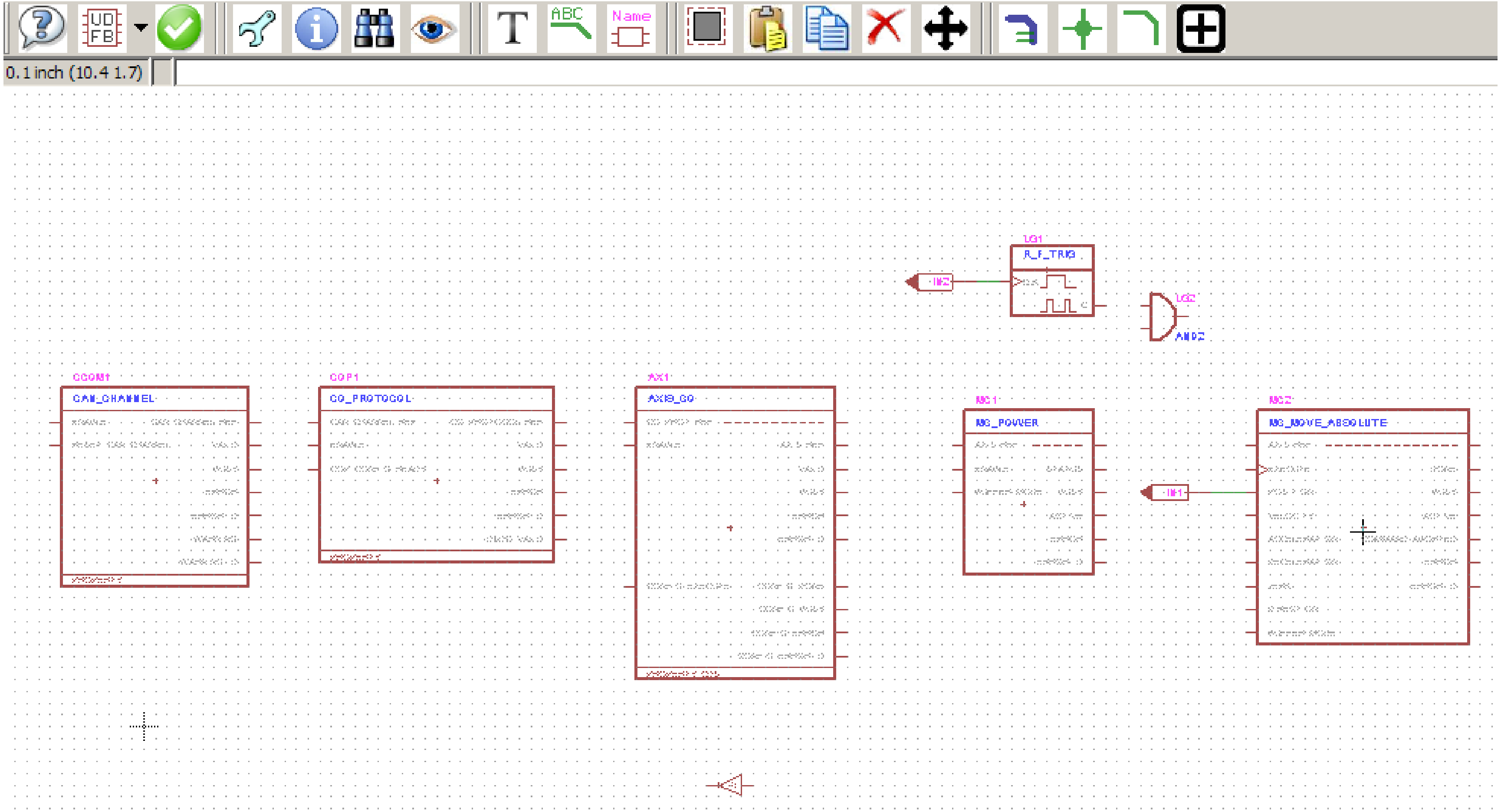


- Click towards the base of the other FB connector to complete connection.



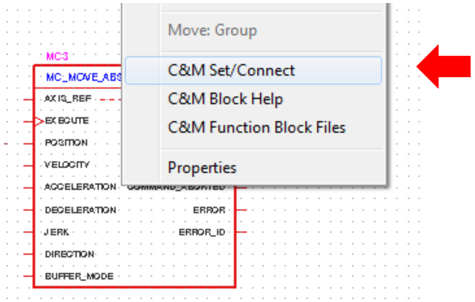
- IMPORTANT! If you do not click towards the base of the connector the connection will not work.

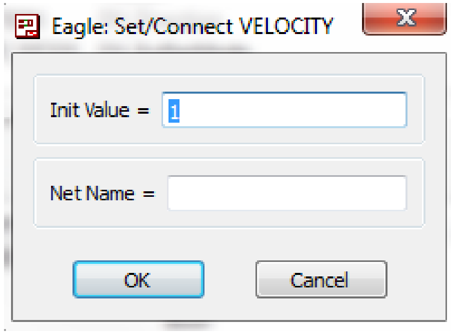
Diagram so far



Set Values for MC_Move_Absolute FB

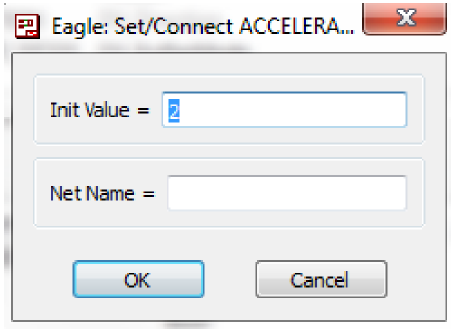
- Right click the 'MC_Move_Absolute' FB.
- Select 'C&M Set/Connect'.
- Double-click 'In4 Velocity' set Int Value=1, click 'OK'.
- Double-click 'In5 Acceleration' set Int Value=2, click 'OK'.
- Double-click 'In6 Deceleration' set Int Value=2, click 'OK'.
- Double-click 'In7 Jerk' set Int Value=10, click 'OK'.
- Click 'OK'.




Eagle: Set/Connect VELOCITY

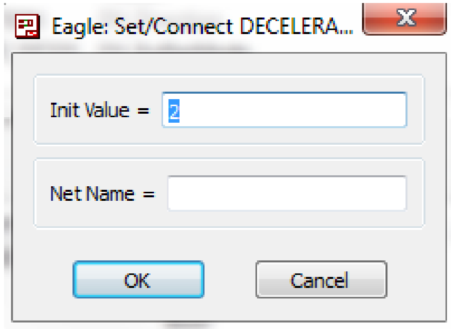
Init Value = 1

Net Name =


Eagle: Set/Connect ACCELERA...

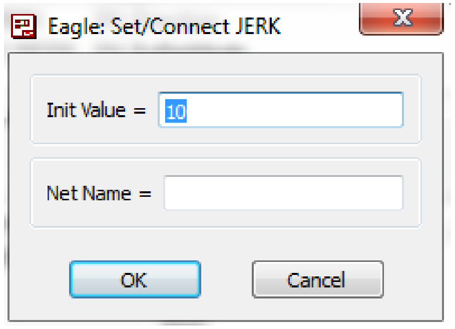
Init Value = 2

Net Name =


Eagle: Set/Connect DECELERA...

Init Value = 2

Net Name =

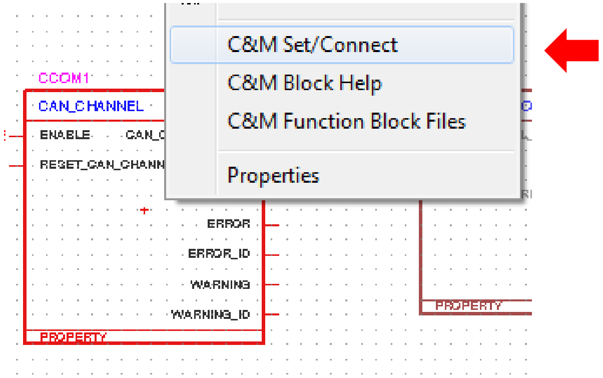

Eagle: Set/Connect JERK

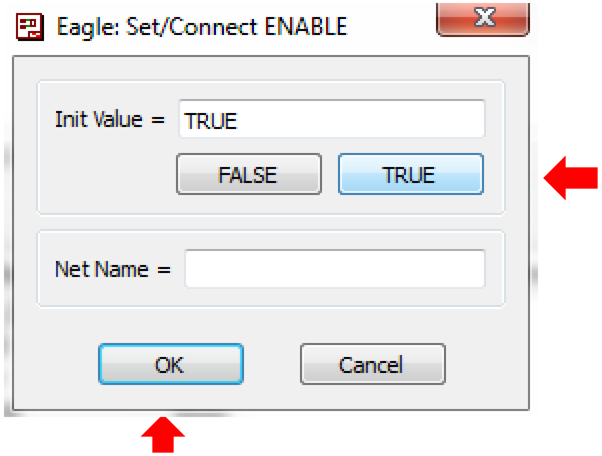
Init Value = 10

Net Name =

Set Values for CAN_CHANNEL FB

- Right click the 'CAN_CHANNEL' FB.
- Select 'C&M Set/Connect'.
- Double-click 'In1 ENABLE' set Int Value=TRUE, click 'OK'.
- Click 'OK'.



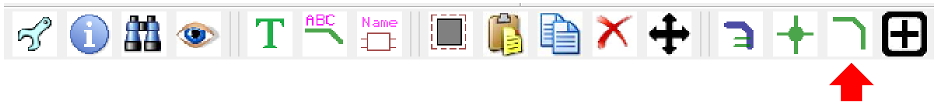

Eagle: Set/Connect ENABLE

Init Value = TRUE

Net Name =

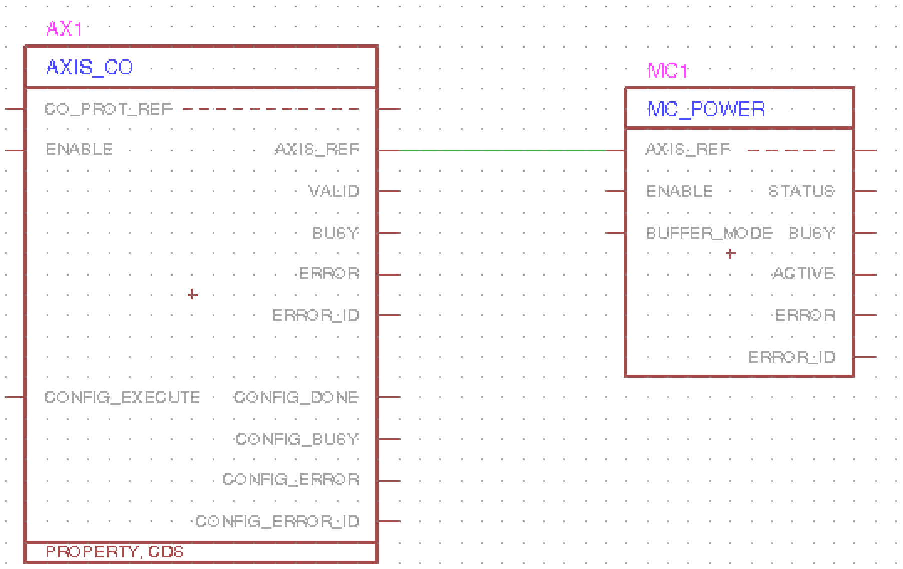
Create Signal Flow and Data Flow Connections (AXIS_REF -> MC_POWER)

- Click the ‘Connections’ icon.



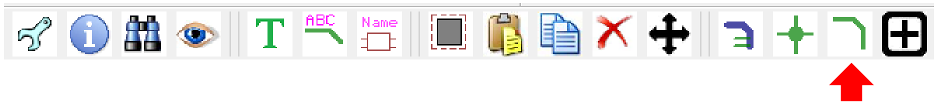
- Click on the ‘AXIS_REF’ connector on the ‘AXIS_CO’ FB.

- Connect to the ‘AXIS_REF’ connector on the ‘MC_POWER’ FB.



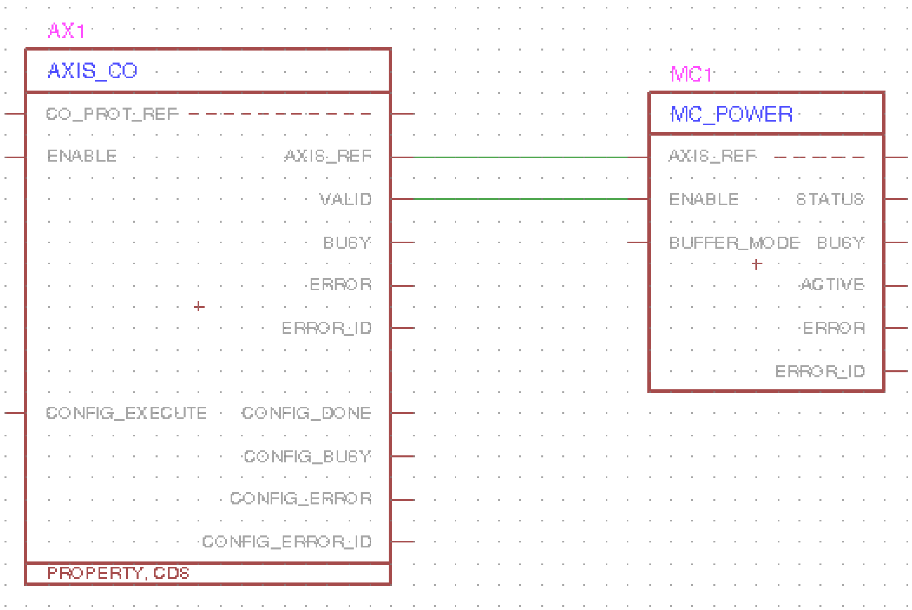
Create Signal Flow and Data Flow Connections (AXIS_REF -> MC_POWER)

- Click the ‘Connections’ icon.



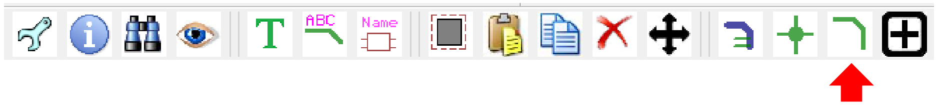
- Click on the ‘VALID’ connector on the ‘AXIS_CO’ FB.

- Connect to the ‘ENABEL’ connector on the ‘MC_POWER’ FB.

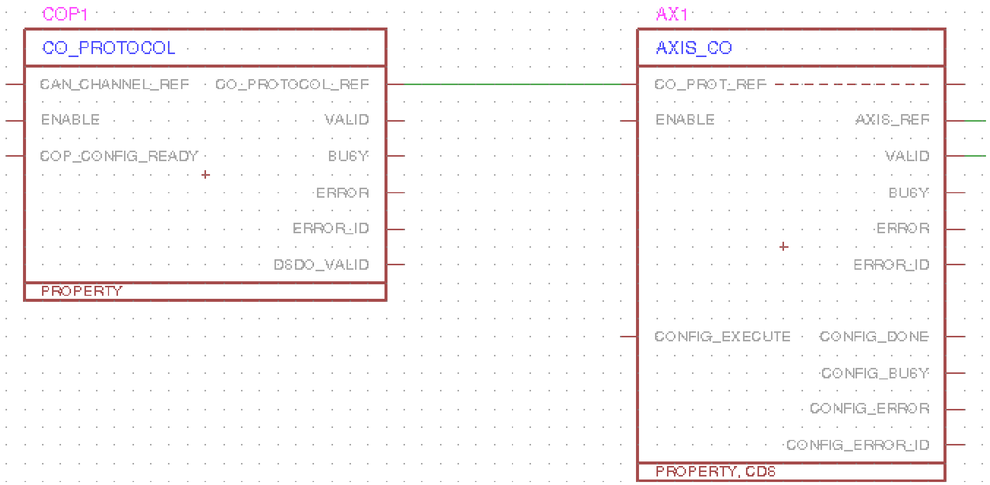


Create Signal Flow and Data Flow Connections (AXIS_REF -> CO_PROTOCOL)

- Click the ‘Connections’ icon.



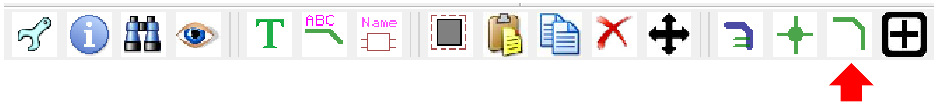
- Click on the ‘ENABLE’ connector on the ‘AXIS_CO’ FB.



- Connect to the ‘VALID’ connector on the ‘CO_PROTOCOL’ FB.

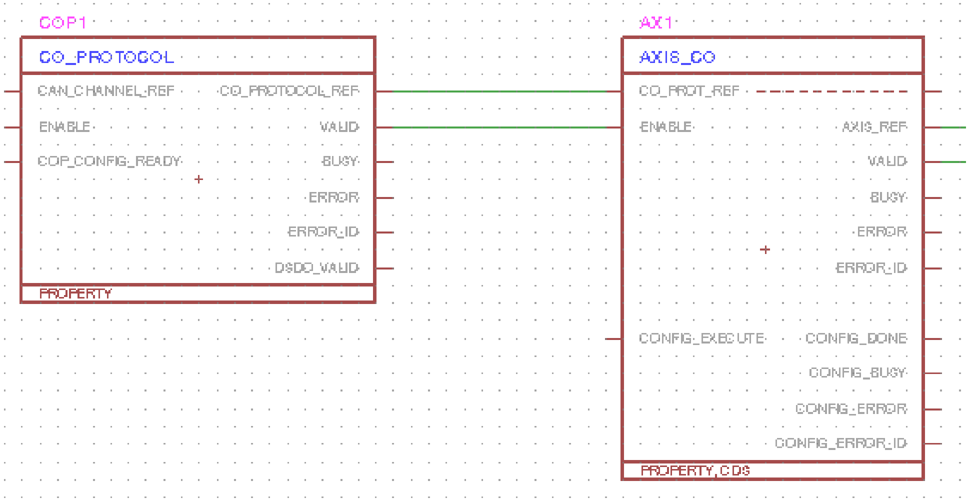
Create Signal Flow and Data Flow Connections (AXIS_REF -> CO_PROTOCOL)

- Click the ‘Connections’ icon.



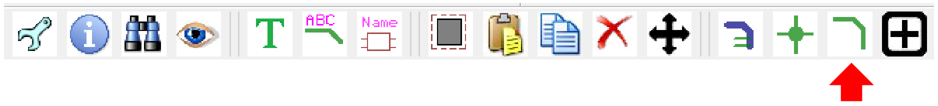
- Click on the ‘CO_PROT_REF’ connector on the ‘AXIS_CO’ FB.

- Connect to the ‘CO_PROTOCOL_REF’ connector on the ‘CO_PROTOCOL’ FB.



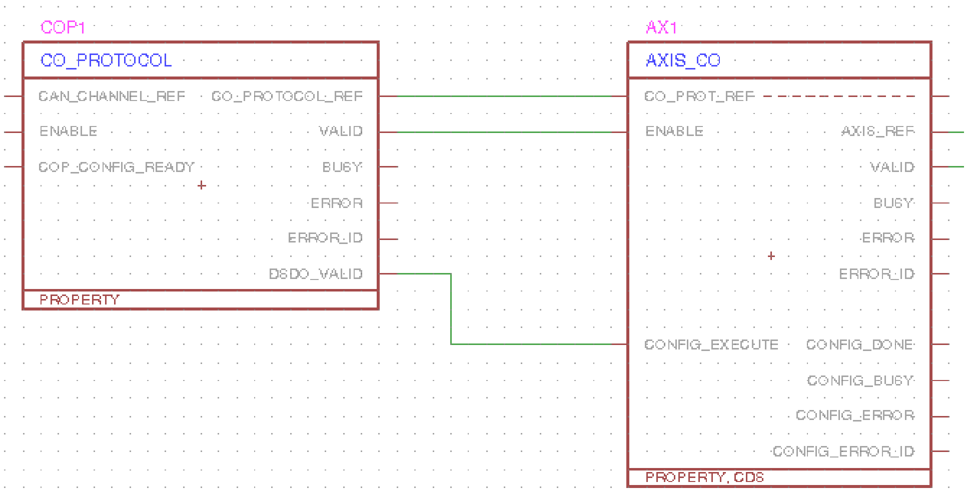
Create Signal Flow and Data Flow Connections (AXIS_REF -> CO_PROTOCOL)

- Click the ‘Connections’ icon.



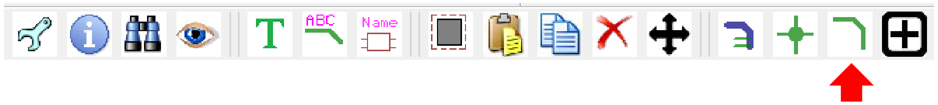
- Click on the ‘CONFIG_EXECUTE’ connector on the ‘AXIS_CO’ FB.

- Connect to the ‘DSDO_VALID’ connector on the ‘CO_PROTOCOL’ FB.

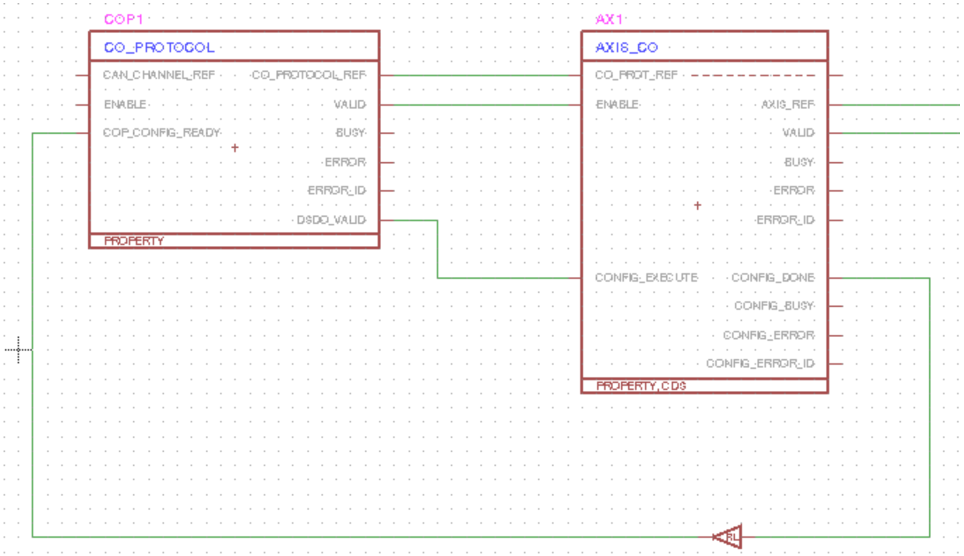


Create Signal Flow and Data Flow Connections (AXIS_REF ->Rloop-> CO_PROTOCOL)

- Click the ‘Connections’ icon.

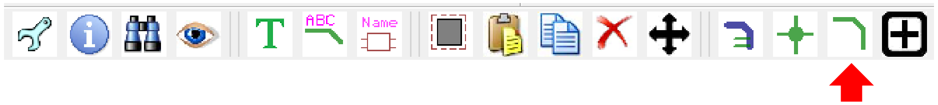


- Click on the ‘CONFIG_DONE’ connector on the ‘CO_PROTOCOL’ FB.
- Connect to the very end of the right connector on the ‘Rloop’ FB.
- Connect the left connector on the ‘Rloop’ FB to the‘COP_CONFIG_READY’ connector on the ‘CO_PROTOCOL’ FB.



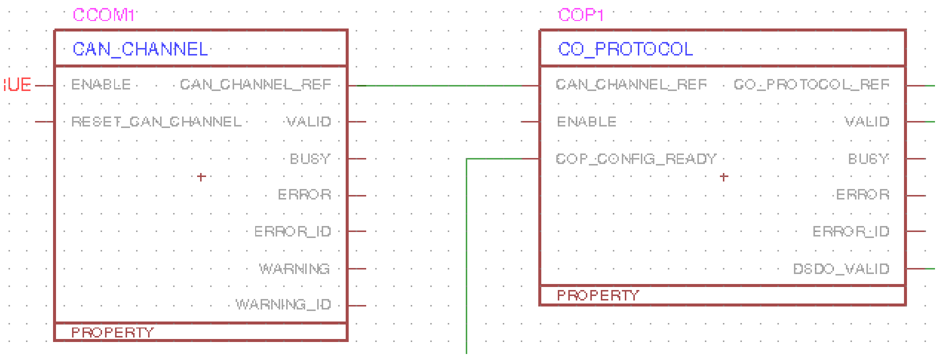
Create Signal Flow and Data Flow Connections (CO_PROTOCOL->CAM_CHANNEL)

- Click the ‘Connections’ icon.



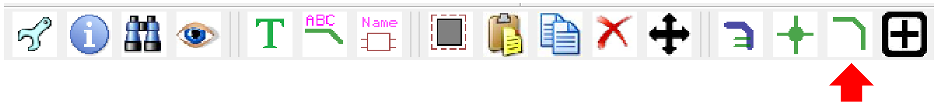
- Click on the ‘CAN_CHANNEL_REF’ connector on the ‘CO_PROTOCOL’ FB.

- Connect to the ‘CAN_CHANNEL_REF’ connector on the ‘CAN_CHANNEL’ FB.

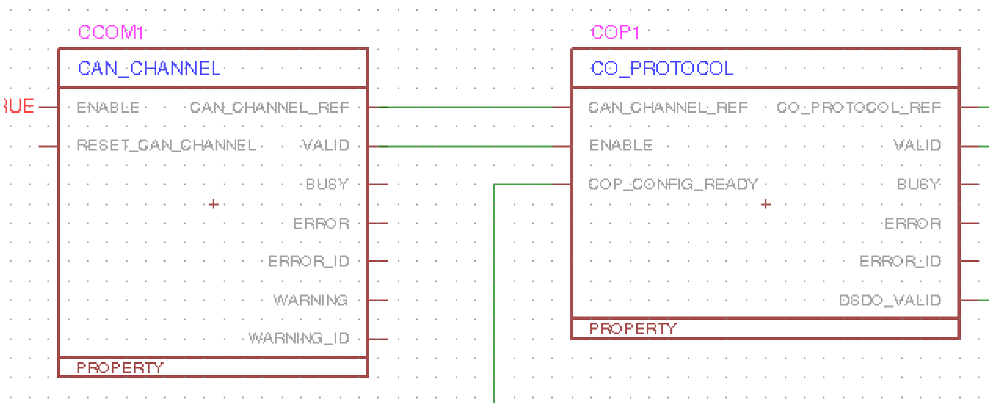


Create Signal Flow and Data Flow Connections (CO_PROTOCOL->CAM_CHANNEL)

- Click the ‘Connections’ icon.



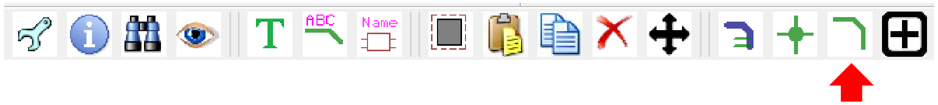
- Click on the ‘ENABLE’ connector on the ‘CO_PROTOCOL’ FB.



- Connect to the ‘VALID’ connector on the ‘CAN_CHANNEL’ FB.

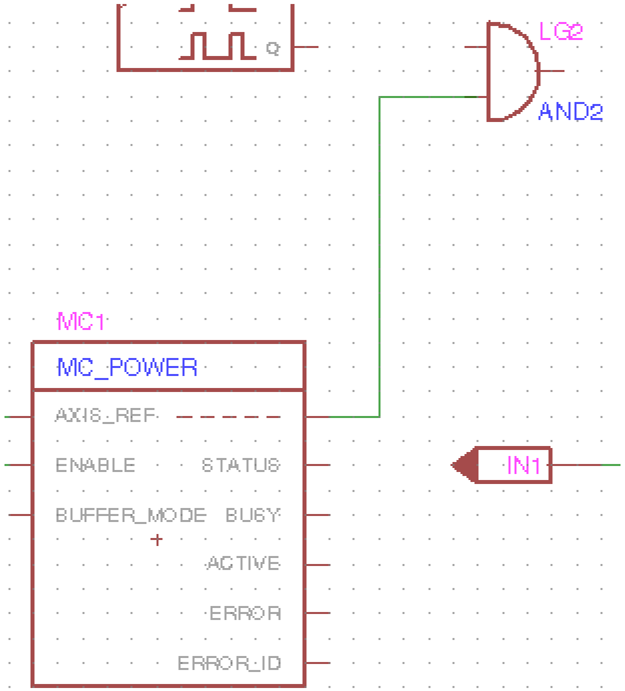
Create Signal Flow and Data Flow Connections (MC_POWER->AND2)

- Click the ‘Connections’ icon.



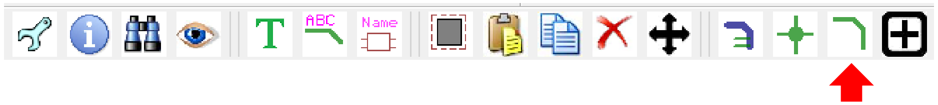
- Click on the ‘STATUS’ connector on the ‘MC_POWER’ FB.

- Connect to the bottom left connector on the ‘AND2’ FB.



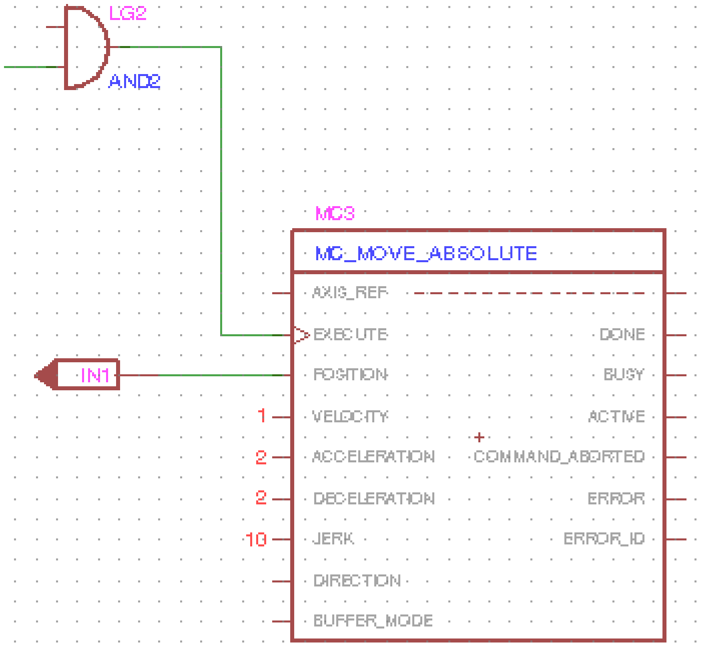
Create Signal Flow and Data Flow Connections (AND2 -> MC_MOVE_ABSOLUTE)

- Click the ‘Connections’ icon.



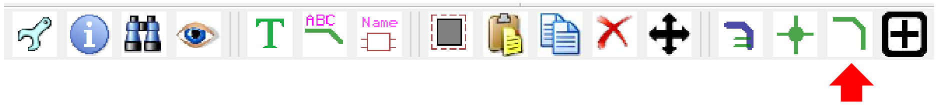
- Click on the right connector on the ‘AND2’ FB.

- Connect to the ‘EXECUTE’ connector on the ‘EXECUTE’ FB.



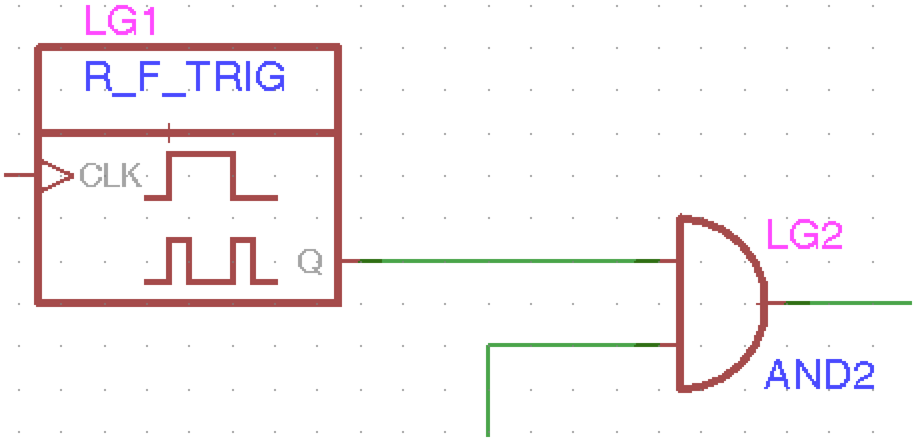
Create Signal Flow and Data Flow Connections (AND2 -> RF_TRIG)

- Click the ‘Connections’ icon.

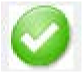


- Click on the top left connector on the ‘AND2’ FB.

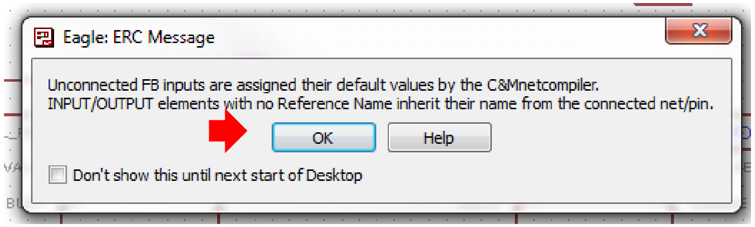
- Connect to the ‘Q’ connector on the ‘RF_TRIG’ FB.



Error Checking



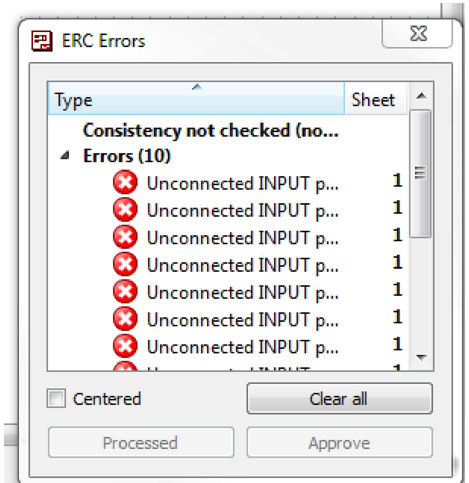
- ERC Checks the wire connections between nodes
- Click this to check that the connections are valid
- Click OK



Eagle: ERC Message

Unconnected FB inputs are assigned their default values by the C&Mnetcompiler.
INPUT/OUTPUT elements with no Reference Name inherit their name from the connected net/pin.

- Then you should receive this box



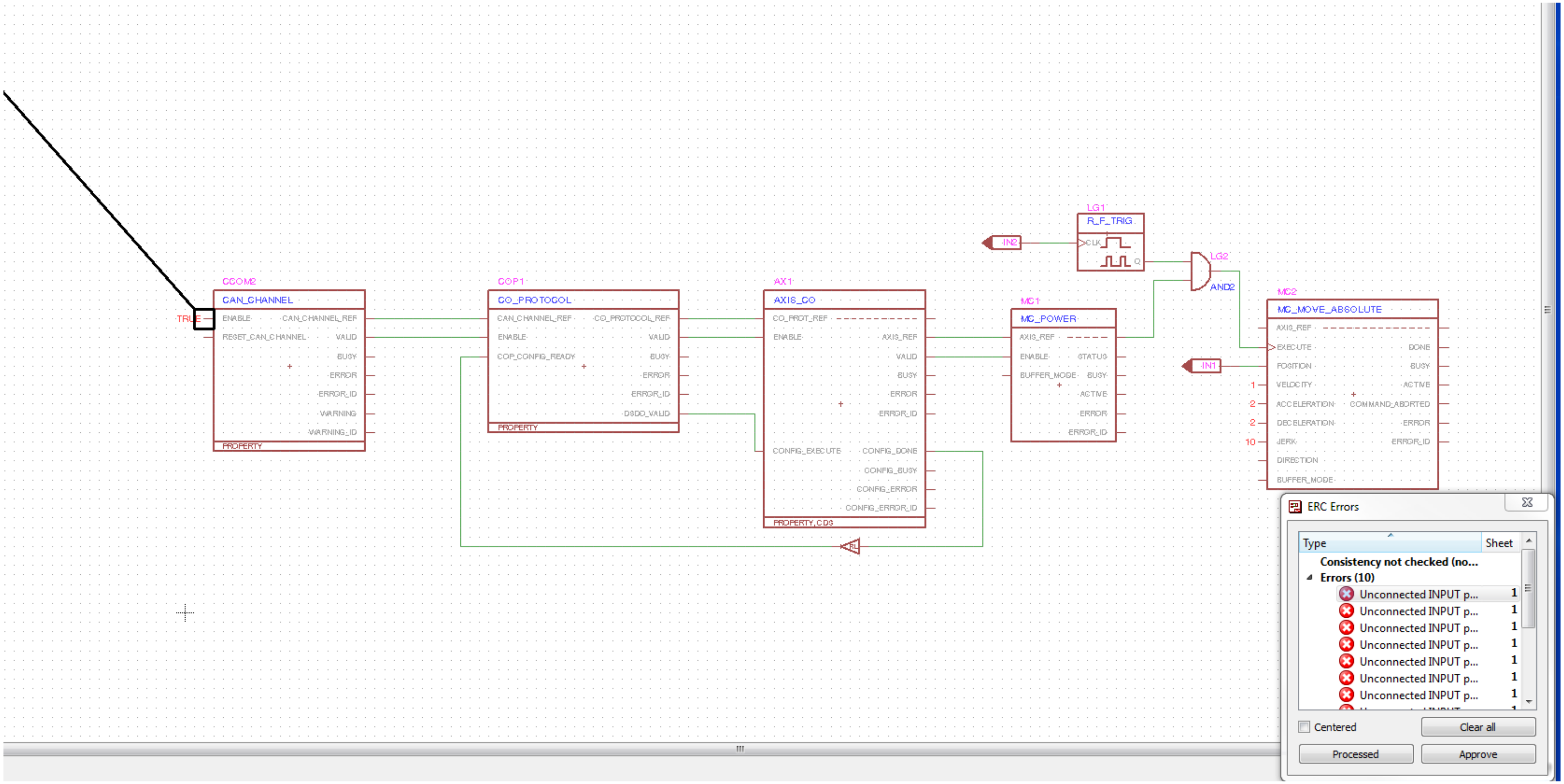
ERC Errors

Type	Sheet
Consistency not checked (no...	
Errors (10)	
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1

Clicking on the box you will see this

The magnifying line shows which nodes are not connected

Verify that the nodes with wire are all connected, if not reconnect nodes



ERC Errors

Type	Sheet
Consistency not checked (no...	
Errors (10)	
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1
Unconnected INPUT p...	1

☐ Centered

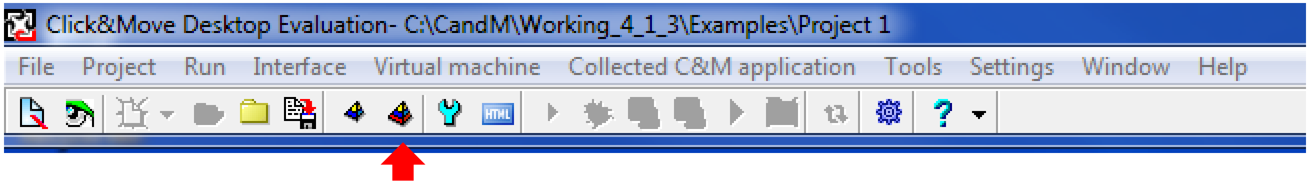
Clear all

Processed

Approve

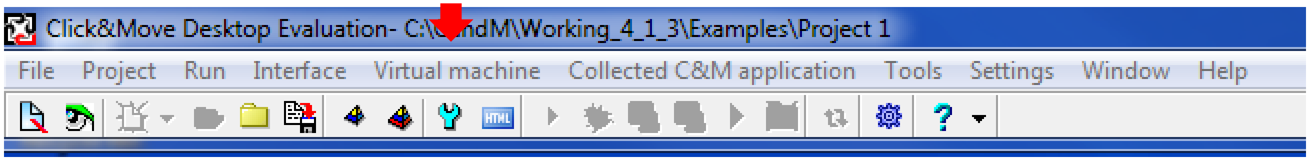
Save the project then close the window

- Once the window is closed, re-compile the project

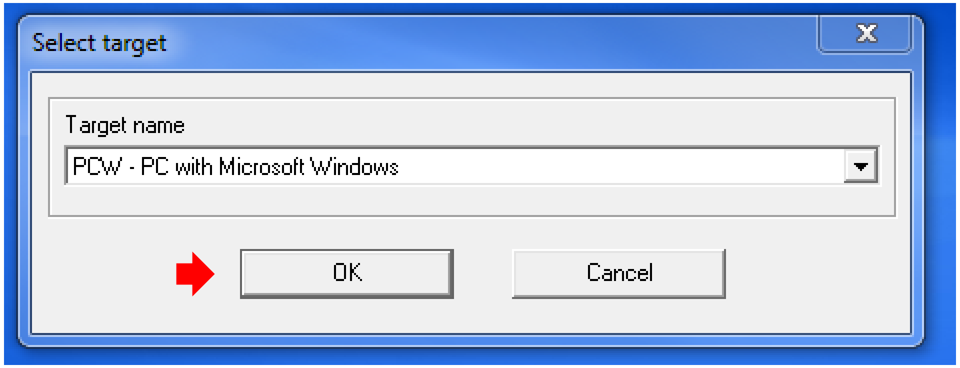

Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 1
File
Project
Run
Interface
Virtual machine
Collected C&M application
Tools
Settings
Window
Help
HTML

- Crazy things will happen please wait
- Once finished, you can close the message window

- Now we are going to create a virtual machine

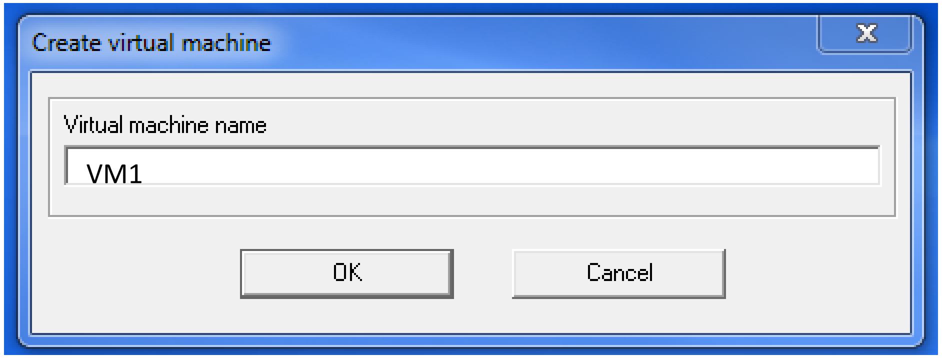

Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 1
File
Project
Run
Interface
Virtual machine
Collected C&M application
Tools
Settings
Window
Help

- Click virtual machine then go to create
- Select PCW-PC with Microsoft Windows then click ok



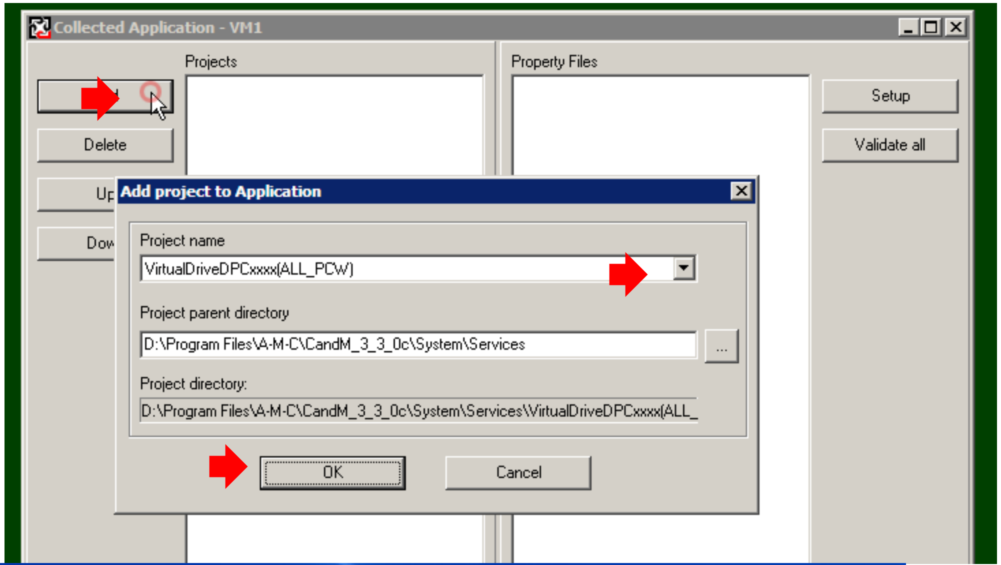
Target name: PCW - PC with Microsoft Windows

- Name your virtual machine then click OK



Virtual machine name: VM1

- Click Add then select VirtualDriveDPCxxxx(ALL_PCW) Then hit OK

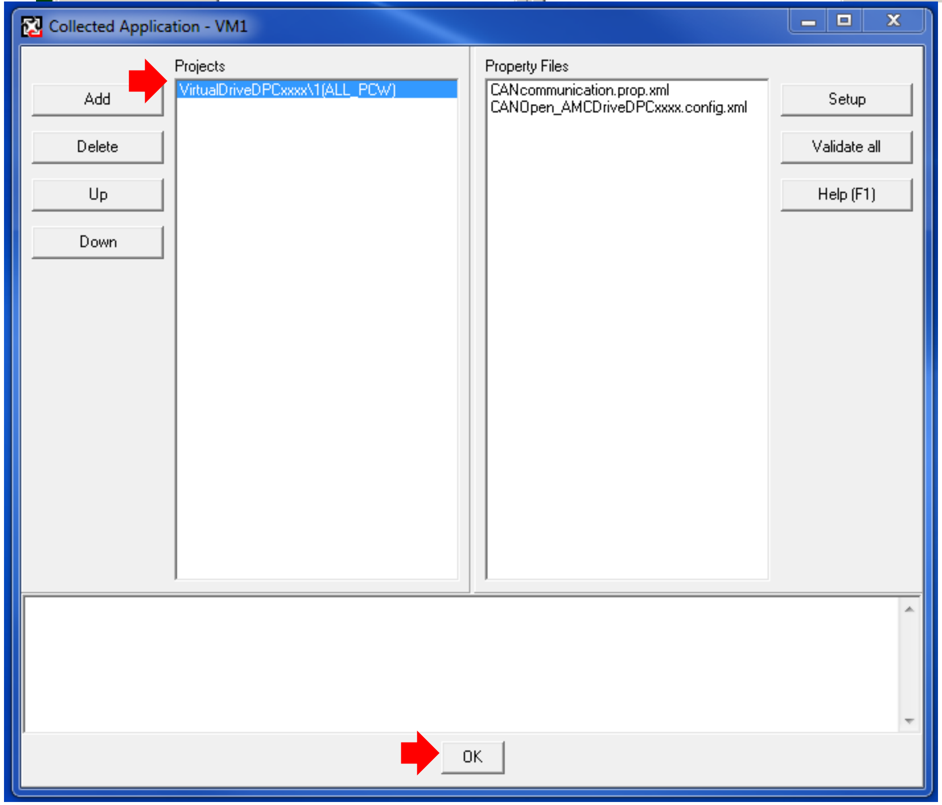


Project name
VirtualDriveDPCxxxx(ALL_PCW)

Project parent directory
D:\Program Files\A-M-C\CandM_3_3_0c\System\Services

Project directory:
D:\Program Files\A-M-C\CandM_3_3_0c\System\Services\VirtualDriveDPCxxxx(ALL_

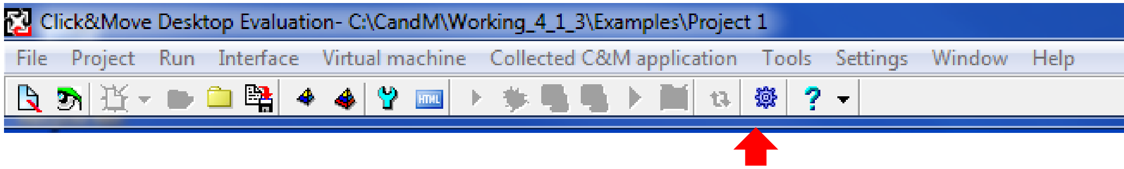
- Highlight the newly created project then hit OK



Projects
VirtualDriveDPCxxxx\1(ALL_PCW)

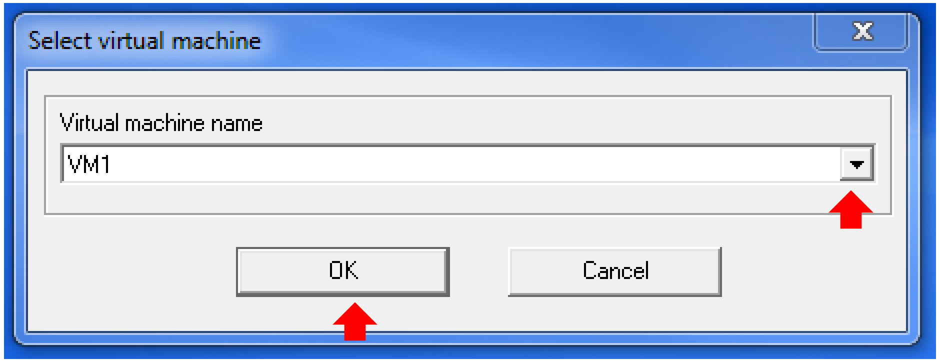
Property Files
CANcommunication.prop.xml
CANOpen_AMCDriveDPCxxxx.config.xml

- Launch the vertical axis by clicking the gear



Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 1

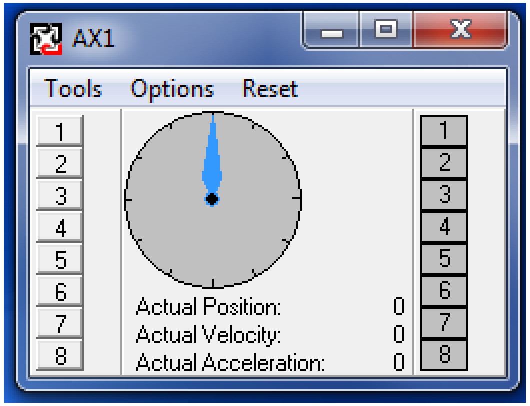
- Select your virtual machine and hit ok



Select virtual machine

Virtual machine name: VM1

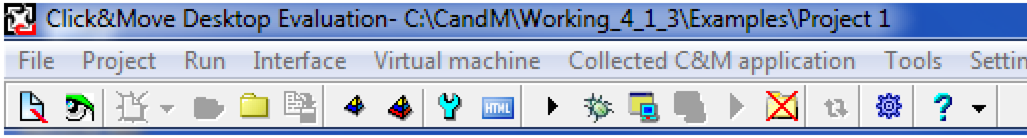
- You should now see this window
- This is your virtual machine



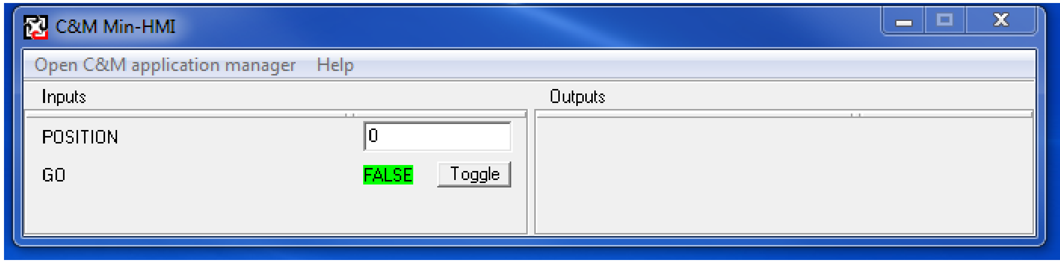
AX1

Actual Position: 0
Actual Velocity: 0
Actual Acceleration: 0

- Click load C&M Application and Run it looks like the play button


Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 1
File
Project
Run
Interface
Virtual machine
Collected C&M application
Tools
Settin

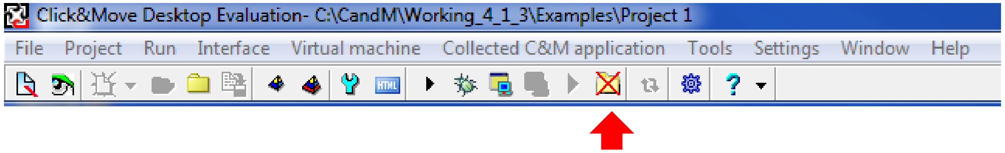
- You will now see this window



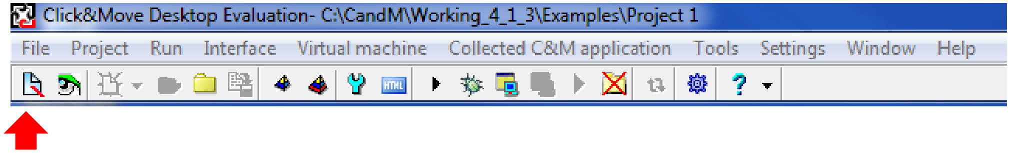
POSITION 0

GO FALSE

- You can type the position number and click toggle to watch the axis spin
- Click the close all button to end the test

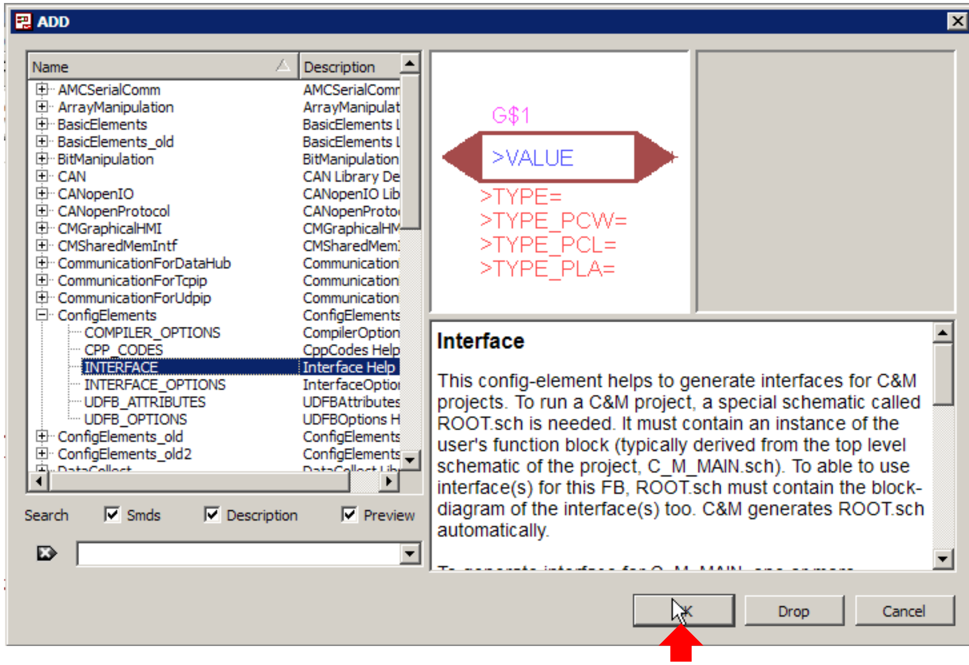

Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 1
File
Project
Run
Interface
Virtual machine
Collected C&M application
Tools
Settings
Window
Help

- Reopen the FBD schematic


Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 1
File
Project
Run
Interface
Virtual machine
Collected C&M application
Tools
Settings
Window
Help

Add the interface block

- Select ConfigElements
- Then select the interface block
- Click OK



Name	Description
AMCSerialComm	AMCSerialComm
ArrayManipulation	ArrayManipulat
BasicElements	BasicElements L
BasicElements_old	BasicElements L
BitManipulation	BitManipulation
CAN	CAN Library De
CANopenIO	CANopenIO Lib
CANopenProtocol	CANopenProto
CMGraphicalHMI	CMGraphicalHM
CMSharedMemIntf	CMSharedMemI
CommunicationForDataHub	Communication
CommunicationForTcpip	Communication
CommunicationForUdpip	Communication
ConfigElements	ConfigElements
COMPILER_OPTIONS	CompilerOption
CPP_CODES	CppCodes Help
INTERFACE	Interface Help
INTERFACE_OPTIONS	InterfaceOptio
UDFB_ATTRIBUTES	UDFBAttributes
UDFB_OPTIONS	UDFBOptions H
ConfigElements_old	ConfigElements
ConfigElements_old2	ConfigElements

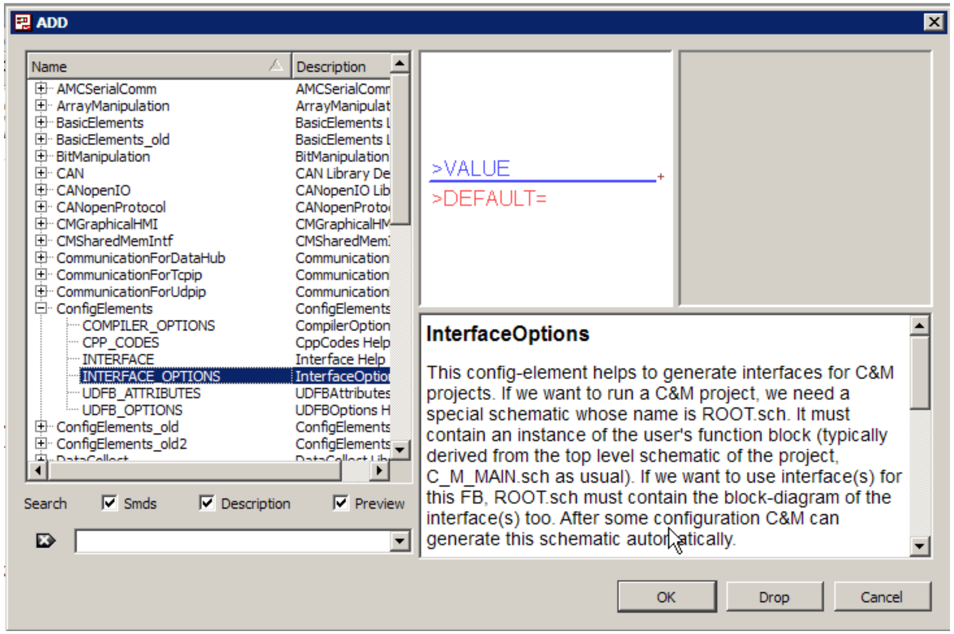
Interface

This config-element helps to generate interfaces for C&M projects. To run a C&M project, a special schematic called ROOT.sch is needed. It must contain an instance of the user's function block (typically derived from the top level schematic of the project, C_M_MAIN.sch). To able to use interface(s) for this FB, ROOT.sch must contain the block-diagram of the interface(s) too. C&M generates ROOT.sch automatically.

- Ad this block above the CAN_Channel

- And another Function block
- this block is located in ConfigElements
- then select the interface_option block

- Place this block to the right of the interface block



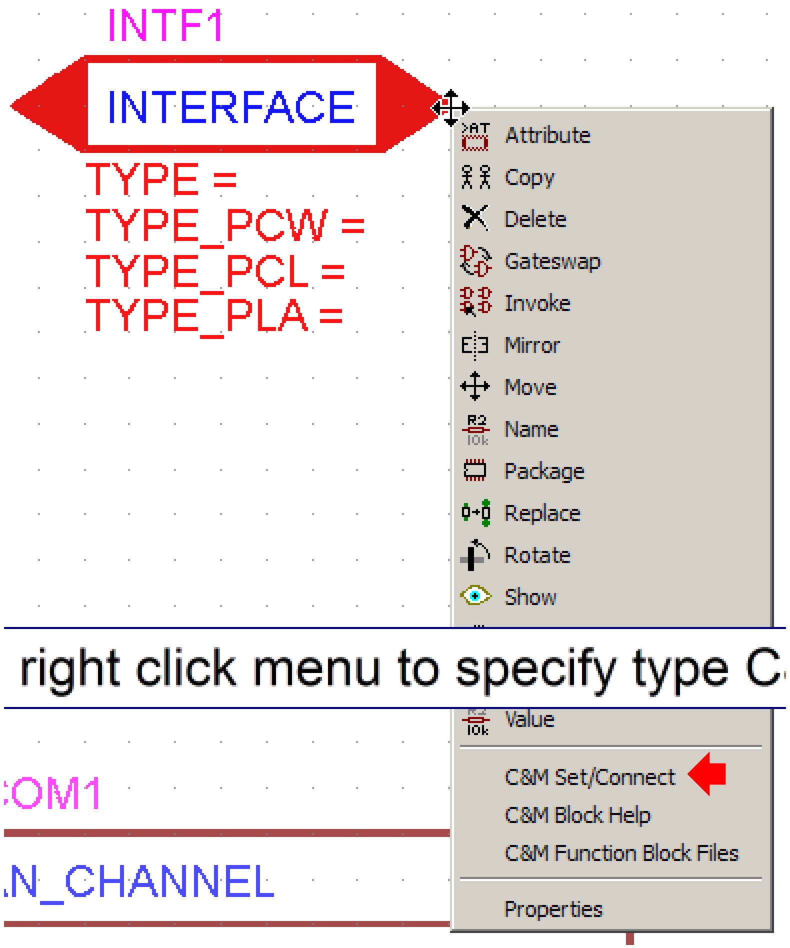
Name	Description
AMCSerialComm	AMCSerialComm
ArrayManipulation	ArrayManipulat
BasicElements	BasicElements l
BasicElements_old	BasicElements l
BitManipulation	BitManipulation
CAN	CAN Library De
CANopenIO	CANopenIO Lib
CANopenProtocol	CANopenProto
CMGraphicalHMI	CMGraphicalHM
CMSharedMemIntf	CMSharedMem
CommunicationForDataHub	Communication
CommunicationForTcpip	Communication
CommunicationForUdpip	Communication
ConfigElements	ConfigElements
COMPILER_OPTIONS	CompilerOption
CPP_CODES	CppCodes Help
INTERFACE	Interface Help
INTERFACE_OPTIONS	InterfaceOptio
UDFB_ATTRIBUTES	UDFBAttributes
UDFB_OPTIONS	UDFBOptions H
ConfigElements_old	ConfigElements
ConfigElements_old2	ConfigElements
DataCollect	DataCollect Lib

Search ☑ Smds ☑ Description ☑ Preview

InterfaceOptions

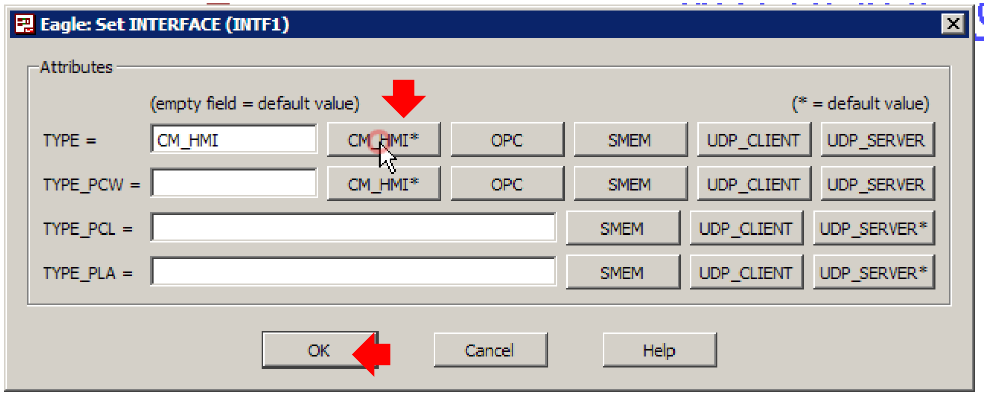
This config-element helps to generate interfaces for C&M projects. If we want to run a C&M project, we need a special schematic whose name is ROOT.sch. It must contain an instance of the user's function block (typically derived from the top level schematic of the project, C_M_MAIN.sch as usual). If we want to use interface(s) for this FB, ROOT.sch must contain the block-diagram of the interface(s) too. After some configuration C&M can generate this schematic automatically.

- Right click the interface cross hair
- Then select C&M Set/Connect



right click menu to specify type C

- Click CM_HMI*
- Then click OK

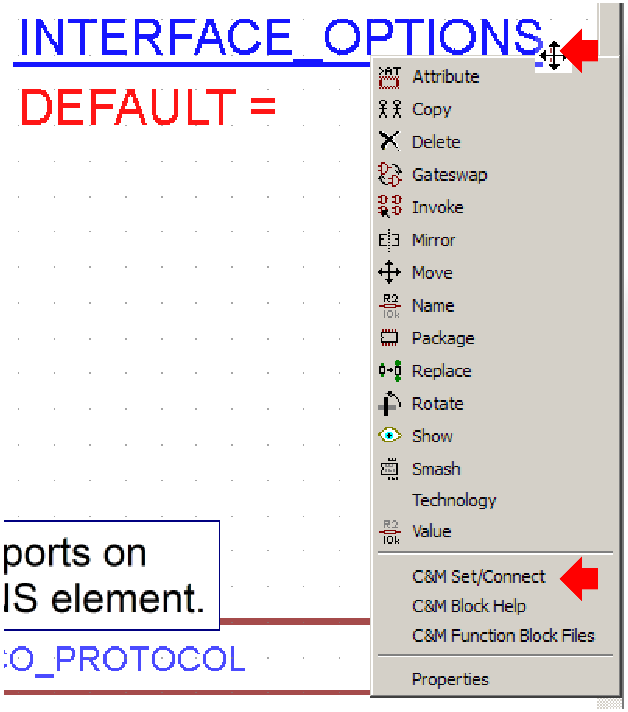

Eagle: Set INTERFACE (INTF1)

Attributes

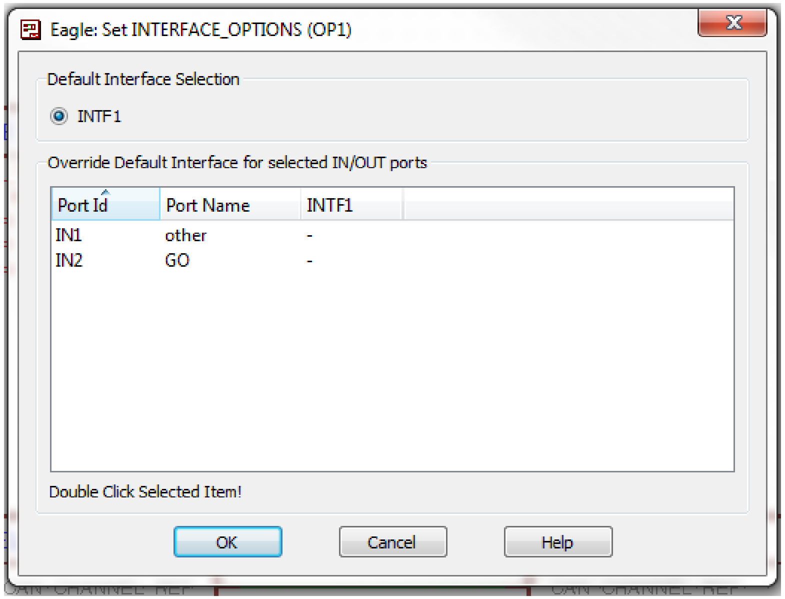
(empty field = default value) (* = default value)

TYPE =	CM_HMI	CM_HMI*	OPC	SMEM	UDP_CLIENT	UDP_SERVER
TYPE_PCW =		CM_HMI*	OPC	SMEM	UDP_CLIENT	UDP_SERVER
TYPE_PCL =				SMEM	UDP_CLIENT	UDP_SERVER*
TYPE_PLA =				SMEM	UDP_CLIENT	UDP_SERVER*

- Right click the interface option
- Then select C&M Set/Connect



- Click INTF1
- Then select OK


Eagle: Set INTERFACE_OPTIONS (OP1)

Default Interface Selection

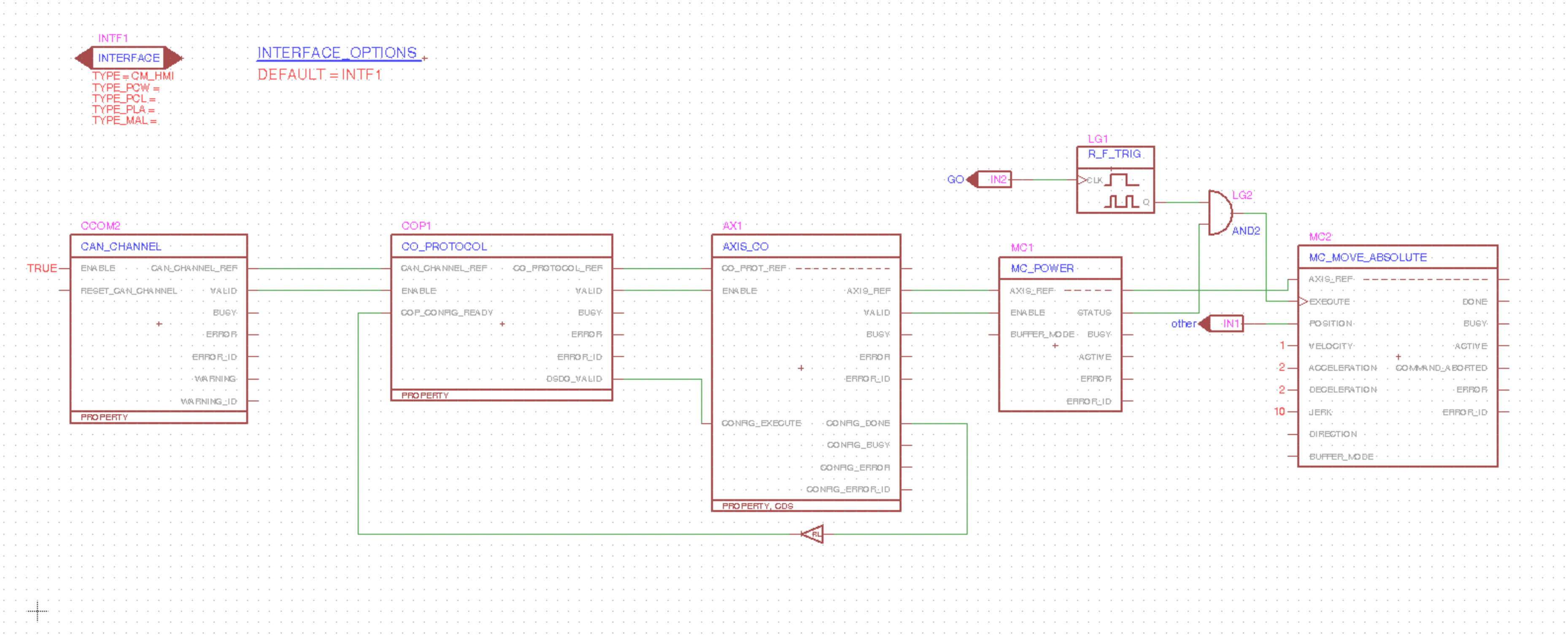
● INTF1

Override Default Interface for selected IN/OUT ports

Port Id	Port Name	INTF1
IN1	other	-
IN2	GO	-

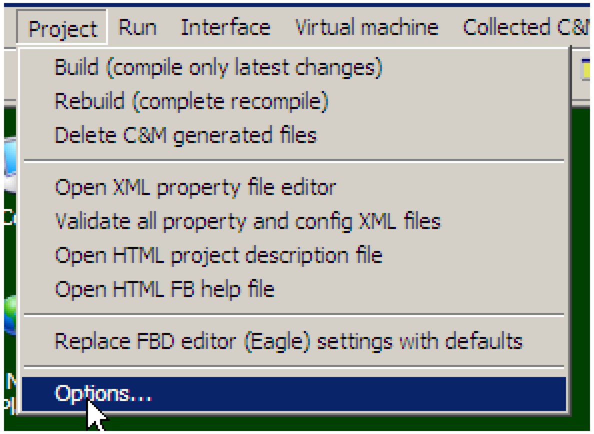
Double Click Selected Item!

This is what the final diagram should look like

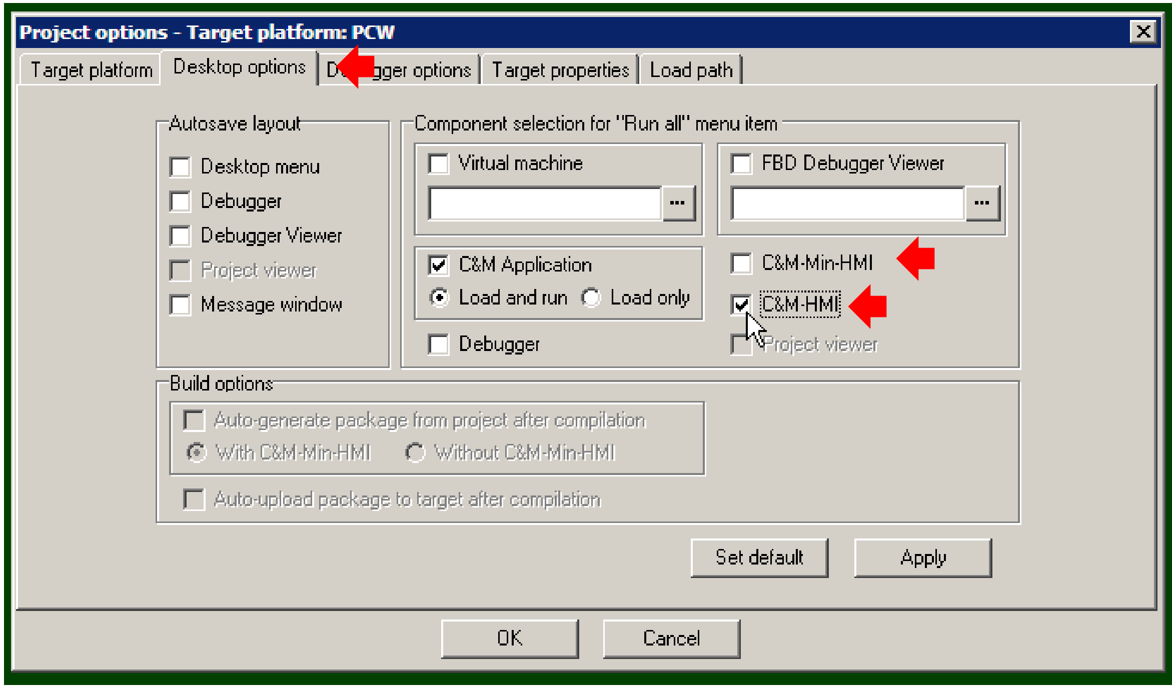


Save and exit the FBD Schematic

- Go to Project options…


Project
Run
Interface
Virtual machine
Collected C&
Build (compile only latest changes)
Rebuild (complete recompile)
Delete C&M generated files
Open XML property file editor
Validate all property and config XML files
Open HTML project description file
Open HTML FB help file
Replace FBD editor (Eagle) settings with defaults
Options…

- Click Desktop Options
- Then Uncheck C&M-Mini-HMI
- Then check C&M-HMI



Autosave layout

- □ Desktop menu
- □ Debugger
- □ Debugger Viewer
- □ Project viewer
- □ Message window

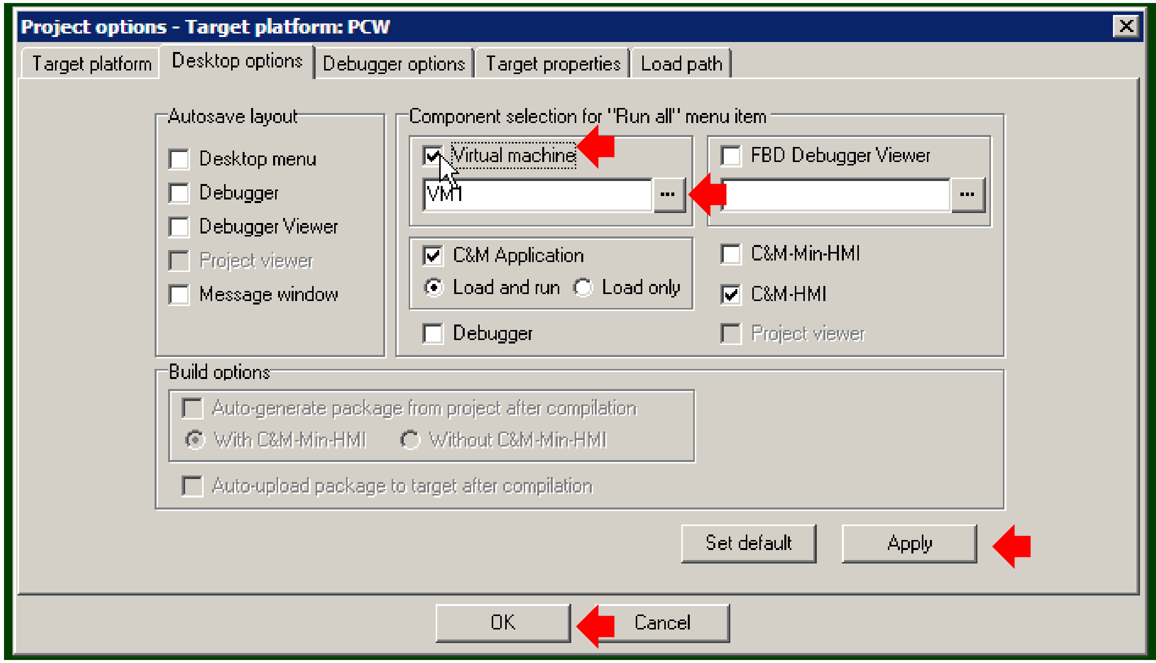
Component selection for "Run all" menu item

- □ Virtual machine
- □ FBD Debugger Viewer
- ☑ C&M Application
  - ● Load and run ○ Load only
- □ Debugger
- □ C&M-Min-HMI
- ☑ C&M-HMI
- □ Project viewer

Build options

- □ Auto-generate package from project after compilation
  - ● With C&M-Min-HMI ○ Without C&M-Min-HMI
- □ Auto-upload package to target after compilation

- Check mark Virtual Machine
- Then select your Virtual Machine



Project options - Target platform: PCW

Target platform | Desktop options | Debugger options | Target properties | Load path

Autosave layout

- □ Desktop menu
- □ Debugger
- □ Debugger Viewer
- □ Project viewer
- □ Message window

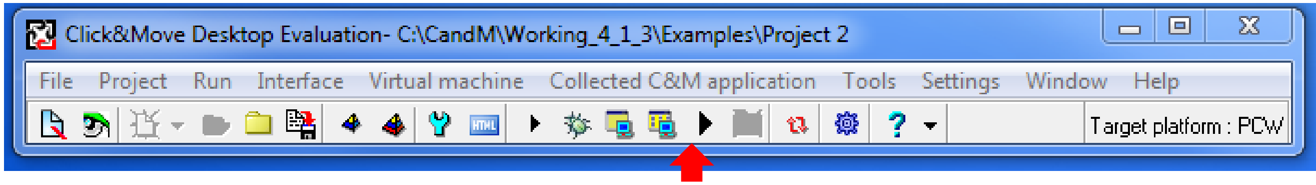
Component selection for "Run all" menu item

- ☑ Virtual machine: VM1 ...
- □ FBD Debugger Viewer: ...
- ☑ C&M Application: ● Load and run ○ Load only
- □ C&M-Min-HMI
- ☑ C&M-HMI
- □ Debugger
- □ Project viewer

Build options

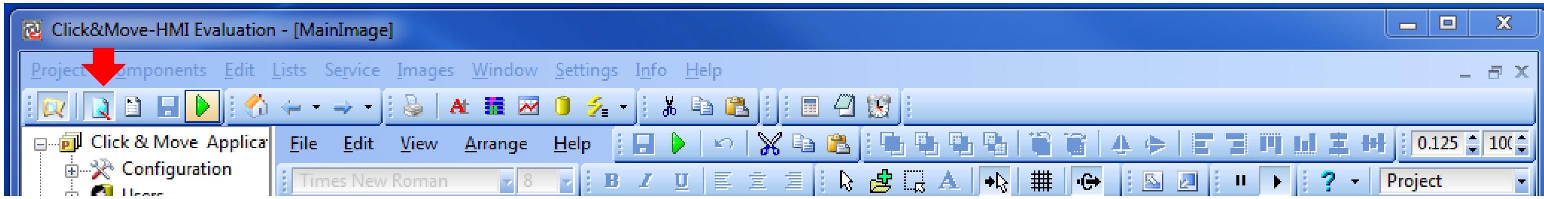
- □ Auto-generate package from project after compilation
  - ● With C&M-Min-HMI ○ Without C&M-Min-HMI
- □ Auto-upload package to target after compilation

- Click Apply then OK

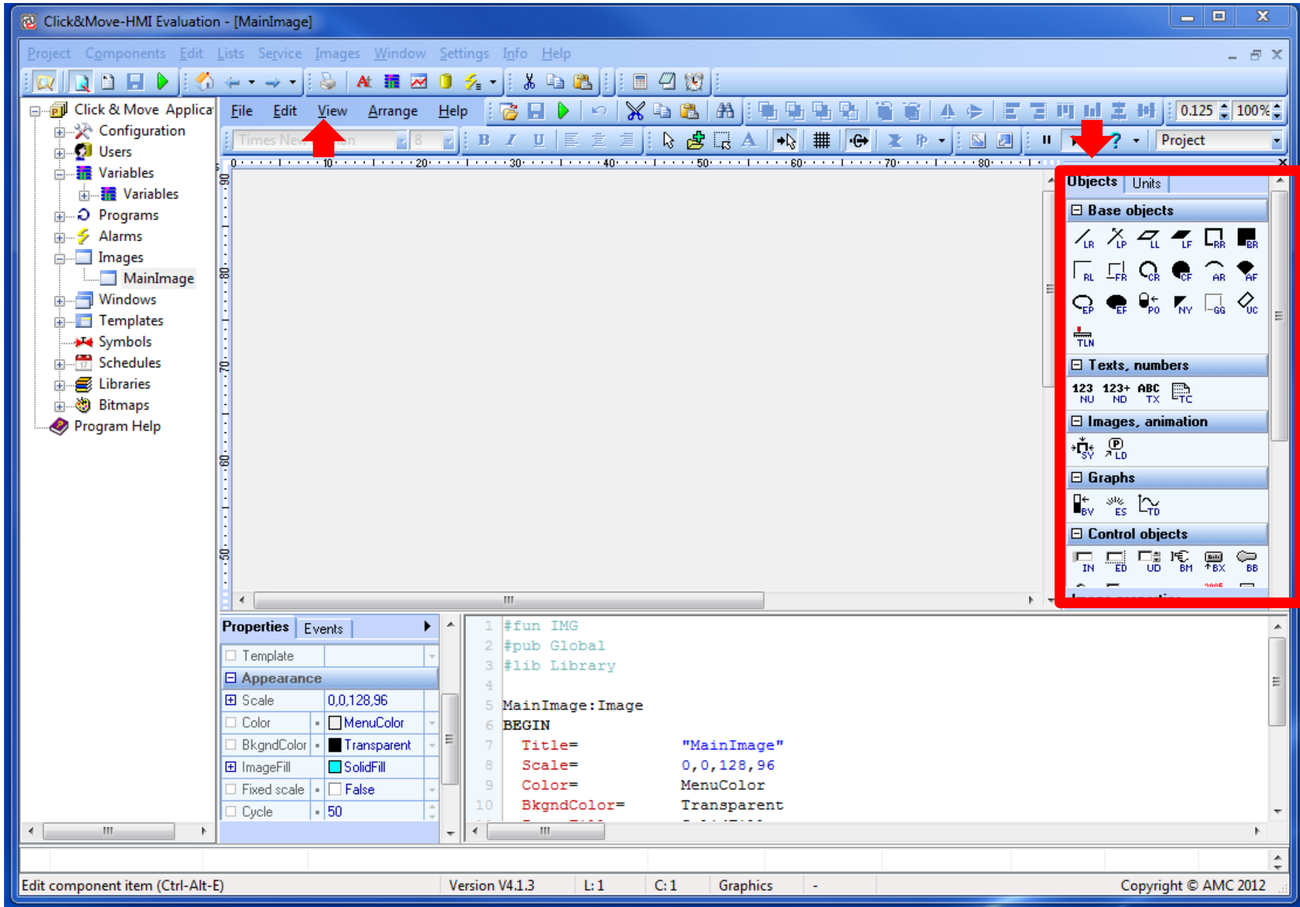


Click&Move Desktop Evaluation- C:\CandM\Working_4_1_3\Examples\Project 2

- Click Run, the larger play button



- Click Edit Component Item
- If the objects box is missing, click view then Object Browser

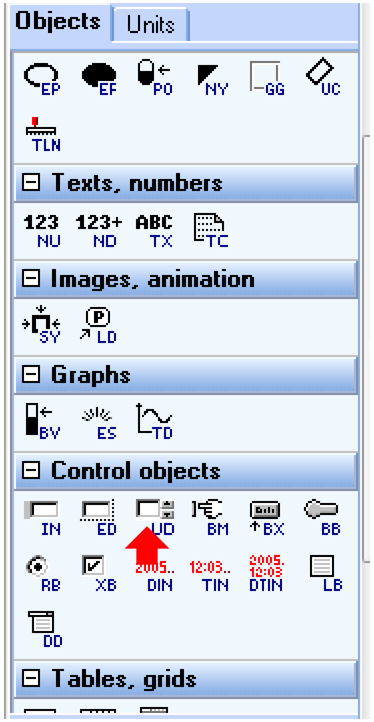


Properties	
Template	
Appearance	
Scale	0,0,128,96
Color	MenuColor
BkgndColor	Transparent
ImageFill	SolidFill
Fixed scale	False
Cycle	50

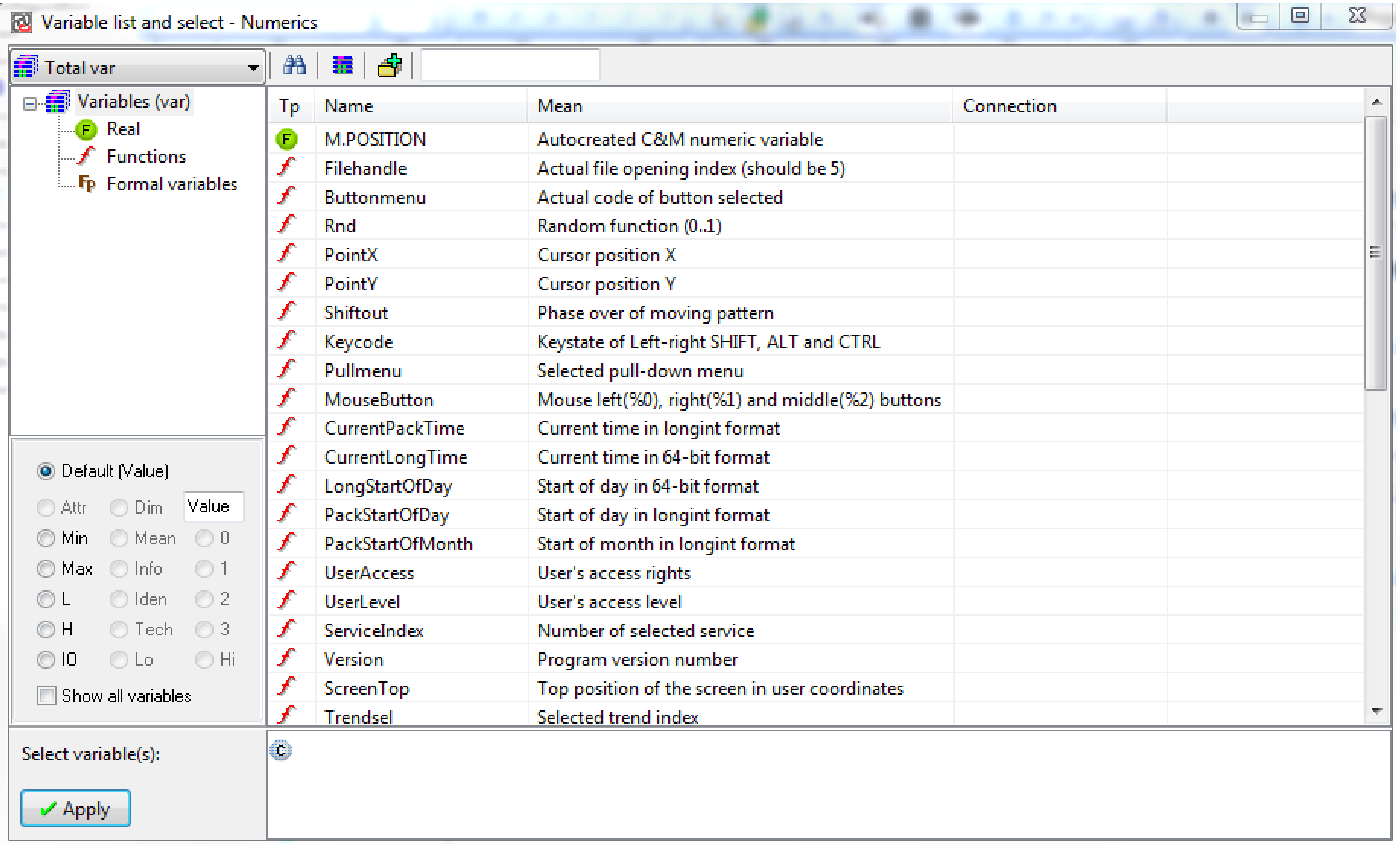
```
#fun IMG
#pub Global
#lib Library

MainImage:Image
BEGIN
  Title=          "MainImage"
  Scale=          0,0,128,96
  Color=          MenuColor
  BkgndColor=     Transparent
```

Edit component item (Ctrl-Alt-E) Version V4.1.3 L:1 C:1 Graphics - Copyright © AMC 2012

- Navigate to control objects and drag an up down object to the main screen



Then you should see this window



Tp	Name	Mean	Connection
F	M.POSITION	Autocreated C&M numeric variable	
f	Filehandle	Actual file opening index (should be 5)	
f	Buttonmenu	Actual code of button selected	
f	Rnd	Random function (0..1)	
f	PointX	Cursor position X	
f	PointY	Cursor position Y	
f	Shiftout	Phase over of moving pattern	
f	Keycode	Keystate of Left-right SHIFT, ALT and CTRL	
f	Pullmenu	Selected pull-down menu	
f	MouseButton	Mouse left(%0), right(%1) and middle(%2) buttons	
f	CurrentPackTime	Current time in longint format	
f	CurrentLongTime	Current time in 64-bit format	
f	LongStartOfDay	Start of day in 64-bit format	
f	PackStartOfDay	Start of day in longint format	
f	PackStartOfMonth	Start of month in longint format	
f	UserAccess	User's access rights	
f	UserLevel	User's access level	
f	ServiceIndex	Number of selected service	
f	Version	Program version number	
f	ScreenTop	Top position of the screen in user coordinates	
f	Trendsel	Selected trend index	

● Default (Value)

○ Attr ○ Dim Value

○ Min ○ Mean ○ 0

○ Max ○ Info ○ 1

○ L ○ Iden ○ 2

○ H ○ Tech ○ 3

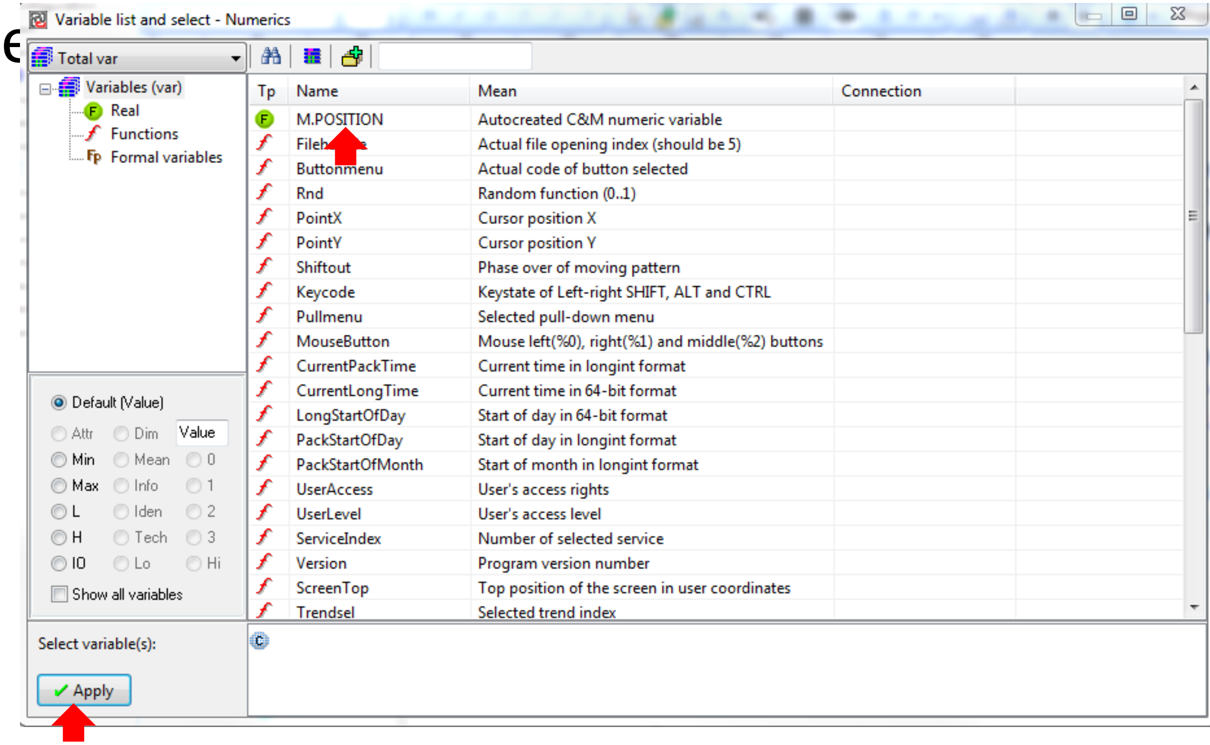
○ IO ○ Lo ○ Hi

□ Show all variables

Select variable(s):

✓ Apply

- Click M.Position the click Apply



Variable list and select - Numerics

Total var

- Variables (var)
  - Real
  - Functions
  - Formal variables

Tp	Name	Mean	Connection
F	M.POSITION	Autocreated C&M numeric variable	
f	Filehandle	Actual file opening index (should be 5)	
f	Buttonmenu	Actual code of button selected	
f	Rnd	Random function (0..1)	
f	PointX	Cursor position X	
f	PointY	Cursor position Y	
f	Shiftout	Phase over of moving pattern	
f	Keycode	Keystate of Left-right SHIFT, ALT and CTRL	
f	Pullmenu	Selected pull-down menu	
f	MouseButton	Mouse left(%0), right(%1) and middle(%2) buttons	
f	CurrentPackTime	Current time in longint format	
f	CurrentLongTime	Current time in 64-bit format	
f	LongStartOfDay	Start of day in 64-bit format	
f	PackStartOfDay	Start of day in longint format	
f	PackStartOfMonth	Start of month in longint format	
f	UserAccess	User's access rights	
f	UserLevel	User's access level	
f	ServiceIndex	Number of selected service	
f	Version	Program version number	
f	ScreenTop	Top position of the screen in user coordinates	
f	Trendsel	Selected trend index	

● Default (Value)

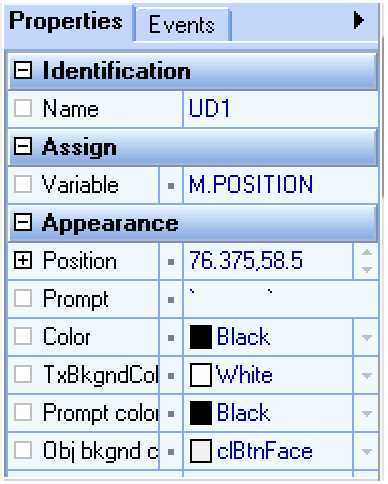
○ Attr ○ Dim Value
○ Min ○ Mean ○ 0
○ Max ○ Info ○ 1
○ L ○ Iden ○ 2
○ H ○ Tech ○ 3
○ IO ○ Lo ○ Hi

□ Show all variables

Select variable(s):

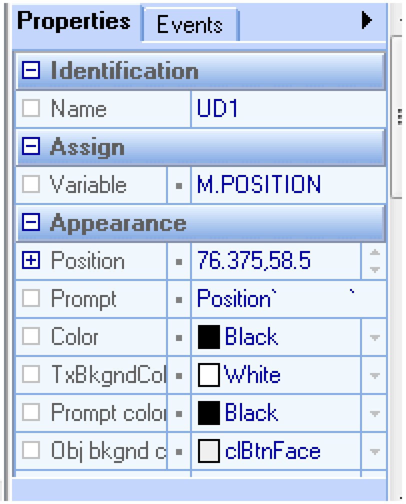
✓ Apply

- In the Properties Window, you can name the box by editing the Prompt
- The apostrophes denote the length the box Type in front of them



Properties | Events

⊟ Identification	
□ Name	UD1
⊟ Assign	
□ Variable	M.POSITION
⊟ Appearance	
⊞ Position	76,375,58,5
□ Prompt	`  `
□ Color	■ Black
□ TxBkgndCol	□ White
□ Prompt color	■ Black
□ Obj bkgnd c	□ clBtnFace

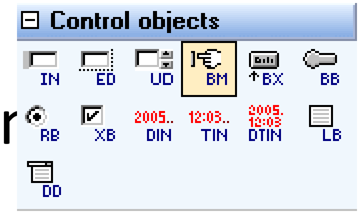


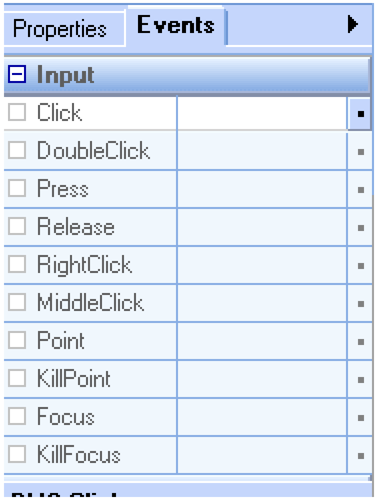
Properties | Events

⊟ Identification	
□ Name	UD1
⊟ Assign	
□ Variable	M.POSITION
⊟ Appearance	
⊞ Position	76,375,58,5
□ Prompt	Position`  `
□ Color	■ Black
□ TxBkgndCol	□ White
□ Prompt color	■ Black
□ Obj bkgnd c	□ clBtnFace

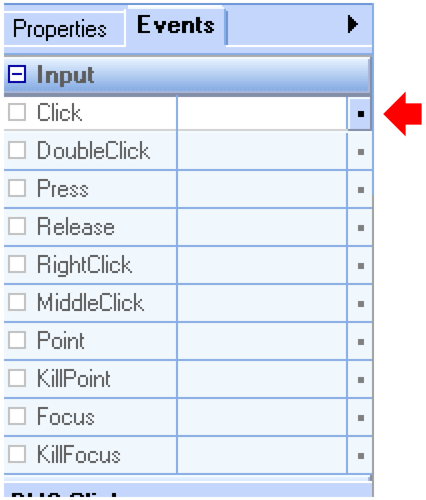
Adding a button

- Under Control Objects, Button Men
- Right Click this to bring up the button types
- Select the start button then drag it to the main screen
- Where the properties menu is select the events tab



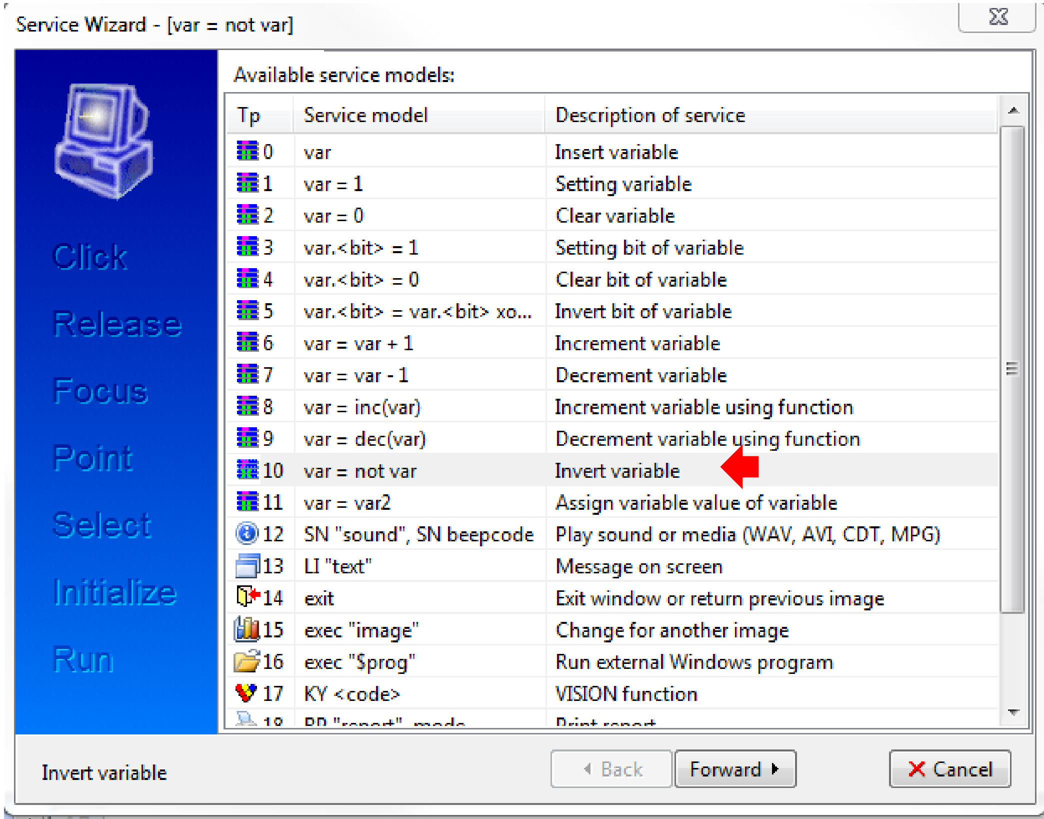


- Click the dot on the click line



□ Click	▪
□ DoubleClick	▪
□ Press	▪
□ Release	▪
□ RightClick	▪
□ MiddleClick	▪
□ Point	▪
□ KillPoint	▪
□ Focus	▪
□ KillFocus	▪

- Click Invert Variable



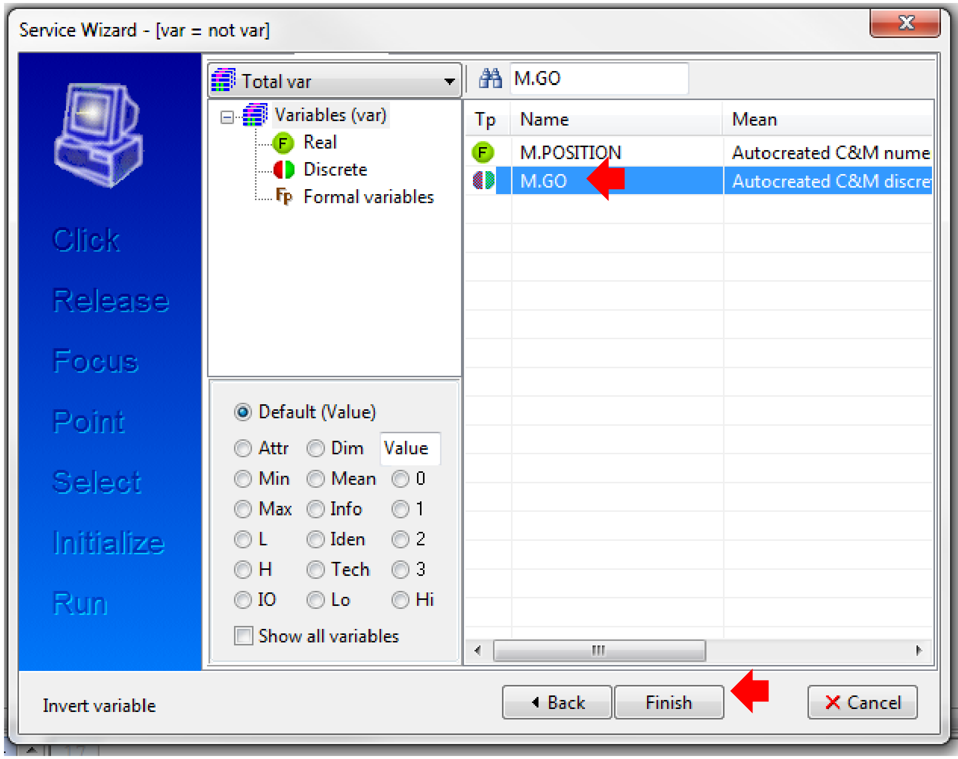
Service Wizard - [var = not var]

Available service models:

Tp	Service model	Description of service
0	var	Insert variable
1	var = 1	Setting variable
2	var = 0	Clear variable
3	var.<bit> = 1	Setting bit of variable
4	var.<bit> = 0	Clear bit of variable
5	var.<bit> = var.<bit> xo...	Invert bit of variable
6	var = var + 1	Increment variable
7	var = var - 1	Decrement variable
8	var = inc(var)	Increment variable using function
9	var = dec(var)	Decrement variable using function
10	var = not var	Invert variable
11	var = var2	Assign variable value of variable
12	SN "sound", SN beepcode	Play sound or media (WAV, AVI, CDT, MPG)
13	LI "text"	Message on screen
14	exit	Exit window or return previous image
15	exec "image"	Change for another image
16	exec "$prog"	Run external Windows program
17	KY <code>	VISION function
18	PP "report", mode	Print report

Invert variable

- Click M.Go then click Finish



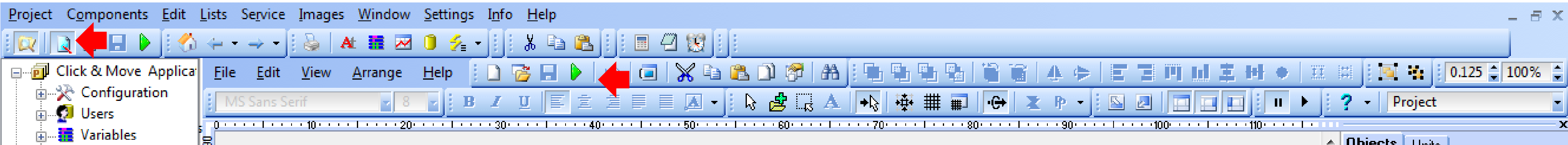
Service Wizard - [var = not var]

Tp	Name	Mean
F	M.POSITION	Autocreated C&M nume
	M.GO	Autocreated C&M discre

Now Click the green play button

Click on the edit component item

Click Save on the pop up menu



Project Components Edit Lists Service Images Window Settings Info Help
Click & Move Applica
Configuration
Users
Variables
File Edit View Arrange Help
MS Sans Serif
8
0.125
100%
Project
Objects
Units

- Congratulations You’re done
- You can now move the axis with this menu