ENGINEERING YOUR SUCCESS.

PARKER AUTOMATION CONTROLLER

Motion in Minutes Tutorial

Effective: June 2016
Document Number: Rev 3
Knowledge Level: Tier 1

Important User Information

Please read and follow all safety information for the Parker Automation Controller (PAC), including the warning and caution statements in this guide, before installing or operating the system.

Safety Information

WARNING: The PAC is used to control electrical and mechanical components of motion control systems in industrial environments. To avoid serious injury or damage to equipment, test the motion system for safety under all potential conditions.

WARNING: The PAC and PAC Input\Output (PACIO) Modules are not fault-tolerant and are not designed or intended for any use in any systems, machines, or applications where failure or fault of any kind of the Products could reasonably be seen to lead to death or serious bodily injury of any person, or to severe physical or environmental damage ("High Risk Use"). You are not permitted to use, distribute, or sublicense the use of these Products in High Risk Use. High Risk Use is STRICTLY PROHIBITED.

WARNING: The PAC contains no user-serviceable parts. To avoid personal injury or damage to the product, do not attempt to open the case or to replace any internal component of the PAC, Modules, or accessories.

WARNING: USER RESPONSIBILITY- Failure or improper selection or improper use of the products described herein or related items can cause death, personal injury and property damage.

Contents

About this Project

The project is intended for first time users of the PAC who have some familiarity with the Parker Automation Manager (PAM) integrated development environment (IDE).

Project BOM

PAC320-CXN31-3A Runtime 3.5.5.20

Compax3 Drive - S025V2F12I31T11M00 or Parker P-Drives

MPE Motors - MPE0401-KCN1

PACIO-450-03

Parker Automation Manager 1.2.1 and 1.2.1 Flashback

Motion_In_Minutes.projectarchive

Expected Programming Level and Project State

The project is intended for first time users of the PAC who have some familiarity with the Parker Automation Manager (PAM) integrated development environment (IDE).

This project will start from scratch, but Motion_In_Minutes.projectarchive has the solution for both the simulation mode and online mode.

Users must have familiarity with and understand the purpose of:

- The PAM Device Tree and it's sub components, specifically:
    - Task Configurator
    - Applications
    - EtherCAT_Master
        - EtherCAT_Master.EtherCAT_Task
    - Library Manager

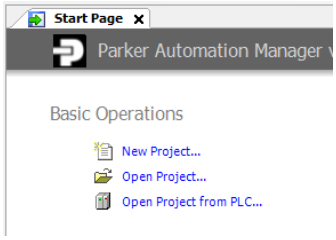
If this is your first time using PAM, please consider reviewing the "Understanding the PAM Environment" tutorial first which covers these topics.

Creating a New Project

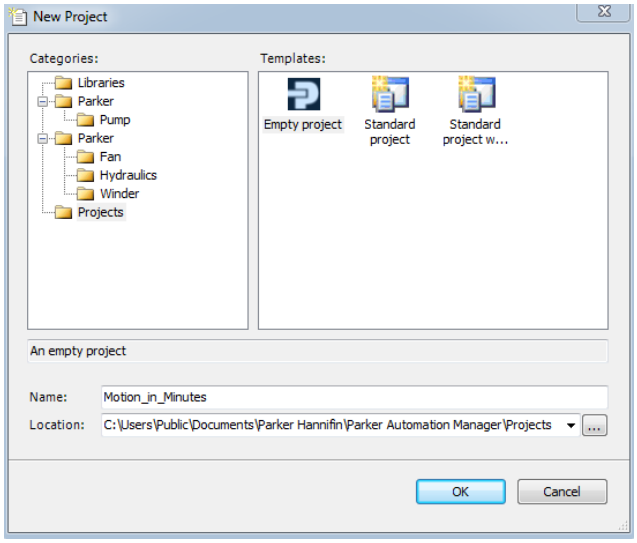
1. Launch Parker Automation Manager (PAM) .

Don't have PAM?
No worries, PAM is available for download on our website, parker.com/emn or bity.ly/Parker_PAC to go directly to the product page.

2. Select "New Project..."



3. Name your project, we used "Motion_in_Minutes".



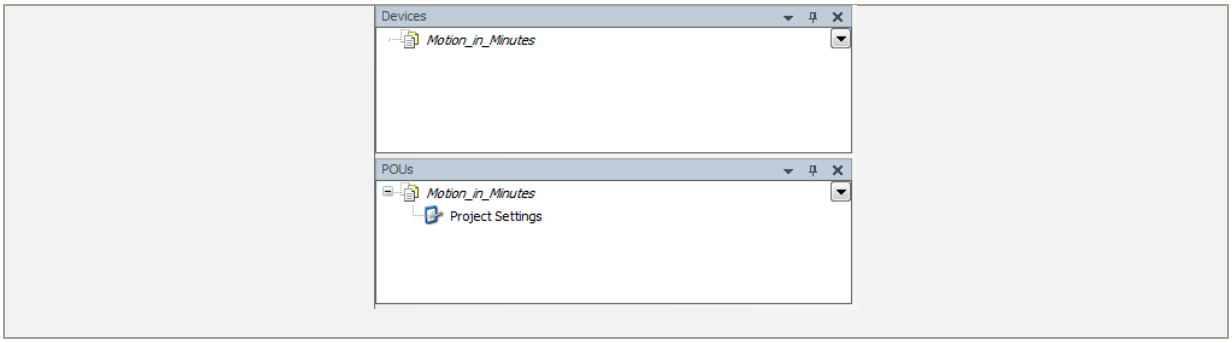
4. Select "Empty project" and hit "Ok"

Setting up your PAM Environment
If you a first time PAM user, we recommend having both the Device and POUs window visible. It is also helpful to have them stacked on top of another to enable easy click-and-drag functionality between them. We'll be using this functionality later in this walk-through guide.

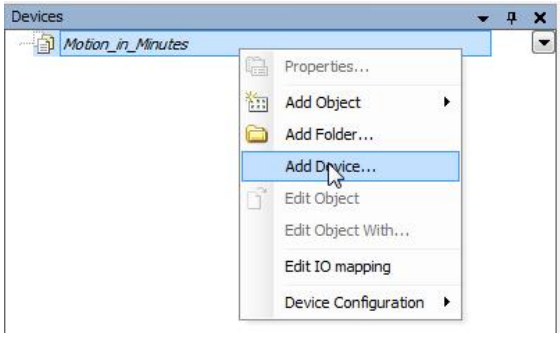
To make sure these windows are visible to go "View -> Devices" and "View -> POUs".

You can click and drag the windows to "snap" them into whatever placement you prefer. The Tac icon in the upper left hand toggles the window between Visible and Auto-hide modes.

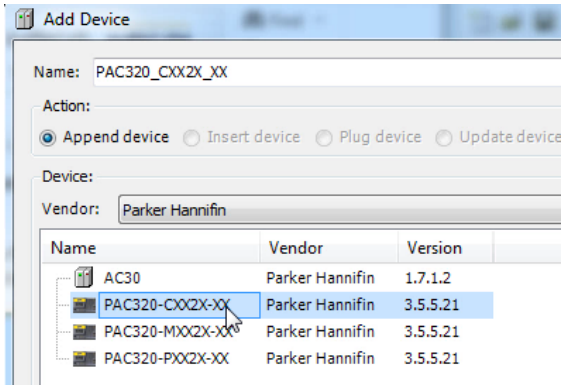
Below is an example of the recommended first-time user layout:



5. Since the user wants to program the PAC it needs to first be added into the system, right click the project and select "Add device"



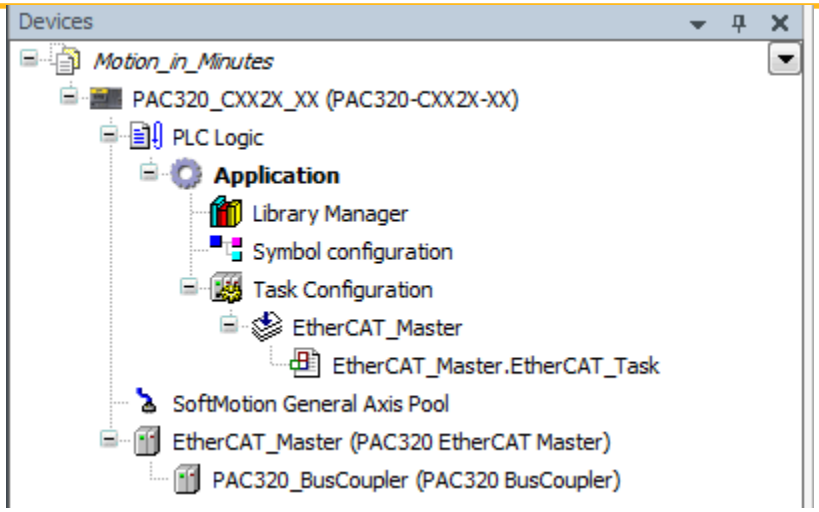
6. A new window will pop up allowing users to select both the vendor and pre-loaded product types.



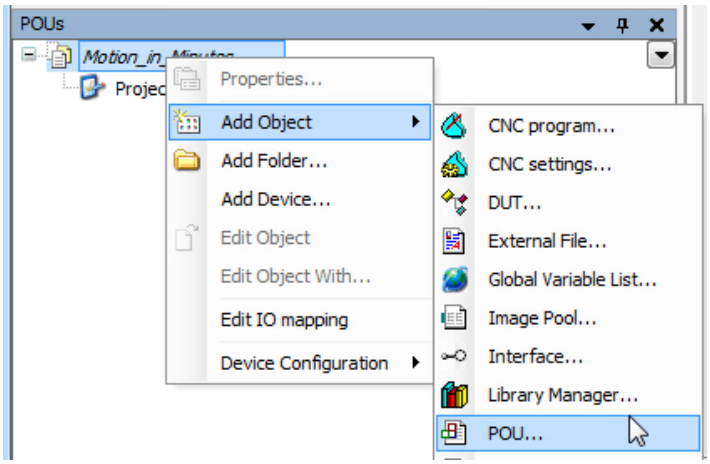
7. Select the PAC model that matches what you intend to program.

No devices needed for development
This tutorial first covers how to work with physical devices available to the user. The last section of this tutorial discusses how to use the same program in Simulation Mode.

8. The device tree will automatically populate with pre-made tasks and hardware configuration menus that are necessary for configuring and programming the PAC.



9. Double click on *PAC320_CXX2X_XX (PAC320-CXX2X-XX)*. Several configuration menus and settings are provided here. Several of these tabs will be used later in the demo to set up our connection and other preferences.
10. Right click "Motion_in_Minutes." In the POUs windows. Select "Add Object" then "POU"

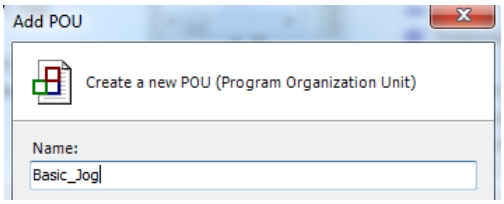


What is a POU?
POU stands for Programmable Organizational Unit. This can include, programs, function blocks, and functions. Each of these can be implemented in one of the several IEC 61131-3 supported languages: Structured Text (ST), Continuous Function Chart (CFC), Ladder (LD), FBD (Function Block Diagram), IL (Instruction List), and Sequential Function Chart (SFC).

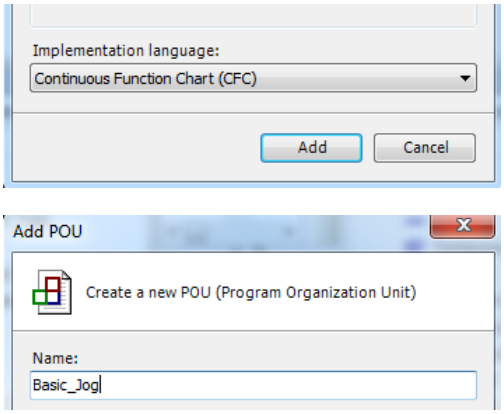
Each language has its own benefits and some work for certain programming functions better than others. For instance, SFC is typically used for creating state machines or where sequential programming is required, ST is perfect for recipe or data management or where math is required.

We chose CFC for programming this demo as it is a very visible language, some compare it to LabView. It is easy to debug as one can clearly see the overall function of the system without needing to know how to program, making it perfect for maintenance-level personnel. It is very easy to debug as it's clear what "inputs" are needed to provide a specific "output." Many prefer to do aspects motion programming in this language as technicians are often responsible for debugging motor issues and faults.

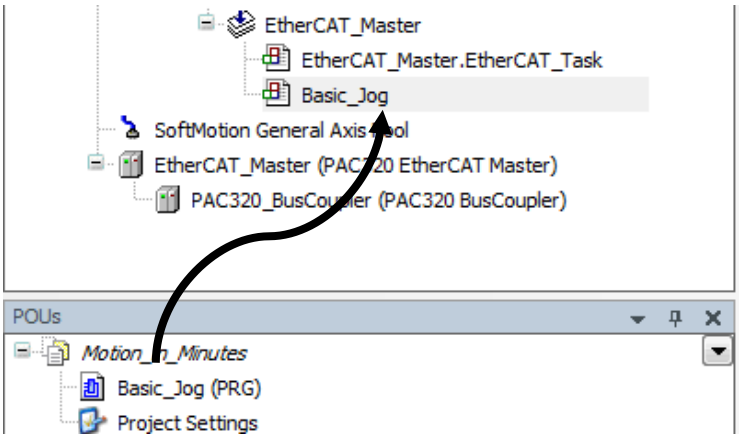
11. Name your program "Basic_Jog" and then select "Add".



12. A new window will appear asking the user to select the type of program. Select "CFC – Continuous Function Chart".



13. Click and drag the Basig_Jog program under the EtherCAT_Master task



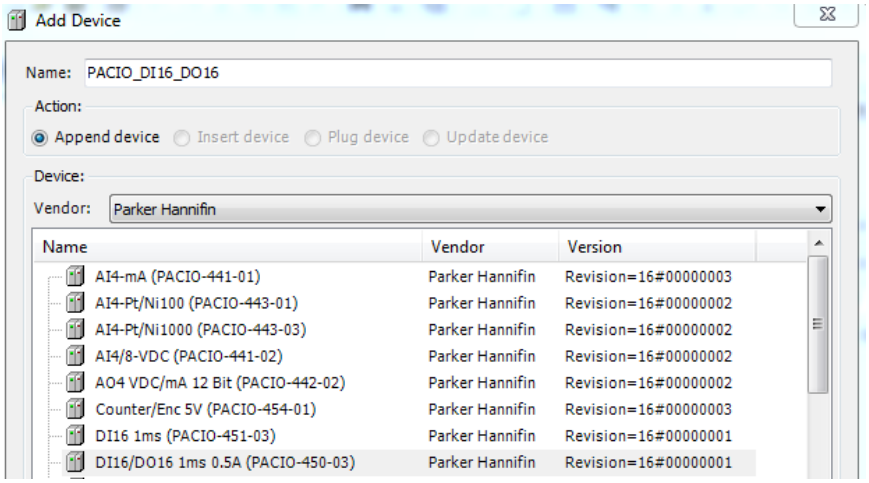
* **Note that the POUs and Device Tree are separate. This Basic_Jog program is not apart of the Task, thus not a part of the application. This means that upon downloading the project to the controller, this program would not be apart of that download.**

By clicking and dragging it under the EhterCAT_Master task we are making a reference to the Basic_Jog program and will be included upon download.

The major benefits of organizing this way is that if you had multiple devices referencing the same program, function, etc a programmer would only need to edit it in one place in order for those changes to be made globally.

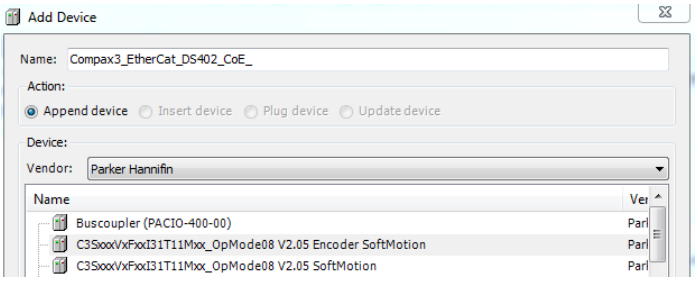
14. If you right click the PAC320-BusCoupler select "Add Device" the list will automatically populate with all the PACIO modules currently available (assuming you're using the latest version of PAM).

Remember!
EtherCAT_Master (PAC320 EtherCAT Master) acts as a representation of the actual RJ-45 EtherCAT port. Whereas the PAC320_BusCoupler (PAC320 BusCoupler) is a representation of anything connected to the E-bus connector such as PACIO.



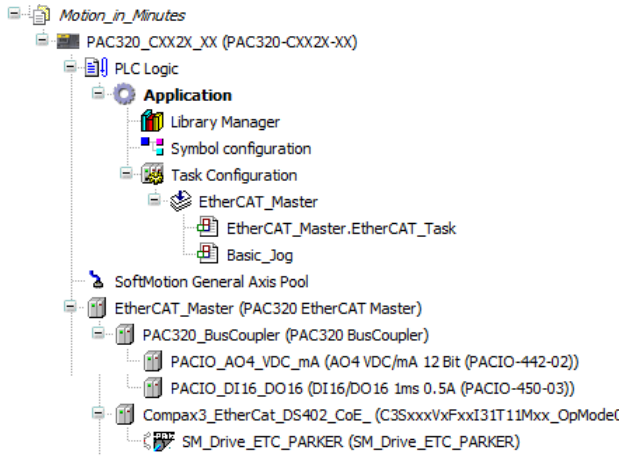
15. Add the PACIO attached to your unit by selecting the part number and clicking "Add Device." In this demo we used PACIO_DI16_DO16 as it will be used in our program later. (We'll go into how to use the "Scan" function later on which makes adding PACIO even easier.)

16. Lastly, right click "EtherCAT_Master (PAC320 EtherCAT Master)" and select "Add Device". A new window will open and a list will fully populate with all the EtherCAT slaves currently supported by Parker:



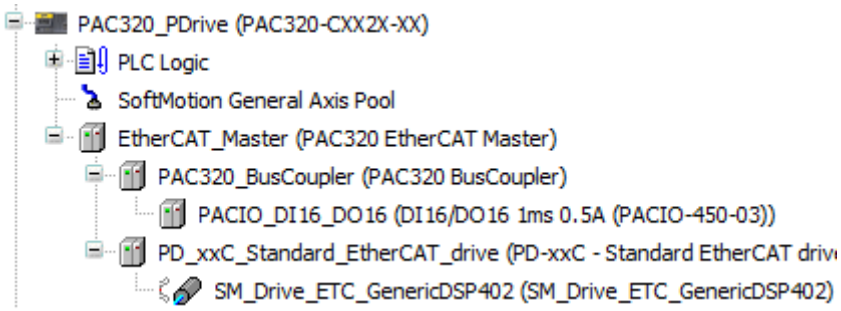
Adding a Compax3

17. If you have a P-drive skip to Step 20
18. Select "C3SxxxVxFxxI31T11Mxx_OpMode08 V2.05 Softmotion" and Add Device. This is representative of the Compax3 Drive.
19. After adding the appropriate devices in your system, your device tree should be similar to the one listed below:



Adding a P-drive

20. Select "PD-xxC – Standard EtherCAT Drive (CoE,EoE,FoE)" and Add Device. This is representative of the P-drive.
21. Close out of the "Add Device" window.
22. Right click the newly created "PD_xxxC_Standard_EtherCAT_drive) and select "Add SoftMotion CiA402 Axis"
23. After adding the appropriate devices in your system, your device tree should be similar to the one listed below:

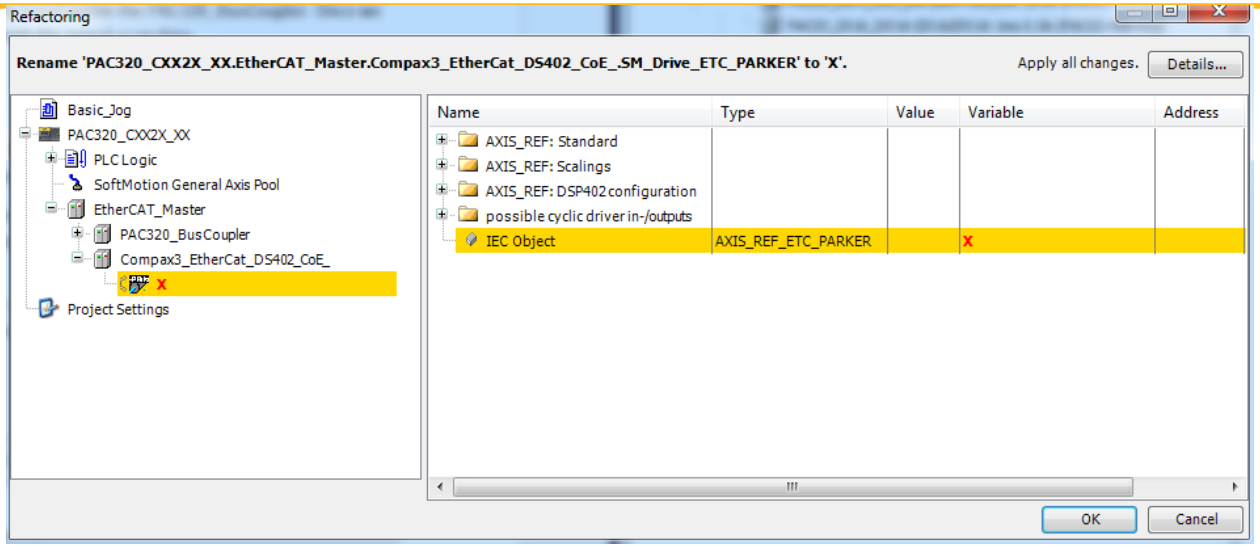


Writing the Program

In this program we'll create a basic jog controlled by the digital inputs of PACIO_DI16_DO16.

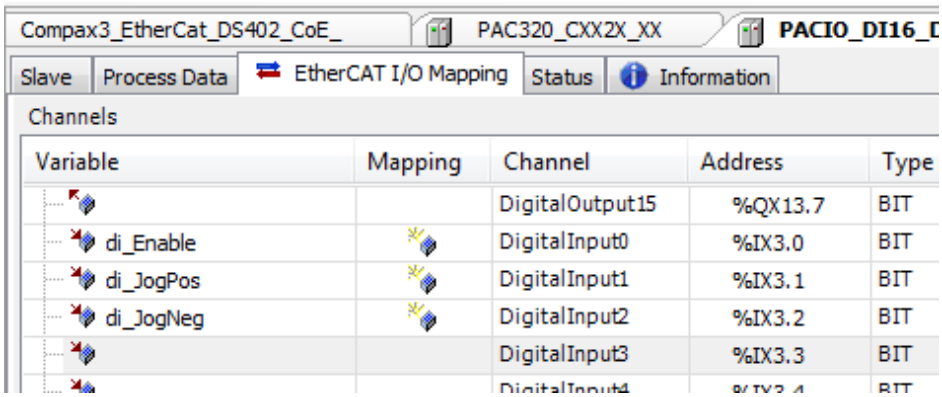
1. First, let's rename our motor axis so it's easier to call in our program, slow double click "SM_Drive_ETC_Parker…" and rename "X". A new window will pop up asking if you want to Refactor.
2. Select "Yes". The PAC will indicate all areas where the name was changed. In this case we only are using it in one place:

Refactoring?
Refactoring scans all programs and dependencies that use the changing variable and renames it automatically. This makes it so users don't have to manually comb the program and make the changes yourself.



3. Select "OK".
4. Double click "PACIO_DI16_DO16" Select the EtherCAT I/O Mapping tab.
5. This is where we will tie our inputs to specific variable names to use them in our program. Use the following names:

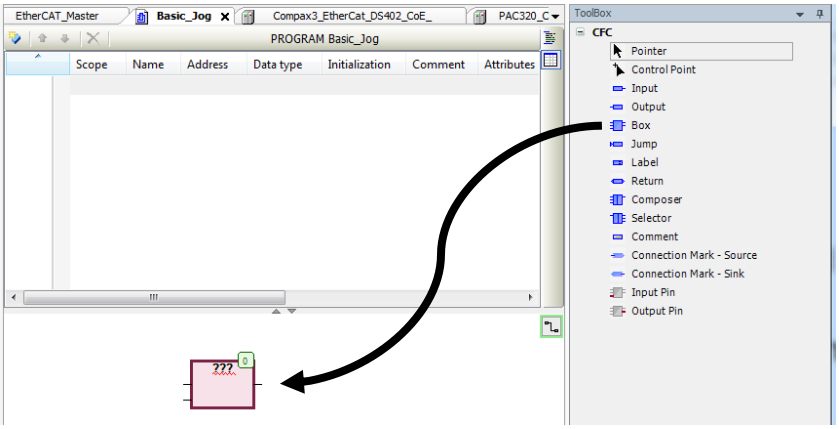
Channel	Variable	Program Purpose
DigitalInput0	di_Enable	Enables the drive/motor
DigitalInput1	di_JogPos	Input will jog the motor in the positive direction
DigitalInput2	di_JogNeg	Input will jog the motor in the reverse direction



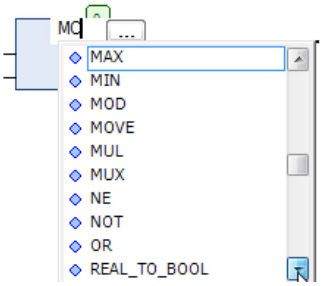
Device names and I/O variables are GLOBAL. This means that we do not need to declare them as we begin programming below. If we did—these local variables would override the I/O variables and our program would not be able to utilize the digital inputs.

6. Open up the Basic_Jog program by double click on it. (Remember, you must do this from the POU window as the Jog_Basic under our EtherCAT_Master task is just a reference!)
7. This will open up a programming space.

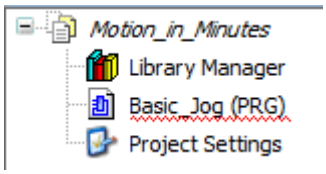
8. From the newly-opened Toolbox window, Click and drag a Box into the programmable field:



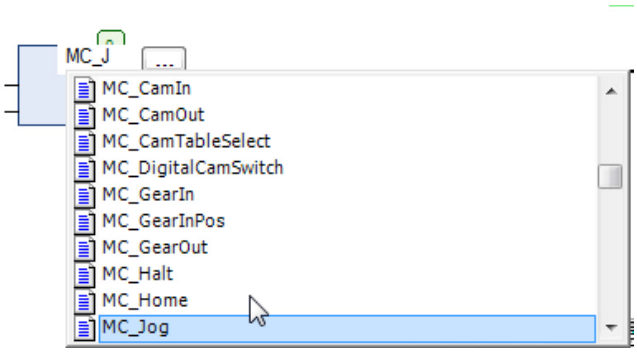
9. Go ahead and start typing "MC_" a box will automatically pop up with matching functions. Nothing should "MC_..."



10. The reason is that our program has no libraries that it's referencing. Copy and paste the "Library Manager" from the "Application" to our POU tree:



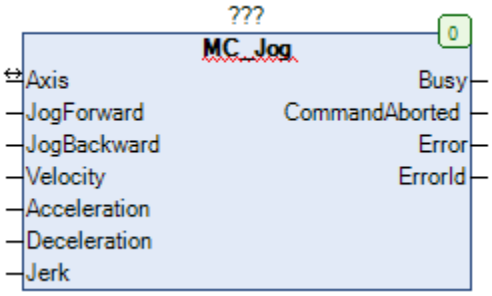
11. Now re-type "MC_J" note many motion related functions are automatically recommended helping the programmer. In this case select "MC_Jog".



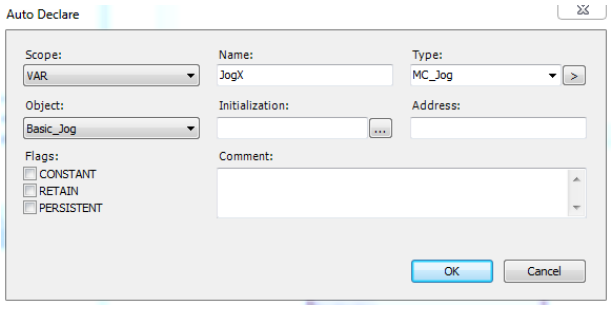
Still not working?
If you're working from a fresh install of PAM you may not have the correct auto-complete settings enabled.

From the file menu go to "Tools" -> "Options" In the options window go to "SmartCoding". Be sure that every item under SmartCoding is checked and click "OK". Try again.

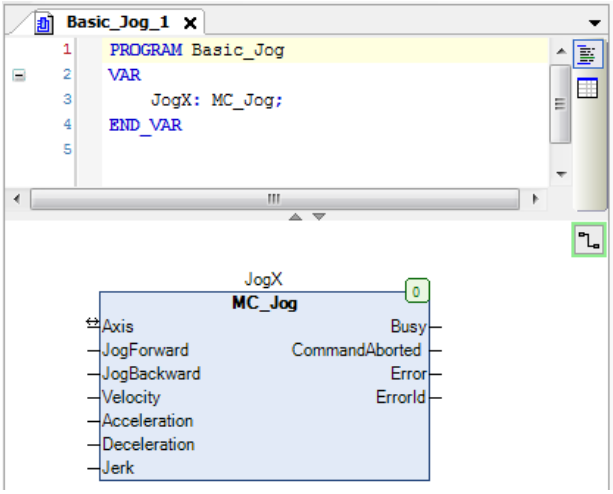
12. The box will transform into the MC_Jog function with required function block inputs and outputs with small leads, shown as lines, used to connect it to other function blocks or variables, etc.:



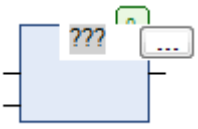
13. Slow double click on the ??? and name our function block "JogX". Since this variable has not yet been declared, a new Auto Declare window will appear asking to declare the variable. Select "OK"



14. The variable now populates on the variable list:



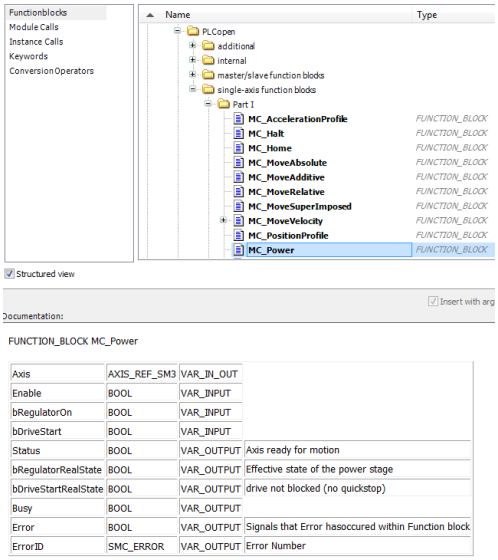
15. Click and add another box to the programming window. Instead of manually typing "MC_..." select the "..." next to the "???" text box.



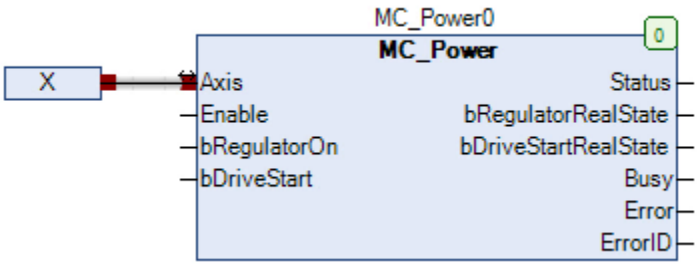
16. This offers an alternative way to find function. You'll note the same libraries in our Library Manager is listed. Open the Categories Tab and open the tree:

"Function Blocks" -> "SM3_Basic" -> "POUs" -> "PLCopen" -> "single-axis function blocks" -> "Part I" and select "MC_Power"

The documentation window automatically populate information based upon the selected function block helping the users determine what function block is needed to solve their application.

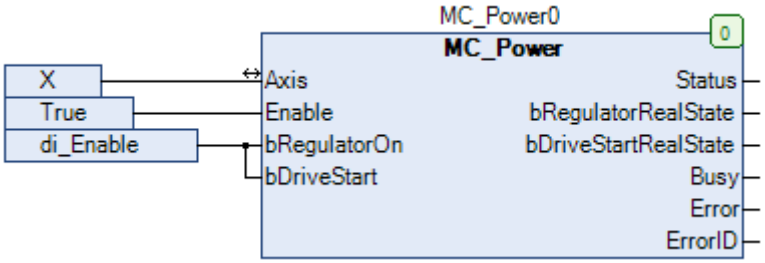


17. Select OK and the function block will be added to our program. Name it "MC_Power0" and allow the program to auto declare it.
18. Click and drag an input to the program and slow-click in the text box and name it "X". Click and drag the lead from "X" to the "Axis" leadoff MC_Power0.



This selects the "X" axis, or rather, the motor connected to the Compax3 is the motor that will be controlled by this function block. NO NEED TO DECLARE "X"! Like our I/O, it's already a "global variable" that is tied to the motor either on our Compax or P-Drive.

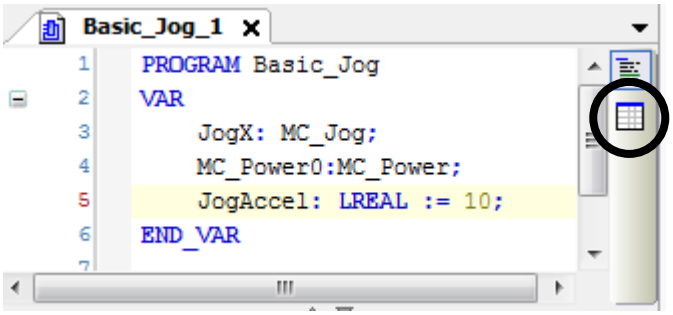
19. Complete the rest of the function block as shown below:



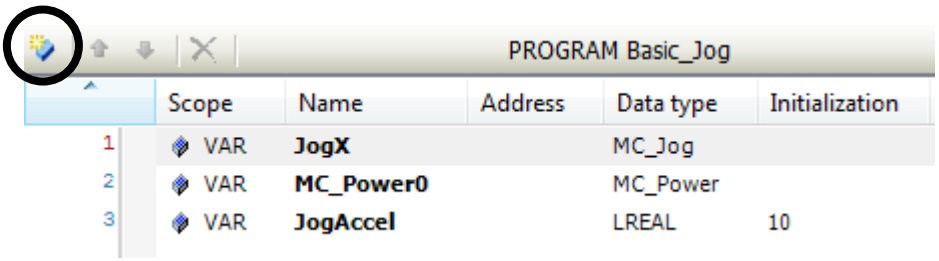
20. The JogX function requires several variables that have not been created yet and are not global. Type "`JogAccel: LREAL := 10;`" in the variable window above your program:

```
1  PROGRAM Basic_Jog
2  VAR
3      JogX: MC_Jog;
4      MC_Power0:MC_Power;
5      JogAccel: LREAL := 10;
6  END_VAR
```

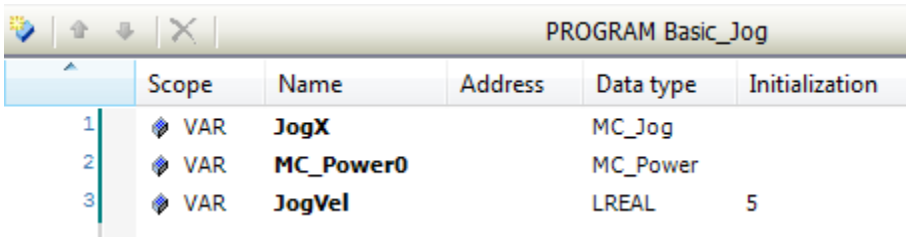
21. Alternatively, variables can also be entered in using a tabular format. Use the "Tabular" button in the upper right corner of the variable window to expose this option.



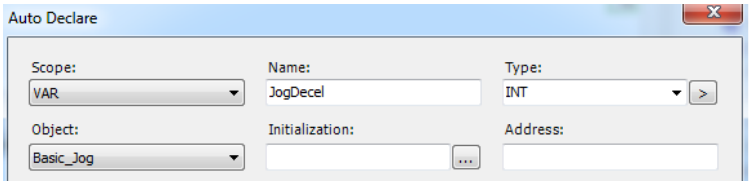
22. To create a new variable, select the "Insert" button in the variable list above the program window.



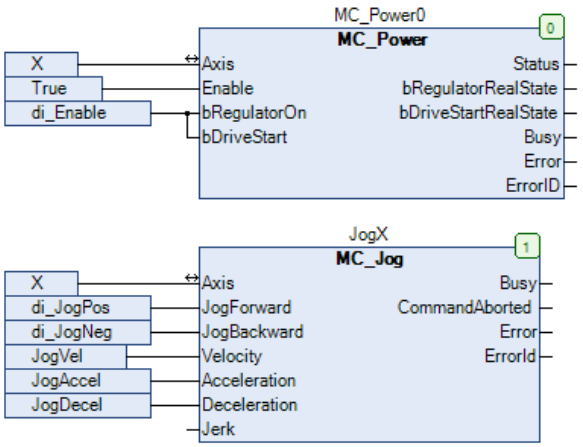
23. Name the new variable "JogVel" and set the initial value to 5. Change the input type to an "LREAL" or a long real.



24. For the last variable we'll use one more method of declaration. Click and drag an input onto the program. Start typing the desired variable name, in this case, "JogDecel" after pressing enter, an Auto Declare box will open.



25. Set the data type to LREAL with the initial value of 10.
26. With your newly minted declarations, complete the program. IT should look like the following function blocks below:



*** REMEMBER, Device names and I/O variables are <u>GLOBAL</u>. Do NOT declare X, di_JogPos, di_JogNeg, di_Enable in your program.**

Don't have an I/O Module?

THEN you'll want to declare di_Enable, di_JogPos, and di_JogNeg as BOOL. In this case we'll "force" these variables (as shown in "Simulation Mode" pg. 16-19.) No need to jump to this step yet.

Important Configuration Settings

Configure the EtherCAT_Master Task

1. In the Devices tree, double click the PLC Logic-> Application-> Task Configuration -> EtherCAT_Master node to bring up the Task Configuration dialog.

2. In the Configuration dialog, change the Type to be External and the External event to be EVENT_ECM_DC. Ensure that the Priority is set to 0.

What are we doing here?
In theory, our project can run using one of two clocks, either the clock in the real time operating system OR a higher precision external clock (sometimes referred to as a boundary clock). Because of the precision required for EtherCAT and especially motion control applications, the external clock should always be used. Priority 0 puts this task as the "highest priority task and cannot be interrupted by other tasks or events. Thus, all our motion critical and high precision tasks should be here.

Protip: Don't set the update interval value here. It's better to do it in the EtherCAT_Master (PAC320 EtherCAT Master) settings. The reason is that the EtherCAT_Master (PAC320 EtherCAT Master) interval setting automatically changes the interval setting here—and they must match.

Configure EtherCAT_Master Node

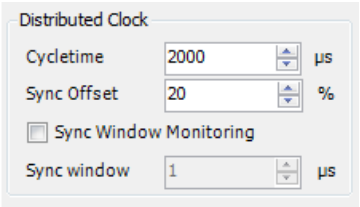
3. In the Devices tree, double click the EtherCAT_Master node to bring up the EtherCAT_Master Configuration dialog.

4. Ensure that the Autoconfig Master/Slave checkbox is checked.

5. Change the Distributed Clock Cycletime to the EtherCAT cycle time that you wish to achieve. (2000 usec in this example).

6. The rest of the settings can be left to default.

Cycle time is limited by slaves
For an EtherCAT network that includes a Compax3, the Distributed Clock Cycletime should not be lower than 500 usec and must be a multiple of 250 usec. A Distributed Clock Cycletime of 1000 usec is a reasonable starting point.

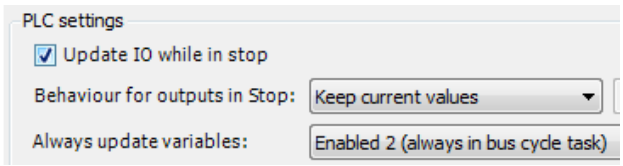


Configure PLC Settings

7. Double click on the PAC320_CXX2X_XX (PAC320_CXX2X_XX). Select the "PLC Settings" Tab.

8. Check "Update IO while in stop"

9. Change "Always update variables" to "Enabled 2 (always in bus cycle task)



How will these settings change the behavior of the PLC?

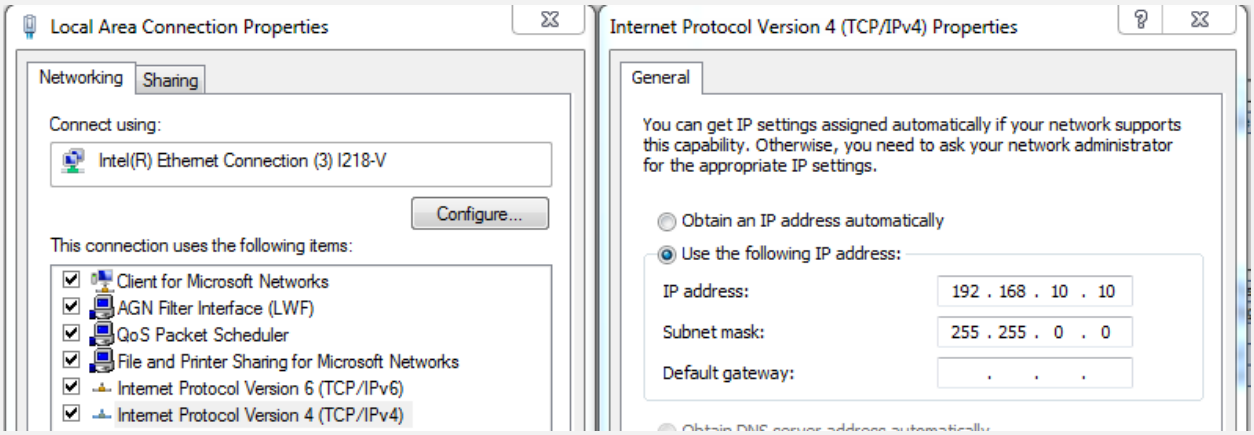
Although a preference, some users expect to see the IO update even with the PLC is in "stop" mode. For instance, this may help the user manually test if the inputs are wired up correctly by toggling them manually and seeing them change in the watch window. To ensure this behavior select "Update IO while in stop".

Typically it is undesirable to have all variables update every bus cycle as not all variables are in use at the same time. However, since the program we're writing has so few variables, the EtherCAT cycle will not be bogged down by selecting this option.

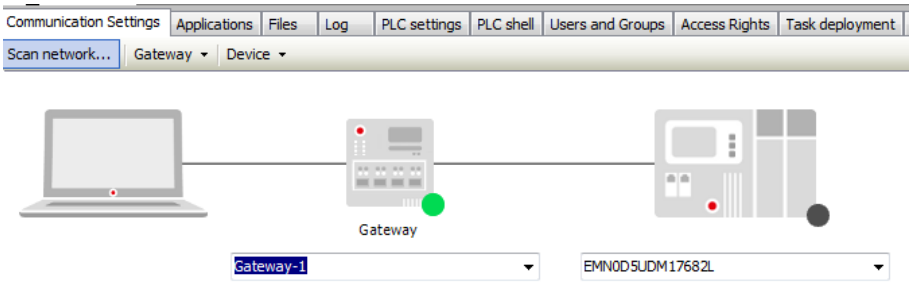
Setting up a Connection

Preparing your PCs Network Settings

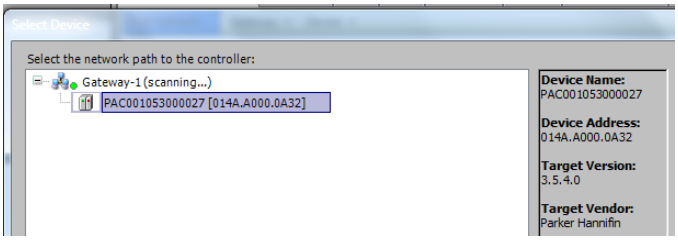
In order to connect with the PAC, the PC used for programming needs to have a static IP set up on their Ethernet port. The default static IP address of the PAC is 192.168.10.50 with a subnet of 255.255.0.0 The Ethernet port used to connect to the PAC needs to be in the same subnet, for example:



1. Double click on "Device (PAC320-MXX2X-XX)." Under the Communications Settings tab, select Scan Network.

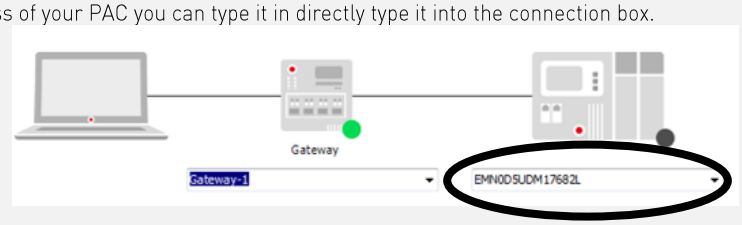


2. After the scanning process finds the PAC, select the PAC and click OK.



I can't find my PAC...

If you know the IP Address of your PAC you can type it in directly type it into the connection box.

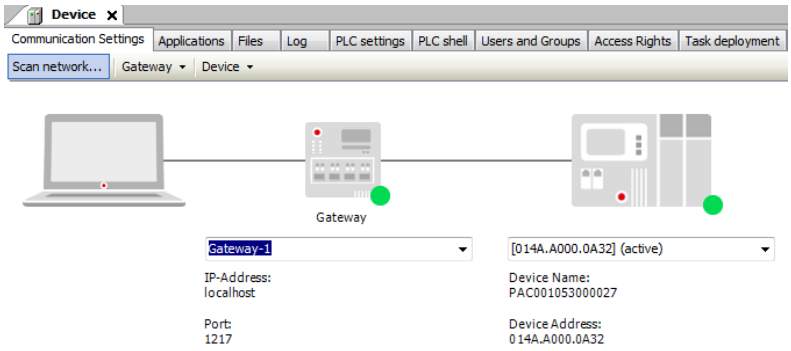


If you still cannot connect to the PAC, confirm that you can "PING" it using the Windows Command Prompt.

If you cannot receive replies form the PAC it may mean that either your Network Settings, your IT department, administration settings, or firewall are preventing connection.

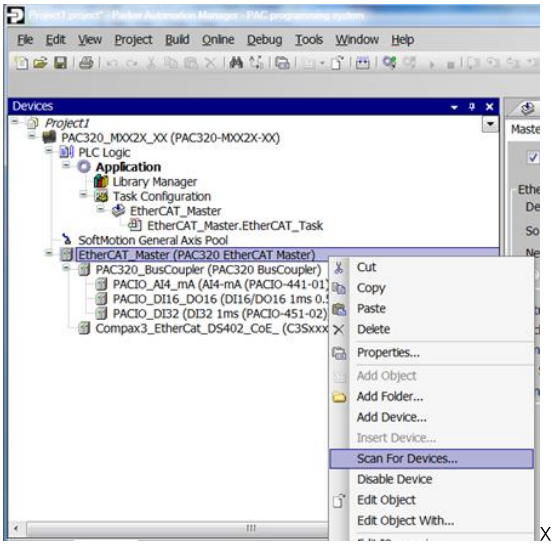
Lastly, the IP address of the PAC may not be what you expect. You can determine the IP address of the PAC by reading it form the SD card. Full instructions on how to do this are in the user manual.

3. The Green dot on the device indicates that the device was found and you are now connected to the PAC.



Scanning and Comparing EtherCAT Devices

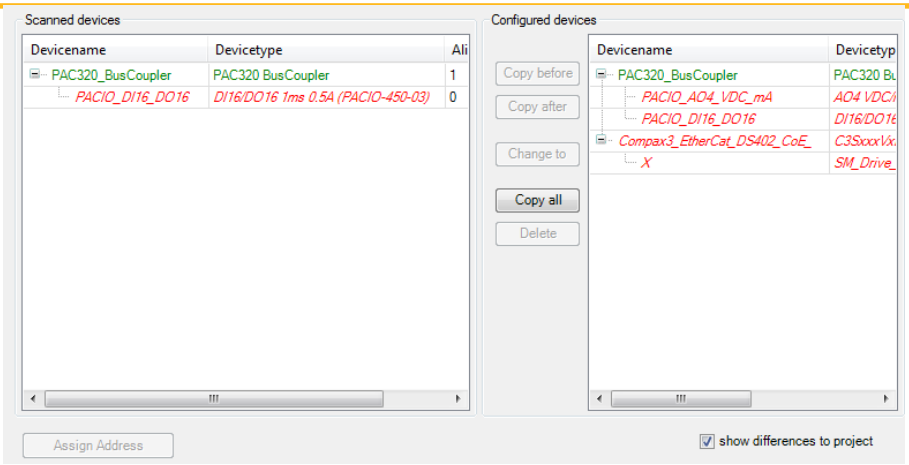
1. In the **Devices** tree, right-click the **EtherCAT_Master (PAC320 EtherCAT Master)** node and select **Scan For Devices...** This scans for all devices specifically on the EtherCAT port including PACIO, drives, or other EtherCAT devices.



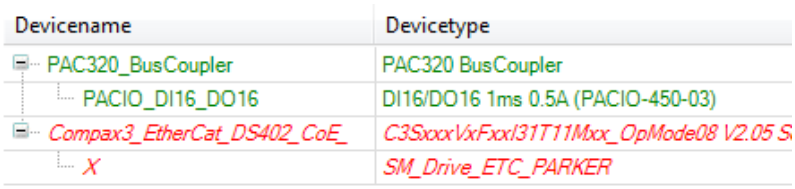
2. A "Scan Devices" dialog box will open showing all the devices available on the EtherCAT network.

My Scan Isn't Working
If the first connection since a flash back or connecting to the PAC for the first time you must first login before proceeding with scanning. This is due to the PAC not yet being programmed with the EtherCAT master configuration—this occurs on the first download.

3. In the lower right hand corner, select the "show differences in project". Differences between the devices listed in the project and what actual devices are on the network show up in red.



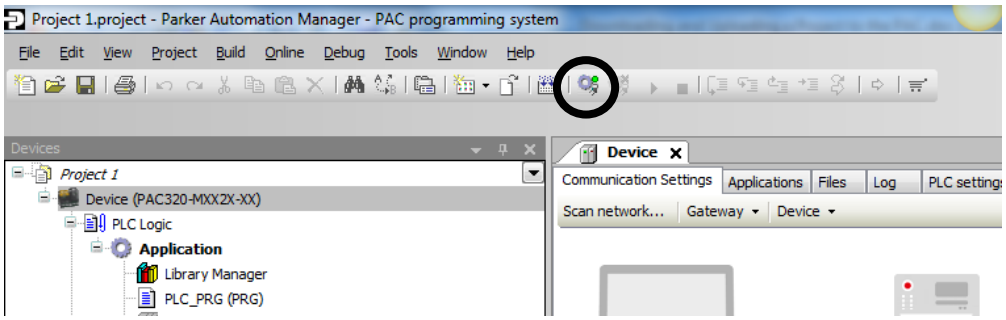
4. If you have added all your devices correctly the whole list would appear as green. For this example a PACIO_DI16_DO16 and Compax3 EtherCAT drive were used.



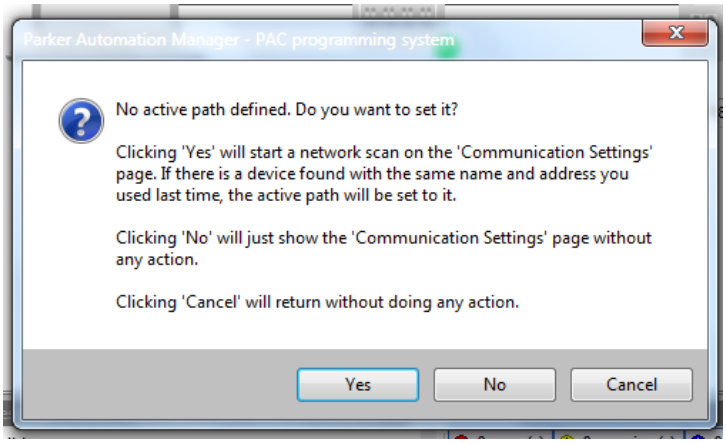
That's easier! Why didn't we just scan the I/O in the first place?
Previously we had manually added the I/O and Compax3 manually to our project. If you have a system that is already fully connected this step is unnecessary as you can simply Scan them in. The benefit of adding on modules manually is that one can begin programming before hardware has arrived.

Running the Demo

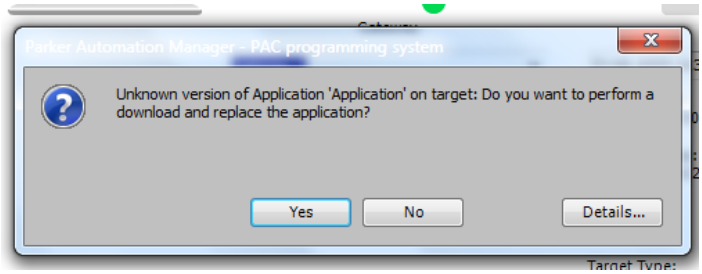
1. To download the project execution code, select the **Login** Icon on the Toolbar.



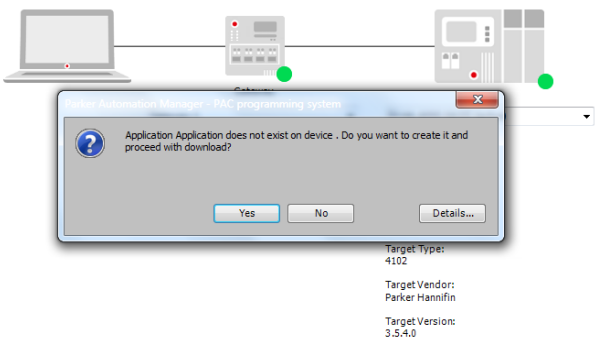
2. You may be prompted to find an active path. To have the PAC automatically scan, select **Yes**. To open the Communications Settings Page, select **No** and then proceed to scan the network for the PAC.



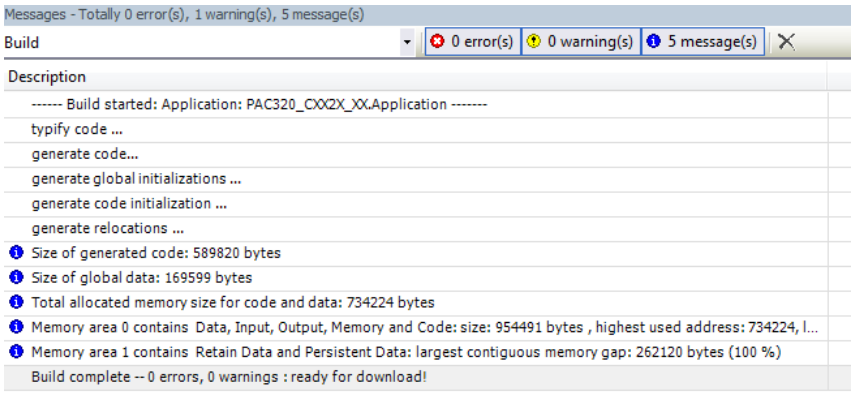
3. If there is a project already on the PAC and you want to replace it with this application, click **Yes**.



4. If you are downloading the project the first time to a PAC without a project, it may prompt you with a pop up window asking if you want to download the application, click **Yes**. The project is now downloaded to the PAC.



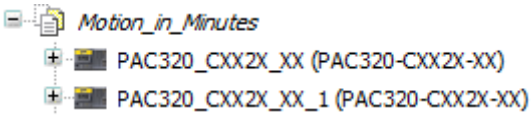
5. Lastly, if there are any errors in your program, they will be populated in the messages window. (Go to View -> Messages to open up this window)



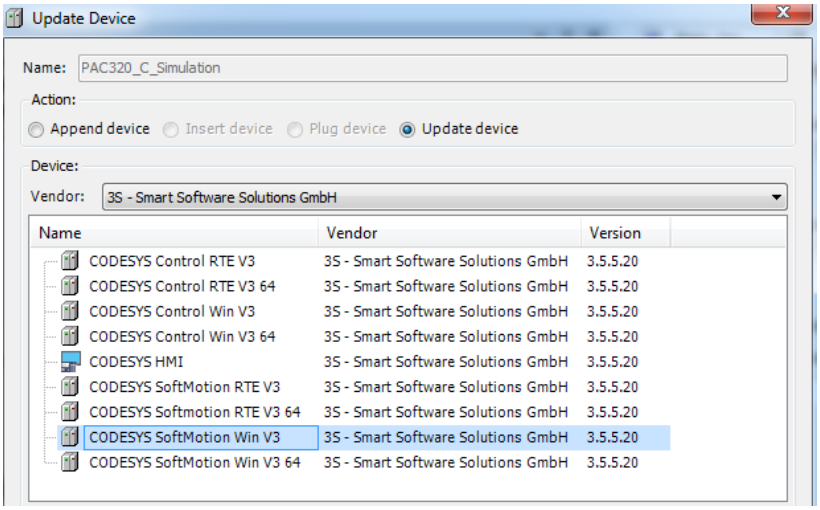
6. Press Start or use F5 as a short cut key to run the program.
7. If you have toggle switches hooked up to your demo, you are now able to control the motor with these inputs. Enable by toggling input 0, jog forward by toggling input 1 and reverse by toggling input 2.
8. If you do not have I/O—see Simulation Mode Steps 16-19 to obtain motion.

Simulation Mode

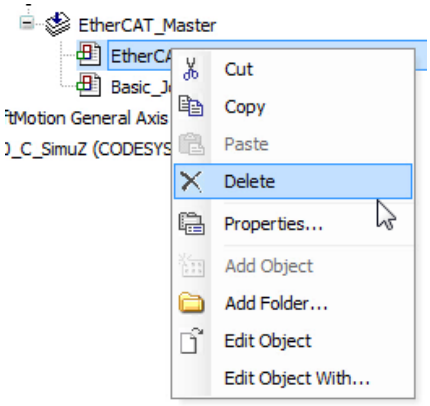
1. First, copy and paste a your PAC320-C device under the Motion_in_Minutes Root.



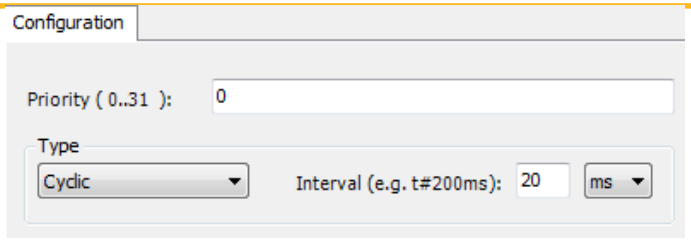
2. Slow-click to rename the copied device to "PAC320_C_Simulation". This will help to identify which device is being run in simulation mode and which utilizes actual devices on the bus.
3. Right click "PAC320_C_Simulation" and select "Upgrade Device".



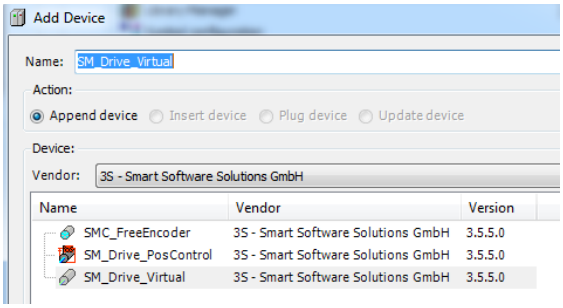
4. Specifically select the "CODESYS SoftMotion Win V3". And "Update Device". Close when the process is completed. The Device is now renamed to "PAC320_C_Simulation (Codesys SoftMotion Win V3).
5. As this is a simulation—three is no actual EtherCAT bus to be serviced. Delete the "EtherCAT_Master. EtherCAT_Task" under the EtherCAT_Master.



6. Continuing that logic—instead of using the external clock, the simulation will use the real-time operating system's clock to simulate the PAC. Double click the EtherCAT_Master and change from "external" to cyclic.



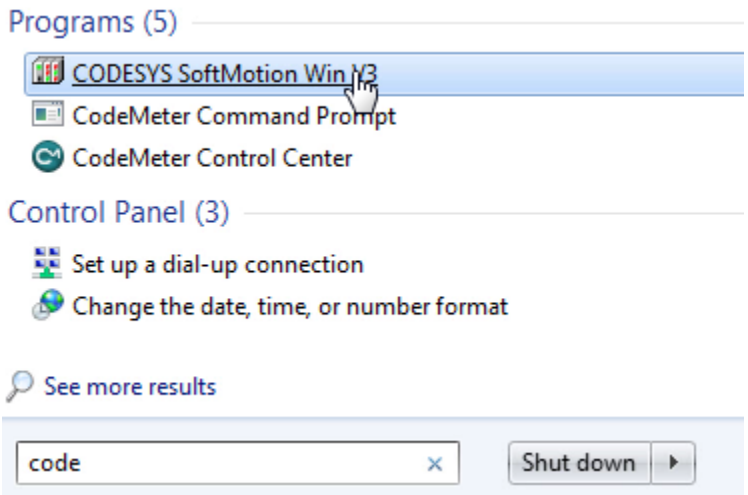
7. The embedded OS clock doesn't have the same capabilities as the external clock so the cycle time may needs to be increased. Set it to 20 ms.
8. The EtherCAT_Master (PAC320 EtherCAT Master) has been fully replaced by the "SoftMotion General Axis Pool". To add an axis to the simulation device, right click the "SoftMotion General Axis Pool" add the "SM_Drive_Virtual".



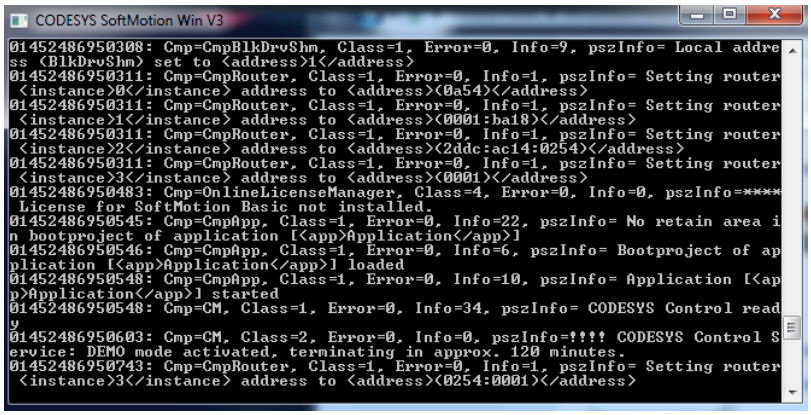
9. Once added, rename it "X" so that it is compatible with the Basic_Jog motion program already written.
10. Since no I/O is available our inputs can instead be simulated by forcing variables. Open up the Basic_Jog (PRG) and declare these additional variables in addition to the already existing ones:

Variable	Type	Initialization
di_Enable	BOOL	0
di_JogPos	BOOL	0
di_JogNeg	BOOL	0

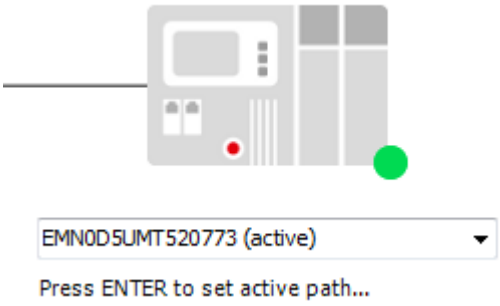
11. As there can only be one active application per project, right click the simulation mode application and "Set as Active Application".
12. To run in simulation mode, PAM needs to process PLC and motion behavior in the background. To do this open CODESYS SoftMotion Win V3. From Windows -> Start -> Programs menu.



13. A new window will execute various code. This window is working with windows in the background to simulate the PAC. **Minimize** this window, DO NOT close this window as the simulation cannot function without it.



14. Go back to PAM. Double click the "PAC320_C_Simulation" to open the Communication Settings. As there is no actual, physical PAC, change the connection to the name of your computer. This should be populated automatically. In the authors case, their computer name was "EMN0D5UMT520773." This step can be thought of as the PAM project connecting to the simulation processing in the background. A green light, like the one shown below implies that the connection is completed.



15. Login to the simulation in the same manner as real, physical connection to the PAC. Either select "Start (F5)" to run the application.
16. Open up the Basic_Jog (PRG) and open the variable declaration table. Both the program and this table are now "live" showing the current state of all variables.
17. To simulate toggling a switch, click on the "prepared value" table next to di_Enable and di_JogPos to set them to "true."

PAC320_CXX2X_XX_1.Application.Basic_Jog

Expression	Type	Value	Prepared value
MC_Power_0	MC_Power		
MC_Jog_0	MC_Jog		
JogVel	LREAL	5	
JogAccel	LREAL	10	
JogDecel	LREAL	10	
di_JogPos	BOOL	FALSE	TRUE
di_JogNeg	BOOL	FALSE	
di_Enable	BOOL	FALSE	TRUE

18. To force these values either go to the File Menu -> Debug -> Force Values or press "F7".

19. Double click on “X”. The Position [u] of the SoftMotion Drive: Basic should be increasing, showing our simulation is jogging in the positive direction.