Micro800[™] and Connected Components Workbench[™]

Application Guide



Table of Contents

Chapter 1: Flash Updating Micro800 Firmware Chapter 2: Importing and Exporting User-Defined Function Blocks Chapter 3: Creating a New Function Block Program Chapter 4: Creating a New Structured Text Program Chapter 5: Using CCW with PanelView Component Chapter 6: Using CCW with PowerFlex Drives Chapter 7: Using CCW with Temperature Controllers

Requirements

Hardware Requirements:

Micro810, 2080-LC10-12QWB.

Micro830, 2080-LC30-16QWB

Micro830 Plug-In, 2080-SERIALISOL

Standard USB Cable

Software Requirements:

Connected Components Workbench (CCW), Release 1.0

RSLinx, v 2.57

Chapter 1 – Flash Updating Micro800 Firmware

Flash Updating Micro800 Firmware

This chapter will show you how to flash update the firmware in a Micro800 controller using ControlFLASH. ControlFLASH is installed or updated with the latest Micro800 firmware when Connected Components Workbench software is installed on your computer.

1. First verify successful RSLinx Classic communications with your Micro800 controller via USB using RSWho (Micro810 12-pt. uses the 12PtM810_xxxxx driver and the Micro830 uses the AB_VBP-x driver).



2. Start ControlFLASH and click Next:



- Catalog Number Enter the catalog number of the target device: 2080-LC10-12QWB Control 2080-LC10-12AWA 2080-LC10-12DWD ٠ FLAS: 2080-LC10-12QBB 2080-LC10-12QWB 2080-LC30-10QVB 2080-LC30-10QWB 2080-LC30-10QWB 2080-LC30-16AWB 2080-LC30-16QVB 2080-LC30-16QWB 2080-LC30-24QBB 2080-LC30-24QVB 2080-LC30-24QWB 2080-LC30-48AWB 2080-LC30-48QBB • < Back Next > Cancel Help
- 3. Select the catalog number of the Micro800 that you are going to update and click **Next**:

4. Select the controller in the browse window and click **OK**:



5. If you get the following screen (Micro810 only), leave the Slot Number at 0 and click OK:



6. Click **Next** to continue, verify the revisions, then click **Finish** and **Yes** to initiate the update:

Firmware Revision
Control
< Back Next > Cancel Help
Summary Provide the standard
 K Back Finish Cancel Help ControlFLASH
Are you sure you want to begin updating the target device?

7. The next screen should show the download progress:

Progress				
Catalog Number: Serial Number:	2080-LC10-12QWB FFFFFFFF			
Current Revision: New Revision:	1.2 1.4			
Transmitting update 2 of 6 block 317 of 2253				

8. If you get the following error message instead, check to see if the controller is faulted or in Run mode. If so, clear the fault or switch to Program mode, click **OK** and try again.

AB_ASA.	DLL
8	Failed to update firmware. Either the target device does not support Flash updates using this programming tool or the target hardware revision is not compatible with the selected version of firmware.
	OK Help

9. When the flash update is complete, you should get a status screen similar to the following. Click **OK** to complete:

Update Status	×
Catalog Number: 2080-LC10-12QWB Serial Number: FFFFFFF	ОК
Current Revision: 1.4 New Revision: 1.4	View Log
Status: Update complete. Please verify this new firmware update before using the target device in its intended application.	Help

Chapter 2 – Importing and Exporting User-Defined Function Blocks

Importing and Exporting User-Defined Function Blocks

This chapter will show you how to create and export a SIM_FB User Defined Function Block (UDFB) so that it can be imported into other projects.

1. Create a new Micro830 project.



2. Under Project Organizer, right click on Function Blocks, select Add then New ST :Structured Text:



3. Right click on UntitledST, select Rename and type in "SIM_FB":



4. Double click on SIM_FB and type in the following:



5. Below SIM_FB, double click on Local Variables and enter in the following:

G	<u></u>	M_FB-¥	AR SIM_FB-P	OU						
			Name		Data	Туре	Direc	tion	Dimen	sion
				- A*		- A*		- A*		- A*
		B_IN			REAL	-	VarInput	•		
		B_O	JT		REAL	•	VarOutput	-		
		i			DINT	-	Var	•		
		j			DINT	•	Var	-		
		🕂 Buffe	er		REAL	-	Var	-	[120]	

6. Right click on SIM_FB and select Build:



If you get any Build errors, correct the errors and Build again until you succeed with no errors.

7. Under Project Organizer, right click on SIM_FB, select Export and then Export Program:



8. Click Export:

Import Export		×
📑 Import Exchange File	눩 Export Exchange File	
Export Variables Only		
Set Password		
Password		
Password		
Confirm Password		
Element Exported Controll	er.Micro830.Micro830.SIM_FB	3
	[Export Cancel

9. Browse to the saving folder location and click Save:

Save As						? ×
Save in:	🗀 Lab Files		•	3 🤣	• 🖭 🥙	
Recent Desktop My Documents My Computer	Controller.Micr	o830.Micro830.TRAFF	IC_CONTROLLE	R_FB.7z		
My Network	, File name:	Controller.Micro830.N	4icro830.SIM_F	B.7z	-	Save
	Save as type:	SevenZip files (*.7z)			•	Cancel

10. To use the SIM_FB in a future project, create a new project and right click on Micro830 under Project Organizer, select Import, then Import Exchange File:



11. Click **Browse**, navigate to the folder location, select the file and click **Open**:

Import Export	×
📸 Import Exchange File	
File Name	Browse
Select Elements to Import	
Name	Select All
	Deselect All
Destination Controller.Micro830	
Import	Cancel

Select Import Ex	change File					<u>? ×</u>
Look in:	🗀 Lab Files		•	G 🦻	• 🖭 👏	
C Recent	Controller.Micro	0830.Micro830.SIM_FB.72 0830.Micro830.TRAFFIC_	CONTROLLE	R_FB.7z		
Desktop						
My Documents						
My Computer						
My_Network	File name:	Controller.Micro830.Micr	0830.SIM_F	B.7z	•	Open
Places	Files of type:	Exchange files (*.7z; *.zi	p)		•	Cancel

12. With **SIM_FB** checked, click **Import** and verify in the **Output** window that the import was successful:



13. Click **Cancel** to close the Import Export screen.

Chapter 3 -Creating a New Function Block Program

Creating a New Function Block Programming

This section will show you how to create a new function block program. In this function block program, the PID standard function block will be used. A User Defined Function Block will be imported to simulate the process value.

1. Start the Connected Component Workbench from the Start Menu: Start →All Programs → Rockwell Automation → CCW → Connected Components Workbench.





Alternatively, double click on the shortcut on the Desktop

2. At the Connected Component Workbench window, drag **2080-LC30-16QWB** from the **Device Toolbox Catalog** window into the **Project Organizer** window - a new project will be created.



3. In the Name field within Project Organizer, enter FBD_Program

Project	Organizer	👻 🕂	×
Name:	FBD_Program		

4. Under Project Organizer, right click on the Programs select Add and select New FBD: Function Block Diagram.



5. Right click on UntitledFBD and select Rename:



6. Type in Process_SIM and Enter:



7. Right click on the **Micro830** in **Project Organizer** and from the popup menu select **Import** → **Import Exchange File** as shown.



8. The **Import/Export** Window will appear, browse for the file Micro830.Micro830.SIM_FB.7z. Select **SIM_FB**, and press **Import** to import the file. Then, close the window. Note: If you don't have SIM_FB, refer to the previous chapter for details on how to create this user defined function block.

Import Export	×
👔 Import Exchange File	
File Name C:\Lab Files\Micro830.Micro830.SIM_FB.7z Select Elements to Import	Browse
Name	Select All
✓ SIM_FB	Deselect All
Destination Controller.Micro830.Micro830 Import	Cancel

9. The Function Block, SIM_FB will be imported into the Project Organizer.



The contents of the SIM_FB Structured Text program is as follows:



10. Double click on **Process_SIM** within the **Project Organizer** to start editing the Function Block Program.

- 💐 Connected Components Workbench File Edit View Build Debug Format Tools Communications Window Help 🍟 🕍 🕨 🗢 🕾 🗊 🖓 🎭 🧌 関 🐻 🆓 🦓 🖉 Cycle Timing (ms): - B I U * A E = = | H . Project Organizer **→** ‡ × Process_SIM-POU 🗸 🗙 Toolbox • 4 × 🖃 FBD Name: FBD_Program Pointer Micro830 - Variable Toolbox for FBD 🗗 Block 🖃 --- 🌅 Programs -≫ Jump programming L Process_SIM -O Return 🖆 Comment Local Variables 👁 Label He Rung Global Variables - Left Power Rail DataTypes Programming workspace - Right Power Rail 🛨 Vertical Bar Function Blocks -O- Direct Coil -Ø- Reverse Coil SIM_FB -S- Set Coil Local Variables -®- Reset Coil - I ⊢ Direct Contact +/- Reverse Contact -IP- Pulse Rising Edge Contact HH- Pulse Falling Edge Contact 🖃 General There are no usable controls in this group. Drag an item onto this text to add it to the toolbox. Project Organizer Device Toolbox 📯 Toolbox Ready
- 11. Function Block Diagram (FBD) Programming Toolbox is required for programming.

12. The following program logic will be developed.



- The Average Function Block will be used as the sampling rate for the analog input simulation.
- The PID Function Block will be for producing a Control Value (CV) that results in the Process Value (PV) tracking the Setpoint Value (SV).
- The SIM_FB is a simulator block using the concept of FIFO, delaying the feedback to the PID function block.

13. Double click on the Local Variables in the Project Organizer under the Process_SIM,



14. Enter the following variables into the Process_SIM-VAR Tab.

Name	Data Type	Initial Value
sv	REAL	10.0
FB	REAL	0
PID1_G	GAIN_PID	-
PID1_AT	AT_PARAM	-
AUTO_RUN	BOOL	-
INIT	BOOL	-
PID1_AT_EXEC	BOOL	-

Upon completion, the variables table should be as follows:

4	Pr	rocess_SIM-VAR Process_SIM-POU*										
			Name	Data Type		Dimension	Alias	Comment	Initial Value			
			- A	*	A*	- A*	- A*	<i>▼ A</i> *	- A*			
	Þ		SV	REAL	•				10.0			
			FB	REAL	*				0.0			
		Ð	PID1_G	GAIN_PID	-							
		Ŧ	PID1_AT	AT_PARAM	*							
			AUTO_RUN	BOOL	-							
			INIT	BOOL	*							
			PID1_AT_EXEC	BOOL	-							

15. Double click on the **Process_SIM**, the programming workspace will appear.



16. Select **Block** from the Toolbox and drag into the Programming Workspace



17. The Instruction Block Selector window will appear.

6004	M Instruction Block Selector: N/A									
r	Controller : 2080LC3016QWBA									
			Name	0	Category	1	Туре	<u> </u>		
		Y	•			▼ A*	▼ A [*]			
		-			Arithmetic		õP	Substraction of two or more		
	× Arithmetic				Arithmetic		ā	Multiplication of two or more		
		1			Arithmetic		öP	Division of two or more inte		
		+			Arithmetic		5	Addition of two or more inte 🚩		

18. Select the **Average** function block from the pull down menu.

🔤 In	🖻 Instruction Block Selector: AVERAGE								
Co	Controller : 2080LC3016QWBA								
	Name 🙎	Category	1	Туре					
	AV ASIN ASIN ASIN_LREAL ATAN ATAN_LREAL AVERAGE	▲ a Manipulation	▼ 0 ^{R*}	▼ A*	Running average over N sample				
	AWA AWT CHAR								

19. The instance AVERAGE_1 will be created, click **OK** to proceed.

🔤 Instru	🔤 Instruction Block Selector: AVERAGE								
Controlle	r : 2080LC3016Q\	WBA							
AVE	Name 🙎 AVERA 🔻 🖋 ERAGE	Category Data Manipulation	<mark>זער דער דער דער דער דער דער דער דער דער ד</mark>	e Af Runni	ing average over N sample				
Raramet		m			>				
Paramet	Name	Data Type	Dimension	Alias					
	v a Sum Size Index Init Ⅲ	REAL · SINT · SINT · BOOL ·	* #1	Sum Size Inde Inite	Sum of all values Size of the fifo Index in the fifo				
Instance: Inputs: Scope:	AVERAGE_1		▼ V Sh	ow Param	oK Cancel				
				<i>AVERA</i> (AVERA IN	GE_1 AGE XOUT				

The function block will appear in the workspace.



20. Select **Block**, and drag another block into the program workspace.

21. Select IPIDCONTROLLER function block from the pull-down menu.



22. The Instance IPIDCONTROLLER_1 will be created.

6004	🔤 Instruction Block Selector: IPIDCONTROLLER 📃 🗖 🔀											
	Cont	rolle	r : 2080LC3016	QWB4	4							
			Name 🔒		Cal	egory	1	Тур	e			
			IPIDCO 🔻 🖋		_		▼ A*	v 0	A*			
		IIPIL	DCONTROLLE	Proc	cess Co	ntrol		Ċ.	Propo	tional Integral	Derivativ	/e.
	1											
												2
F	Para	met	ers									_
			Namo		Data	Tupe	Dimor	sion	Aliae		_	
			Name	A*	Data	- A*	Dimer	- A *	- A			
		Ŧ	RESERVED_IF	PIDC	SINT	*	[1184]		RDIC	Reserved pa	rameter	
			Process		REAL	•			Р	Process Valu	ie	
			SetPoint		REAL	*			SP	Set Point		-
	4				BEAL	*			FR	FoodRack	•	
				0115	D 1			C Ch	ou Dorom	otoro		
Inst	anc	e:	IFIDCONTR		.n_1			✓ JU	uw raiain	eters		
Inpu	uts:		9 9									
Sco	pe:		Process_SIM	4								
										ок	Cano	cel
									_			.:



23. The function block will be shown in the programming workspace.



24. Select **Block**, and drag another block into the program workspace.

25. Then select **SIM_FB** function block from the pull-down menu.

80	Ins	struction Blo	ock Se	lector: N/A			
ſ	Con	troller : 2080LC	3016QV	VBA			
		Name	0	Category	1	Туре	<u> </u>
		4	A.		▼ A*	▼ A [*]	
		RTC_SET	ŀ	Anmetic		ÖP	Substraction of two or more
		ISLALER		hmetic		ōP	Multiplication of two or more
		SHR		hmetic		ÖP	Division of two or more inte
		SIM_FB		nmetic		5	Addition of two or more inte 🚩
	<	SIN_LREAL					

26. The Instance SIM_FB_1 will be created.

🔤 Ins	M Instruction Block Selector: SIM_FB								
Cont	roller : 2080LC	3016QWB	A						
	Name	0	Cal	egory	1	Туре			
	SIM_FB	▼ A*			▼ A	* 🔻 🖪	i e		
	SIM_FB	(Us	er defin	ed)					
									>
Para	meters								
	Nam	ie	Data	Туре	Dime	ension	Alias		<u> </u>
		- A*	•	·* A*		▼ A*	- A*		=
	i		DINT						
	i		DINT	•					
	🕂 Buffer		REAL	•	[020]				-
	RIN		BEAL						
Instanc	e: SIM_FI	B_1			*	🔽 Sho	w Param	ieters	
Inputs:	\$1								
Scope:	Proces	s_SIM							
								ОК	Cancel



27. After completing Steps 15-26, the programming workspace should have 3 function blocks as shown below.

28. Select the **Variable** from the Toolbox, and drag to the programming workspace. Connect it to the **SetPoint** of IPIDCONTROLLER_1 Function Block as shown below:



29. Then select SV from the Local Variable-Process_SIM, to assign to the Setpoint of the IPIDCONTROLLER_1.

🔜 Va	🗏 Variable Selector									
SV Use	ne r Global Variab	oles - Micro830	Type REAL	ariables - Proces	. :s_SIM	Global Scope Micro830 System Variables	• Micro830 1//	Local Scope Process_SI	M	
	Name	Data T	une	Dimension	Alias	Initial Value	Attribute	Comment	String Size	
	Traine	D'dia 1	ype	Dimension	Allus	mitiar Falac	Attribute	Commerte	String Size	
	SV SV	REAL	~			10.0	ReadWrite			
	FB	REAL	*			0.0	ReadWrite 🔹			
*			*							
.,	DK Cancel									

30. SV will pass the parameter value to the SetPoint of the IPIDCONTROLLER_1



31. Repeat Steps 28-30 for the parameters shown for IPIDCONTROLLER_1

IPIDCONTROLLER Parameter	Local Variable – Process_SIM	Value
Feedback	FB	
Auto	AUTO_RUN	
Initialize	INIT	
Gains	PID1_GAINS	
AutoTune	PID1_AT_EXEC	
ATParameters	PID1_AT	
ErrorMode		0



32. After completing, the IPIDCONTROLLER_1 should appear as shown below:

33. Click on the **Output** of IPIDCONTROLLER_1, then connect to the **B_IN** of the SIM_FB_1.

Process_SIM-POU*		
		IPIDCONTROLL IPIDCONTROLL Process Output

As shown.


34. Then connect **B_OUT of the SIM_FB_1** to the **XIN** of the AVERAGE_1.

AVERAGE_1 AVERAGE • RUN XOU		<i>SIM_I</i>	<i>FB_1</i> FB
XIN	<u> </u>	B_IN	B_OUT
N			

35. Connect a variable at **N** of the AVERAGE_1 and enter a sample cycle value of 5. Insert a TRUE variable for **RUN**.



36. Then connect **XOUT** of AVERAGE_1 to the Process of the IPIDCONTROLLER_1.





37. Click on the **Output** of IPIDCONTROLLER_1 again, then connect to **FeedBack** of IPIDCONTROLLER_1.

38. The complete program should appear as follows:



39. Finally, build and save the Function Block Program. Right click on the Micro830 icon in Project Organizer and select **Build**.

Project	Organizer				•	џ	х
Name:	FBD_Prog	ram					
	Micro830						
	Pr	Ĩ	Ор	en			
			Bui	ld			

40. At the **Output** window at the bottom center of the screen, the build should show succeeded.

Output
Show output from: Build 💿 😽 🖓 🖓 🖘
Build resource: MICR0830 Configuration: MICR0830 Compiling for 2080LC3016QWBA SIM_FB PROCESS_SIM Linking for 2080LC3016QWBA MICR0830: 0 error(s), 0 warning(s) Compiling for SIMULATOR SIM_FB PROCESS_SIM Linking for SIMULATOR MICR0830: 0 error(s), 0 warning(s) CONTROLLER: 0 error(s), 0 warning(s)
======================================

Click on **Save** icon **Save** to save your work.

Testing the Function Block Program

This section will show you how to test the Function Block Program created, proceed with the steps shown below.

1. In the **Project Organizer**, right click on **Micro830**, and select **Download**.



2. From the Connection Browser, select 2080-L30-16QWB, and click on OK.

Connection Browser	
Autobrowse Refresh	
 Workstation, ROCKWELL-D80AEF Linx Gateways, Ethernet AB_VBP-1, 1789-A17/A Virtual Chassis 00, Workstation, RSLinx Server 16, Micro830, 2080-LC30-16QWB USB 16, Micro830, 2080-LC30-16QWB 	
	OK Cancel

3. The following dialog box will appear for confirmation of the downloading if the controller is in RUN mode click **Yes** to proceed.



4. If the download is successful the Output window will display Succeeded



5. The following window will appear to change from Program Mode to Run Mode. Click **Yes** to proceed.



6. Click on the k at the Debug Toolbar, the programming workspace will change from a white to beige background.



At the same time, the status and value of the parameter will be display on screen.

	IPIDCONTROLL IPIDCONTROLL Process Output
REAL SV	9.99523 SetPoint Absolut. 0.00476
FB REAL 9.99523 AUTO RUN	FeedBa ATWarn 0
eool Faise INIT BOOL Faise	Auto OutGains - False Initialize
PID1_G GAN_PI. False	Gains
PIDT_AT_EXEC	AutoTu
	ErrorMo

7. To change the **SV** value of the IPIDCONTROLLER_1, double click on **SV**. The following Variable Monitoring window will appear.

	Va	riable Moni	itoring					×
ſ	Global Variables - Micro830 Local Variables - Process_SIM System Vari							
		Nam	e	Logical Value	Physical V	'alue	Lock	
			· A*	× A*		* 🕂	- A	≡
		SV		10.0	N/A			
		FB		10.0178	NZA			
		+ PID1_G						
		♦ PID1_AT						-
							•	
					<u>0</u> K		ancel]

8. Change the SV to 15.0 by clicking on the Logical Value field, then hit enter.

e!	Va	riable Monito	oring					
	Glot	oal Variables - Mic	ro830	Local Variables - F	Process_SIM	Syster	m Vari 🔨	>
		Name		Logical Value	Physical V	'alue	Lock	
			× A*			× ∦*	- A'	
	I	SV		15.0	N/A			
		FB		10.029	NZA			
		+ PID1_G						
		♦ PID1_AT						r
					<u>0</u> K		ancel	

9. Monitored the Output Value of the IPIDCONTROLLER_1, you will be able to see the value increase.

	IPIDCONTROLL IPIDCONTROLL Braccon
REAL 15.0	SetPoint Absolut. 4,6501
FB REAL 10.3504 AUTO_RUN	FeedBa ATWarn 0 Auto OutGains
INIT	Initialize

10. To stop monitoring the variable, click on 🧾 at the Debug Toolbar.

11. Then from the Micro830 tab, click on **Disconnect** to go offline.

Micro830	Process_	5IM-VAR	Process_SIM	1-POU	SIM_FB-POU				
Micro8	30		R	temote 1ode:	○ Program ● Run	Major Fault: Controller Mode:	Not Faulted Run (Remote)	Disconnect	Connected
J. Download	1 Upload	Variables	📻 Program			2	2080-LC30-	16QWB	

Chapter 4 -Creating a New Structured Text Program

Creating a New Structured Text Program

This chapter will show you how to create a new structured text program for creating menu selections and simple mathematical calculations.

1. Start the Connected Component Workbench for the Start Menu: Start →All Programs → Rockwell Automation → CCW → Connected Components Workbench.



Connected Components Workbench

Alternatively, double click on the shortcut on the Desktop

2. At the Connected Component Workbench window, drag **2080-LC30-16QWB** from the **Device Toolbox Catalog** window into the **Project Organizer** window. A new project will be created.



3. At the Name field, under the Project Organizer, enter ST_Program.

Project	Organizer	→ ‡ ×
Name:	ST_Program	

4. Under the **Project Organizer**, right click on the **Programs** select **Add** and select **New ST: Structured Text**.

Project Organizer	• 4	X			
Name: ST_Project					
Micro830					
Programs					
Global Va		Add	•	Ξ	New ST : Structured Text
	ß	Paste		- 6 0	New LD : Ladder Diagram
	Ŀ.	Proper	ties		New FBD : Function Block Diagram

Micro830		
Programs		
		Build
	1	Download
Global Variables	8	Upload
DataTypes	1	Online Change
Function Blocks	Ĩ	Open
		Import •
		Export •
	¥	Cut
	Ð	Сору
	B	Paste
	\times	Delete
		Rename

5. Right click on **UntitledST** and select **Rename**:

6. Type Selection and Enter:



- 7. Double click on **Selection** within the **Project Organizer** to start editing the Structured Text program.
- 8. Click at the Line no. "1" at the Selection-POU* tab.



9. Enter the following sentence "(*Simple Selection Program with CASE Statement*)", then hit enter.

Selection-PO	U*					
1	(*Simple	Selection	Program	with	CASE	Statement*)

Note: For entering comments use "(* comments *)"

10. Click at Line no. "2" at the Selection-POU* tab, enter the following program.



Note: All Structured Text Reserve word will be represented in magenta, and comments will be represented in Green.

When entering the IO variable, we are able to select from the pull down menu as shown



For Boolean expression, True is "1" and False is "0".

11. Double click on the Local Variables under the Selection programs to define a new variable.



12. Create an integer variable select_no as shown:



13. At the Project Organizer, double click on the Global Variables to create the Alias for the outputs.



14. At Micro830-VAR tab, enter Output_0 at Alias for _IO_EM_DO_00 and Output_1 at Alias for _IO_EM_DO_01.

4	Micro830-VAR Selection-VAR Selection-POU*									
		Name	Data Type	Dimension	Alias					
		- A*	× A*	→ A ⁺	- A					
		_IO_EM_DO_00	BOOL 🔹 👻		Output_0					
	I	_IO_EM_DO_01	BOOL 🔷 👻		Output_1					

15. Finally, build and save the structured text programming. Right click on the Micro830 icon in **Project Organizer** and select **Build**.



16. At the **Output** window at the bottom center of the screen, the build should show succeeded.



Click on **Save** icon **I** to save your work.

Inserting a Function Block in a Structured Text Program

This section will show you how to insert at function block in the existing Structured Text Program.

- 1. Double click on the **Selection**, to edit.
- 2. At Line 10 of the Selection-POU* tab, enter the following sentences

```
9

10 IF _IO_EM_DO_OO THEN

11 i:= a*b*c;

12 ELSE IF _IO_EM_DO_O1 THEN
```

- 3. At line 13 of the Selection-POU* tab, enter "AV" and select AVERAGE from the pull down menu.
 - 13
 Av

 1
 Image: Ascil and the second sec
- 4. Then key in "(" the following pull down menu will appear. Select the <Create New Instance>



5. The following dialog box will appear, AVERAGE_1 is created.

80+	ns	tru	ction Blo	ck S	elector: AVER	AGE								
_ Cc	ont	rolle	r : 2080LC3	0160	WBA									
			Name	0	Categor	y	1	TJ	ре					^
			-	A*		-	A*	•	A.					
		AT/	AN		Arithmetic			5	F	Arc ta	angenl	t		
		AT/	AN_LREAL		Arithmetic			- 2	7	Perfo	rm 64-	bit i	real arctang	e
		AVI	ERAGE		Data Manipulatio	n			в	Runn	ing av	era;	gelover Nisa	•
		AW	Ά		Communications			-17	1	Write	a strir	ng v	vith charact	e 💙
													>	
	_													
Pa	ara	met	ers											
Шг				Nar	ne	Data	a T u	ne		Direc	tion	1	Dimensi	
					- A*		-	d'			-	d'		
			XIN		0-	REAL		+	Varl	nout		+		
			N			DINT		-	Varl	nput		•		
		Ŧ	Fifo			REAL		*	Var			•	[0127]	
			Index			SINT		-	Var			+		Υ.
	۹,													
			AVED 40							D				
Instar	nc	e:	AVERAL	2E_1		~		×	onow	гагал	neters			
Input	s:		\$3											
Scop	e:		Selectio	n										
											OK		Car	ncel

Note: 3 Inputs are required for Average function block, similar to Ladder Logic Representation.



RUN, XIN, N parameter will be required.

6. Click **OK** to create an instance. When entering the instance, the popup box will indicate the parameter needed for the Function block.



7. Please end the parameter as shown:

'AVERAGE_1(_IO_EM_DO_01,a,3)' Where: RUN = _IO_EM_DO_01 XIN = a N = 3

8. Then assign the output of the AVERAGE to j, as per shown. Close the IF statement with END_IF.

```
9
10 IF _IO_EM_DO_OO THEN
11 i:= a*b*c;
12 ELSE IF _IO_EM_DO_O1 THEN
13 AVERAGE_1(_IO_EM_DO_O1,a,3);
14 j:= AVERAGE_1.XOUT;
15 END_IF;
16 END_IF;
```

Notes:

• The mathematical equation can be expressed by entering it as is. If doing the calculation in ladder, you might need a few function blocks to complete the equation.

Example: i := a + b + c; or circumference := $2^{3},142^{r}$; (with r is the variable) or r:= circumference/($2^{3}.142$);

• When using IF statement, we must also close with an END_IF, in the case if there is an ELSE_IF statement used, we must also close the ELSE_IF statement with END_IF.

9. In completion of writing the program, variables used must be created. Double click on the **Local Variables** under the Selection programs to create variable.



10. Create the following variables for the program

Name	Data Type	Initial Value			
а	Real	0.0			
b	Real	1.5			
с	Real	3.142			
i	Real	2.0			
j	Real	0.0			

The Selection-VAR tab should look like the following:

/	M	licro	830-VAR / S	election-	VAR	Selection-PC				
Γ			Name	Data Ty	pe	Dimension	Alias	Comment	Initial ¥alue	
			▼ A*		A*	▼ A*	- A*	- A*	- 1	F*
			select_no	INT	•					1
			i	REAL	*				0.0	1
			a	REAL	*				1.5	1
			Ь	REAL	•				3.142	1
			с	REAL	*				2.0	1
			j	REAL	•					1
		Ŧ	AVERAGE_1	AVERAGE	•					1
	*				•					

11. Finally, build and save the structure text programming. Right click on the Micro830 icon in **Project Organizer** and select **Build**.



12. At the **Output** window at the bottom center of the screen, the build should show succeeded.



Click on **Save** icon **b** to save your work.

Testing the Function Block Program

This section will show you how to test the Function Block Program created. In continue to the steps in Creating New Function Block Program, proceed with the steps shown below.

1. In the Project Organizer, right click on Micro830, and select Download to download the program:

Project	Organi	izer			•	ц	×
Name:	ST_Pro	ojecl	t				
	Micro8	30 Pro		Open Build Download	4		

2. From the Connection Browser, select 2080-L30-16QWB, and click on OK.

С	onnection Browser
	Autobrowse Refresh
	E III Workstation, ROCKWELL-D80AEF 효사율 Linx Gateways, Ethernet
	AB_VBP-1, 1789-A17/A Virtual Chassis AB_00, Workstation, RSLinx Server
	E → IIII 16, Micro830, 2080-LC30-16QWB E → USB
	II- 16, Microssu, 2080-LC30-16QWB

3. The following dialog box will appear for confirmation of the downloading if the controller is in RUN mode. Click on **Yes** to proceed.



4. In the completion of downloading the program, the Output window will display Succeeded



5. The following window will appear to change from Program Mode to Run Mode. Click on **Yes** to proceed.



6. Click on the *k* at the Debug Toolbar, the programming workspace will change from white background to gray background.



7. To simulate the variable, run over the 📼 and the following popup dialog box will appear. Click on the dialog box to monitor.

CASE select_ny OF 1: _IO_EM_DO_OO:= TRUE;	
2: IO_EM_DO_01:= TRUE;	CASE select no OF
ELSE	1: IO_EM = TRU
IO_EM_DO_00:= FALSE;	2: IO_EM = TRU
IO_EM_DO_01:= FALSE;	ELSE

8. The Variable Monitoring window will appear.

V a	riable Monito	ring						×
Glob	al Variables - Mic	ro830	Local Variable:	s - S	election	System Va	ariable 📢	>
	Name		Logical Val	ue	Physic	al Value	Lock	
		· ≁	*	A*		▼ A*	- × A	≣
	select_no		1		N/A			
	а		1.5		N/A			
	Ь		3.142		N/A			
	с		2.0		N/A			-
					<u>0</u> K		ancel]

9. Change the value at the Logical Value of the variable select_no. to simulate the program.

V a	riable Monito	oring					×
Glob	oal Variables - Mic	cro830	Local Variables -	Selection	System V	ariable ≺	>
	Name		Logical Value	Physic	alValue	Lock	
		· → ∂{*	- A		- A*	- × A	≡
I	select_no		2	N/A			
	а		1.5	N/A			
	Ь		3.142	N/A			
	с		2.0	N/A			-
				<u>0</u> K		ancel]

Simulation for **select_no** using the demo kit output indicators:

- In the demo kit, Output 0 should be lit when the select_no variable is 1. At the Logical Value of select_no. change to 2. Now, Output 0 should turn off, and Output 1 should lit.
- Change the value of select_no variable to 0 or 3, both Output 0 and Output 1 should turn off.
- The program logic is written so that if the value is not 1 or 2, both Output 0 and Output 1 should turn off.

10. To simulate the mathematic calculation, at the **Variable Monitoring** Window, change the value of a, b and c.

👥 Va	riable Monitoring							
Glob	al Variables - Micro830	Local Variables - 9	Selection	System V	ariables -	Micro830	0 1/0	- Micro83 🔨 🕨
	Name	Logical Value	Physic	al Value	Lock	Data	Туре	Dimension
	× A	- A*		- A*	- A*		* A*	▼ A*
	select_no	2	NZA			INT		
	a	1.5	N/A			REAL	-	
	Ь	3.142	N/A			REAL	•	
	с	2.0	N/A			REAL	-	
	i	9.426	N/A			REAL	•	
	i	1.5	N/A			REAL	-	
	+ AVERAGE_1					AVERA	GE 🝷	
						<u>0</u> K		<u>C</u> ancel

Simulation for the equation i := a*b*c;

Initial values of a is 1.5, b is 3.142 and c is 2.0, change the values as shown below.

 Vario	able Monitoring						
Global \	Variables - Micro830	Local Variables - 9	delection System V	/ariables -	Micro830	1/0	- Micro83 🔨 🕨
	Name	Logical Value	Physical Value	Lock	Data Ty	pe	Dimension
		<i>A</i> [*]		- A*	*	A*	
	select_no	2	N/A		INT	•	
	a	8.0	N/A		REAL	•	
	Ь	9.0	N/A		REAL	•	
	с	10.0	N/A		REAL	•	
	i	9.426	N/A		REAL	•	
	i	8.0	N/A		REAL	•	
+	AVERAGE_1				AVERAGE	•	
					<u>0</u> K		Cancel

However, we expected i to equal 720.0. We need to change the value of the select_no to **1** to execute the equation $i := a^*b^*c$;

💀 Variable Monitoring										
Glot	bal Variables - Micro830	Local Variables - Selection System \		System V	System Variables - Micro830 🛛 1/0 - Micro83 <					
	Name	Logical Value	Physic	al Value	Lock	Data Type	Dimension			
	- A*	- A*		- A*	- A*	× 🚜	× ∂₹*			
I	select_no	1	N/A			INT 🔷				

When the **select_no**'s value is changed to 1, the equation will be executed. The value will be shown in the **Variable Monitoring** window.

💀 Variable Monitoring									
Global	Variables - Micro830	Local Variables - 9	election	System Va	ariables -	Micro830	1/0	- Micro83 🔹 🕨	
	Name	Logical Value	Physic	alValue	Lock	Data	уре	Dimension	
				- A*	- • A*		- A*	- A*	
	select_no	1	N/A			INT			
	а	8.0	N/A			REAL	-		
	Ь	9.0	N/A			REAL	•		
	с	10.0	N/A			REAL	-		
	i	720.0	N/A			REAL	•		
•	i	8.0	N/A			REAL	-		
+	AVERAGE_1					AVERA0	GE 🔻		
								•	
						<u>0</u> K		Cancel	

The program is written in such:

IF _IO_EM_DO_00 THEN

i := a*b*c;

Therefore, only when the Output 0 = 1 will the equation be executed.

- 11. To stop the monitoring of the variable, click on <a>Image at the Debug Toolbar.
- 12. Then from the Micro830 tab click on **Disconnect** to go offline.

Micro830 Process_SIM-VAR Process_SIM-POU SIM_FB-POU									
Micro8	30		F N	lemote 1ode:	○ Program ● Run	Major Fault: Controller Mode:	Not Faulted Run (Remote)	Disconnect	Connected
Jownload	1 Upload	Variables	<u>न</u> Program			2	080-LC30-	16QWB	

Chapter 5 -Using Connected Components Workbench with PanelView[™] Component

Using Connected Components Workbench with PanelView Component

Before you begin, you should already have a general knowledge of how to use the Connected Components Workbench software and how to create an application for you Micro800 controller. If you do not have this knowledge, please review the Micro800 and CCW Getting Started Guide, Publication 2080-QR001B-EN-P.

The recommended Modbus RTU network topology for a Micro800 and PanelView Component is to configure the Micro800 controller as a slave device, and the PanelView Component as the master device. Therefore that is the configuration that will be discussed and configured in this guide.

Mapping Variables to Modbus Registers

The Micro800 supports the following Modbus registers.

Address	Range	Data Type	Access
Output Coils	000001-065536	Boolean	Read/Write
Input Coils	100001-165536	Boolean	Read Only
Input Registers	300001-365536	Word (16-bit)	Read Only
Holding Registers	400001-465536	Word (16-bit)	Read/Write

1. Create a new CCW project for your Micro800 controller, and create a Global Variable called DATA with data type INT and attribute ReadWrite.

		ODINI .	Readwrite	
	SYSVA_MAJ_ERR_HALT	BOOL 🔷	Read	*
	SYSVA_ABORT_CYCLE	BOOL 🔷	Read	-
	DATA	INT 📼	ReadWrite	▼
*		+		*

2. Open the Modbus Mapping table by following the steps below.



3. Add a Variable to the Mapping table by following the steps below.

	Double-click here to launch Variable Selector window.	the
Properties Variable Name DataType Ac	ldress Addresses Used	
Select the User Global Variables tab.	Then cli DATA v	ck here to select the rariable.
Name Type Bool	Glebal Scope	
Name Data Type Dimension * # * # * #	stem Variables - Micro830 1/0 - Micro830 0 Alias ▼ ∂⊈*	Petined Words - Controller Comment
Then click OK .		
		OK Cancel

4. Map the **DATA** variable to register address 400001.



 Repeat steps 3 and 4 for variables, _IO_Embedded_Digital_Output_0 (I/O – Micro830 tab), __SYSVA_CYCLECNT (System Variables – Micro830 tab), and __SYSVA_REMOTE (System Variables – Micro830 tab), and map them to the register addresses as shown below.

Variable Name	Data Type	Address	Addresses Used						
DATA	Int	400001	400001						
_IO_Embedded_Digital_Output_0	Bool	000001	000001						
SYSVA_CYCLECNT	Dint	300001	300001 - 300002						
SYSVA_REMOTE	Bool	100001	100001						
•									
	,								
Notice that this variable uses two consecutive Modbus registers – this is because it is a 32-bit variable.									

6. You have completed mapping variables to Modbus registers. Save your project.

Configure Micro800 Serial Port

You will be configuring your Micro800 controller as a Modbus RTU slave device. The PanelView Component will be configured as the Modbus RTU Master.

1. Open the Serial Port properties panel.



2. Configure the Serial Port Properties with the following values:

-Properties		
Driver:	Modbus RTU	
Baud Rate:	19200 💌	
Parity:	None	
Unit Address:	1	
Modbus Role:	Modbus RTU Slave	

3. Expand **Advanced Settings** to configure the **Protocol Control** properties with the following values:

ΓĒ	Advanced Settings			
	Media:	R5232	RTS Pre-Delay:	0
	Data Bits:	8	RTS Post-Delay:	0
	Stop Bits:	1		
	Response Timer:	1000		
	Broadcast Pause:	1000		
	Inter-Char Timeout:	0		

If you are using RS485, you can set the **Media** property to RS485 and leave the remaining settings the same.

4. You have completed configuring your serial port for Modbus. Build and save your project, and then download it to your controller.

Create an Offline PanelView Component Application

1. Add a PanelView Component device to your project.



2. Launch PanelView Component Design Station.





3. Select the PanelView Component platform and create a new application.

- Select the Communication tab. PanelView Component **•** × PVc DesignStation Startup PVc DesignStation - PVc Communication Tags Security Settings Languages Screens Alarms Recipes 😺 🥎 📙 😭 🖄 🐁 🖻 ີ 🗙 🕑 🕑 Application Language: English (United States) (1033) Load Last Saved Driver Configuration Protocol Serial DF1 • C Ethernet Allen-Bradley SLC/PLC -Driver USB / Ethernet Use Ethernet Encapsulation: 🗖 PanelView Component Settings Uptimization Port Write Opt
 Baud Rate
 Data Bits
 Party Bits
 Flow
 Report
 Link Settings

 19200
 8
 None
 1
 None
 Frows?
 Station Address
 Protocol
 Only Accept Responses For Station Address
 Slave Poll Delay

 19200
 8
 None
 1
 None
 If
 2
 Full Duplex
 00
 Controller Settings Delete Selected Controller(s)

 Ascending

 Ascending Add Controller Name Controller Type Address Timing Auto-Demotion Description Protocol Settings Slot
 Error Checking Method Swap PLC-5 Float Words? Request Size Disable N File Floats Configurat
 CRC Large Large R ... PLC-1 MicroLogix 1 Device: 2711C-T6T
- 4. Setup **Communication** settings to configure your PanelView Component as a Modbus Master to communicate to your Micro800 controller.

Panel¥iew Component	Select Modbus fro	om I
PVc DesignStation Startup PVc DesignStation - PVcApplication1	dropdown list.	
Settings Communication	Tags Scre	
😺 🥎 🔒 😫 🖄 🐘 🔒 🚺 🗡		
Load Last Saved Driver Configuration		
Protocol		
Serial DF1		
C Ethernet DF1 DH485		
OEMax NX Plus Series		
Use Etherne Modbus Ascii		
TIWAY Host Adapter - TIWAY1 TIWAY Host Adapter - UNILINK MHIU		
TIWAY Host Adapter - UNILINK HIU Mitsubishi FX	Flow Report	
Mitsubishi FX Net	Control Errors? Stat	
RS232 Mitsubishi	ne 🔽 2	
Siemens S5 Siemens S7 MPI		

Configure the Driver settings as shown below – the default settings will work for RS232. If using RS485, change the Port settings to **RS422/485 (Half-duplex)**.

RS232

	Driver USB / Ethernet										
U	Use Ethernet Encapsulation:										
P	anelView Componer	nt Settings									
	Write Optimization										
	Port	Baud Rate	Data Bits	Parity	Stop Bits	Flow Control	Report Errors?				
	RS232	19200 -	8	None	1	None	\checkmark				

RS485

Driver USB / Ethernet										
Use Ethernet Encapsulation:										
PanelView Compon	ent Settings									
Write Optimization			-							
Port	Baud Rate	Data Bits	Parity	Stop Bits	Flow Control	Report Errors?				
RS422/485 (Half Dupl	+ 19200	8	None	1	None					
5. In the Controller Settings, configure a controller with settings as shown below.

		Ev. as the	erything ca default ex first three	an be left cept for settings.	
Controller Settings					
Add Controller Delete Selected Controller(s)					
Port by Name	Ascending				
Name Controller Type Address	Timing Auto-Demotion	Description Settings	Block Sizes	Modbus TCP Framing	Deactivate tags on illegal address exception
Micro800 Modbus 1					

6. Create tags addressed to the tags you created earlier in your Micro800. Refer to the section called "Mapping Variables for Modbus Registers" for details on how to create the Micro800 tags.

					C	lick the Tags tab.	
PVc DesignStation Startu	p PVc DesignStation -	PVcApplication1					
Settings	Communio	cation	Tags		Screens	Security	Alar
🔈 🍫 🔒 I	📅 🔊 😽		× 5	6	0		Appl
External Mem	nory System e Tag(s)	Global Conne	ctions				
	ag Name	Da	ata Type		Addre	SS	Contro
	\backslash						
	Cli	ck Add Tag					

Create the following tags as shown below - make sure to choose the correct data type.

Ext	External Memory System Global Connections							
Add	Add Tag Delete Tag(s)							
	Tag Name	Data Type	Address	Controller				
1	Output_0	Boolean	0000001	MICR0800				
2	Cycle_Count	32 bit Integer	3000001	MICRO800				
3	Remote_Status	Boolean	1000001	MICR0800				
4	DATA	16 bit Integer	4000001	MICRO800				

7. Create a screen display with objects linked to the tags you just created.



Create a maintained pushbutton linked to tag, **Output_0**. This is not typical practice, as a direct output should not be turned on/off directly, but is done for demonstration purposes.







Create a numeric display object linked to tag, Cycle_Count.





Create a multistate indicator object linked to tag, **Remote_Status**.





Create a Numeric Input Enable object linked to tag, DATA.





Add a Goto Config button to your display.



Your display should look like the following.



8. You are done creating your PanelView Component application. Save your application.

Transferring an Offline PVc Application to a PVc Terminal

Hardware Used

PanelView Component C600 - 2711C-T6T

This section will demonstrate how to transfer an offline PVc Application to a PVc terminal. Transferring the file involves copying the application to a USB or SD flash media, and then inserting it into the PVc terminal, and copying it to the terminal.

1. From your CCW project, launch the PVc DesignStation Startup pane.



- 2. Insert either a USB flash drive, or SD card into your computer.
- 3. Set up a file transfer to copy the application to your USB/SD flash media.

	Click File Transfer.
elView Component Dashboard File Transfer Help Quit	
Click New Transfer . This will launch the File Transfer Wizard.	
Allen-Bradle	y I
	File Trar
Terminal Transfer New Transfer Transfer files to and from the terminal storage	e media
Terminal Cleanup Delete File Permanently remove files from Internal, SD, or US	B storage devices

Configure the File Transfer as shown below, and then click Transfer.

File Transfer Wizard	×
From: Internal Storage (PVc DesignStation) File: Application - Modbus_Example To: My Computer	
(1) Select file destination	_
My Computer	
	-
Back Next Transfer Cancel	

Browse to the root of your flash media, and then click **Save**. This will save the PVc application file (.CHA) to your flash media.



- 4. Remove the flash media from your computer and insert into PanelView Component terminal.
- 5. Copy the application from your flash media to your PanelView Component.



Select Intern Source, and y that your app been copied t terminal.	al as your you'll notice lication has to your				
Allen-Bradley				PanelView	C600
		-			
		File Manager		-	
	Source: 🖣	To:		_	
	🔼 Internal 🚺	Internal	Copy		
	Modbus_Example	;	Delete		
		R	Set As Startup		
	Copy Recipe	Copy Alarm Histo	ory Run		
	Startup Application				
		Rese	t Main		

6. You have completed transferring an offline application to your PVc terminal.

Cabling the Micro800 to a PanelView Component

Hardware Used

PanelView Component C600 – 2711C-T6T

RS232 Cable, 1761-CBL-xxxx or 2711-CBL-PMxx

RS485 Adapter, 1763-NC01

1. For RS232 communications, you will need an 8-pin Mini-DIN to 9-pin D-shell null modem cable. See table below for recommended cables.

0.5 m (1.6 ft)	1761-CBL-AP00
2 m (6.6 ft)	1761-CBL-PM02
5 m (16.4 ft)	2711-CBL-PM05
10 m (32.8 ft)	2711-CBL-PM10

For RS485 communications, you will need to use a 1763-NC01 adapter, and wire the recommended twisted pair shielded cable as shown below. The recommended cable is Belden 3105A or equivalent (two-wire shielded twisted pair with drain). **Note:** Because both devices' serial ports are non-isolated, connect the shield/drain wire at one end only to prevent a ground loop.



There is no need for terminating resistors. The PanelView Component has an internal 121 ohm resistor across the R and R- terminals, and the Micro800 is terminated by jumpering TERM to A on the 1763-NC01 adapter.

2. Connect cables, and test the application.

RS232 - Connect the serial cable from the 8 pin Mini-DIN port on the Micro830 to the D-shell connector on the PanelView Component terminal.

RS485 – Connect the 1763-NC01 adapter to the Mini-DIN port on the Micro830 controller. Connect the RS485 cable from the 1763-NC01 adapter to the RS485/422 port on the PanelView Component.

- 3. Confirm the controller is in RUN mode and that no faults exist.
- 4. Load PanelView Component application.

Allen-Bradley	, 	PanelView C600	Click File Manager.
	Main File Manager Communication Display Display System Information	Goto te and Time y Hour 14 th Minute 55 ar Second 1 26	
Select you and click F	ur application, Run.	PanelView C600	
	File Manager Source: To: Internal Internal Modbus_Example Copy Recipe Copy Alarm History Startup Application Reset	Copy Delete Set As Startup Run	



6. You have finished testing your application.

Chapter 6-Using Connected Components Workbench with PowerFlex® Drives

Hardware Used

- PowerFlex 4-Class Drive
- 1203-USB
- Modbus Cable (Flying leads to RJ45)

Adding a PowerFlex 4 Drive to a CCW Project

This chapter will show you how to add a PowerFlex 4-Class Drive to a CCW Project.

1. Review the Getting Started Guide (Pub# 2080-QR001B-EN-P) to learn how to create a new project and add a controller. Once that's done, the screen should look like the following and click on **Device Toolbox**.

File Edit View Build Debug	Tools Communications W	indow Help	
🗄 🖆 🖼 - 🔙 🖌 🖬 🛍	9 • 9 • 5 • 5	🚬 🚰 🋠 🚚 🎬 🕨 🗉 🗢 🗐 🗐 🥵 🧌 🗏 🗐 🍪 🦓 🕼 🖉 💋 Cycle Timing (ms): 🗾 🚽
Project Organizer 🛛 🛨 🕇 🗙	Micro830	•)	C Device Toolbox - 4 ×
Name: Project2	Micro830	Remote Program Major Fault: Connect Mode: Run Controller Mode: Disconnected	Discover
Programs	Download Upload	2080-LC30-16QWB	E Catalog
Global Variables			Controllers
DataTypes	Micro830		2080-LC10-12AWA
Function Blocks			2080-LC10-12QBB
			2080-LC10-12QWB
			2080-LC30-10QVB
		000000000	2080-LC30-10QWB
			2080-LC30-16AWB
	General	Vendor Name: Allen-Bradley	2080-LC30-16QVB
	 Communication Ports Serial Port 	Catalog ID: 2080-LC30-16QWB	2080-LC30-16QWB
	USB Port Date and Time	Firmware OS Rev: 1.3	2080-LC30-24QBB
	Startup/Faults	Name:	2080-LC30-24QV8
	Embedded I/O	Description:	2080-LC30-48AWB
	< Empty > < Empty >	Boot Revision: 0.0	2080-LC30-48QBB
			2080-LC30-48QVB
			2080-LC30-48QWB
			Drives
	Output		
	Show output from:		
			Device Toolbox
Ready	1		L kt

2. Expand the Drives folder within Device Toolbox:



3. Click on the PowerFlex 4 icon, hold and drag it across to the **Project Organizer**, then release:



<u>Note:</u> The default name for the drive is **PowerFlex4_1*** to change this, just right click on it and select rename to enter the desired name. Also, notice the asterisk (*) next to the project name and the drive that indicates the project has been modified and needs to be saved. Once the project is saved, the asterisk will disappear.

4. Double click on the PowerFlex 4_1 icon and you should see the Device Configuration screen:

PowerFlex 4			∽ onnect	Discor	inecteo
Download Upload	III Parameters	i Properties	+ _* + Wizards	2 Manual	Weight Heip
Powerflex.					
1P110V.25	5HP				
Bevision: 61	001				
Status					
-Feedback					

Connect to a PowerFlex 4 Drive using a 1203-USB Device

This section will show you how to add 1203-USB to the CCW project to be able to connect to the PowerFlex 4 Drive added in the previous section.

1. Once you are on the PowerFlex 4 Device Configuration window, click on the **add+** tab:

PowerFlex 4_1* Micro830	▼ X
PowerFlex 4	Connect Disconnected
Download Upload Parameters	↑, ⁺ 2 Properties Wizards Manual
PowerFlex:	
1P110V.25HP	
Series: A	
Revision: 6.001	
Status	
Feedback	
PowerFlex 4 add+	

2. Double-click on the 1203-USB:

PowerFlex 4_1* Micro830	▼ ×
PowerFlex 4	Connect Disconnected
DSI COMMs (Single-dri	ve)
1203-U5B 22-5CM-232	22-COMM-D 22-COMM-P
22-COMM-E 1769-5M2	22-WIM-Nx 22-COMM-C
22-сомм-в	
нім	
22-HIM-A3 22-HIM-A3 SER C	
Port:	Add
PowerFlex 4 add+	

3. Click on the **1203-USB** tab added on the bottom:

11

PowerFlex 4	1 - 1203-USB	add+	

4. Before connecting to the Drive, you must install the 1203-USB drivers and configure a new DF1 connection in RSLinx (refer to publication DRIVES-UM001B-EN-P for more details). Click the **Connect** button:

PowerFlex 4_1* Micro830	•	×
PowerFlex 4	Connect Disconnected	
Download Upload Parameters	Properties Manual Help	
1203-USB		
Series: A		
Revision: 1.004		L
Status		
Port: 1		
	+ 1	
	<u>. </u>	

5. Expand the DF1 connections and look for the **01,AB DSI** representing the 1203-USB, select it and then click **OK**.

Connection Browser	
 ✓ Autobrowse ☐ Workstation, ROCKWELL-D80AEF ☐ ♣ Linx Gateways, Ethernet ☐ ♣ AB_DF1-1, Data Highway Plus ☐ ♣ AB_DF1-2, Data Highway Plus ☐ 00, Workstation, DF1-COM3 ☐ 01, AB DSI ☐ ♣ AB_ETH-1, Ethernet ☐ ♣ AB_ETHIP-1, Ethernet ☐ ♣ AB_VBP-1, 1789-A17/A Virtual Chassis ④ ↓ USB 	
	OK Cancel

6. Notice the green background around the drive in the Project organizer meaning that you are now connected to the PowerFlex 4 using the 1203-USB. Click on the **PowerFlex 4** drive tab:



7. Select the Wizards as shown:



8. Select the PowerFlex 4 Startup Wizard and then click Select.



9. The following screen will show. Click Next to skip this welcome screen:

PowerFlex 4 Startup Wizard	- (1 of 10)	X
Wizard Step Image: Reset Parameters Image: Reset Parameters Image: Reset Parameters Image: Stop / Brake Mode Image: Stop / Brake Mode Image: Reset Parameters Image: Reset Parameters	Welcome Image: Comparison state Wizard Revision 3.0.19	The PowerFlex 4 Startup Wizard assists in setting commonly used drive parameters. It is recommended that the wizard steps be performed sequentially. Tip: Hovering the cursor over an icon I of or text box on a wizard page will display additional information as a tip message.
		Cancel < Back Next > Finish >>

10. Click on Reset Parameters :

PowerFlex 4 Startup Wizard	- (2 of 10)	×
Wizard Step ✓ ∰ Welcome ✓ ∰ Reset Parameters ∰ Motor Data ∰ Stop / Brake Mode ∰ Direction Test ∰ Ramp Rates / Speed Limits ∰ Speed Control ∰ Digital Inputs ∰ Relay Output ∰ Pending Changes	Reset Parameters Licking the Reset Parameters button will cause parameters to be changed immediately (set to default values based on the input supply selected and provides a known stating point (all parameters at default settings) for future edits. If you want to keep the existing parameter settings, then this step can be skipped by clicking Next>. Reset Parameters Parameters Reset: Unknown	
	Cancel < Back Next > Finish >	»] //

10. Click Yes and then click Next:



11. In this quick start we will use the default Motor Data. Click Next:

PowerFlex 4 Startup Wizard -	(3 of 10)				×
Wizard Step ✔☷ Welcome	Motor Data				
✓ Image: Version of the second se	Motor OL Current:	2.3	Amps		
📰 Stop / Brake Mode 📰 Direction Test	Motor NP Volts:	230	Volt		
E Ramp Rates / Speed Limits	Motor NP Hertz:	60	Hz		
📰 Digital Inputs					
📰 Pending Changes					
]					
			Cancel	K Back N	lext > Finish >>

12. Select as shown and then click **Next**:

PowerFlex 4 Startup Wizard ·	(4 of 10)
Wizard Step ✔ Welcome	Stop Mode / Brake Type
✓ III: Reset Parameters ✓ III: Motor Data ✓ III: Stop / Brake Mode III: Direction Test III: Ramp Rates / Speed Limits	DB Resistor Sel: Disabled 💌 Stop Mode: Ramp, CF
Speed Control Digital Inputs Relay Output B:: Pending Changes	DC Brake Level: 0.1 Amps DC Brake Time: 0.0 Secs
	Cancel < Back Next > Finish >>

13. To complete the Direction Test follow these steps:

- a. Click O to clear the present fault (F048) if showing.
- b. Enter the desired reference. For this quick start we'll use 30Hz and then click
- c. A speed reference acknowledgement window will appear to accept a parameter change. Click **Yes.**
- d. By now the motor should be rotating at reference speed. Verify that the motor direction of rotation is correct and then select the **Yes** radio button.
- e. You are done with the direction test. Click **Next** to continue.

≫PowerFlex 4 Startup Wizard - (5 of 10)	×		
Wizard Step	Direction Test			
 ✓ E: Welcome ✓ E: Reset Parameters ✓ E: Motor Data ✓ E: Stop / Brake Mode ✓ E: Direction Test E: Ramp Rates / Speed Limits E: Speed Control E: Digital Inputs E: Relay Output E: Pending Changes 	Danger: This test will cause the motor to rotate. Misuse may result in death, injury or damage equipment. You should have an external safe method of stopping the motor nearby when using feature. Image: Ensure that Motor Data is correct before proceeding with this page. Direction Test causes some parameters in the drive to change immediately. When you leave this page the device will be sto When you leave this page the drive will be stopped. Set the Jog Reference to a positive value and JOG the drive. The motor should rotate in the forward direct Verify that the direction of rotation is correct. Digital Ins will be set to NotUsed during the test.			
	Reference Image: Constraint of the application? At Reference Jog Reference Jog Office 30.0 Hz Jog No Image: Constraint of the application? 30.0 Hz			
	Close < Back Next > Finish :	›>]		
	1.0			

Speed Reference					×
The Speed Reference is currently not set to the comm port. To u set to 5. Do you want it changed?	use the referen	ce velocity from	this wizard page the Sp	beed Reference	(parameter 38) must be
[Yes	No			

14. Select as shown and then click Next:



15. Select Comm Port and then click Next:

斧PowerFlex 4 Startup Wizard - (7 of 10)					×
✓ PowerFlex 4 Startup Wizard - (Wizard Step ✓ Image: Welcome ✓ Image: Stop / Brake Mode ✓ Image: Stop / Brake Mode <td>Speed Contro</td> <td>Comm Port</td> <td></td> <td></td> <td></td> <td>X</td>	Speed Contro	Comm Port				X
			Close	< Back	Next >	Finish >>
16. Set the **Start Source** to Comm Port to eventually trigger the **Preset Freqs** shown below. Select as shown and then click **Next**:

PowerFlex 4 Startup Wizard - (8 of 10)	×
 PowerFlex 4 Startup Wizard - (Wizard Step Welcome Reset Parameters Motor Data Stop / Brake Mode Stop / Brake Mode Ramp Rates / Speed Limits Speed Control Speed Control Digital Inputs* Relay Output Pending Changes 	B of 10) Digital Inputs Stop Source: Ramp, CF Start Source: Comm Pot Direction Digital Common Digital In 1: Not Used Digital In 2: Not Used Preset Freqs: 0 0.0 Hz 1 5.0 Hz 2 10.0 Hz 3 20.0 Hz	×
	Cancel < Back Next > Finish >>	

17. Select as shown and then click Next:

PowerFlex 4 Startup Wizard -	9 of 10)		×
Wizard Step ✓ III Welcome ✓ III Reset Parameters ✓ III Motor Data ✓ III Stop / Brake Mode ✓ III Direction Test ✓ III Ramp Rates / Speed Limits ✓ III Speed Control ✓ III Digital Inputs" ✓ III Relay Output III Pending Changes	Relay Output	R1 R2 R3	
		Close < Back Next	> Finish >>

18. Click Finish:

PowerFlex 4 Startup Wizard - ((10 of 10)	×
Wizard Step ✓ E:: Welcome ✓ E:: Reset Parameters ✓ E:: Motor Data ✓ E:: Stop / Brake Mode ✓ E:: Direction Test ✓ E:: Ramp Rates / Speed Limits ✓ E:: Digital Inputs* ✓ E:: Relay Output ✓ E:: Pending Changes	Applied and Pending Changes Below is a list of changes that have already been made. Wizard Step "Reset Parameters" Parameters were reset Wizard Step "Direction Test" Direction Test completed successfully. No changes were made by direction test Below is a list of changes that will be made if you click Finish. Wizard Step "Digital Inputs" Change parameter "52 - [Digital In2 Sel]" value from "Preset Freq" to "Not Used". Change parameter "51 - [Digital In1 Sel]" value from "Preset Freq" to "Not Used".	
5	Close < Back Next >	Finish >>

19. Save the project by clicking 🛃 and the following window will appear. Click **Yes** to upload the drive parameters.

Upload Online Devices?
Do you want to upload the following devices:
PowerFlex 4_1
Yes No

Configuring the Controller for Modbus Communication with a PowerFlex 4

This section will show you how to configure the Micro830 for Modbus communication using the Serial plug-in module.

1. To configure the controller plug-ins, double click on the Micro830 icon in the **Project Organizer** to bring up the following screen:



2. Add an isolated serial plug-in to slot 1 by right clicking on the graphic of the first plug-in slot and selecting **2080-SERIALISOL**:



3. The device configuration window will now look like this:

Micro830 Remote Mode: Program Major Fault: Connect Disconnected Download Lipload 2080-LC30-16QWB Manuals Help Micro830 Image: Status Image: Status Help Micro830 Image: Status Image: Status Image: Status Help General Image: Status Image: Statu	Micro830 PowerFlex 4_1*					• ×
Mode: Run Controller Mode: Controller Mode: <thcontroller mode:<="" th=""> <thcontroller mode:<="" th=""></thcontroller></thcontroller>	Micro830	Remote	Program	Major Fault:		
Download Upload 2080-LC30-16QWB Manuals Help Micro830 Image: Comparison of the start o		Mode:	🔴 Run	Controller Mode:		Connect
Micro830 Image: Second Secon	Download Upload		2080-LC	30-16QWB		2 00 Manuals Help
General Memory Communication Ports Serial Port USB Port Date and Time Interrupts Properties Properties Driver: ASCII 19200 Parity: None Parity: None Parity: None	Micro830	Alter-divadity Alter-	000000			
<pre>Empty ></pre>	General Memory General Serial Port USB Port Date and Time Interrupts Protection Startup/Faults Modbus Mapping Embedded I/O Plug-In Modules 2080-SERIALISOL Centry >	Properties - Driver: Baud Rat Parity:	e:	ASCII 19200 None	v v	

4. Double Click the **2080-SERIALISOL** plug-in and verify the settings are the same as shown below.

General	Properties		
Memory			+ 2080-SERIALISOL
Communication Ports	Driver:	Modbus RTU	
- Serial Port			
USB Port	Baud Rate:	9600 💌	
Date and Time		,	
	Davitur	None	
Duckashing	Pancy.	Intone	
Protection		100	
	Unit Address:	100	
Modbus Mapping			
Embedded I/O	Modbus Role:	Modbus RTU Master	
⊡ Plug-In Modules			
2080-SERIALISOL			
< Empty >			
Compey >	L Advanced Settings		
,			
	- Droportion		
jee General	Propercies		
Memory			+ 2080-SERIALISOL
🚊 Communication Ports	Driver:	Modbus RTU	
Serial Port			
USB Port	Baud Rate:	9600 💌	
Date and Time		,	
Interrupts	Parity:	None	
Protection	· andy ·		
	Unit Address:	100	
Modbus Manning		J	
Embedded I/O	Advanced Cettings	1	
Endedded IV	Auvanceu Settings		
	-Protocol Control	·	
< Emplu >			DTS Dro Dolour
< Empty >	Media:	R5485	RTS Pre-Delay: jo
		,	
	Data Bits:	8	RTS Post-Delay: U
		,	
	Stop Bits:	1	
	Dop bics.	1 ¹	
		1000	
	Response Timer:	1000	
		Lease	
	Broadcast Pause:	1000	
	Inter-Char Timeout:	U	
]			

5. Right Click on Micro 830, then select **Build**.



Programming the Controller for Modbus Communication with a PowerFlex 4

This section will show you how to program the Micro830 for Modbus messaging with a PowerFlex 4.

1. Start by creating a new ladder diagram program by right clicking on **Program**. Move the cursor over the **Add** tab and select **New LD :Ladder Diagram**.



2. A new ladder icon will appear as shown:



3. Double click on the new ladder icon:



4. Open the **Toolbox** tab if it is not already open

Project Organizer 🔍 🗸 🛱 🗙	UntitledLD-POU Micro830 PowerFley 4 1*	Toolbox 🗸 🕂 🗙
Name: PwerFlex_4*		🗉 Ladder
		R Pointer
Micro830		~~어 Rung
Programs		-O Return
		->> Jump
		Branch
Local Var		-O- Direct Coll
		- Reverse Coll
Global Variables		-GF Set Coll
DataTypes		-Reset Coll
		-Rulse Falling Edge Coil
Eunction Blocks		+ F Direct Contact
PowerFlex 4 1*		-I/- Reverse Contact
		-IP- Pulse Rising Edge Contact
		⊣∾⊢ Pulse Falling Edge Contact
		Elock
		🖻 General
		Drag an item onto this text to add it to the
		toolbox.
	Culput • • • • ×	
	Show output from:	
1		Pevice Toolbox Toolbox
Ready		14 I

UntitledLD-POU Micro830 PowerFlex 4_1*	Toolbox	
1	E Ladder	
	Rung	
	-≫ Jump	
	🗂 Branch	
	-O- Direct Coil	
Instruction Block Selector: N/A	-Ø- Reverse Coil	
Project 8/03c3e9 3e8c 4020 9457 77/1e2/d00fc : 2080LC3016Qw/BA	-9- Set Coil	
	-®- Reset Coil	
Name 🐒 Category 났 Lype	-@- Pulse Rising Edge Coil	
	-®- Pulse Falling Edge Coil	
Antimetic Substraction of two or more	+ ⊢ Direct Contact	
	+/I- Reverse Contact	
	내라 Pulse Rising Edge Contact	
	-INF Pulse Falling Edge Contact	
	Block	
Parameters	🖯 General	

5. Drag and drop a **Block** on the rung. The **Instruction Block Selector** will now open:

6. Type in **MSG** in the text box under Name and **MSG_MODBUS** will appear:

60 4	🔤 Instruction Block Selector: N/A						
	Project_8f03c3e9_3e8c_4020_9457_77f1e2fd00fc : 2080LC3016QWBA						
	Name 隆	Category	1 Ту	ре		Ī .	
		- ·	▼ <i>d</i> t* ▼	at a			
	MSG_MODBUS	Communications	10	F Senda	a modbus message.		
	. (
	•]	<u>•</u>		



7. Double click on the **MSG_MODBUS** and the following function block will appear:

8. To use the block, you need to configure it. To find help on the instruction blocks, in this case the **MSG Modbus**, go to Help, Search, click on Local Help, and enter **MSG Modbus** in the search box.

🕜 MSG_MODBUS - Online Help - Micr	osoft Document Explorer
File Edit View Tools Window	Help
🗄 🚱 Back 🔘 📓 🛃 🍓 🗚 🗐 🕢 H	ow Do I 👻 🔍 Search 🔥 Index 🧒 Contents 🔀 H
Index - 🕈 🗙	MSG_MODBUS Search
Filtered by:	URL: ms-help://CCW.v10/Coll_ISa5_acf/lrsb_ISa5/
(no filter)	
Look for:	Connected Components Workbench
msg modbus	MSG_MODBUS
MSG_MODBUS function block multiplication operator MUX4B function MUX8B function naming conventions constants function blocks functions programs variables Neg operator NOT operator NOT operator NOT_MASK function operators 1Gain addition AND ANY_TO_BOOL ANY_TO_BOOL ANY_TO_DATE ANY_TO_DINT ANY_TO_DWORD ANY_TO_INT ON_TO_LINT ON_TO_LINT	Description: This function block can be used to send a M MSG_MODB NR Cancel Error Local ErrorID Target LocalA

Parameter	Parameter Type	Data Type	Description
IN	Input	BOOL	If Rising Edge (IN turns from FALSE to TRUE), start the function block with the precondition that the last operation has been completed.
Cancel	Input	BOOL	TRUE - Cancel the execution of the function block.
LocalCfg	Input	MODBUSLOCPARA See <u>MODBUSLOCPARA</u> <u>Data Type</u> .	Define structure input (local device).
TargetCfg	Input	MODBUSTARPARA See <u>MODBUSTARPARA</u> <u>Data Type</u> .	Define structure input (target device).
LocalAddr	Input	MODBUSLOCADDR	Define local address (125 words).
Q	Output	BOOL	TRUE - MSG instruction is finished. FALSE - MSG instruction is not finished.
Error	Output	BOOL	TRUE - When error occurs. FALSE - No error.
ErrorID	Output	UINT	Show the error code when message transfer failed. See <u>MSG_MODBUS Error Codes</u> .

9. Here you will find the information on the inputs and outputs of the block.

10. For the Cancel parameter, click on the upper part of the blue box and double click on the input from the Micro830 you want to assign, in this case, **Input 0** will be selected.



11. To create the other variables for the function block, double click on the <u>bottom</u> of the next blue box which will open the **Local Variables**.

🔡 Yariable Selector					
Name Ty		Global Scope — Micro830	_	Local Scope	
User Global Variables - Micro	330 Local Variables - U	UntitledLD Sys	tem Variables	s - Micro830 1/0 -	Micro830 D
Name	Data Type	Dimension	Alias		
- A*	MODBUSL 🔻 🔫	× ∂**	* A *		
*	τ.				
•					>
				OK	Cancel

12. If the **MSG_MODBUS_1** variable is not showing, click on the filter as shown below.



13. We now need to create variables for the other function block inputs. Click on the light blue box to the right of the asterisk. Type in **LocalCfg**. Tab over to Data Type.

U	lse	r Global Variables - Micro	830 Local Variab	iles - l	JntitledLD	Sys	stem Variable	s - Micro830
		Name	D ata Type		Dimensi	on	Alias	
		<i>~</i> 0 ₹ *			*	A*	- A*	
		MSG_MODBUS_1	MSG_MODBUS	-				
	Ø	LocalCfg	BOOL	•				
Ľ		s	9					

14. Type in **MODBUSLOCPARA**. See step 9 for where this data type assignment came from. You will note as you begin typing, the name will populate. Pay attention to the last half of the word to ensure you have the correct data type. Press Enter.

ſ	Use	r Global Variables - Micro	830 Local Variables -	UntitledLD	Sys	stem Varia	bles - Micro830
		Name	Data Type	Dimensi	on	Alias	
		- o#*	× ==	•	A*	- A*	
		MSG_MODBUS_1	MSG_MODBUS 🔹				
	Þ	ElecalCfg ElecalC	MODBUSLOCPAR 👻				
	*		*				

15. Type in **TargetCfg** in the light blue box to the right of the asterisk. Type in **MODBUSTARPARA** under data type. Press Enter.

U	se	r Gl	obal Variables - Micro	830	Local Vari	ables	- Ur	titledLD	Syste	m Variable	es - Micro830
			Name		Data Ty	ре		Dimen	sion	Alias	
			× 0€*						* A *	- A*	
		Ŧ	MSG_MODBUS_1	MSG	_MODBU:	5	•				
		÷	LocalCfg	MOD	BUSLOCE	PARA	*				
	Σ	Ŧ	TargetCfg	MOD	BUSTAR	PARA	•				
	ŧ						*				

16. Type in **LocalAddr** in the light blue box to the right of the asterisk. Type in **MODBUSOCADDR** under data type. Hit Enter and then click OK on this window to go back to the function block view.

ĺ	Use	r Global Variables - Micro	830 Local Variables - Un	ititledLD	Syste	m Variable	es - Micro830
		Name	Data Type	Dimen	sion	Alias	
		<i>~ 0</i> ₽*	* ==		* A*	- A*	
		■ MSG_MODBUS_1	MSG_MODBUS 🔹 🔹				
		🗄 LocalCfg	MODBUSLOCPARA 💌				
		🗉 TargetCfg	MODBUSTARPARA 🔻				
	Þ	🛨 LocalAddr	MODBUSLOCADDR 💌				
	*		*				



17. Assign the appropriate variables to each of the Input boxes by clicking on top of the box for each and select the corresponding variable.

18. Complete the selection to look like this.



UntitledLD-POU*	Toolbox 👻 🕈 🗙
	🗉 Ladder
	R Pointer
	ને Rung
	-🗢 Return
	->> Jump
	🗂 Branch
	-O- Direct Coil
d Divital los	-Ø- Reverse Coil
o_orgina_np-	-6- Set Coil
	-®- Reset Coil
LocalCfg	-@- Pulse Rising Edge Coil
	-®- Pulse Falling Edge Coil
	+ F Direct Contact
🔛 Variable Selector	+/- Reverse Contact
	-IP- Pulse Rising Edge Contact
Name type a local scope Local scope	⊣N⊢ Pulse Falling Edge Contact
	: Block
	🗉 General
User Global Variables - Micro830 Local Variables - UnititedLD System Variables - Micro830 1/0 - Micro830 Defined Words -	
Name Data Type Dimension Alias Comm	There are no usable controls in this
	add it to the toolbox.
*	
OK Cancel	

19. To trigger the message, drag and drop a **Direct Contact** to the left of the msg function block from the Toolbox as shown below. Notice the **Variable Selector** will appear.

20. In the Variable Selector, click on the I/O – Micro830 tab.

🔜 V	ariable Sele	ctor							×
Nam	e	Type BOOL			Global Scope Micro830	•	Local Sco N/A	ope	
User	Global Variables - Micro	830 Local Va	ariables - Un	titledLD Syste	m Variables	- Micro830 1/0 -	Micro830	efined Words - Cc💶 🗌	
	Name	Data	Гуре	Dimension	Alias			Comr 📥	
		BOOL	× ==	• A*	* A*				
	_IO_EM_DO_00	BOOL	•						
	_IO_EM_DO_01	BOOL	•						
	_IO_EM_DO_02	BOOL	•						
	_IO_EM_DO_03	BOOL	•						
	_IO_EM_DO_04	BOOL	•						
	_IO_EM_DO_05	BOOL	•					Y	
								▶	
							<u></u>]	K <u>C</u> ancel	

21. Double click on the Input you need to trigger the message. In this case, select by double clicking _IO_EM_DI_01 and the selector will close.

Use	r Global Variables - Micro	830 🗍 Local Variables
	Name	Data Type
	* <i>0</i> **	BOOL 🔹
	_IO_EM_DO_00	BOOL
	_IO_EM_DO_01	BOOL
	_IO_EM_DO_02	BOOL
	_IO_EM_DO_03	BOOL
	_IO_EM_DO_04	BOOL
	_IO_EM_DO_05	BOOL
	_IO_EM_DI_00	BOOL
	_IO_EM_DI_01	BOOL
	_IO_EM_DI_02	BOOL
	_IO_EM_DI_03	BOOL
	_IO_EM_DI_04	BOOL
	_I0_EM_DI_05	BOOL



22. After assigning the Direct Contact, the ladder now looks like this. Double click (bottom of the box) on one of the Local Variable Inputs to Display the Variable Selector.

23. Once the Variable Selector window appears, complete the following steps:

- a. Expand the Local Variables created (LocalCfg, TargetCfg...).
- b. You may have to use the scroll bar at the bottom of the variable tab to see the Initial Values. For ease of use, you can move the **Initial Value** column by dragging the top of the column and moving it next to Data Type.
- c. Set up the variables by clicking on the initial value field for each variable and enter the values shown in steps i, ii, and iii. For more information on the initial values refer to the message instruction CCW Help file:
 - i. Channel = 2 (2 is for the embedded serial port and 5 9 would be for different slot numbers the serial port could be located)
 - ii. Cmd = 3 (3 is for read holding registers and 16 would be for writing multiple registers. For a complete list and description of all the commands refer to the message instruction CCW Help file.)
 - iii. For more information on PowerFlex 4 address and node settings refer to drives Publication 22A-UM001I-EN-E.

🔛 Variable Selector					<u>- 0 ×</u>
Name Type USINT	Global Scope Micro830	•	Local Scope UntitledLD	• <u> </u>	•
Local Variables - UntitledLD System V	ariables - Micro830 1/	0 - Micro830 De	fined Words - C	ontroller	
Name	Data Type	Initial Value	Dimension	Alias	
- <i>d</i> **	× E	· A*		× ∂**	
💽 🖸 LocalCfg	MODBUSLOCPAR -				
LocalCfg.Channel	UINT	5			
LocalCfg.TriggerType	USINT	0			
LocalCfg.Cmd	USINT	3			
LocalCfg.ElementCnt	UINT	1			≡
TargetCfg	MODBUSTARPAR 🔻				
TargetCfg.Addr	UDINT	8449			
🌈 TargetCfg.Node	USINT	100			
🖃 🕀 LocalAddr	MODBUSLOCADD 🚽	·	2		
*	-				_
			OK		Cancel

Configuring the Embedded Serial Port on the Micro830.

1. For the embedded serial port on the Micro830, click the Serial Port under Communication Ports, and change the Driver to Modbus RTU. If necessary, change the other properties to match the screen shot below.



Micro830

General	Properties		
Memory	100 M		
Communication Ports	Driver:	Modbus RTU	~
- Serial Port		-	
USB Port	Baud Rate:	19200	~
Date and Time			
Interrupts	Parity:	None	~
Modbus Mapping	Unit Address:	0	
Embedded I/O			
🖹 Plug-In Modules	Modbus Role:	Modbus RTU Master	~
2080-SERIALISOL			
2080-IF4			

2. Open the Advanced settings and select RS485 for Mode.

: General	Properties			
Memory				
Communication Ports	Driver:	Modbus RTU		
	David Datas	locoo 📃		
USB Port	Daud Rate;	9000		
Interrupts	Parity:	None		
Startup/Faults				
Modbus Mapping		In		
Embedded I/O	Advanced Settings			
2080-SERIALISOL	Media	R5485	RTS Pre-Delay: 0	_
2080-IF4	- Houlds			
	Data Bits:	8	RTS Post-Delay: U	
	Stop Bits:	1		
	Perpopse Timer:	1000		
	Response filler.	1.000		
	Broadcast Pause:	1000		
	Inter-Char Timeout:	ln In		•

3. Go to the Variables section and change the LocalCfg Channel to 2.

		Name	Data Type	Initial Value
		<i>▼</i> A*	* ==	- A*
Ŧ	MSG_N	IODBUS_1	MSG_MODBUS 🔹 🔻	
•	LocalC	fg	MODBUSLOCPAR. 👻	
		LocalCfg.Channel	UINT	2
		LocalCfg.TriggerType	USINT	0
		LocalCfg.Cmd	USINT	3
		LocalCfg.ElementCnt	UINT	1
	Target(Cfg	MODBUSTARPAR 👻	
		TargetCfg.Addr	UDINT	8449
		TargetCfg.Node	USINT	100
+ I	LocalA	ddr	MODBUSLOCADD 👻	

4. Build the project.

Cabling the Controller to a PowerFlex 4 Class Drive

This quick start will show you how to physically connect a PowerFlex 4 class Drive to the Micro830.

1. Wire the Micro830 Modbus Plugin to the drive as shown below. The PowerFlex 4 comes with a built-in RS485 DSI port where Modbus Communication is available. In order to communicate between the Micro 830 and PowerFlex 4, the Serial Communication port on the Micro830 will be configured as RS485 for the communication media. Below is the basic connection between the Micro830 and PowerFlex 4 using the recommended Belden 3105A twisted pair cable.



2. Connect the USB cable to the USB port shown below to establish communication between the PC and the Micro830. If this is the first time you connect to the controller, refer to the Getting Started Guide, Pub# 2080-QR001B-EN-P, to establish communications between RSLinx and a Micro830 via USB.



Testing Modbus Communication with a PowerFlex 4 Class Drive

In this section, you will test the Modbus message created in the previous sections. The rung below will trigger an input on the controller that will execute a Modbus read message.

1. Start with the rung shown below. For information on how to create this rung refer to the previous sections starting on Chapter 6.



2. Build and download to the controller. If you are not familiar with the download steps, refer to the Getting Started Guide, Pub# 2080-QR001B-EN-P.

- _IO_EM_DI_01 MSG_MODB. MSG_MODB. IN Q _IO_EM_DI_00 False False Error Cancel LocalCfg 5 Local... ErrorID TargetCfg 8449 Target... LocalAddr 0 LocalA_
- 3. Once connection is established with the controller, start debugging by pressing 🕨 in the top menu bar. The following should show the Modbus message in debug mode:

4. Trigger Input **_IO_EM_DI_01** to read a Modbus message from the controller. Notice that the function block input **LocalAddr** now is displaying a WORD value.



5. Stop debugging by clicking on the stop button on the top menu. To use the WORD value read from the drive in the previous step, a copy of this value needs to be assigned to a new variable using a **1 gain** function block. In the **Toolbox**, click and drag a **Block** as shown below to the end of the rung.



6. Type 1, select the 1 gain function block and click OK.

• Instruct	on Block S	elector: 1 gain				
- Controller	2080LC301	6QWBA				
	Name	Category	y <u>1</u> T	уре		
1 gaii		Arithmetic	- *	Assig	nment of o	ne variable to a
						<u> </u>
- Parameter	s					
	N	ame	Data Type	Direc	tion 1	Dimension
		- o#*	- A*		- A*	- A*
î			•	VarInput	*	
0	1			VarOutput	-	
*			*		Υ.	
•						•
				Chow Darar	toro	
nstance:	<u> </u>			Show Falai	neters	
nputs:	<u>×</u> 1		$\overline{\mathbf{v}}$	EN / ENO		
cope:	UntitledLD					
					ОК	Cancel

7. A new **1 gain** function block has now been added. Double click on the bottom of the input box to add the desired input to be copied.



8. Once the Variable Selector opens, select the local variable LocalAddr[1] as shown below and click OK.

<u>Note:</u> LocalAddr[1] is the variable holding the WORD value read in step 4.

¥arial	ble Selector								_ 0
Vame - LocalA	sddr[1]	ype MORD	•	Global S Micro83	cope)	•	Local Scope		•
ser Glo	obal Variables - Micro830	Local Variables	- UntitledLD	System \	/ariables - Micro	830 170) - Micro830 🗍 D	efined Wo	ord:
	Name		D ata T	уре	Dimension		Alias		Init
		- A*		- A *	× ₀₹*			▼ A*	
Ŧ	MSG_MODBUS_1		MSG_MODB	US 🝷					
+	CopyOfLocalCfg		MODBUSLO	CPAR 🝷					
Ð	LocalCfg		MODBUSLO	CPAR 🝷					
•	TargetCfg		MODBUSTA	RPAR 👻					
	LocalAddr		MODBUSLO	CADD 🔻					
	🕨 🔲 LocalAddr[1]	N	WORD						
	LocalAddr[2]	43	WORD						
	LocalAddr[3]		WORD						
	LocalAddr[4]		WORD						
	LocalAddr[5]		WORD						
	LocalAddr[6]		WORD						
	LocalAddr[7]		WORD						
	LocalAddr[8]		WORD						_
	LocalAddr[9]		WORD						
	LocalAddr[10]		WORD						
	LocalAddr[11]		WORD						
_	LocalAddr[12]		WORD						_
							ОК	C	ancel

9. Double click on the output box.



10. Create a new local variable as shown below. For this quick start type *Logic_Status* as the name of the variable, select **WORD** as its data type and click **OK**.

Variable Selector Name Type Logic_Status BOOL		Global Scope — Micro830	Local Scope UnitledLD	
User Global Variables - Micro830 Local Variables	s - UntitledLD System \	/ariables - Microl	830 1/0 - Micro830 Defined W	ords - Controller
Name	Data Type	Dimension	Alias	Initial Value
- <i>A</i> *	* <u>=</u> =	- A*	- A	<i>~ ∂</i> **
■ MSG_MODBUS_1	MSG_MODBUS 🔹			
 CopyOfLocalCfg 	MODBUSLOCPAR 👻			
🕢 🕢 LocalCfg	MODBUSLOCPAR 🔻			
🛨 王 TargetCfg	MODBUSTARPAR 👻			
🖃 🕀 LocalAddr	MODBUSLOCADD 🔻			
IFI CTU 1	CTII 👻			
🎜 Logic_Status	WORD T			
*		1		
				•
			OK	Cancel

11. To interpret the data read by the message refer to **Appendix A**, *Reading (03) Logical Status Data* table. This table can be used to determine the meaning of each of the 16 bits of the WORD. Start by adding a **Rung** as shown below.



12. Select, drag and drop a Direct Contact from the toolbox to the start of the newly added rung.



13. Once the Variable Selector displays, under the **Local Variables** click on the filter as shown below to display the variables.

Variable Selector			ilobal Scope — Micro830	Local S Untitled	cope	
User Global Variables - Micro830 Local Varia	ables - UntitledLD) System V	ariables - Micro	330 1/0 - Micro830 Def	ined Words	- Controller
Name	Data	Туре	Dimension	Alias	In	itial Value
	A* BOOL	* == *	~ off*		* <i>0</i> ₹*	~ A*
					ок	Cancel

14. Select the *Logic_Status* variable created before.



15. Go to the name field and type *Logic_Status.0* for the bit *0* of the Logic Status WORD. Then click OK.

🖁 ¥ariable Selector								_ 🗆 ×
Name Logic_Status.0]	Type WORD	<u> </u>	Glo	bal Scope rro830	•	Local Scop)e	<u>•</u>
User Global Variables - Micro830	Local Variables	- UntitledLD	System \	/ariables - Micro	830 1/0 - M	icro830 Def	ined Wo	rds - Cc 🔸 🕨
Name		D ata T	уре	Dimension		Alias		Initial Valu
	- A*			- A*			~ A *	- 6
*			Υ.					
•								
						OK		Cancel



16. Select, drag and drop a **Direct Coil** to the end of the rung as shown below.

17. In the Variable Selector I\O Micro830 tab, select _IO_EM_DO_00 as the embedded output for this rung. Then click OK.

🔡 Yariable Selector				_ 🗆 ×
Name Type _IO_EM_DO_00	Global Scope Micro830		Local Scope	•
User Global Variables - Micro830 Local Variables	: - UntitledLD System \	/ariables - Micro830	1/0 - Micro830 De	fined \
Name	Data Type	Dimension	Alias	
- A*	BOOL - 📼	- A*		- A*
• 🚺 _IO_EM_DO_00	BOOL 👻			
_IO_EM_DO_01 13	BOOL 🔹			
_IO_EM_DO_02	BOOL 🔹			
_IO_EM_DO_03	BOOL 🔹			
_IO_EM_DO_04	BOOL 🔹			
_IO_EM_DO_05	BOOL 🔹			
_IO_EM_DI_00	BOOL 🔹			-
				\rightarrow
			ОК	Cancel



18. Repeat steps 11 – 17 from this quick start to add an additional Logic Status rung that will read bit 7 to determine whether the drive is faulted or not.

19. Build and download at this time. Now you are ready to start debugging by clicking the button on the top menu bar and the ladder should look as shown below.



20. Trigger Input 1 (_IO_EM_DI_01) as shown below and notice that the Modbus message will return the status WORD shown before to be 1037. Now it can be shown that at the bit level *Logic_Status.0* is true stating that the Drive is READY and showing this status by enabling Output 0 (_IO_EM_DO_00).

1		
	LIO_EM_DI_01 MSG_MODB. IN EN EN EN EN EN EN EN EN EN	
	LocalAddr[1] False Cancel Error False 1037 (1 o1 1037	
	Local_tg 5 Local_ ErrorID 0	
	8449 Target.	
2	1037 LocalA	
2	Logic Status 0)
3	Logia Status 7	

Sending a Write Message to the Drive to Start, Stop and Change Speed

In this section, you will change the Modbus message to write to the drive and be able to control the drive. To do this, you will need to add some additional ladder logic to be able to Start, Stop and set a Speed on the drive. This section of the quick start will modify the modbus message used in the previous section to read the drive's status.

- 1 _IO_EM_DI_01 _IO_EM_DI_00 _IO_EM_DI_00 Cancel Error LocalCfg LocalCfg Local. ErrorID TargetCfdLocalCfg : AnyStructure ______Target.. LocalA..
- 1. Start by double clicking on the *LocalCfg* input as shown:

2. Once the **Variable Selector** displays change the **Initial Value** of the *LocalCfg.Cmd* from a read value of 3 to a value of **16** for writing holding registers. The *LocalCfg.Channel* should be set to 2 if you are using the embedded serial port, otherwise enter the slot number of the serial port you are using.

📕 Variable Selector					
Name Type User Global Variables - Micro830 Local Variables	s - UntitledLD Sustem 1	Global Scope Micro830 Variables - Micro8	30 1/0 - Miero	Local Scope UntitledLD	ontroller
Name	Data Type	Initial Value	Dimension	Alias	
- A*	MODBUSL 👻 🖛	~ A*	× 0€*		- A*
📃 🗟 LocalCfg	MODBUSLOCPAR 🔻				
LocalCfg.Channel	UINT	5			
LocalCfg.TriggerType	USINT	0			
🖉 🗾 LocalCfg.Cmd	USINT	16			
LocalCfg.ElementCnt	UINT	2			
*	×.				
				ОК	Cancel

 Also Change the *TargetCfg.Addr* to 8193 as shown below. Remember that the Micro830 uses 1-based Modbus addressing, therefore for writing logic command data (Appendix A), use 8192 +1 = 8193 (Reference Publication 22A-UM001I-EN-E).

🖶 Variable Selector					
Name Type Model Model	STARPARA 💌	Global Scope Micro830	_	Local Scope UntitledLD	•
User Global Variables - Micro830 Local Vari	ables - UntitledLD Sys	tem Variables - Mi	cro830 1/0 - M	icro830 Defined Words	- Controller
Name	Data Type	Initial Value	Dimension	Alias	
- 02 *	MODBUS 🕆 🖙	⊤ ∂ ‡*	× ∂**		- A*
	MODBUSTARPAR 💌				Rea
TargetCfg.Addr	UDINT	8193			Rea
		100	-		Rea
				ОК	Cancel
4. Now that the Modbus message will be writing to the drive, create a rung that will trigger a 1 gain copy function block to hold the value that will be written to the drive. Therefore toggling the bit in front of the 1 gain will determine if the controller sends a Start, Stop, or Speed change message. Start by selecting, dragging and dropping a **Rung** as shown below.



5. Select, drag and drop a Block.



6	Type 1.	select the '	1 gain	function	block	and	click	OK.
υ.	iype i,	301001 1110	i gani	Turiotion	DIOCK	anu	CHOR	UIV.

Instruc	tion Bloc	k Selea	tor: 1 gain					
- Controlle	r : 2080LC	3016Q\	VBA					
	Name	0	Category	v <u>1</u>	Гуре			
		• A*		▼ A**	▼ 0¶*			
1 ga	in ^µ ⊽	A	Arithmetic		ÖP	Assignmer	nt of o	ne variable to a
						1		<u>•</u>
Paramete	are							
raramete	30							
		Name	•	D ata Type	•	Direction	1	Dimension
			<i>~</i> A*	* d	£^		° off*	- A*
	1] -1				Vari	nput	· ·	
*	01				van	Julpul		
								• • •
istance:				-	Shov	/Parameter	s	
nuts:	-	_		 V	EN Z	ENG		
	Untitlac	10						
cope.	Unideo	LD						
						OK		Cancel

7. Select, drag and drop a **Direct Contact** as shown below.

/UntitledLD-POU*	▼ X	Toolbox – 🕂 🗙
		🖃 Ladder
		Revenue Pointer
		How Rung
_IO_EM_DI_01 (MSG_MODB.		-🗢 Return
MSG_MODB_		->> Jump
		🗂 Branch
		+O+ Direct Coil
		-Ø- Reverse Coil
LocalCfg		-®- Set Coil
		-®- Reset Coil
Local. ErrorID		- -(P- Pulse Risina Edae Coil
TargetCfg		-®- Pulse Falling Edge Coil
Tarnet		H F Direct Contact
localádir		-1/1- Reverse Contact
		HPH Pulse Rising Edge Contact
		HNH Pulse Falling Edge Contact
2		T Block
		- General
		There are no usable controls in this group.
		Drag an item onto this text to add it to the
		00000.
		Previce Techor (1) Techor
<u> </u>		

IO_EM_DI_01		Global Scope — Micro830	(ariables - Mioro	Local Scope	T Define 1
Name		Data Type	Dimension	Alias	D CHIT!
	- ∂ ** BC)OL - 🖂	- A*		- 6
_IO_EM_DO_00	BOOL				
_IO_EM_DO_01	BOOL				
_IO_EM_DO_02	BOOL				
_IO_EM_DO_03	BOOL				
_IO_EM_DO_04	BOOL				
_IO_EM_DO_05	BOOL				
_I0_EM_DI_00	BOOL				
	BOOL	. *			
_I0_EM_DI_02	BOOL				
_IO_EM_DI_03	BOOL				
_I0_EM_DI_04	BOOL				
_IO_EM_DI_05	BOOL				
_I0_EM_D1_06	BOOL				
_IO_EM_DI_07	BOOL				
_IO_EM_DI_08	BOOL				
_IO_EM_DI_09	BOOL				
*		*			

8. From the I/O – Micro830 tab, select _IO_EM_DI_01 (Input1).

9. Select, drag and drop a **Direct Coil** as shown below.



10. Create a variable **Run_Fwd** as shown below. Then click **OK.**

🔡 Yariable Selecto	r				_ 🗆 ×
Name	Type	Global Scope	•	Local Scope UntitledLD	T
User Global Variable	es - Micro830 Local Varia	bles - UntitledLD Syste	m Variables - Micro	5830 1/0 - Micro830	Defin
	Name	Data Type	Dimension	Alias	
	*	A* -	- A		~ A *
🛛 🕞 MSG_MOD	BUS_1	MSG_MODBUS	•		
🕒 🕒 LocalCfg		MODBUSLOCPAR	*		
🕢 🕢 🕀 🕞		MODBUSTARPAR	•		
🛨 LocalAddr		MODBUSLOCADD	-		
2 Run_Fwd	Ţ	BOOL			
-					
4					
				ОК	Cancel

11. Now click on top of the input box for the **1 gain** function block and type **18** as the WORD that will be copied to trigger the <u>forward command</u>. Refer to **Appendix A** *Writing (06) Logic Command Data* to determine why when the forward command bit is high it equals decimal 18.



12. Double click on the output box.



Variable Selector Name Type LocalAddr[1] WORD		Global Scope	Local Scope UntitledLD	
User Global Variables - Micro830 Local Varia	ables - UntitledLD System \	/ariables - Micro8	30 1/0 - Micro830 Defined Wo	ords - Controller
Name	Data Type	Dimension	Alias	Initial Value 📥
-	A* • A*	- A*	- <i>d</i> f*	- A*
■ MSG_MODBUS_1	MSG_MODBUS 🔻			
💽 🖅 LocalCfg	MODBUSLOCPAR -			
🕞 🕞 TargetCfg	MODBUSTARPAR 🔻			
LocalAddr	MODBUSLOCADD -			
LocalAddr[1]	WORD			
LocalAddr[2]	WORD			
LocalAddr[3]	WORD			
LocalAddr[4]	WORD			_
	W000			
			OK	Cancel

13. Once the Variable Selector displays, select LocalAddr[1]. Then click OK.

14. Now that the new rung is complete, click on top of the direct contact to the left of the Modbus message and select the new variable created **Run_Fwd**.



2		Run Fwd
		0
	18 LocalAddr[1]	-0
3		
	_IO_EM_DI_02 I gain	Run_Rev
		-0
	LocalAddr[1]	
4		
	_IO_EM_DI_03 1 gain	Stop
	EN ENO	-0
	LocalAddr[1]	

15. Just as the previous rung, add the two rungs shown below to trigger the reverse command (**Run_Rev**) and the stop command (**Stop**).

16. Also, create two additional rungs that will trigger two preset frequencies as shown below. This example uses pre-set frequencies 1 and 2 as **Freq** and **Freq2** respectively, or as what can be consider a slow and fast speed for the drive depending on the application.

5		
	_IO_EM_DI_04 1 gain	Freq
	EN ENO	_0
	20480 LocalAddr[1]	Ŭ
	- i1 o1 -	
6		
	_IO_EM_DI_05 1 gain	Freq_2
		<u> </u>
	24576 LocalAddr[1]	-

UntitledLD-POU*	
Run_Fwd MSG_MODB MSG_MODB	▲ Ladder N Pointer I → Rung → Return → Jump
LocalCfg LocalCfg LocalAddr LocalAddr	Direct Coll O Direct Coll O Direct Coll O Posec Coll O Posec Coll O Posec Coll O Pose Rising Edge Coll O Puse Falling Edge Coll I Direct Contact I/+ Reverse Contact I/+ Reverse Contact I/+ Reverse Contact I/+ Reverse Contact

17. Select, drag and drop a branch under the Run_Fwd direct contact as shown below.

18. Now add a new **Direct Contact** to the newly added branch.



19. Once the Variable Selector displays, select the variable Run_Rev. Then click OK.

📴 Va	riable Selector							<u>_ X</u>
Ru	me n_Rev	Type BOOL	_	Global Micro8	Scope	_	Local Scope UntitledLD	•
Use	r Global Variables	- Micro830	.ocal Variables	- UntitledLD	System V	/ariables - Micro	830 1/0 - Micro830	Defin_◀ ▶
		Name		Data 1	уре	Dimension	Alias	
			× ∂**	BOOL	• ==	* o#*		- A*
	Run_Fwd			BOOL	*			
	Run_Rev			BOOL	~			
	Stop			BOOL	*			- 1
	Freq			BOOL	•			
	Freq_2			BOOL	*			
*					*			
								· · · · ·
							ок	Cancel

1		
	Run_Fwd	
	Run_Rev	
	Stop	LocalCfg
	Freq	TargetCfg Target
	Freq_2	LocalAddr LocalA.

20. Repeat steps 18 and 19 for Stop, Freq, and Freq_2 branches.

21. Your ladder is now complete and it should look like this:



- 5 _IO_EM_DI_04 1 gain Freq EN ENO О ł 20480 LocalAddr[1] i1 01 6 _IO_EM_DI_05 Freq_2 1 gain ENO 11 ΕN Ο 24576 LocalAddr[1] o1
- 22. Now you are ready to start debugging. Save, build and download your project. Click on the button on the top menu bar. Trigger input 4 (_IO_EM_DI_04) to write pre-set frequency 1 (Freq).

23. Now that a frequency is set, trigger input 1 (**_IO_EM_DI_01**) to enable the forward command in the drive as shown below. This verifies the message is working as intended feel free to toggle the other bits to test Stop, Run_Rev, and Freq_2.



Appendix A

1Gain

Description:

Directly links the input to output. When used with a Boolean negation, moves a copy of i1 to o1 .



Arguments:

Parameter	Parameter Type	Data Type	Description
EN	Input	BOOL	Function enable. When EN = TRUE, execute the direct link to an output computation. When EN = FALSE, there is no computation.
i1	Input	BOOL - DINT - REAL - TIME - STRING - SINT - USINT - INT - UINT - UDINT - LINT - ULINT - DATE - LREAL - BYTE - WORD - DWORD - LWORD	Input and output must use the same format.
01	Output	BOOL - DINT - REAL - TIME - STRING - SINT - USINT - INT - UINT - UDINT - LINT - ULINT - DATE - LREAL - BYTE - WORD - DWORD - LWORD	Input and output must use the same format.
ENO	Output	BOOL	Enable out.

Reading (03) Logic Status Data

The PowerFlex 4 Logic Status data can be read via the network by sending Function Code 03 reads to register address 8448 (Logic Status).

	Logic Status				
Address (Decimal)	Bit(s)	Description			
	0	1 = Ready, 0 = Not Ready			
	1	1 = Active (Running), 0 = Not Active			
	2	1 = Cmd Forward, 0 = Cmd Reverse			
	3	1 = Rotating Forward, 0 = Rotating Reverse			
	4	1 = Accelerating, 0 = Not Accelerating			
	5	1 = Decelerating, 0 = Not Decelerating			
	6	1 = Alarm, 0 = No Alarm			
0440	7	1 = Faulted, 0 = Not Faulted			
0440	8	1 = At Reference, 0 = Not At Reference			
	9	1 = Reference Controlled by Comm			
	10	1 = Operation Cmd Controlled by Comm			
	11	1 = Parameters have been locked			
	12	Digital Input 1 Status			
	13	Digital Input 2 Status			
	14	Not Used			
	15	Not Used			

Writing (06) Logic Command Data

The PowerFlex 4 drive can be controlled via the network by sending Function Code 06 writes to register address 8192 (Logic Command). P036 [Start Source] must be set to 5 "RS485 (DSI) Port" in order to accept the commands. In addition to being written, register address 8192 can be read using Function Code 03.

Logic Command				
Address (Decimal)	Bit(s)	Description		
	0	1 = Stop, 0 = Not Stop		
	1	1 = Start, 0 = Not Start		
	2	1 = Jog, 0 = No Jog		
	3	1 = Clear Faults, 0 = Not Clear Faults		
		00 = No Command		
	5.4	01 = Forward Command		
	5,4	10 = Reverse Command		
		11 = No Command		
	6	Not Used		
	7	Not Used		
		00 = No Command		
	9,8	01 = Accel Rate 1 Enable		
		10 = Accel Rate 2 Enable		
8192		11 = Hold Accel Rate Selected		
	11,10	00 = No Command		
		01 = Decel Rate 1 Enable		
		10 = Decel Rate 2 Enable		
		11 = Hold Decel Rate Selected		
		000 = No Command		
		001 = Freq. Source = P036 [Start Source]		
		010 = Freq. Source = A069 [Internal Freq]		
	14 10 10	011 = Freq. Source = Comms (Addr 8193)		
	14,13,12	100 = A070 [Preset Freq 0]		
		101 = A071 [Preset Freq 1]		
		110 = A072 [Preset Freq 2]		
		111 = A073 [Preset Freq 3]		
	15	Not Used		

Chapter 7 – Using Connected Components Workbench with Temperature Controllers

Hardware & Software Versions Used

- Temperature Controller, 900-TC8 or 900-TC16
- Simple Temperature Control Connected Component Building Block, Pub# CC-QS005A-EN-P
- Appropriate communication module for the 900-TC per application
 - 900-TC8COM
 - 900-TC16NACCOM

Configuring and Programming the Controller for Modbus Communications to a 900-TC Temperature Controller

This chapter will show you how to configure and program the Micro830 controller with the 2080-SERIALISOL and the 900-TC termperature controller.

 With the assumption you have the Micro830 controller selected in the project file, you can now go to the controller window and select the 2080-SERIALISOL plug-in card. For more information on how to create a new Micro830 project, review the Getting Started Guide (Pub# 2080-QR001B-EN-P).



2. Click the down arrow, and select Modbus RTU

Micro830	Remote Mode:	ProgramRun	Major Fault: Controller Mode:	1 <u>111</u> 17	Connect
Download Upload		208	80-LC30-16Q	WB	
Micro830	Alter-Braffy A	0000000			
General Memory Communication Ports Serial Port USB Port Date and Time Interrupts	Propert Drive Bauc	ies er: d Rate:	Modbus RTU ASCII Modbus RTU Shutdown	•	
USB Port Date and Time Interrupts Protection Startup/Eaulto	Parit	1 Rate: :y:	None	×	

3. Change the rest of the parameters to the following listed below, using the down arrows.

Properties		
		_
		+ 2080-SERIALISOL
Driver:	Modbus RTU 💌	
Baud Rate:	9600	
basarrator		
Devile		
Parity:	None	
Unit Address:	U	
Modbus Role:	Modbus RTU Master	
+ Advanced Settings		

4. Expand the Advanced Settings and change the **Media** to **RS485**. Leave the rest of the parameters as shown below.

ΓΞ	Advanced Settings			
	Media:	R5485	RTS Pre-Delay:	
	Data Bits:	8	RTS Post-Delay:	0
	Stop Bits:	1		
	Response Timer:	1000		
	Broadcast Pause:	1000		
	Inter-Char Timeout:	0		<u>-</u>

5. Right click on Micro830, then select Build.



6. Right click on programs. Move the cursor over the Add tab, to New LD: Ladder Diagram.



The following will now appear.



7. Double click the ladder icon.

Project Organizer 🔍 🚽 🗙	UntitledLD-POU Micro830	▼ ×
Name: 900TC_Q5*	1	
Micro830		
Program		
Global Variables		

8. Open the Toolbox tab if it is not open.



9. Click and drop a **Block** on the rung. The Instruction Block Selector window will now open.

💐 Connected Components Workb	ench	
File Edit View Build Debug	Tools Communications Window Help	
! 🎦 💕 🎟 - 🖬 👗 🛍 🛍	ッ・ページー 💷 🔍 🔹 👘 🐘 👘 📳 🔮 👘 🖓 👘 👘 関 🔮	🍓 🍓 🖋 🖋 Cycle Timing
	B I ∐ — K A Ē Ē Ē Ē Ę	
Project Organizer 🔍 👻 🕂 🗙	UntitledLD-POU* Micro830	Toolbox
Name: 900TC_QS*		🗆 🕒 Ladder
74444		Revenue Pointer
Micro830		
Programs		-🔿 Return
		-≫ Jump
	Instruction Block Selector: N/A	🗖 Branch
	Project_d0dada6c_4b06_436c_be3e_3083b35c8773 : 2080LC3016QWBA	-O- Direct Coil
Local Va	Name ⁰ Category ¹ Type	-Ø- Reverse Coil
Global Variables		-6- Set Coil
	- Arithmetic	-®- Reset Coil
	* Arithmetic 100 Multiplication of two or more	-@- Pulse Rising Edge Coil
No. and a straight day	/ Arithmetic 📅 Division of two or more inte	-®- Pulse Falling Edge Coil
Function Blocks	+ Arithmetic , 🖅 Addition of two or more inte	- I ⊢ Direct Contact
		-//- Reverse Contact
		HP- Pulse Rising Edge Cont
	Parameters	-IN- Pulse Falling Edge Con
		Block
1		

10. Type in MSG in the cell under Name.

	📑 Ins	Instruction Block Selector: N/A						
	Pro	ject_d0dada6	с_4ЬО6_	436c_be3e_3083b	35c8773	: 2080LC	C3016QWBA	
l		Name	0	Category	1	Туре		
I		📝 msg	▼ #*		▼ A*	▼ #*		
		(All) (Custom) (Empty) - *		nmunications		6	Send a modbus message	
		+						

11. MSG_MODBUS will now appear as one of the available instruction blocks.

🔤 Instruction Block Sele	Instruction Block Selector: N/A						
Project_d0dada6c_4b06_	436c_be3e_3083b3	35c8773 :	: 2080LC	3016QWBA			
Name 🔍	Category	1	Туре				
🗾 📝 msg 🛛 🔻 🗨		▼ A [*]	▼ A*				
MSG_MODBUS	Communications		č.	Send a modbus message			
•					►		



12. Double click on MSG_MODBUS and the following will appear.

 To use the block, you need to configure it. To find help on the instruction blocks, in this case the MSG Modbus, go to Help, Search, click on Local Help, and enter MSG Modbus in the search box.

Ø MSG_MODBUS - Online Help - Microsoft Document Explorer							
File Edit View Tools Window	Help						
🔅 😋 Back 🌍 📓 🛃 🍓 A [‡] 🛛 🔞 H	🔾 😋 Back 🏐 📓 🛃 🦓 A 🕯 🛛 🥑 How Do I 👻 🔍 Search 🚺 Index 🧒 Contents 👿 H						
Index 🚽 🗸	MSG_MODBUS Search						
Filtered by:	URL: ms-help://CCW.v10/Coll ISa5 acf/lrsb ISa5/-						
(no filter)							
Look for:	Connected Components Workbench						
msg modbus	MSG_MODBUS						
MSG_MODBUS function block multiplication operator MUX4B function MUX8B function naming conventions constants function blocks functions programs variables Neg operator not equal operator NOT operator NOT operator NOT_MASK function operators 1Gain addition AND ANY_TO_BOOL ANY_TO_BOTE ANY_TO_DATE ANY_TO_DINT ANY_TO_INT ANY_TO_INT	Description: This function block can be used to send a M MSG_MODB NR Q Cancel Error Local ErrorID Target LocalA						
ANY_TO_INT	Anoumonte						

14. Here you will find the information on the inputs and outputs of the block.

Parameter	Parameter Type	Data Type	Description
IN	Input	BOOL	If Rising Edge (IN turns from FALSE to TRUE), start the function block with the precondition that the last operation has been completed.
Cancel	Input	BOOL	TRUE - Cancel the execution of the function block.
LocalCfg	Input	MODBUSLOCPARA See <u>MODBUSLOCPARA</u> <u>Data Type</u> .	Define structure input (local device).
TargetCfg	Input	MODBUSTARPARA See <u>MODBUSTARPARA</u> <u>Data Type</u> .	Define structure input (target device).
LocalAddr	Input	MODBUSLOCADDR	Define local address (125 words).
Q	Output	BOOL	TRUE - MSG instruction is finished. FALSE - MSG instruction is not finished.
Error	Output	BOOL	TRUE - When error occurs. FALSE - No error.
ErrorID	Output	UINT	Show the error code when message transfer failed. See <u>MSG_MODBUS Error Codes</u> .

Arguments:

15. For the Cancel parameter, click on the upper part of the blue box and double click on the input from the Micro830 you want to use, in this case, Input 0.



16. To create the other variables for the function block, double click on the bottom of the next blue box and open up the Local Variables.

User Global Variables - Micro830	Local Varia	ables - UntitledLD	Syst	em Variables - M	1icro830 1/0 - Mi
Name		Data Typ	e	Dimension	Alias
	⊤ A*	MODBUSL	* ==	- A*	- A*
*			•		

17. We now need to create variables for use with the function blocks. Click on the light blue box to the right of the asterick. Type in LocalCfg. Tab over to Data Type.

User Global Variables - Micro830			Local Variables	s - UntitledLD	Syst	em \
		Name		Data T	уре	
			- A*		•	
		MSG_MODBUS_1		MSG_MODBL	JS	•
	I	LocalCfg		BOOL		•

18. Type in MODBUSLOCPARA. See step 15 for where this data type assignment came from. You will note as you begin typing, the name will populate. Pay attention to the last half of the word to ensure you have the correct data type. Hit enter.

Use	er Global Variables - Micro830	Local Variables	s - UntitledLD Syste	em V
	Name		Data Type	
		- A*	*	
	MSG_MODBUS_1		MSG_MODBUS	•
	🛨 LocalCfg		MODBUSLOCPAR	Ψ.
*				•

19. Type in TargetCfg in the light blue box to the right of the asterisk. Type in MODBUSTARPARA under data type. Hit enter.

Us	er Global Variables - Micro830	Local Variables	s - UntitledLD Syste	em V	aria
	Name		Data Type		D
		~ A*	*		
	📧 MSG_MODBUS_1		MSG_MODBUS	•	
	王 LocalCfg		MODBUSLOCPAR	•	
	📧 TargetCfg		MODBUSTARPAR	•	

20. Type in LocalAddr in the light blue box to the right of the asterisk. Type in MODBUSOCADDR under data type. Hit enter.

Use	r Global Variables - Micro830	Local Variables	s - UntitledLD	System \	/ariabl
	Name		Data T	уре	Din
		- A*			
	MSG_MODBUS_1		MSG_MODBL	JS 🝷	
	∃ LocalCfg		MODBUSLOC	PAR 🝷	
	🗉 TargetCfg		MODBUSTAF	RPAR 🔻	
	🕒 LocalAddr		MODBUSLOC	add 🝷	
*				*	

21. Assign the appropriate variables as listed below by clicking the top half of the box and selecting the variable.



22. Complete the selection to look like this.



- 23. Now, to trigger the message, the addition of a direct contact will be used.
- 24. Click, hold and drop the direct contact to the left of the msg function block from the Toolbox.



25. Click the I/O – Micro830 tab.

🔡 Va	💀 Variable Selector							
Name	e	BOOL	ariables . I In		ilobal Sco Micro830 m Variable	Local Scope		
	Name	Data	Type	Dimension	Alias			
	· 0#*	BOOL	× ==	- A*	* A*	*		
	_IO_EM_DO_00	BOOL	•					
	_IO_EM_DO_01	BOOL	•					
	_IO_EM_DO_02	BOOL	•					
	_IO_EM_DO_03	BOOL	•					
	_IO_EM_DO_04	BOOL	•					
	_IO_EM_DO_05	BOOL	•			×		
						→		
						<u> </u>		

- 26. Click and hold the right side of the Name column. This changes the grouping and order of the Digital Inputs and Outputs.
- 27. Double click on the Input you need to trigger the message. After the double click, the selector will close, and the program ladder will open.

Use	User Global Variables - Micro830 🗍 Local Variables						
	Name	Data Type					
	- A*	BOOL 🔹					
	_IO_EM_DO_00	BOOL					
	_IO_EM_DO_01	BOOL					
	_IO_EM_DO_02	BOOL					
	_IO_EM_DO_03	BOOL					
	_IO_EM_DO_04	BOOL					
	_IO_EM_DO_05	BOOL					
	_IO_EM_DI_00	BOOL					
	_IO_EM_DI_01	BOOL					
	_IO_EM_DI_02	BOOL					
	_IO_EM_DI_03	BOOL					
	_IO_EM_DI_04	BOOL					
	_IO_EM_DI_05	BOOL					



28. Set up the parameters as shown below by clicking on the Initial Value box for each variable. You may have to use the scroll bar at the bottom of the variable tab to see the Initial Value column. For ease of use, you can move the Initial Value column by click and holding the top of the column and moving the column to where you want it. These settings are based on the 900-TC settings used and found in Publication CC-QS005A-EN-P. Information on the message variables can be found in the CCW Help.

/	Un	titl	edL	D-VAR Micro830 UntitledLD-POU					
				Name	Initial ¥alue	Data Type	Dimension	Alias	
					× A*	- A*	- A*	- A*	- A*
		-	MS	G_MODBUS_1			MSG_MODBUS 🔻		
		-	Loc	alCfg			MODBUSLOCP4 👻		
				LocalCfg.Channel		5	UINT		
				LocalCfg.TriggerType		0	USINT		
				LocalCfg.Cmd		3	USINT		
				LocalCfg.ElementCnt		10	UINT		
		-	Tar	getCfg			MODBUSTARP4 👻		
				TargetCfg.Addr		1	UDINT		
				TargetCfg.Node		17	USINT		
		Ŧ	Loc	alAddr			Modbuslocai 👻		
	*						Ψ.		

29. Build and download the program.

Configuring the Embedded Serial Port on the Micro830

1. For the embedded serial port on the Micro830, click the Serial Port under Communication Ports, and change the Driver to Modbus RTU. If necessary, change the other properties to match the screen shot.

Micro830



; General	Properties	
- Memory		-
Communication Ports	Driver:	Modbus RTU 💌
Serial Port		
USB Port	Baud Rate:	9600 💌
- Date and Time		
Interrupts	Parity:	None
Startup/Faults		
Modbus Mapping	Unit Address:	0
Embedded I/O		
🗄 Plug-In Modules	Modbus Role:	Modbus RTU Master
- 2080-SERIALISOL		
2080-IF4		

2. Open the Advanced settings and select RS485 for Media.

General	Properties			
Memory				
Communication Ports	Driver:	Modbus RTU		
Serial Port				
USB Port	Baud Rate:	9600		
- Date and Time	Devileur	Name		
	Parity;	INone		
		0		
Embedded I/O	- Advanced Settings			
🖻 Plug-In Modules			RTS Pre-Delay:	
- 2080-SERIALISOL	Media:	R5485		
2080-IF4			RTS Post-Delay: 0	
	Data Bits:	8	. ,	
	Stop Bits			
	Deep bies.			
	Response Timer:	1000		
	Broadcast Pause:	1000		
	Inter-Char Timeout:	0		

3. Go to the variables window and change the LocalCfg Channel to 2.

Micro830 UntitledLD-YAR UntitledLD-POU							
	Name	Initial Value	Data Type				
	- A	- of	- A*				
🔚 🕀 MSG_	MODBUS_1		MSG_MODBUS 👻				
🔄 🖃 Locald	Efg		MODBUSLOCP4 👻				
	LocalCfg.Channel	2 🥌	UINT				
	LocalCfg.TriggerType	0	USINT				
	LocalCfg.Cmd	3	USINT				
	LocalCfg.ElementCnt	10	UINT				
🛨 🕂 Targe	tCfg		MODBUSTARP4 👻				
🔚 🕀 LocalA	Addr		Modbuslocai 👻				
*			*				
+ MSG_ - Local(MODBUS_1 Cfg LocalCfg.Channel LocalCfg.TriggerType LocalCfg.Cmd LocalCfg.ElementCnt tCfg Addr	 2 0 3 10 	MSG_MODBUS MODBUSLOCP# UINT USINT UINT MODBUSTARP# MODBUSLOCAI				

4. Build the project.

Cabling the Controller for a 900-TC Temperature Controller and Testing the Controller Program.

This section will show you how to configure and program the Micro830 controller with the 2080-SERIALISOL and the 900-TC temperature controller.

1. For this section, program the 900-TC as listed in the Simple Temperature Control Connected Components Building Block, Publication CC-QS005A and Temperature Controllers User Manual, Publication 900-UM007D.

Follow the steps below for the 900-TC communication setup:

- Communication protocol: Mod
- Communications unit no.: 17

This parameter sets a unique unit number for each temperature controller, letting the host identify the temperature controller during communication. When two or more temperature controllers are used, do not use the same unit number. This building block uses unit numbers (nodes) 17...24.

- Communication baud rate: 9.6 kbps
- Communications parity: NoNE
- Send data wait time: 20

2. Follow the basic wiring connections shown below, select the appropriate drawings based on the 900-TC you are using. When using the 2080 SERIALISOL module, ground the shield/drain to the chassis of the controller.



900-TC8 & 900-TC16

Terminator (120Ω, 1/2 W)

Note: If using the 1763-NC01 cable, wire the same for the 900-TC, connect the following way.



900-TC8 & 900-TC16

Terminator (120Ω, 1/2 W)



Note: Grounding Your Analog Cable

Use shielded communication cable, such as the Belden #3105A. The Belden #3105A cable has two signal wires (White/Blue Stripe and Blue/White Stripe), one drain wire, and a foil shield. The drain wire and foil shield must be grounded at end of cable.

3. Assuming you have created the program from the previous sections starting in Chapter 7, built and downloaded the program on the Micro830, you can now proceed.

- 4. Verify the program by running the debugger.
- 5. View the variable tab. Energize input 1 on the Micro830. You should get something similar to this. LocalAddr(2) is the process variable, LocalAddr(3) is the lower status word, LocalAddr(4) is the upper status word, and LocalAddr(6) is the set point.

ntit	ledLD-VAR Micro830) UntitledLD-POU					
		Name	Logical Value	Physical ¥alue	Lock	Initial ¥alue	Data Type
		₹ A*	- 0 ⁴	- A*	- A	· At	<i>▼ 0</i> ₽*
-	MSG_MODBUS_1						MSG_MODBUS 🝷
-	LocalCfg						MODBUSLOCP# 🔻
-	TargetCfg						MODBUSTARP# 👻
-	LocalAddr						Modbuslocai 👻
	LocalAddr[1]		0	N/A			WORD
	LocalAddr[2]		82	N/A			WORD
	LocalAddr[3]		768	N/A			WORD
	LocalAddr[4]		24576	N/A			WORD
	LocalAddr[5]		0	N/A			WORD
	LocalAddr[6]		75	N/A			WORD
	LocalAddr[7]		0	N/A			WORD

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 USA, Tcl: (1) 414.382.2000, Fax: (1) 414.382.4444 Europe/Middle East/Africa: Rockwell Automation, Vorstlaan/Boulevard du Souverain 36, 1170 Brussels, Belgium, Tcl: (32) 2 663 0600, Fax: (32) 2 663 0640 Asia Pacific: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tcl: (852) 2887 4788, Fax: (852) 2508 1846