Micro800[™]and Connected Components Workbench[™]

Getting Started Guide



Copyright © 2012 Rockwell Automation Technologies, Inc. All Rights Reserved. This program is protected by U.S. and International copyright laws as described in the about box

Powered by Visual Studio





Copyright © 2011 Rockwell Automation, Inc.

Table of Contents

- Chapter 1: Software Requirements and Installing the Software
- Chapter 2: Create a New CCW Project
- Chapter 3: Configuring Controller Plug-in Modules
- Chapter 4: Creating a User Defined Function Block
- Chapter 5: Creating a New Ladder Diagram Program
- Chapter 6: Downloading a Project and Troubleshooting Faults
- Chapter 7: Testing a Running Program
- Chapter 8: Saving and Closing a Project
- Chapter 9: Connecting to Existing Controllers
- Chapter 10: Using Micro810 Smart Relay Functionality (no software)

Introduction

In this document you will learn the basic Micro800 and CCW functions by creating a working "Traffic Light" project. The Traffic Light project is a four way intersection control program that monitors traffic flow with car sensors. You will need the software and hardware specified below.



Requirements

Hardware Requirements:

Micro810, 2080-LC10-12QWB.

Micro830, 2080-LC30-16QWB

Standard USB Cable

Software Requirements:

Connected Components Workbench (CCW), Release 1.01 and higher

RSLinx, v 2.57 and higher

Chapter 1 – CCW Software Requirements and Installing the Software

Hardware & Software Versions Used

Software Requirements

Connected Components Workbench software has been successfully tested with the following operating systems versions and service packs:

Operating system compatibility

Microsoft® Windows® XP® SP3 or later (except Home editions)

Microsoft Windows Vista® SP2 or later

Microsoft Windows 7®

Hardware requirements

To use Connected Components Workbench effectively, your personal computer must meet the following hardware requirements:

Component	Minimum requirement	Recommended
Processor	Pentium 3 or better	Pentium 4 or better
RAM Memory	512 MB	1.0 GB
Hard Disk Space	3.0 GB free	4.0 GB free
Optical Drive	DVD-ROM	DVD-ROM
Pointing Device	Any Windows-compatible pointing device	Any Windows-compatible pointing device

Installing the CCW Software (Standard Version)

This chapter will show you how to install the CCW software standard version.

1. Insert the Connected Component Workbench (CCW) DVD-ROM. In case the software doesn't launch automatically. Explore the content of the CD and double click on



the Connected Components Workbench Setup icon CCW5etup.exe

2. Once CCW setup wizard is launched, follow through the setup instruction. Select setup language and click **Continue.**

Connected Components Workb	ench Setup 1.02.00
	Welcome to the Connected Components Workbench Setup Wizard. Select setup language: English (United States)





Copyright © 2011Rockwell Automation, Inc.

3. Select the type of installation (typical installation recommended) and click Next.

🚰 Connected Components Workbench Sett	ıp 1.02.00	×			
Connected Components Workbench Setup Connected Components Workbench 1.02.00					
Configuring Products:	Select one or more features to install	USTERN, THINN, SOLVE,			
Connected Components Workbench Rockwell Software Common	C C Lustom	ram features will be installed.			
	Rockwell Components RSLinx Classic Connected Components Workbench v1.02 ControlFLASH v10.00.11 Virtual COM Port Device Driver v6.3a	2.00 View Release Notes for Selected Product			
	Hard drive space required to install:				
Connected Components Workbench Step 1 of 2 - Feature Selection	Rockwell Components All selected products	2520.8MB 2520.8MB			
Explore CD Contents		< Back Next > Cancel			

4. Check the boxes to create shortcuts on desktop and then click Next.

7 Connected Components Workbench Set	up 1.02.00	×
Connected Com	ponents Workbench Setup Connected Components Workbench 1.02.00	
Configuring Products:	Create shortcuts on the desktop for one or more of the following pro	oducts
Connected Components Workbench Rockwell Software Common	Connected Components Workbench v1.02.00	
Connected Components Workbench Step 2 of 2 - Desktop shortcuts		
Explore CD Contents	<u><b< u="">a</b<></u>	ick <u>N</u> ext > Cancel

5. Enter customer information in the fields and then click Next.

🚏 Connected Components Workbench Setu	p 1.02.00		×
Rockwell Softwa Connected Co	are Common Set mponents Workbench 1.02	up .00	
Configuring Products:	Please enter the following information		listen, think, solve:"
✓ Connected Components Workbench	User Name:		
Rockwell Software Common	Rockwell Automation, Inc.		
	Organization:		
	Rockwell Automation, Inc.		
Rockwell Software Common			
Step 1 of 3 - Customer Information			
Explore CD Contents		< <u>B</u> ack	Next > Cancel

6. Review the terms in the license agreement. Accept and click Next.

🚏 Connected Components Workbench Setup	1.02.00 ×
Rockwell Softwa Connected Cor	mponents Workbench 1.02.00
Configuring Products:	Please read the following license agreement carefully
 Connected Components Workbench 	END-USER LICENSE AGREEMENT (Rev 12/2009)
Rockwell Software Common	
	IMPORTANT-READ THIS AGREEMENT CAREFULLY: This End-User License Agreement ("EULA") is a legal contract between you (either an individual or a single entity) ("You or Licensee") and Rockwell Automation, Inc. ("Rockwell Automation") for the software product or products that Rockwell Automation licenses to You, which includes computer software and may include associated media ("Software"). Rockwell Automation may also license to You printed materials, and "online" or electronic documentation ("Documentation"). An amendment or addendum to this EULA may accompany the Software. ROCKWELL AUTOMATION IS WILLING TO LICENSE THE SOFTWARE AND DOCUMENTATION TO YOU ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS AND CONDITIONS IN THIS AGREEMENT. YOU ACCEPT AND AGREE TO BE
Rockwell Software Common	○ I do not accept the terms in the license agreement
Step 2 of 3 - License Agreement	
Explore CD Contents	< <u>B</u> ack <u>N</u> ext > Cancel

7. Use default installation location or change it by clicking **Change.** Once installation location is defined, click **Install.** Full installation process may take up to 2 hours.

🚏 Connected Components Workbench Setur	1.02.00		×
Rockwell Softwa Connected Con	are Common Setup		
Configuring Products:	Select location where all products will be installed		LISTEIN, THIINK, SOLVE,
✓ Connected Components Workbench	C:\Program Files\Rockwell Automation	Change	
Rockwell Software Common			
Rockwell Software Common			
Step 3 of 3 - Install Location			
Explore CD Contents		< <u>B</u> ack	Install Cancel

8. Installation in progress, the whole process may take up to 2 hours. If this is a new installation, it will go through all four sections shown below.



9. It is recommended to install ControlFLASH v10.00.11. This is necessary for downloading firmware. To install ControlFLASH, click **Next.**

🚏 Connected Components Workbench Setup 1.02.00	X
ControlFLASH Setup Connected Components Workbench 1.	02.00
Installing Products:	@ControlFLASH 10.00.11
✓ Rockwell Automation USB CIP Driver v3.16.02 ✓ RSLinx Classic v2.59.00	Welcome to the ControlFLASH 10.00.11 Setup Wizard
✓ Connected Components Workbench v1.02.00 ♥ ControlFLASH v10.00.11	The installer will guide you through the steps required to install ControlFLASH on your computer.
Installing ControlFLASH v10.00.11	WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.
	Cancel Ageck Next>
Explore CD Contents	<back next=""> Cancel</back>

10. Review terms in the license agreement. Select "I Agree" and click Next.

🔂 ControlFLASH 10.00.11		
License Agreemen	t	Factory Talk
Please take a moment to read th Agree'', then "Next". Otherwise	ne license agreement now. If yo click "Cancel".	u accept the terms below, click "I
END-USER LICE IMPORTANT-READ TH Agreement ("EULA") is single entity) ("You or Automation") for the so licenses to You, which is media ("Software"). Ro materials, and "online"	NSE AGREEMEN a legal contract between y Licensee") and Rockwell oftware product or product ncludes computer software ockwell Automation may or electronic documenta	Γ (Rev 12/2009) LLY : This End-User License rou (either an individual or a Automation, Inc. ("Rockwell is that Rockwell Automation and may include associated also license to You printed tion ("Documentation"). An
◯ I <u>D</u> o Not Agree		
	Cancel	< Back <u>N</u> ext >

11. Use default installation location or change it by clicking **Browse**. Click **Next** to continue.

🔂 ControlFLASH 10.00.11			
Select Installation Folde	ſ	Factory	
The installer will install ControlFLASH to t To install in this folder, click "Next". To in	he following folder. Istall to a different fo	lder, enter it below o	r click "Browse".
Eolder: C:\Program Files\ControlFLASH\			Browse
	Cancel	< <u>B</u> ack	<u>N</u> ext >

12. For new users, it is <u>not recommended</u> to check "Enable FactoryTalk Security". Click **Next** to continue installation.



13. Once Installation is complete, click Close. It is not necessary to launch ControlFLASH at the moment.



14. Installation is now complete for CCW (V1.02). Click Finish.

🚰 Connected Components Workbench Setup	1.02.00
Rockwell Softwa Connected Components We	are Setup orkbench 1.02.00
Installation Summary: ✓ Rockwell Automation USB CIP Driver v3.16.02 ✓ RSLinx Classic v2.59.00 ✓ Connected Components Workbench v1.02.00 ✓ ControlFLASH v10.00.11	Connected Components Workbench Setup Complete IMPORTANT: Registering your interest in Connected Components Workbench will allow us to notify you of key information, updates and related products. If you would like to register, please go to http://ncckwellautomation.custhelp.com/app/utils/create_account Thank you.
Explore CD Contents	< Back. Einish Cancel

Chapter 2 – Creating a New CCW Project

Creating a New CCW Project

This chapter will show you how to create a new CCW project.

1. Start by opening the Connected Components Workbench software.



Double click on the **Connected Components Workbench** icon Workbench or from the **Start** menu, select **Programs > Rockwell Automation > CCW > Connected Components Workbench**.

2. Begin a new project by clicking on **Catalog** and expanding the **Controllers** folder in the **Device Toolbox**, which is located on the right-hand side of the Workbench screen. Double click on a Controller. For this example we will select the **2080-LC30-16QWB**.



3. A new Micro830 project based on this controller has now been created. The Micro830 should show up in the **Project Organizer** on the left-hand side of the Workbench screen.



4. You can Change the **Name** for the project on top of the **Project Organizer.** For this example we will use, Traffic Light.



5. Select **File→Save Project as...** In the "Save Project As..." box, type in the **Name** and **define** save location then click **OK.** A file with the assigned name will be created at the defined location.

🔡 Save Project As				×
Name:	Traffic Light		_	_
Location:	C:\Documents and Settings\Labuser\My Documents\CCW\Traffic Light		Browse	
		ОК	Cancel	

Chapter 3 – Configuring Controller Plug-in Modules

Configuring Controller Plug-ins

This chapter will show you two examples of plug-ins and how they are configured in the controller.

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

Micro830		•
Micro830	Remote C Program Major Fault: S Connect Disconnecte	d
Download Upload	2080-LC30-16QWB	Manuals Help
Micro830	Slot 1 Slot 2 Slot 2	
General	Properties	
Memory	Vendor Name: Allen-Bradley	
- Serial Port	Catalog ID: 2080-LC30-16QWB	
USB Port	Firmware OS Rev: 1.3	
Interrupts Startup/Faults	Name:	
Modbus Mapping — Embedded I/O ⊡ Plug-In Modules	Description:	
<pre> < Empty > < Empty ></pre>	Boot Revision: 0.0	

2. Assuming Plug-in slot 1 has an isolated serial port module installed in it. Right click on the graphic of the first plug-in slot and select **2080-SERIALISOL**.



3. Assuming Plug-in slot 2 has 4-channel analog input module installed in it. Right click on the graphic of the second plug-in slot and select **2080-IF4**.

Alten-Bradley	0000000	00000
-704E1 AN ONC	D Also-Bradley	2080-TE2
/and come		2080-IF4
Micro830	1 2 1 4	2080-OF2
	A DESCENT REPORTS	2080-TC2
	0000	2080-RTD2
		2080-TRIMPOT6
۲	0000000	2080-SERIALISOL

4. Notice that the Micro830 graphic changes to show the installed plug-ins. Now, if you needed to change the **Channel 0 Input Type** for the analog input module, just select under **Properties** from either **Current** or **Voltage** as well as the ability to change the **Frequency** and **Input State**.



5. In the event that an RTC module needs to be configured, repeat step 2, but select the **2080-MEMBAK-RTC** and the screen should look similar to the following.

Note: RTC Plug-ins 2080-MEMBAK-RTC can ONLY be in slot 1 of the Micro830.

MICro830	▼ ×
Micro830	Remote Program Major Fault: SP Mode: Run Controller Mode: Disconnected Disconnected
Download Upload	2080-LC30-16QWB
Micro830	
General Memory Communication Ports Serial Port USB Port Date and Time Interrupts Startup/Faults Modbus Mapping Embedded I/O Plug-In Modules C800-MEMBAK-RTC C Empty >	Properties Real Time Clock Real Time Clock Enabled Date: Saturday , January 01, 2000 Time: 12:00:00 AM Sync RTC with PC Clock Show Current.RTC Parameters Download RTC Parameters Download RTC Parameters Back Up Whole Restore Whole Back Up Fault Log Restore Fault Log

6. For more information on available plug-ins refer to:

http://ab.rockwellautomation.com/Programmable-Controllers/Micro800-Plugin-Modules

Chapter 4 – Creating a User Defined Function Block (UDFB)

Creating a User Defined Function Block (UDFB)

This chapter will show you how to create a User Defined Function Block for controlling a Traffic Light. User Defined Function Blocks are not required for every project but can be used to capture repetitive code for easy re-use throughout your project.

1. Under Project Organizer, right click on Function Blocks, select Add and select New LD : Ladder Diagram.



2. Right click on **UntitledLD** and select **Rename**.



3. Type **TRAFFIC_CONTROLLER_FB** (the name given to the Ladder Diagram file will be the name of the UDFB) and Enter.



4. The general Traffic Controller algorithm for the function block to be implemented in the ladder diagram program is as follows:

If the traffic light over at **North (N)** and **South (S)** is **red** and a car is waiting at **either side** of the road for **5 seconds**, the lights over at **East (E)** and **West (W)** change from **green to yellow**, **hold for 2 seconds** and then it change from **yellow to red**. As the lights at **East (E)** and **West (W)** change to **red**, lights over at **North (N)** and **South (S)** change to **green**.



 The most important thing to define up front when creating a UDFB is what inputs are required, and what outputs will be produced, by this function block. These inputs and outputs are defined in the function block's Local Variables. Therefore, under TRAFFIC_CONTROLLER_FB, double click on Local Variables.



6. Right click on the top row and select Reset Settings so all of the necessary columns are displayed.



7. For this UDFB, we need four Boolean inputs (for a car sensor in each of the four positions in the intersection) and six Boolean outputs (for red, yellow and green lights in each of the two directions). The inputs are entered in with **Direction VarInput** and the outputs are entered in with **Direction VarInput**. Enter in the variables with the **Names**, **Data Types** and **Directions** as shown.

4	TR	AFFIC_CONTROLLER_FB-VA	R							
		Name	Data	Туре	Dimension	Alias	Comment	Initial Value	Directio	'n
		- A		- A*	- A*	- A*		- A*	-	A*
		N_CAR_SENSOR	BOOL	-					VarInput	+
		S_CAR_SENSOR	BOOL	•					VarInput	-
		E_CAR_SENSOR	BOOL	-					VarInput	+
		W_CAR_SENSOR	BOOL	•					VarInput	*
		NS_RED_LIGHTS	BOOL	*					VarOutput	-
		NS_YELLOW_LIGHTS	BOOL	•					VarOutput	*
		NS_GREEN_LIGHTS	BOOL	*					VarOutput	-
		EW_RED_LIGHTS	BOOL	•					VarOutput	*
		EW_YELLOW_LIGHTS	BOOL	-					VarOutput	•
		EW_GREEN_LIGHTS	BOOL	•					VarOutput	•

8. Double click on TRAFFIC_CONTROLLER_FB to begin editing the ladder logic program.





9. We want the first rung to work as follows:

lf:

the North/South Red Lights and East/West Green Lights are on, and

a car trips either the North Sensor or the South Sensor for at least five seconds,

Then:

change the East/West Lights from Green to Yellow.

10. Click on the **Toolbox** tab in the lower right-hand corner and click on the **+** in front of **LD** to list the available ladder instructions.



11. To implement the first bullet, we need two **Direct Contacts** in series (since the logic is North/South Red Lights AND East/West Green Lights). Click and drag a **Direct Contact** instruction from the **Toolbox** to the rung and release.



12. When you release the mouse button, the **Variable Selector** screen appears. Select **NS_RED_LIGHTS** and click **OK**.

🔛 Va	riable Selector				
Nar NS Loca	ne Type _RED_LIGHTS	9 DU ROLLER_FB (9	Global Micros	Scope	Local Scope TRAFFIC_CONTROLLE
	Name	Data Type	Dimension	Alias	Com
	- <i>d</i> t*	BO 💌 🖬	<i>~ ∂</i> ₹*	- A*	
	N_CAR_SENSOR	BOOL 🔹			
	S_CAR_SENSOR	BOOL 🔹 💌			
	E_CAR_SENSOR	BOOL 🔹			
	W_CAR_SENSOR	BOOL 🔹 💌			
	NS_RED_LIGHTS	BOOL 👻			
	NS_YELLOW_LIGHTS	BOOL 🔹 👻			
	NS_GREEN_LIGHTS	BOOL 🔻			
	EW_RED_LIGHTS	BOOL 🔹			
	EW_YELLOW_LIGHTS	BOOL -			
	EW_GREEN_LIGHTS	BOOL 🔹 👻			
*		Υ.			
					•
					OK Cancel

13. Similarly, add a second **Direct Contact** and assign **EW_GREEN_LIGHTS** to this contact. So far, your rung should look like this.



14. To implement the second bullet, we need two **Direct Contacts** in parallel (since it's North Car Sensor *OR* South Car Sensor). First, click and drag a **Branch** from the **Toolbox** to the end of the rung and release.

Local Variables-VAR TR	AFFIC_CONTROLLER_FB-PO	U*		▼ X	Toolbox	≁ ‡ ×
1					Ladder Pointer	
	NS_RED_LIGHTS	EW_GREEN_LIGHTS			⊖ Return	
					-≫ Jump	
			 		🖽 Branch	
					-O- Direct Coil	
					-Ø- Reverse Coil	

15. Click and drag a Direct Contact instruction from the Toolbox to the upper branch and release. Assign variable N_CAR_SENSOR to this contact. Then, click and drag a Direct Contact instruction from the Toolbox to the lower branch and release. Assign variable S_CAR_SENSOR to this contact. Now your rung should look like this.

AFFIC_CONTROLLER_FB-PO	U*	
NS_RED_LIGHTS	EW_GREEN_LIGHTS	N_CAR_SENSOR
		S_CAR_SENSOR

16. Next we need a 5 sec delay, so click and drag a **Block** instruction from the Toolbox to the right of the branch and release. The **Instruction Block Selector** screen appears. Under the **Category** column, select **Time** to list all the Time-based instructions. Select TON for On-delay timing and click OK.

Mark Instruction Blo	ck Selector: N/	(4	
Project : 2080-LC3	30-16QWB				
Na	ame 🙎	Category	1	Туре	
DOY TDF TOF TON		Time Boolean operations Communications Comparators Counter Data conversion Data Manipulation Input/Output	▼ <i>6</i> ^{4*}	∰ ∰ 100	Turn on output w Compute time diffe Off-delay timing On-delay timing
Parameters		Process Control Program Control String manipulation Time	Ţ		

17. Click on the **TON** block and hit the **F1** key to bring up context-sensitive help in order to get an explanation of the block inputs and outputs.



18. Click on the top of the **PT** input block and enter in a programmed/preset time of "T#5s" (for a time format of 5 seconds) as shown below and press enter key.



19. To implement the last bullet, we need two **Coils** in parallel (since we need to turn off the East/West green lights *AND* turn on the East/West yellow lights). First, click and drag a **Branch** from the **Toolbox** to the end of the rung and release.



20. Click and drag a Reset Coil instruction from the Toolbox to the upper branch and release. Assign variable EW_GREEN_LIGHTS to this coil (this will turn off the East/West green lights). Then, click and drag a Set Coil instruction from the Toolbox to the lower branch and release. Assign variable EW_YELLOW_LIGHTS to this coil (this will turn on the East/West yellow lights). Now your completed rung should look like this.

NS_RED_LIGHTS EW_GREEN_LIGHTS	N_CAR_SENSOR	IN	TON_1 TON	
	S_CAR_SENSOR	T#5s		EW_YELLOW_LIGH
		PT	ET 🗕	

- 21. Now document the rung by double clicking in the green block above the rung and entering in "If the North/South red lights and East/West green lights are on and a car trips either the North sensor or the South sensor for at least five seconds, then change the East/West lights from green to yellow."
- 22. We want the second rung to work as follows:

lf:

the East/West Yellow Lights are on for at least two seconds,

Then:

change the East/West Lights from Yellow to Red and the North/South Lights from Red to Green.

23. Enter in the second rung to look like the following.

the East/West yellow lights are on for at least two seconds, then change the East/W	est lights from yellow to red and the North/South lights from red to green.
V_YELLOW_LIGH TON_TON	EW_YELLOW_LIGH
T#2s	EW_RED_LIGHTS
	NS_RED_LIGHTS
	NS_GREEN_LIGHTS
	<u> </u>

24. The next two rungs have the same form as the first two rungs, so we will take advantage of cut and paste, then just edit the variable assignments for each instruction. Select the two rungs by clicking in the dark blue region to the left of rung 1, then hold the **Shift** key down and click in the dark blue region to the left of rung 2. Right click and select **Copy**.



25. Now select just rung 1, then right click and **Paste**. The two rungs are inserted prior to the original two rungs. Double click on the **TON** in rung 3 and within the **Instruction Block Selector** screen, edit the **Instance** from **TON_1** to **TON_3** and click **OK**.

🔤 Instru	Instruction Block Selector: TON				
Project	:: 2080-LC30-16QWB				
	Name	<u> </u>			
		▼ p#*			
ИТ 📃 📗	ND		Progra		
TC)F		Time		
🕨 🕨 TO	DN		Time		
ТС	DNOFF		Time		
Parame	eters				
	Name	Da	ta Typ		
	- A	*	- 0		
	Pdate	TIME			
	Redge	B00	Ľ		
	IN	B00	Ľ		
	PT	TIME	:		
	1111				
Instanc	e: TON_3		•		

Similarly, change the **Instance** of the **TON** in rung 4 from **TON_2** to **TON_4**.

26. In rungs 3 and 4, change each **EW** variable to **NS** and each **NS** variable to **EW**, so that the rungs appear as follows (don't forget to modify the rung comments to match!).

East/West red lights and North/South change the North/South lights from gr	green lights are on and a car trips either een to yellow.	the East sensor or the West sensor for at least	five seconds,
RED_LIGHTS NS_GREEN_LIGHTS	E_CAR_SENSOR	TON_3 TON	NS_GREEN_LIGHTS
	W_CAR_SENSOR	T#5s	NS_YELLOW_LIGH
North/South yellow lights are on for at	least two seconds, then change the Nor	h/South lights from yellow to red and the East/V	Vest lights from red to green.
ELLOW_LIGH	V.4		NS_YELLOW_LIGH
T#2s	°		NS_RED_LIGHTS
PT	ET		EW_RED_LIGHTS
			EW GREEN LIGHTS

27. We need to add one more rung to handle "initial conditions". When the program is first downloaded to the controller and run, none of the lights are initially turned on. This last rung will check for this condition (all lights off) and turn on the North/South red lights and the East/West green lights. Click and drag a **Rung** from the **Toolbox** to the white space below rung 4 and release. Click and drag six **Reverse Contacts** from the **Toolbox** onto the new rung and assign one **...LIGHTS** variable to each contact.

NS_RED_LIGHTS	NS_YELLOW_LIGH	NS_GREEN_LIGHTS	EW_RED_LIGHTS	EW_YELLOW_LIGH	EW_GREEN_LIGHTS
<u> </u>	/I	// _	// _	//	// _

28. Next, add two **Set Coils** in parallel to turn on **NS_RED_LIGHTS** and **EW_GREEN_LIGHTS**. Complete rung 5 by documenting its operation.

To initialize the program after the initial download, if all the lights are turned off, turn on the North/South red lights and the East/West green lights.									
NS_RED_LIGHTS	NS_YELLOW_LIGH	NS_GREEN_LIGHTS	EW_RED_LIGHTS	EW_YELLOW_LIGH	EW_GREEN_LIGHTS		NS_RED_LIGHTS		
<u> </u>	//	// _	// _	//	//		©		
							EW_GREEN_LIGHTS		

29. Finally, build and save the five-rung program.



Right click on TRAFFIC_CONTROLLER_FB in Project Organizer and select Build.


30. You should get verification in the **Output** window at the bottom center of the screen that the build succeeded.



Click the Save icon \blacksquare to save your work.

Chapter 5 – Creating a New Ladder Diagram Program

Creating a New Ladder Diagram Program

This chapter will show you how to create a new Ladder Diagram Program that uses the **TRAFFIC_CONTROLLER_FB** User Defined Function Block created in the previous chapter.

1. Under Project Organizer, right click on Programs select Add and select New LD : Ladder Diagram.







Copyright © 2009 Rockwell Automation, Inc.

2. Right click on UntitledLD and select Rename.



3. Type Traffic_Light_Control and Enter.



4. Double click on **Traffic_Light_Control** within **Project Organizer** to start editing the ladder logic program.

5. Click on the **Toolbox** tab in the lower right-hand corner and click on the **+** in front of **LD** to list the available ladder instructions

Toolbox 🗣 🗣	x
🗉 LD	
Rointer	
[···어 Rung	
-🔿 Return	
->> Jump	
🗂 Branch	
-O- Direct Coil	
-Ø- Reverse Coil	
-9- Set Coil	
-®- Reset Coil	
-®- Pulse Rising Edge Coil	
-®- Pulse Falling Edge Coil	
⊣⊢ Direct Contact	
-I∕I− Reverse Contact	
⊣P⊢ Pulse Rising Edge Contact	
⊣¤⊢ Pulse Falling Edge Contact	
Ellock	
🗉 General	
	_
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox	-



6. Click and hold on the **Block** instruction within the **Toolbox** and drag a Block onto the rung as shown.

7. When you release the mouse button, the instruction Block Selector screen ap

Main Struction Block Selector	: N/A		
Project : 2080-LC30-16QWB =			
Name	Category	1 Туре	
	A	▼ A* ▼ A*	
· ·	Arithmetic	ōP	Substraction of tv
×	Arithmetic	ōP	Multiplication of t
	Arithmetic	ÖP	Division of two or
+	Arithmetic		Addition of two or
J			
Parameters			
J			
Instance:	🔽 Scope: Traff	ic_Light_Control	
Inputs: 🛨 0	EN/EN		
			Show Parameters 🔽
		0	Cancel
			///

8. Under **Name**, type **tr** and note that only the instructions starting with **tr** are listed. Click on **TRAFFIC_CONTROLLER_FB** and notice that all of the **Parameters** associated with this function block are listed below it.

🔤 Instruction Block Selector: TR	AFFIC_COM	TROLLER_FB		_ 🗆 ×
Project : 2080-LC30-16QWB				
Name 9	Cat	egory 1	Туре	
tr 🗸		▼ A*	▼ A**	
TRAFFIC CONTROLLER	(User define	ed)	1	
TRIMPOT_READ	Input/Outp	ut	5	Read the Trimpot va
	Arithmetic		sī	Truncate decimal par
I I				•
Parameters				
Name	Data Typ	e Dimensio	n Alias	:
- Æ*	6	A* - 0	A* - d	t*
TON_1	TON	-		
TON_2 TON_2	TON	*		
■ TON_3	TON	•		
TON_4	TON	*		
N_CAR_SENSOR	BOOL	•		
S_CAR_SENSOR	BOOL	*		
E_CAR_SENSOR	BOOL	÷		
W_CAR_SENSOR	BOOL	*		
NS_RED_LIGHTS	BOOL	÷	_	
NS_YELLOW_LIGHTS	BOOL	*		
NS_GREEN_LIGHTS	BOOL	•		
EW_RED_LIGHTS	BOOL	*		
EW_YELLOW_LIGHTS	BOOL	-	_	
EW_GREEN_LIGHTS	BOOL	•		
*		*		
				×
		T (G 1 () (G	Cantral	
	Scop	e: Tramc_Light_	Control	
nputs: 🛫 4	EN	/ ENO 🗵		
				Show Parameters 🔽
			OK	Cancel

9. Make sure that EN/ENO is checked and then click **OK**. The Traffic Controller function block should be displayed on the rung as shown below.



10. By convention, function blocks list inputs on the left-hand side of the block and outputs on the righthand side of the block. In order to see the full names and data types of the variables that these inputs and outputs are associated with, move your cursor over the block - you should get the following listing.

TRAFFIC_CONTROLLER_FB TRAFFIC_CONTROLLER_FB_1	
Inputs EN: N_CAR_SENSOR: Bool S_CAR_SENSOR: Bool F_CAR_SENSOR: Bool	
ULT CAR_SENSOR: Bool	
ENO: NS_RED_LIGHTS: Bool NS_YELLOW_LIGHTS: Bool NS_GREEN_LIGHTS: Bool EW_RED_LIGHTS: Bool EW_YELLOW_LIGHTS: Bool	
EW_GREEN_LIGHTS: Bool	

- 11. The first function block input that connects directly to the ladder rung is the function block enable (EN) bit. The remaining four function block inputs are "real world" inputs that indicate whether a car is waiting at a red light in any of the four possible directions North, South, East and West. These inputs get mapped to four Boolean input variables local to the function block: N_CAR_SENSOR, S_CAR_SENSOR, E_CAR_SENSOR and W_CAR_SENSOR. You are going to assign four Micro830 controller inputs to these function block inputs.
- 12. Click on the top of the input variable block that connects to **N_CAR...** and you will get a dropdown menu of all the existing variable names that could be assigned to **N_CAR_SENSOR**. Scroll down and select **_IO_EM_DI_00** and enter.



13. Notice that because of the length of embedded input 0 variable name, it is hard to tell what input is actually assigned to **N_CAR_SENSOR** when viewing the function block. One way to tell is to position your cursor over the block as shown below.



14. Another way is to assign shorter Alias names to these variables. Double click on the first input block

– this brings up the Variable Selector screen. Go ahead and type in Alias names for the six outputs
(DO0-DO5) and the first six inputs (DI0-DI5).

🔡 Va	ariable Sele	ctor						
Name	eEM_DI_00	Type BOOL		•	Global Scope Micro830		Local Scope -	•
User (Global Variables - Micro	830 Local∖	/ariables - N//	A System Var	iables - Micro	830 1/0 - Micro8	30 Defined Wor	ds - Controller
	Name	Data	Туре	Dimension	Alias			Cor 📥
	<i>~ ∂</i> {*	BOOL	× ==	<i>™</i> 0 4 *	<i>d</i> [*]			
	_IO_EM_DO_00	BOOL	*		Output_0			
	_IU_EM_DU_01	BOOL	•		Uutput_1			
	_IU_EM_DU_U2	BUUL	· ·		D02			
	_IU_EM_DU_03	BOOL	- -		D03			
	_10_EM_D0_04	POOL			D04			
	_10_EM_00_05	BOOL	· •		D05			=
	_10_EM_00_00	BOOL			D07			
	10 EM DO 08	BOOL			D08			
	10 EM DO 09	BOOL	-		D09			
J	IO EM DI 00	BOOL	~		DIO			
	_IO_EM_DI_01	BOOL			DI1			
	_IO_EM_DI_02	BOOL	*		DI2			
	_IO_EM_DI_03	BOOL	*		DI3			
	_IO_EM_DI_04	BOOL	•		DI4			
	_IO_EM_DI_05	BOOL	*		DI5			
	_IO_EM_DI_06	BOOL	•					
	_I0_EM_DI_07	BOOL	•					
	_IO_EM_DI_08	BOOL						-
	IO EN DI OO		-					• • •
							<u>0</u> K	<u><u>C</u>ancel</u>



15. Assign the remaining three input variable blocks as follows: _IO_EM_DI_01 to S_Car..., _IO_EM_DI_03 to E_Car..., and _IO_EM_DI_04 to W_Car... (note that we skipped using Input 2!).

16. The first function block output that connects directly to the ladder rung is the function block output enabled (ENO) bit – it reflects the status of the input enable (EN) bit. The remaining six function block outputs are "real world" outputs that connect to the red, yellow and green traffic signal lights for each direction. These outputs get mapped to six Boolean output variables local to the function block: NS_GREEN_LIGHTS, NS_YELLOW_LIGHTS, NS_RED_LIGHTS, EW_ GREEN_LIGHTS, EW_ YELLOW_LIGHTS, and EW_ RED_LIGHTS. Assign the first six Micro830 digital outputs to the output variable blocks starting with _IO_EM_DO_00 to NS_R... and ending with _IO_EM_DO_05 to EW_G....



17. The rung is now complete except for a description of what the rung does. Double click on the green area just above the rung and type in "This rung assigns the Micro830 I/O to the TRAFFIC_CONTROLLER function block."



18. Finally, build and save the one-rung program. Right click on the Micro830 icon in **Project Organizer** and select **Build**.



19. You should get verification in the **Output** window at the bottom center of the screen that the build succeeded.



Click the Save icon \blacksquare to save your work.

Chapter 6 – Downloading a Project and Troubleshooting Faults

Establishing a Connection Between Your Computer and a Micro830 via USB

This section will show you how to configure USB drivers on your computer the first time you connect to any Micro800 controller.

- 1. Normally RSLinx Classic is installed as part of the Connected Components Workbench software installation process.
- 2. Power up the Micro830 controller.
- 3. Plug USB A/B cable directly between your PC and the Micro830.
- 4. Windows should discover the new hardware. Click No, not this time and Next.

Found New Hardware Wizard	
	Welcome to the Found New Hardware Wizard Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission). Read our privacy policy
	Can Windows connect to Windows Update to search for software? O Yes, this time only O Yes, now and every time I connect a device No, not this time Click Next to continue.
	< Back Next > Cancel







Copyright © 2009 Rockwell Automation, Inc.

5. Click Install the software automatically (Recommended) and Next.



6. Wizard will search.



7. Wizard should complete in less than a minute. Click *Finish.*



Connecting to a Micro830 Controller

In this section you will learn how to connect to a Micro830 controller. You must establish communications with the controller before you can download your project to it.

1. Double click on the Micro830 under Project Organizer to open the controller's property tab.



2. Click on the **Connect** button.



3. Browse and Select the controller from the Connection Browser screen.



 Click OK and the status on the device toolbar will indicate Connected. If successfully connected, the button changes to Disconnect and the Controller Mode is displayed. Being Connected means the Connected Components Workbench (CCW) is now connected and/or communicating with the Micro830.



Downloading a Project and Troubleshooting Faults

This section will show you how to download a project to a controller and how to troubleshoot faults that may appear after downloading a project. Prior to downloading a project you must connect to the controller, refer to the previous section for more details on how to connect to the Micro830 controller.

1. Follow the instructions below to build, save, and download your project.

Right click on the Micro830 icon in the Project Organizer and select Build.



 You should get verification in the Output window at the bottom center of the screen that the build succeeded. If you have errors you must go back to your program and fix the error before it will allow you to download.



3. After verifying you don't have errors, click Save.

💐 Connected Components Workb								
File	Edit	View	Project	Build				
1	2	-	1 26 1	a 18.				

4. Right click on the Micro830 icon in the Project Organizer and select Download.



Note: If you don't have the plug-ins configured in Chapter 3 your controller will fault when attempting to download. Modify your controller configuration to match your hardware and attempt downloading again.

5. If your controller is in Remote Run mode you will see the following message. Click **Yes** to continue.



6. After the download is complete you will see a message asking if you want to change to Run mode. Click Yes to begin running your program or No if you need to make further modifications.



Troubleshooting Faults

In this section you will learn how to troubleshoot faults that may appear after you download a project. One of the common faults you may see is a configuration mismatch between your project and the controller. In this section you will learn how to troubleshoot that fault and how to fix it.

1. Double click on the Micro830 icon ion the Project Organizer.



You should see a window with Micro830 information. This window gives you status information for your controller and allows you to configure the plug-ins and other parameters.

2. Below is a screen shot of a faulted Micro830, notice how there is a red icon that displays **Faulted**. This indicates a fault occurred after the project was downloaded.



3. To troubleshoot faults, click on **Startup/Faults** under the **Controller Configuration** tree. The detail of the fault is shown; in this case you will notice there is a plug-in configuration mismatch. To resolve it, click **Clear Major Fault** and match physical plug-ins with configured plug-ins.



Chapter 7 – Testing a Running Program

Testing a Running Program

This chapter will show you how to debug and force a command in a running program.

- 1. This chapter assumes you have a running program. For details on how to create, build, and download a program, refer to earlier chapters.
- 2. Verify you are in Run mode. If you are not in Run mode, you must select the Run radio button.



3. Select Start Debugging from the Debug menu.



Copyright © 2009 Rockwell Automation, Inc.

4. The following screen will then appear under the Traffic_Control_POU tab.

📉 (Running) - Connected Con	nponents Workbench						_ 8	X
File Edit View Build Debua Tool	s Communications Window Helt	C						
1 n n n n n n n n n n n n n n n n n n n		• • • • (1 • • • •		1	Cvcle Ti	mina (ms):	-
Project Organizer	= = = -	Re Liebt Control (- ~	
Name: Traffic Light		fic_Light_Control+	,00				• ^	Ŋ
JIII Micro830	Name	Logical Value F	hysical Value	Lock	Data ⁻	Гуре	Dir	Nic
	- <i>A</i> *	- <i>A</i> *	~ <i>A</i> **	· de		* A*		
					BOOL	•		de la
Traffic_Light_Control	IO_EM_DO_01				BOOL	•		<u>×</u>
Local Variables	IO_EM_DO_02				BOOL	•	=	
	IO_EM_DO_03				BOOL	•		
	IO_EM_DO_04	V			BOOL	*		
Data Types	IO_EM_DO_05				BOOL			
🖃 🔊 Function Blocks	IO_EM_DO_06				BOOL	Ť		
	_IO_EM_DO_07				BOOL	× .		
	_IO_EM_DO_08				BOOL	₹.		
	IO_EM_DO_09				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			
	_IO_EM_DI_06				BOOL			
	_IO_EM_DI_07				BOOL			
	_IO_EM_DI_08				BOOL	•		
			[10]		ROOL		-	
🖪 Output								9
Ready							INS	5

5. To Force the value of a variable, left mouse click on global variables.

- 6. Left mouse click on variable you desire to force. For this example, we are illustrating forcing _IO_EM_DI_00.
- 7. Click on the Lock box for the _IO_EM_DI_00.

4	Micro830 Micro830-VAR Traffic_Light_Control-POU									
	Name		Logical Value	Physical Value	Lock	Data	Type			
		- d*	- 0**	- 01	- at		· Æ			
	_IO_EM_DO_05					BOOL				
	_IO_EM_DO_06					BOOL	*			
	_IO_EM_DO_07					BOOL	*			
	_IO_EM_DO_08					BOOL	*			
	_IO_EM_DO_09					BOOL	*			
	_IO_EM_DI_00					BOOL				
	_IO_EM_DI_01					BOOL	*			

8. Click on Logical Value on the _IO_EM_DI_00.

Mi	Micro830 Micro830-VAR Traffic_Light_Control-POU									
	Name		Logical Value	Physical Valu	e Lock	Data	Type			
		ď.	~ 0 ^{#*}	- 0	** - A*		• A*			
	_IO_EM_DO_05					BOOL	•			
	_IO_EM_DO_06					BOOL	*			
	_IO_EM_DO_07					BOOL	•			
	_IO_EM_DO_08					BOOL	•			
	_IO_EM_DO_09					BOOL	•			
I	_IO_EM_DI_00				V	BOOL				
	_IO_EM_DI_01					BOOL	*			

9. You should now see the output LEDs on the Micro830 changing state (if using the same program).
 You will also see the outputs change state by the under the Logical Value column and the Physical Value column under the global variable tab.

Micro830 Micro830-VAR	Traf	fic_Light_	_Contro	I-POU			
Name		Logical	Value	Physical	Value	Lock	Da
	• of**		· A*		· A*	₹ dt*	
_IO_EM_DO_00							BO
_IO_EM_DO_01				1			BO
_IO_EM_DO_02				V			BO
_IO_EM_DO_03							BO
_IO_EM_DO_04				V			BO
_IO_EM_DO_05				1			BO
_IO_EM_DO_06							BO
_IO_EM_DO_07							BO
_IO_EM_DO_08							BO
_IO_EM_DO_09							BO
IO_EM_DI_00		V				V	BO
							RΟ

- 10. To stop forcing the value, **uncheck** Logical Value. To allow the program or external sources to change the value, **uncheck** Lock.
- 11. To stop debugging, click Debug in the toolbar and select Stop Debugging.

12. You can now leave the Micro830 in remote run mode, can select program mode, or disconnect from the Micro830 by opening the Micro830 tab and selecting the radio dial for your preference.

Chapter 8 – Saving and Closing a Project

Saving and Closing a Project

This chapter will show you how to save and close a CCW Micro830 project.

1. From an open project.

Rockwell Automation

Copyright © 2009 Rockwell Automation, Inc.

2. To save, click on File in the tool bar.

3. Select and click Save. The project is now saved. The project file will be saved in the project folder, under the CCW folder under My Documents.

C:\Documents and Settings\Labuser\My Documents\CCW\Micro8001\Micro8001.ccwsln

Note: Please know where your project is saved to. Under My Documents/CCW/...

4. To close, click on File and select Close.

5. The Connected Components Workbench will now look like this.



Chapter 9 –

Connecting to Existing Controllers

Connecting to an Existing Controller

This chapter will show you how to connect to an existing controller.

To upload and connect a project from a controller to a blank project

- 1. Open CCW software.
- 2. Under Device Toolbox, click on the '+' to open Discover.



3. Click on Browse Connections.







Copyright © 2009 Rockwell Automation, Inc.

4. The following will appear if it is connected to the network, select the controller and click **OK**.

Connection Browser	
Autobrowse Refresh	
🖃 🗐 Workstation, USMKEBGBERNHA2	
由一番 Linx Gateways, Ethernet	
E AB_DF1-1, DF1	
H. AB_ETHIP-1, Ethernet	
AD_VDP-1, 1709-A17/A VICUal Cliassis	
Eleventaria de la construcción	
,	
	ancel

5. The project organizer will then populate with information on the program.



6. If you select your own controller from the Device Toolbox Catalog, and place it in the Project Organizer, click Upload.



7. The following may appear, click yes.



8. After upload is complete, you will see the following in the output window at the bottom of the program tab.

• 🖗 🚑 🛒 🖃			
Start Uploading Resource #1			
-to-date, O skipped			

9. Double click on the Micro830, and the following will appear. Double clicking on the program tab will enable the user to view the upload code.



10. You can now work with your uploaded project.

Connecting to a Controller that has a Different Processor in the Project

1. If trying to connect to a different controller than the one in the project, the following will appear.

Connecte	ed Components Workbench	<
1	Failed to connect with the controller. The actual controller type does not match with the controller type for this project.	ļ
	Project: 2080-LC10-12DWD Online: 2080-LC30-16QWB	
	ОК	

2. The recommendation would be to abort, and either change the controller in the project, or change the controller you are connected to.

Chapter 10 – Using Micro810 Smart Relay Functionality (no software)

Using Micro810 Preprogrammed Features

The Micro810 12-point (8 Inputs & 4 Outputs) controller comes with eight Smart Relay function blocks built-in that can be configured using the optional LCD Display and pushbuttons to control the four relay outputs, without using any software! These built-in function blocks are:

- TON On-delay Timing
- CTU Count Up
- DOY Turning on an output if the value of real-time clock is in the range of Year Time setting.
- **TOW** Turning on an output if the value of real-time clock is in the range of Day Time setting.
- CTD Count Down
- **TONOF** On-delay timing on a true rung, and then Off-delay timing on the false rung.
- **TP** Pulse Timing
- **TOF** Off-delay Timing

This chapter will show you how to configure the Count Up (CTU) function block.

- 1. Power up the Micro810 controller.
 - Upon power up, the Micro810 splash screen is briefly displayed.



The Status display should be showing the PROG status, the day and time, and the I/O Status.
Press and at the same time to navigate to the Main Menu.



3. Press to enter the SR (Smart-Relay) function block program. The function block for controlling Output 0 is displayed.



4. Press once to navigate to the function block controlling Output 1.



5. Press once. The instruction parameter field will be selected.



6. The instruction parameter field shows the CTU (Count Up) instruction.



7. Press Once to select the CLK parameter field. This is the trigger for counting.



8. Press wice to select I04.



9. Press once to select the RESET parameter field. This input will force a counter reset.



10. Press wice to select I05.



11. Press three times to move to the first non-zero entry in the PV (Preset Value for the counter)

parameter field.



12. Press wice to make this digit a zero.



6

13. Press Sonce to position to the next non-zero digit in the PV field and repeat step 12.

- Repeat the same step for the second to last non-zero digit.
- For the last digit, press in five times to make this last digit to a value of 3.



14. Press once to position to the screen selection parameter.



15. Press to submit the parameter changes.

- A screen will confirm your request to save the parameter changes.
- Press to save the changes.



Testing the Count-Up (CTU) Predefined Function

The count-up (CTU) instruction increments the counter whenever input CLK makes a transition from low to high. The instruction will compare the current value CV with the preset value PV, and energize output O1 when $CV \ge PV$. To simulate the operation, we connect a count pushbutton to I4, a reset pushbutton to I5, and a lamp to output O1.



1. Press to return to the Main Menu.



2. Press whree times to get to select Mode Switch and press



3. Press once and press to select RUN mode.



4. Press to confirm the RUN mode selection.





5. The Screen will indicate that the controller is in RUN mode. Press (LESC) to return to the Main Menu

6. Press fines and select SR FUNCTION by pressing . Press once for Output1.



7. Press and release the count pushbutton. The current value CV increments to 00001.



8. Press and release the count pushbutton. The current value CV increments to 00002.



9. Press and release the count pushbutton. The current value CV increments to 00003. Since the current value CV = present value PV, the output O1 is energized, and the lamp is iluminated.



10. Press and release the Reset pushbutton. The current value CV is reset to zero, and output O1 is deenergized. The lamp is no longer iluminated.

