

# **MAKER:** Instructional Module on Use of a Programmable Logic Controller for Smart Traffic Light Control

#### Mr. Jorge A. Piña, Baltimore City Community College

Jorge A. Piña moved to Baltimore from Chile, South America, to study Computer Science. He graduated from Coppin University with a Bachelor's Degree in Computer Science in 1997. Since then, he has been working as the Coordinator of the Mathematics Learning Center under the direction of the Mathematics and Engineering Department at Baltimore City Community College. During this period of time he also has taught developmental mathematics courses from Basic Arithmetic to Intermediate Algebra. In his free time, he enjoys fixing his old BMW M3 and driving it as fast as he can.

#### Dr. Sheng-Jen "Tony" Hsieh, Texas A&M University

Dr. Sheng-Jen ("Tony") Hsieh is a Professor in the Dwight Look College of Engineering at Texas A&M University. He holds a joint appointment with the Department of Engineering Technology and the Department of Mechanical Engineering. His research interests include engineering education, cognitive task analysis, automation, robotics and control, intelligent manufacturing system design, and micro/nano manufacturing. He is also the Director of the Rockwell Automation laboratory at Texas A&M University, a state-of-the-art facility for education and research in the areas of automation, control, and automated system integration.

## MAKER: Instructional Module on Use of a Programmable Logic Controller for Smart Traffic Light Control

#### Abstract

A Smart PLC Traffic Light system using a smart PLC controller logic has been developed. Traffic light system is being controlled for each light based on traffic flow using proximity sensors that detect presence of vehicles at the intersection of roads. The smart traffic light system consists of a Click PLC, proximity sensors, and a traffic light apparatus.

### Motivation



Traffic light systems are available all around the world. They are a great tool for preventing accidents at the intersection of roads and for controlling traffic flow at intersections. In the old system, the traffic control is done by setting predetermined time for each Red, Green and Yellow on each road. Predetermining the time for each light can be impeding traffic flow, because a driver can be at the intersection waiting for the red light to change to green based on the predetermined time while traffics in other directions are with no traffic at all. Traffic light The Smart PLC Traffic Light system using a Click PLC controller was developed to solve the problems identified in the old traffic light cycle system.

At the 1999 Annual International EMME/2 Conference of Chicago,

Illinois, a paper on improving reliability of Traffic using intersection delay was presented by Aashtiani et al.. Delay estimation at intersections was determined with or without signals. This detailed research really highlights the needs of improving traffic light control and also the challenges that we are facing.

The objective of this lesson plan is to develop a Smart PLC Controller logic based traffic light simulator system. The system consists of a several proximity sensors, a Click PLC, power supply, and 2 sets of green-yellow-red lights to simulate the traffic light controlled by a ladder logic program.

As compared to fixed time control systems, the foundation of a dynamic system is actually a detector which is nothing more than a simple device that communicates with the traffic light and informs it about traffic conditions in real time. This time, the traffic light can not only adjust timing, but also solve traffic congestion by changing the cycle of the traffic light as soon as the traffic in the intersection gets heavy with cars.

This project is using proximity sensors to simulate the switches or cameras that in a real scenery will communicate to the PLC about the traffic condition

In this project the cycle of the traffic light will be change by the Smart Traffic Light Controller depending of traffic in the intersection. There will be different times on for the Green and Red lights depending of the input, and the Yellow light will be kept constant on for 2 seconds.

Per example, If inputs, Sensor 1 and Sensor 2 or Sensor 5 and Sensor 6 are on and inputs Sensor 3 and Sensor 4 or Sensor 7 and Sensor 8 are off the PLC controller will run a program M10, if inputs Sensor 3 and Sensor 4 or Sensor 7 and Sensor 8 are on and inputs Sensor 1 and Sensor 2 or Sensor 5 and Sensor 6 are off the PLC controller will run a program M11.

#### **Project System Platform**

An automated system typically consists of controller, sensors, lights, and system structure. Table 1 lists the components of the platform used for this project and cost.

Item	Description	Quantity	Cost
Click PLC	Model C0-00DR-0	1	\$68
Proximity Sensor	Double M12 PW PNP NO	8	\$192
Color Light	24-Volts and 12- Volts	2 red, 2 green, 2 yellow	\$24
Power Supply	24-Volt	1	\$45
Power Supply	12-Volt 25-Watt	1	\$27
PLC Software	Software	1	\$0
Wood Platform	Wood 20x20 inc	1	\$10
Wires	Black/red/green	1 pack	\$7

Table 1. Project Platform and Cost

#### The Smart Traffic Light

This system has been designed to demonstrate how a traffic light in an urban intersection will work controlled by a Smart Traffic Light System. Following is a detailed description including the physical design, input and output assignments, the system schematics, and the ladder logic programing of this smart system. Fig. 1 shows the finished design of the system.

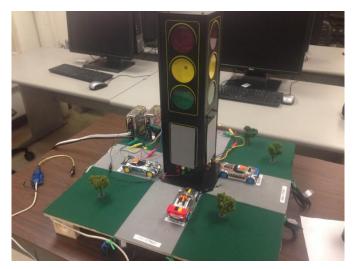


Fig. 1. Smart Traffic Light

#### **Physical System Functionality and Layout**

The Smart Traffic Light system is using a normal cycle light of 10 seconds (Fig. 2), the Smart Traffic Light system will execute this routing until there is not heavy traffic for neither of the street in the intersection.

There are 8 proximity sensors (Sensor1 through 8) installed under the ground to detect the traffic on the intersection and send a sign to the PLC to let it know to act smart

If there is a heavy traffic in the Main street, Sensor 3 and Sensor 4 will be ON or Sensor 7 and Sensor 8 will be ON. In either case the Smart Traffic Light Controller will change the cycle light allowing the green light of the Main street to be a 15 second making the red light of the Side street stay ON longer.

If there is a heavy traffic in the Side street, Sensor 1 and Sensor 2 will be ON, or Sensor 5 and Sensor 6 will be ON. In either case the Smart Traffic Light Controller will change the cycle light allowing the Green light of the Side street to be a 15 second making the Red light of the Main street stay ON longer.

If an emergency vehicle approach to the intersection with the siren ON, the Smart Traffic Light Controller will make the red light of the traffic light blinks for both street warning the drivers coming for all directions to stop allowing the emergency vehicle cross the intersection without any delay and safe.

All of this is possible by the smart decision of the Smart Traffic Light Controller.

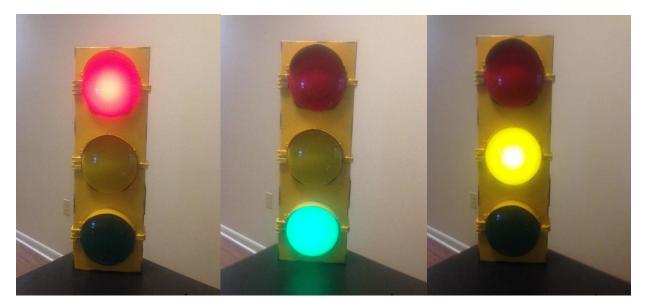


Fig. 2. Traffic Light Cycle

#### **INPUT / OUTPUT Port Assignments**

The system has eight inputs and six outputs. The inputs are the 8 sensor in the ground and the six Output are the 2 traffic lights (2-green, 2-yellow, 2-red) for the intersection.

INPUT ADDRESS	INPUT DEVICE	OUTPUT ADDRESS	OUTPUT DEVICE
	NAME		NAME
I:00/01	Sensor 1	O:00/01	Side Green
I:00/02	Sensor 2	O:00/02	Side Yellow
I:00/03	Sensor 3	O:00/03	Side Red
I:00/04	Sensor 4	O:00/04	Main Yellow
I:00/05	Sensor 5	O:00/05	Main Red
I:00/06	Sensor 6	O:00/06	Main Green
I:00/07	Sensor 7		
I:00/08	Sensor 8		

Table 2 shows the I/O address for each I/O.

Table 2. Input/Output Device Addresses

#### **System Schematics**

The Smart Traffic Light Controller consists of a Click PLC, Model C0-00DR-0, a 24-volt power supply to fit the PLC, 8 proximity sensors located under the ground (Fig.3.) closed to the intersection. A 12-volt power supply is used to power 4 output lights. The other 2 output lights are powered by a 24-volt power supply. This project has 4 output lights of 12-volt and 2 output lights of 24-volt.

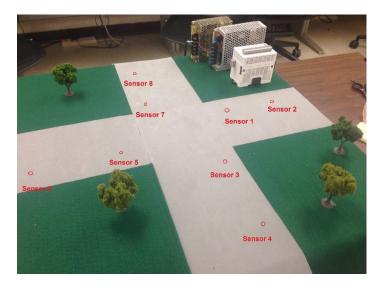


Fig.3. Locations of the 8 sensors

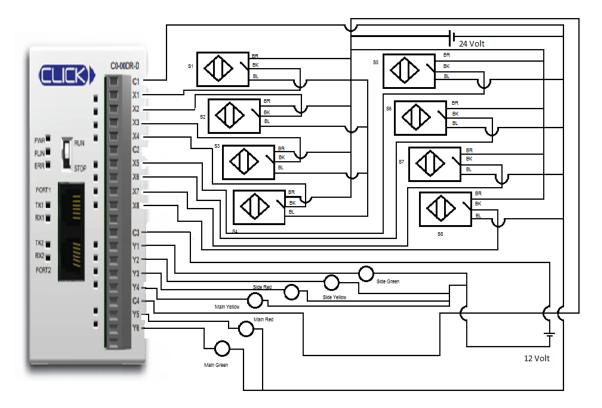


Fig. 4. System Schematic of the Smart Traffic Light

Students complete three activities: 1) Familiarization with the Click PLC; 2) Click Programing Software and Basics of Ladder Logic; and 3) Build and test the Smart Traffic Light Controller

#### Activity 1: Familiarization with the Click PLC

Objectives

- To became familiar with the Click controller
- To run Click software
- To connect the Click controller to the computer

The Click Programmable controller, model: CO-OODR-D permits hardware devices such as timers, counters, and relays to be replaced by a ladder logic program. A set of instructions representing the logic is written and downloaded into the Click PLC memory to execute them following the order of instructions to interpreter the sign from its input devices and operate its output devices.

The Click Controller, model C0-00DR-D from Koyo can be programmed by using the Click programming software from the Automation Direct website (www.automationdirect.com). The Click controller is connected to the computer through the USB port (Fig. 5).

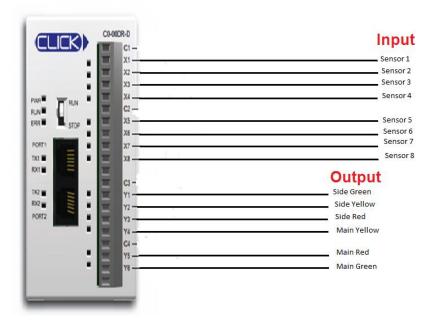


Fig. 5. Adapter to connect the Click to a computer



Fig. 6. Click Controller, Model C0-00DR-D showing the inputs and outputs

PLC inputs X1 through X8 are connect to the proximity sensors and are activated by a 24-VDC power supply, this is an external source that also supply power to the Click.

PLC outputs are marked as Y1 through Y6, they are connect to a 6 output lights and powered by a 24-VDC voltage and a12-VDC voltage.

#### **Procedure Summary**

In this lesson you will familiarize with the Click controller communication, basic functions, such as open a project, and saving a project.

#### Procedure

- 1. Connect the Click to the power supply and the computer.
- 2. Turn on the computer and load the Click programming software.

Creating a New Project or Open a project

- 1. Choose the New command in the File menu to Start a new project or choose Open Project command
- 2. You will Open Project1 to see an example of the ladder logic programming.
- 3. The project 1 is display in the Ladder View Window.

Saving a Project to a file

- 1. Now you will save the project in the computer memory. To save the project to choose the Save or Save As command in the File menu
- 2. This will open the Save As commando where you will type the project filename.
- 3. Click Save button in the dialog box, so the project will save with the extension CKP after the name.
- 4. Click Close Project from the File Menu. Click Exit from the File Menu to close the software.

### Activity 2: The Click Programming Software and Basics of ladder logic

Objectives

- To create a project and edit a project using PLC ladder program
- To print a project
- To learn how to use on delay timer and a subroutine

The Click programming software is used to program and control the Click PLC creating, editing and monitoring PLC ladder program that can stored in the Click memory. The Click programming software is free and can be downloaded from the website of Automation Direct (www.automationdirect.com).

The Click programming software runs under the Microsoft Windows environmental, Fig. 7 shows a view of the program open and the several elements to start a program.

		M / / / 2 2 3 4 4 4 5 1 2 0 1 2 2 3 4 4 4 5 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1	ee ni 🤏 🐂 🧏 🐘 🖏 🛯 🛛 🖾 🛍	~		
Navigation ×	-	A B C D E F	G H I J K L M	AF A	Interaction List	
Program Function PLC	E F				Instruction	
Ladder Program     Ladder Program     Main Program     Subroutine Program	1			-( NOP )	Contact Contact (NO) Contact (NC)	∽ Main bar
- Address Picker	2			-( NOP )	Edge Contact	
Edit Bung Comments	3				Coil	
Local Program Information				( NOP )	er Out er Set	
Cross Reference View	4			( NOP )	4nd Reset	
Monitor	H I			(1007.)	Timer/Counter	<b>\</b>
B 15 Data View	5			-( NOP )	per Counter	1
DataView1					Advanced	1
Override View	6			-( NOP )	Inter Drum	1
System Monitor					sa Shift Register Copy/Search	
	7			( NOP )	Darn Copy	Instructions
1					san Search Program Control	monuctions
	8			( NOP )	Feel Call	
					Pat For Next Next	
	9			-( NOP )	Communication	
	10			( NOP )	ap Receive	
				(NOF)	sa Send	
	11			( NOP )		
				(110) 7		
	12			(NOP) *		
	<	to communication advantation (		>		
++ ++ ++ ++ ++ ++ + F2 F3 12 -F3 5HF2 5	+++-+	*** ***				
Ready		Bit I Integer I Integer(2 words) IF Fic	ine point III Hex T Taxt A Ascii Offline 0000/0000 No select CPU 11A		NUM .	
Search the web and Wi			the second se	~ <b>6</b> 9	3.21 PM	
Search the web and Wi	ndows		<u> </u>	~ 6	7/17/2016	
1		20	1			
Project Tree		Instruction Toolbar	\			
			Ladder View Window			

Fig. 7. The Click software and its elements

The PLC ladder program is the main component of the project. The PLC runs the program following the inputs to control its outputs.

#### Procedures

- 1. Connect the Click to the computer.
- 2. Turn on the computer and load the Click programming software.

Creating a New Project

- 1. Choose the New command in the File menu to Start a new project.
- 2. Name the project and save it.
- 3. Now you are ready to type your first ladder logic program.
- 4. You have several programs to type from the File name Projects.
- 5. After you finish typing it you can click the Save button in the dialog box to save the project. It will be saved with the extension CKP after the name.
- 6. Now you can check for errors, click on Syntax Check under Program menu.
- 7. With no errors you are ready to write the project in the PLC.
- 8. Click Write Project into PLC under PLC menu.

CLICK Programming Software - bill1	2.ckp - [	Main Program]													- • ×
G File Edit View Setup Progra			Window Help												- 8 3
🗋 🖉 🖬 🎯 🖪 🛯 🛓 🖬 💼		2   AA   🖉 😣 📐		y 🔿 🐁 🤇	) 🛃 🚳	- M 61	S	in in 📆 🛛	<b>1</b> 🗛 🕈	6					
		-   a a   - 2 - 5		Nicknam		iress Comme		Comments		Area	1				
	-	A	B	C	D D	E	F	Gommerics	H	nica i	J		AF		-
Navigation ×	<u> </u>	A	в		U	E	F	6	н		J		AF	_^	Instruction List
Program Function PLC															Instruction •
CPU Configuration		■ X003	🖪 X004										B Y001		Contact
- System Configuration - 19 Com Port 1 Setup	1												(out)		Contact (NO)
- 27 Com Port Setup															Edge Contact
- 97 Com Port3 Setup		B X001											B Y002		St Compare
🕂 🕐 Scan Time	2												(OUT)		Coil
- Match Dog Timer	<u> </u>														(en) Out
Password Setup 															+tm) Set +tmn) Reset
Battery Backup Setup     J/O Configuration		<b>■</b> ×002													Timer/Counter
Interrupt Setup															THE Timer
															ENT Counter
🖹 🎯 Software Setup	3												( END )		Advanced
🧑 Software Setup ⊡∭ Manual	<u> </u>												()		호국 Math onum Drum
- Wanual															sn Shift Register
Hardware Manual	4												( NOP )		Copy/Search
-															tery Copy
															suu Search
	5												( NOP )		Program Control
															FOR FOR
	6												(1100)		Nat Next
	в												( NOP )	_	END End
															Communication
	7												( NOP )	~	so Receive
	<													>	and oction
×	-														1
															~
-+	<b> ↓ -</b> H•F3	할 찾 찾 찾	1												
	H+F3														
Ready		B	Bit 🚺 Integer 🔢	Integer(2 word:	s) 👔 Floati			d 🗛 Ascii	Ottline	0009/8		C0-00DD1-D	) /:U:AF		NUM
Search the web and Wir			口 🤤		$\bigcirc$	<u>ه،</u>	<u>ð</u>	8	W	Ø			^	<b>6</b>	□ 🥼 📮 11:34 AM 7/18/2016

Figure 8: Sample 1 - Ladder logic program using an AND and OR statements

CLICK Programming Software - bill File Edit View Setup Programming Software - bill			Window Help										- 0	× ×
				y 🝈 🐁 🥥	¥ 🚳	<i>pa</i>    6	<b>4 4</b> 1	in 🐂 🚯	II 🖪 👹	5				
III @ 🖉 💩 👿 🔳				Nicknames		ess Comment	s Rung O	Comments	Coil A	-				
Navigation ×		A	B	C	D	E	F	G			AF		▲ Instruction L	ist ×
Program Function PLC									Tim	er (ON Delay)	) T1		Instruction	•
E SCPU Configuration			_						Curr	rent Value	Not Retained	<b>■</b> <u></u>	Contact	
- We System Configuration - 19 Com Port1 Setup	1	■ X001	BX002						Unit		sec		H Contact (N	
- 🖅 Com Port2 Setup	'								Enable Set	Point	<b>II</b> 5		🚯 Edge Conta	
- 197 Com Port3 Setup - 10 Scan Time													St Compare Coil	_
									Curr	rent	III TD1		(un) Out	
													(knn) Set (mn) Reset	
🖃 🏧 1/0 Configuration													Timer/Coun	ter
- 🛃 Interrupt Setup 		B∎T1										■ Y001 	THE Timer	
- G Software Setup	2											_(001)	Advanced	
Software Setup													🛓 Math	
🖮 🤟 Manual													sn Shift Regis	ter
													Copy/Search	
	3											—(END)	www.Copy swarSearch	
													Program Cor	ntrol
	4	L										( NOP )	THE Call FOR For	
													NAT Next	
	5											( NOP )	End Communicat	ion
	Ľ,											(1101 )	RD Receive	ion
	<	1										,	Y So Send	
×													1	_
														~
→I-         ★IF         -II-         ★IF         -II-         +II-         +	<b>↓ -</b> H•F3	┇╋ <u>╋</u>												
Ready		B	Bit 🚺 Integer 🔟	Integer(2 words)	Floatin	e point 📔 H	lex ፲ Text	t 🖪 Ascii	Offline	0009/8000	C0-00DD1-D	1:AF	NU	
Search the web and Wi	ndows		0 🤤		0	<u>s</u>	ð 🕘	8	W	ø 🖸		^		1:37 AM 18/2016

Figure 9: Sample 2: Ladder logic program using a Timer

CLICK Programming Software - subr	outine.	ckp - [Main Program]									- o ×
🔄 File Edit View Setup Progra			Window Help								_ 8 ×
🗋 🧉 🔄 🖪 🔔 🛛 👗 🐚 💼	10	e M 🖌 🖉 📐	🔍 🔍 🔍 📝	Õ 🐁 🥥	🛃 🚳 🔤	• 📲 ا ان ا	🗉 🔚 🐘 🖏	🛯 🔼 🐔			
■ 🖩 🥥 🖌 🐱 🔳				Nicknames			Rung Comments	Coil Area			
Navigation ×		A	B	C	D	E F	G	н	JI	AF ^	Instruction List ×
Program Function PLC											Instruction +
Ladder Program     Main Program     Subroutine Program     Subroutine 1     Winterrupt Program     Oram     Address Pricker	1									Call Subroutine1	Contact Contact (NO) Contact (NC) Contact (NC) Compare Coil
	2									(END)	(m) Out (m) Set (m) Reset Timer/Counter
Monitor     Monitor     Monitor     Monitor     Monitor     Monitor     Million     Million     Million     Million     Million     Million	3									( NOP )	าหลัTimer Counter Advanced ฐระ Math
- 편ன Text View - 현재 Override View - 현재 System Monitor	4 5									( NOP )	Drum Sn Shift Register Copy/Search Copy Sull Search
	6									( NOP )	Program Control Int Call FOR For Next
	7									(NOP)	END End Communication RD Receive SD Send
	<	•								>	
×											< v
→F         →F         →F         →F         →F         →F         →F         →F         →F         >H         →F         →F         >H         →F         →F         →F         >H         →F         →F         >H         →F         >H         →F         →F         >H         →F         →F         →F         >H         →F         →F         >H         →F         >H         →F         →F         >H         →F         →F         →F         >H         →F         →F         >H         →F         >H         →F         →F         >H         →F         >H         →F         >H         →F         →F         >H         →F         →F         →F         >H         →F         →F         >H         →F         →F	<b> ↓ -</b> H•F3	<u></u> 발 칼 찾 찾									
Ready		В	Bit 🚺 Integer 🔢 Inte	ger(2 words)	Floating p	oint 📔 Hex [	🚺 Text 🛛 🗛 Ascii	Offline 0009	V/8000 C0-00DD1-	D 1:1:A	NUM
Search the web and Wi			口 🧲 🖡			5	<b>e</b> ) (e)	<b>W</b>		^ 🥳	I2:16 PM 7/18/2016

Figure 10: Sample 3 - Ladder logic program using a Main program and a SubRoutine

CLICK Programming Software - subr File Edit View Setup Program			r Windo	w Help														– 0 ×
		e M 🖌 🖉 📐	•		J 🔿 🖉	<b>N</b>	¥ 🚳	1 M	4	🛚 🖌	<b>%</b>		1					
🔳 🖬 🥥 💋 🔳 🔳			u :	101		cknames		ess Comm	ents	Rung Co	mments		l Area					
Navigation ×		A	в	C		D	E	F		G	н	1	J		К	AF	^	Instruction List ×
Program Function PLC																		Instruction +
Edder Program		<b>■</b> X001														Y002		Contact
	1															(out)		Contact (NO)
Subroutine 1																		🚯 Edge Contact
- Interrupt Program		BX002														■ Y002 ( OUT )		Compare
- 🛃 Edit Rung Comments	2																	en Out
- Cocal Program Information																Return,		•tnd) Set •tnd) Reset
Cross Reference View	3																	Timer/Counter
🗄 🕮 Monitor																		THE Timer
🖻 🙀 Data View	4															( NOP )		Advanced
DataView 1																		효고 Math num Drum
	5															( NOP )		sa Shift Register
																		Copy/Search
	6															( NOP )		sam Search
																		Program Control
	7															( NOP )		FOR FOR
																		Next Communication
	8															( NOP )		RD Receive
	q															( NOP )	~	so Send
	<																>	
×																		^
																		~
H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H         H	₩ <b>₩</b>	゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚゚	F															
Ready			_	nteger 🕅	Integer()	words)	F Floatin	e point 📔	Hex 1	T Text	A Ascii	Offline	0009/8	3000	C0-00DD1	-D 1:1:A		NUM
Search the web and Wir	ndows			9		H)		_	Ô	9	8	W					^ 🔞	🕎 🕼 📮 12:18 PM 7/18/2016

Figure 11: Sample 4 - A Subrouting program

## Activity 3: Building and testing the Smart Traffic Light Controller

Objectives

- To build a Smart PLC Traffic Light Controller
- To create a project using PLC ladder program

The Smart Traffic Light Program

The ladder logic program for the traffic light intersection with lights facing North /South (Main Street) and West /East (Side Street) the normal cycle time is 12 seconds.

For the Main street the Green light is on for 5 seconds, followed by the Yellow light for 2 seconds, and then the Red light that is on for 5 seconds. For the Side street the Red light is on for 7 seconds, followed by the Green light is on for 3 seconds, then the Yellow light for 2 seconds shows the end the normal cycle. This is the synchronization sequence and frequency of the lights for the normal cycle of the Main and Side street this in this project.

If traffic is detected in the Main street the Smart Traffic Light will change the time of the cycle of the traffic light allowing the drivers on the main street go through the intersection making the traffic flow smugly. For Main street the Green light will be on for 15 seconds, following by the Yellow light on for 2 seconds, then the Red light will be on for 5 seconds. While in the Side street the Red light will be on for 17 seconds, followed by the Green light on for 5 seconds, and followed by the Yellow light on for 2 seconds.

If traffic is detected in the Side street the Smart Traffic Light will change the time of the cycle of the traffic light allowing the drivers on the Side street go through the intersection making the traffic flow smugly. For Main street the Green light will be on for 15 seconds, following by the Yellow light on for 2 seconds, then the Red light will be on for 5 seconds. While in the Main street the Red light will be on for 17 seconds, followed by the Green light on for 5 seconds, and followed by the Yellow light on for 2 seconds.

And if an emergency vehicle or a police car with the siren on approach to the intersection the Smart Traffic Light system will flash the Red light on both street to make all traffic from all direction to stop allowing the emergency vehicle cross the intersection fast and safe.

All of these tasks are possible by the 8 sensors installed in the ground closed to the intersection, the ladder logic programing, the PLC who makes all of this possible and the lights that give directions to the drivers how to proceed at the intersection.

This program uses timers to accomplish this task.

#### Ladder Logic Program

One of the major challenges of the smart traffic light system was how to make the light from the main street (cycle green-yellow-red) synchronizes with the light of the side street (cycle red-green-yellow). I could accomplish this task using several OnDelay timers as show in Figs.12-14.

For this project uses a 12 seconds (green-red-yellow) for a normal cycle traffic light, and for heavy traffic will be using a cycle of 22 seconds (green-red-yellow).

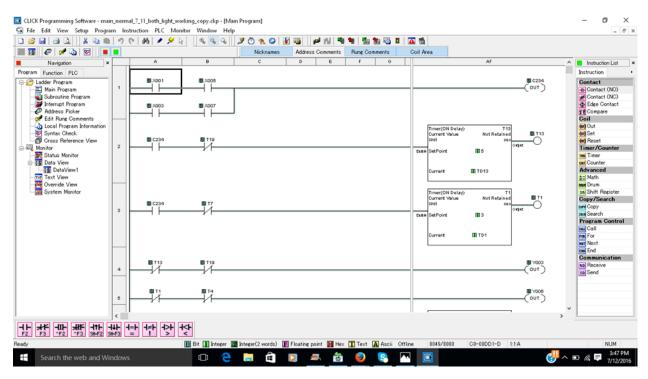


Fig. 12. Ladder logic- Synchronization of the main street light and side street light.

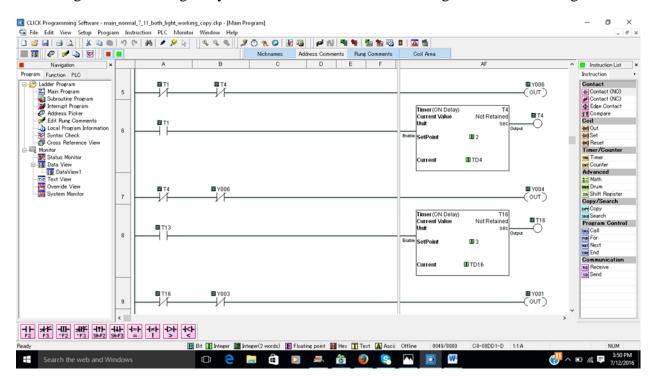


Fig. 13. Ladder logic- Synchronization of the main street light and side street light.

CLICK Programming Software - main File Edit View Setup Progra				gram]					- 0 ×
	0.0			Õ 🐁 🞱 🛃 🛙	8   pi ni ni	- 🐂   🌆 🐂 🚳	I 🔼 🐔		- 0 A
III @ 🖋 💊 💌 🔳		1			Address Comments	Rung Comments	Coil Area		
Navigation ×		A	B	С	D	E F		AF	↑ Instruction List ×
Program Function PLC							Timer (ON (	Delav) T7	Instruction +
E 2 Ladder Program		EN T4					Current Val	Not Retained	Contact
- Subroutine Program - Interrupt Program - P Address Picker	10						Enable SetPoint	00 5	Contact (NC) Edge Contact \$18 Compare
Edit Rung Comments							Current	III TD7	Coil Out
G Cross Reference View ⊟							Timer (ON 0	Delav) T19	en Set en Reset Timer/Counter
Status Monitor Status View Status View Status View 1	11	<b>⊞</b> ⊤16					Current Val Unit		Di Counter Advanced
Text View	''  [						Enable SetPoint	002	tath Math
- Override View System Monitor									sa Shift Register
							Current	III TD19	Copy/Search
									Copy sea Search
		<b>B</b> T19	E Y001	E Y003				■ Y002	Program Control
	12	//	/ĭ	/i				(TOT)	For For
									Next De End
		<b>1□</b> T7	¥004	EN T4				🖪 Y005	Communication no Receive
	13	/\		— I I—				(OUT)	so Send
	14							(END)	
	<								×
-+	<b>↓</b>  - H•F3 =	· * * *							
Ready		B	šit 🚺 Integer 🔢 Inte	eger(2 words) 👔 Fi	oating point 📔 Hex	: 🚺 Text 🚺 Ascii	Offline 0049/	/8000 C0-00DD1-D 1:1:A	NUM
Search the web and Wi	ndows		0 🤤	. 🕯 🖸	<b>-</b> 👼	چ 🧕		W	3:50 PM → □ (k ♥ 3:50 PM 7/12/2016

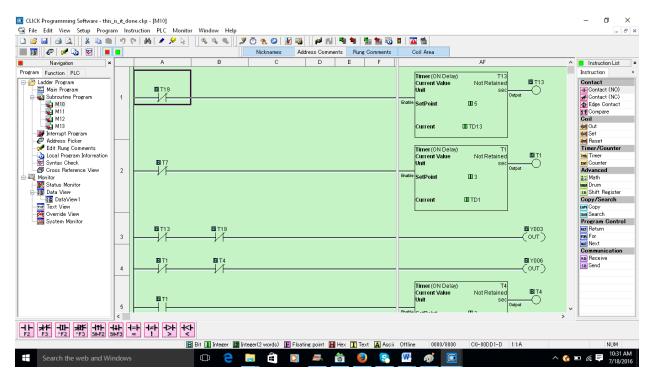
Fig. 14. Ladder logic- Synchronization of the main street light and side street light.

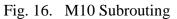
Ladder Logic Program of the Smart PLC Traffic Light Controller

The ladder logic project of the Smart PLC traffic Light Controller consists of one main program and 3 subroutines. The main program will call any of the subroutine depending of the traffic. The sensors will indicate to the PLC controller to call the subroutine needed for the specific traffic situation at the moment. The main program and the subroutines are show in Fig.15 through Fig. 26.

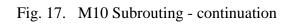
CLICK Programming Software - this, G File Edit View Setup Progra			Window Help												- 0 ×
🔄 nie Eule New Setup Progra				Õ 🐁 🥥	¥ 🚳	- <b>M</b>	<b>Na Na</b> 1 🕅	ha 🐂 🚳	<b>1</b> 🗛 🕈	6					- 0 4
	1 2 3	i ana i e i ve vo		Nickname		ress Commen		Comments	Coil						
Navigation ×		A	B	C	D	E	F	G	н	1		)	⊧ AF	^	Instruction List ×
Program Function PLC															Instruction +
Ladder Program Main Program	1		■×005										Cal	→M10	Contact Contact (NO) Contact (NC)
		B X003	■ ×007												Edge Contact     Compare     Coil
→ ₩13 → ₩ Interrupt Program → ♥ Address Picker → ♥ Edit Rung Comments		<b>B</b> X003	<b>■</b> ×004										Cal	→M11	end Out end Set end Reset Timer/Counter
	2	B ×007												→M11	Timer Counter
Monitor														Math Dum Drum Sn Shift Register	
₩ DataView 1 ₩ Text View ₩ Override View	3	BIX001	×002										Cal	→M12	Copy/Search Mr Copy Search
- 🞆 System Monitor		B ×005	1006												Program Control Int Call Pon For
		BX002	<b>E</b> IX004												Nati Next End Communication
	4												Cal	→M13	no Receive so Send
	5												(END)	)	
	<													~	
F2 F3 -F2 -F3 SHF2 S	≮  ↓ -  + ⊪+F3 =	╪ ╡╪╸╡╱╴ ╴												>	
F2 F3 ^F2 ^F3 SH+F2 S	H+F3 :							_							
		B	Bit 🚺 Integer 🔢 Int		🗉 📔 Floatin					0000/80		C0-00DD	1-D 1:1:A		NUM 10-21 AM
Search the web and Wi	ndows		□ 🤶 🛛		0	<u>,</u>	ð 😣	8	W	<i>🎻</i> 🚺				^ 🔞	■ 🧖 📮 10:21 AM 7/18/2016







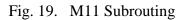
CLICK Programming Software - this_is										- o ×
🔄 File Edit View Setup Program	n Instruction PLC Monito									_ 8 ×
🗋 🧉 🖬 🎿 💁 📖	🔊 🔍 🔥 🖉 😓	🌐 🔍 🔍 🔍 🍠 🕐	🐁 🥥 🛃 🚳	- M M 3	l 🐂   🛅 🛅 🖏	I 🛛	N 🐔			
🔳 🖩  🥥 💆 🔳 🔳		1	Nicknames Add	iress Comments	Rung Comments	0	Coil Area			
Navigation ×	A	B	С	D	E F			AF		▲ Instruction List ×
Program Function PLC			· · · · · · · · · · · · · · · · · · ·				Timer (ON Delay)	<b></b>		Instruction +
Index Program       Main Program       Image: Structure Program	5					Enable	Current Value Unit SetPoint Current	T4 Not Retained Sec 00 00 2	IB) T4	Contact Contact (NO) Contact (NC) Contact (NC) Compare Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil Coil
Local Program Information Syntax Check Groups Reference View Monitor Estatus Monitor Data View Data View System Monitor System Monitor	6 BT13 7 BT13	B YODG				Enable	Timer (ON Delay) Current Value Unit SetPoint Current	T16 Not Retained sec 00 3 00 TD16	E3 Y004 	mil Timer and Counter Advanced Advanced Math mil Drum Shift Resister Copy Search Procram Control ref Return ref Ret
-	8 EB T16	₩ Y003					Timer (ON Delay) Current Value Unit	T7 Not Retained sec ou		Communication     Rol Receive     Sective     Sective
	사 ····································	F								
		Bit 🚺 Integer 😰 Integer				000	e 0000/8000	C0-00DD1-D 1		10.04
Ready		nteger 🔝 Integer	(2 words) 🔡 Floatii		i 🛄 Text 🖪 Ascii			00-00001-0 1	i:n	NUM
Search the web and Wine	dows	🗆 🤤 🥅		۱	👂 🧕	W	<i>i</i>		^ <u>(</u>	i 🗊 🌈 📮 10:32 AM 7/18/2016



CLICK Programming Software - this_ G File Edit View Setup Program													- 0 ×
🕒 🎯 🖬 🎯 🔍 🐇 🐚 🖿	m insi	PLC Monitor	Window Help	Õ 🐁 🞱 🛃	3   pJ m	Sta 🌆	🐜 🕷 🚳		A 🐔				- 6' X
					Address Commer	ts Run	ng Comments		Coil Area				
Navigation ×		A	B	С	D	E	F			AF		^	Instruction List ×
Program Function PLC									Timer (ON Delay	о T7			Instruction +
Ludder Program     Main Program     Main Program     Multicular Program     Mil     Mil	9							Enable	Current Value Unit SetPoint Current Timer(ON Delay Current Value Unit	Not Retained Seco DD 5 DD TD7	ва тт7 дри		Contact  Contact (NO)  Contact (NC)  Contact (NC)  Coil  Coi
Cross Reference View     Monitor     Status Monitor     Go Data View     Go Data View     Go Data View     Text View     Go verside View     Go verside View     Go System Monitor	10	<b>□</b> ■ T19	<b>IS</b> Y003				Enable	<sup>e</sup> SetPoint Current	00 2 00 TD19	utput		Advanced Advanced Advanced Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math Math M	
	11		——//——	1/-							—(out)		FOR FOR
	12		<b>IB</b> Y004	<b>IB</b> Y003							BY005 —(OUT) Return		MIT Next Communication NB Receive SD Send
	13											<b>v</b>	
	<b>₩⊢ -</b>   1+F3	***											
Ready		B	Bit 🚺 Integer 🔟 Inte	eger(2 words) 🛐 FI	loating point 📔	Hex ፲ 1	ext 🔼 Ascii	Offlin	ne 0000/8000	C0-00DD1-D 1	:1:A		NUM
Search the web and Wir	ndows		© 🤶 I	<b>i</b> 🗎 🖸	<b>_</b>	ð (	9	W	<i>ৰ্য</i> 💽		^	6	■ //2 ■ 10:32 AM 7/18/2016

Fig. 18. M10 Subrouting - continuation

CLICK Programming Software - this_												-	
🖼 File Edit View Setup Program	m Ins	truction PLC Monito	r Window Help										_ 8 ×
🗋 💕 🖬 🦪 🔍 🕺 💺 🐚 💼	9	শ 🗛 🖌 🖉 😓	🌒 🔍 🔍 🖉	🕐 🐁 🥥   🛃	🚳 🛛 🍋 🛍 🖗	🥦 🐂   ł	lii 🛍 🗓	I 🛛	s 🐞				
				Nicknames	Address Commer	its Rung	Comments	0	Coil Area				
Navigation ×		A	В	C	D	E	F			AF		🔺 📘 Instru	uction List ×
Program Function PLC			1						Timer (ON Delay)	T13	1	Instructi	ion 🔸
Ladder Program Main Program Main Program Mil Mil Mil Mil Mil Mil Mil Mil	1							Enable	Timer (UN Delay) Current Value Unit SetPoint Current	Not Retained sec 00 5	0utput	∰ Cont ∰ Edge St Com Coil ∯rd Out ∳rd Set	tact (NO) tact (NC) 9 Contact pare
Address Picker     Edit Rung Comments     Local Program Information     Syntax Check     Wonitor     Monitor     Bata View     Data View     Data View     Totat View	2							Enable	Timer (ON Delay) Current Value Unit SetPoint Current	T1 Not Retained sec 00 3 00 TD1	BIT1	106 Time at Cour Advance a Math own Drun 38 Shift Copy/S tee Copy	Counter er ced n n t. Register Search
- 😼 Override View - 📷 System Monitor	3		■ T19 ■ T4 ↓					_			■¥003 (OUT) ■¥1006 (OUT)	NET Retu FOR For NAT Next	m Control m t unication sive
H → H - H + H + H + H + H + H + H + H + H +	5 < ↓↓ ┦								Timer (ON Delay) Current Value Unit	Not Retained sec	BT4	×	
فالقنفا تغنيا تقنيا تحتيا تحتيار			Bit 🚺 Integer 🔟 Inte	eger(2 words) 🔳	Eloating point 🖼	Hex 🔳 Tex	t 🖪 Ascii	Offline	e 0000/8000	C0-00DD1-D	1:1:A		NUM
Search the web and Wir	ndows			-		ð 🕘		W	<i>a</i>			🔇 🗆 🌾 🖣	10-24 AM



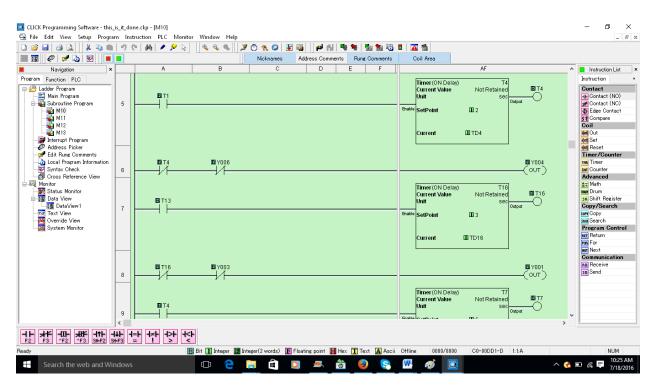


Fig. 20. M11 Subrouting - continuation

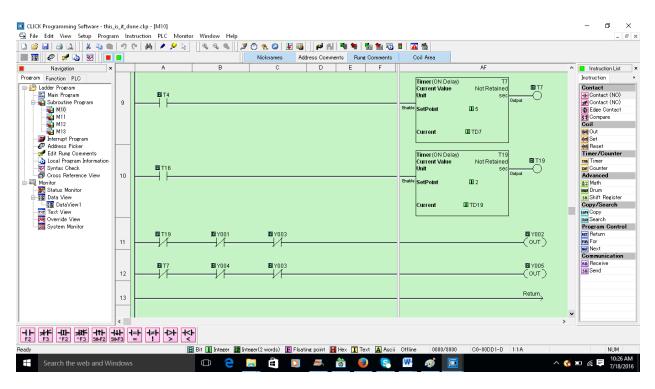


Fig. 21. M11 Subrouting - continuation

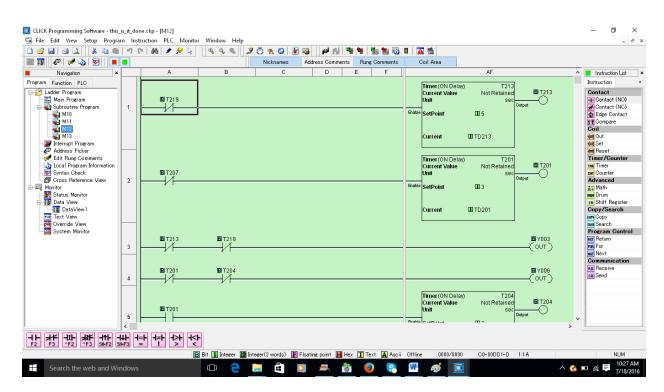


Fig. 22. M12 Subrouting

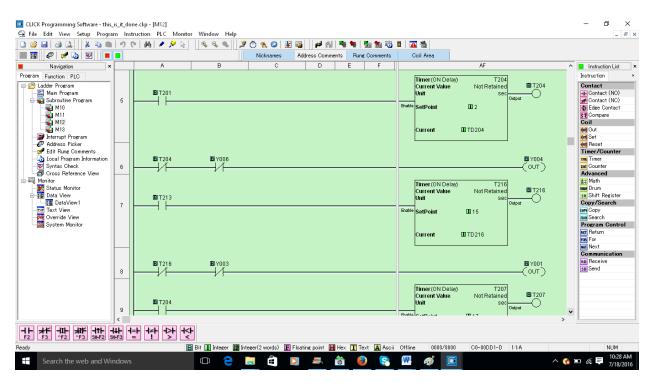


Fig. 23. M12 Subrouting - continuation

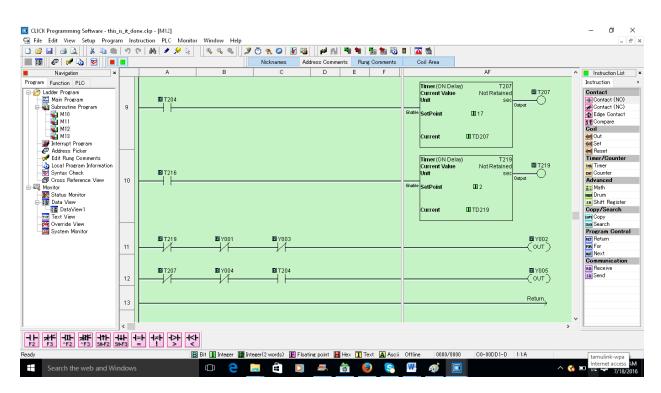
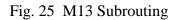


Fig 24. M12 Subrouting - continuation

🖸 CLICK Programming Software - this_jis_it_done.ckp - [M13] — 🖸													
□													
= 🖬   🖉 🕩 💆 📃				Nicknames Ad	ddress Comments	Rung	Comments	(	Coil Area				
Navigation ×		A	B	С	D	E	F			AF		^	Instruction List
Program Function PLC													Instruction +
Adder Program     Main Program     Main Program     Main Program     Mil     Mil	1	B C45							Timer (ON Dela) Current Value Unit SetPoint	Not Retained	utput		Contact Contact (NO) Contact (NO) Contact (NC) Edge Contact Compare
									Current	<b>II</b> TD200			Coil Geni Out Geni Set Geni Reset Timer/Counter
	2	<b>■</b> T200									■ Y006 —( OUT )		Tms Timer Counter
									Timer (Ohl Delay	A TG4			Advanced
	3 —	B T64	19 Y006						Timer (ON Delay Current Value Unit SetPoint Current	Not Retained	UTET TE1		and Drum sn Shift Register Copy/Search terv Copy sant Search Program Control nat Return rent For
	4		<b>IB</b> T64								■Y005 —(out)		Next Communication ND Receive SD Send
	5	B T61 B T64 (out)							—(out)	~			
Ready	Bit 🚺 Integer 🔢 Integ	🚺 Integer 🔢 Integer(2 words) 🕞 Floating point 📔 Hex 頂 Text 🔒 Ascii						0 C0-00DD1-D	1:1:A		NUM		
Search the web and Windows			💷 🤮 📑 🛱 💟 🛲 📸 🥘 🥞 🛄 🛷 🔟 ^ 🚱 🖉								// ■ 10:30 AM 7/18/2016		



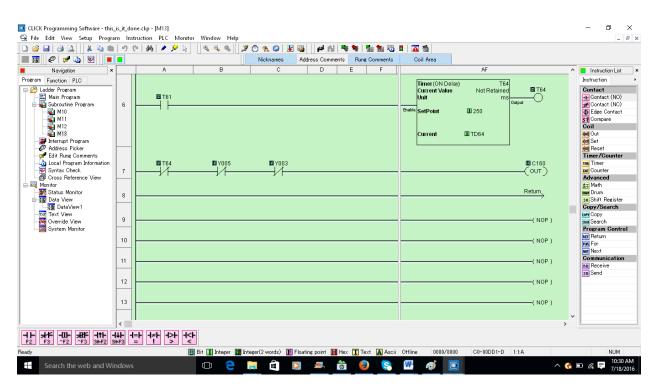


Fig. 26. M13 Subrouting - continuation

#### **Additional Multimedia support materials**

www.youtube.com/watch?v=qDuCusvU-mY

 $\underline{www.youtube.com/watch?v=LM8U9FCMDx0\&list=PLdnqjKaksr8qRPCFkU2Q8XQe0bfo99rs}{\underline{6}}$ 

Relays

www.youtube.com/watch?v=STKH6Rp7rX8

Introduction to PLCs

www.youtube.com/watch?v=d8jASbyqO6M

How to program a PLC

www.youtube.com/watch?v=1CMA9g27QmI

#### **Conclusion and Future Directions**

This learning activity provides an excellent opportunity for students to integrate their knowledge of PLC, input, ladder programing logic, output and the connection between them to create a Smart PLC system to control a Traffic Light.

<u>Challenges of the Smart PLC Traffic Light Controller</u>: This project intentionally leaves a margin for students to go farther with their work. Student teams can modify or expand the project considering the pedestrian crossing the intersection, and also thinking in drivers who want to do a left turn in any of the streets.

#### ACKNOWLEDGEMENTS

This material is based upon work supported by the Research Experiences for Teachers Program (RET) under the National Science Foundation under Grant No. 1300779. Any opinion, findings, and conclusion or recommendation expressed in this material are from the author(s) and do not necessarily reflect the view or opinion of the National Science Foundation.

#### REFERENCES

- 1. Hsieh, S. and Hsieh, P.Y., "Web-based Modules for Programmable Logic Controller Education," *Computer Applications in Engineering Education*, 13(4), Dec 2005, pp. 266279.
- 2. Hsieh, S. and Hsieh, P.Y., "An Integrated Virtual Learning System for Programmable Logic Controller," *Journal of Engineering Education*, 93(2), April, 2004.
- 3. Hsieh, S. and Hsieh, P.Y., "Animations and Intelligent Tutoring Systems for Programmable Logic Controller Education," *International Journal of Engineering Education*, 19(2), 2003.

- 4. Hsieh, S., "Reconfigurable and Scalable Automated Systems Projects for Manufacturing Automation and Control Education," 2011 ASEE Annual Conference [PDF].
- 5. Wikipedia, https://en.wikipedia.org/wiki/Programmable\_logic\_controller
- 6. Plc-scada-dcs.blogspot.com, http://plc-scada-dcs.blogspot.com/2013/12/basic-plc-ladder-programming-training\_6827.html#ixzz4Cu6ghNNO
- 7. Automation Direct, <u>www.automationdirect.com</u>