LABORATORY REPORT

LAB 10: Systems Integration (Microcontroller, PLC and Computer Systems).

GROUP H

PROGRAMME: MECHATRONICS ENGINEERING

GROUP MEMBERS:	MATRIC NO:
1. SYEDA SAMIA	2123536
2. ATHIRAH HAZIQAH BINTI BADERULHISHAM	2210240
3. SITI HEIDI AMIRA BINTI AZRUL	2111970

DATE OF SUBMISSION:
Friday, 24th May 2024

ABSTRACT

This project involves creating a ladder diagram for a blinking LED using OpenPLC Editor, then executing it via OpenPLC runtime on a Raspberry Pi. The Raspberry Pi will control the LED through its GPIO pins, ensuring the LED blinks continuously. This demonstrates the integration of ladder logic with hardware control, showcasing OpenPLC's capability to manage real-world automation tasks efficiently using a cost-effective Raspberry Pi setup.

Table of Contents

INTRODUCTION

This project focuses on creating a blinking LED system using OpenPLC, a widely adopted open-source PLC (Programmable Logic Controller) platform that provides a cost-effective alternative to traditional, proprietary PLCs (Programmable Logic Controllers). It supports standard PLC languages, such as ladder logic, allowing users to create and execute control programs for various automation tasks. By designing a ladder diagram in the OpenPLC Editor and deploying it on a Raspberry Pi running the OpenPLC runtime, we aim to demonstrate the practical application of ladder logic in controlling physical devices.

The Raspberry Pi is a small, single-board computer developed by the Raspberry Pi Foundation to promote computer science education and enable various electronics projects. Despite its compact size, the Raspberry Pi is a powerful and versatile device capable of running a full Linux operating system and interfacing with a wide array of sensors, actuators, and other peripherals through its GPIO (General-Purpose Input/Output) pins.

In this project, we will leverage the capabilities of the Raspberry Pi to control an LED. The process will involve creating a precise control sequence using the OpenPLC Editor, uploading the program to the OpenPLC runtime on the Raspberry Pi, and establishing the necessary hardware connections. This hands-on approach not only provides a clear example of how ladder logic can be applied to real-world tasks but also showcases the seamless integration between OpenPLC and Raspberry Pi, highlighting the potential for scalable and flexible automation solutions in various applications.

PROCEDURE

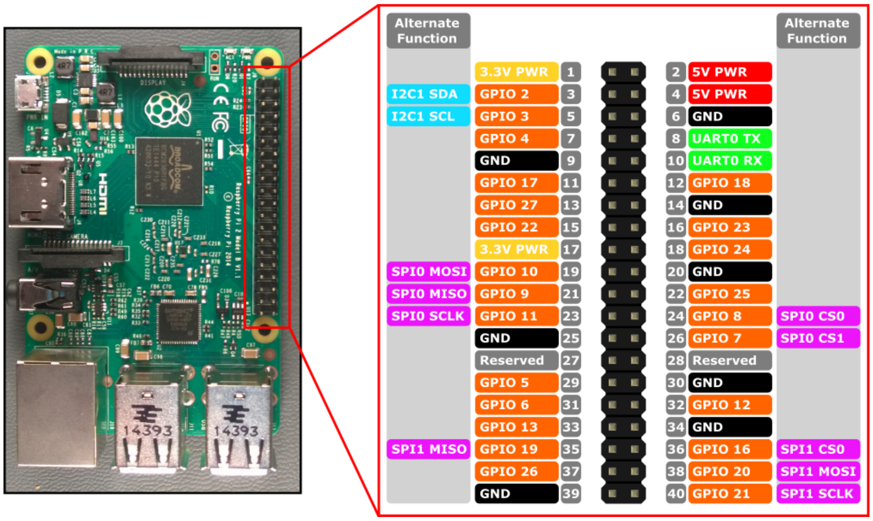
Materials And Equipment:

- Raspberry Pi
- Monitor
- Keyboard
- Mouse
- LAN cable
- LED
- Breadboard
- Jumper wires
- Resistor
- Power supply for the Raspberry Pi

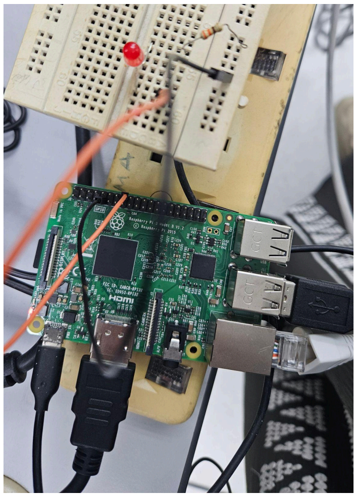
Experimental Setup:

1. Connect the keyboard, mouse, and monitor to the Raspberry Pi using USB cables.
2. Connect the LED to the Raspberry Pi.
    a. Connect one pin to pin 16 which is GPIO 23
    b. Connect the other pin to pin 9 which is the ground.
3. Connect the LAN cable to the Raspberry Pi.

4. Connect the Raspberry Pi to a power source.



Alternate Function					Alternate Function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK



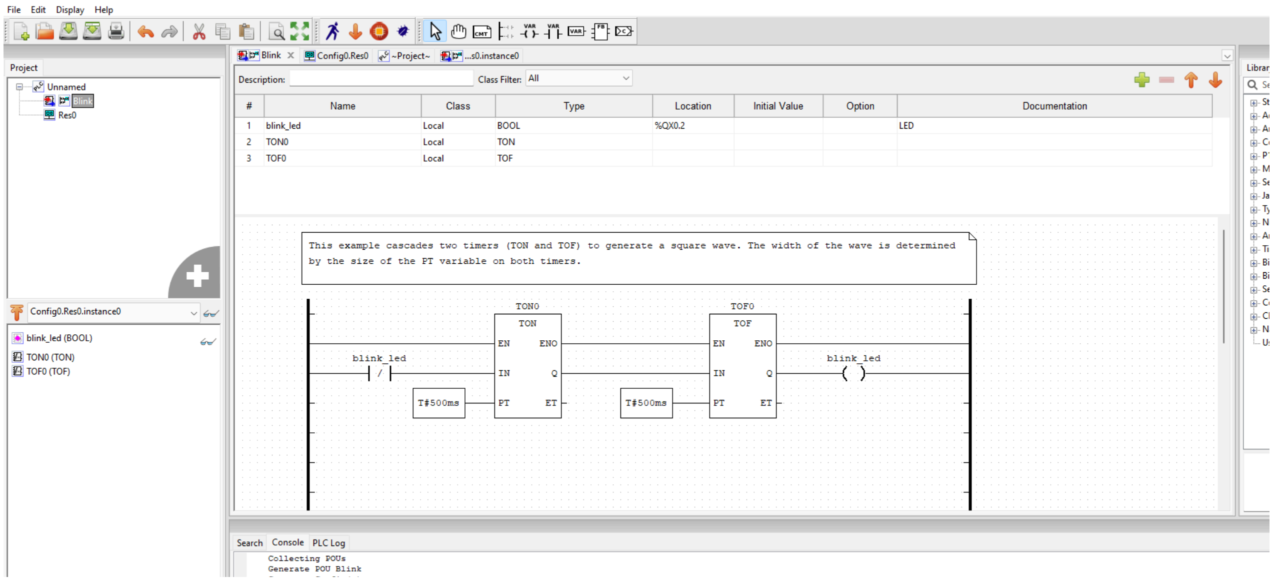
Methodology:

1. Power up the Raspberry Pi.
2. Open the command window and type:
    a. sudo apt install git
    b. git clone https://github.com/thiagoralves/OpenPLC_v3.git
    c. cd OpenPLV_v3/
    d. ls
    e. ./install.sh rpi
    f. ifconfig (to get the IP address)
3. Reboot Raspberry Pi after installation.
4. Access Open_PLC server.
    a. Open a browser.
    b. Search: (IP):8080
5. Login into the server (default login:

    username: openplc

    password: openplc)

6. Install OpenPLC Editor on a laptop.
7. Create a new program in the Editor.
8. Create the ladder diagram below and change the location of the blink_led to %QX0.2:



#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	blink_led	Local	BOOL	%QX0.2			LED
2	TON0	Local	TON				
3	TOF0	Local	TOF				

```
This example cascades two timers (TON and TOF) to generate a square wave. The width of the wave is determined
by the size of the PT variable on both timers.
```

```
Collecting POUs
Generate POU Blink
```

9. Save the program file.
10. On the Open_PLC server go to Programs then click on choose file and insert the ladder diagram save file.
11. Click the Upload program button.

RESULTS

The ladder diagram was uploaded and executed without errors on the Raspberry Pi. Furthermore, the LED blinked continuously at specified intervals, following the ladder logic. Moreover, the system operated reliably over an extended period of time, with the LED continuing to blink without interruption. Last but not least, the Raspberry Pi's GPIO pins successfully connected to the LED, allowing for precise control as specified by the ladder logic.

DISCUSSION

The experiment aimed to demonstrate the practical application of ladder logic programming in controlling a physical device, specifically an LED, using the OpenPLC platform deployed on a Raspberry Pi. The successful execution of the experiment validated the hypothesis that open-source tools can effectively facilitate real-world automation tasks.

The deployment of the ladder diagram on the Raspberry Pi and running the OpenPLC runtime proceeded smoothly without encountering any errors. The LED connected to the Raspberry Pi's GPIO pins blinked continuously as expected, demonstrating the successful execution of the control logic. The stable operation of the system over an extended period further underscored the reliability of both the OpenPLC runtime and the Raspberry Pi hardware. Moreover, the experiment highlighted the versatility of the Raspberry Pi as a platform for hardware interfacing and automation projects. Its GPIO pins provided the necessary interface for connecting and controlling the LED, showcasing the Raspberry Pi's capability to serve as a control device for various applications.

CONCLUSION

In conclusion, the experiment successfully demonstrated the practical capabilities of OpenPLC and Raspberry Pi in creating scalable and flexible automation systems. It provided valuable insights into the integration of software-based control systems with physical hardware, emphasizing the potential of open-source tools for realizing cost-effective automation solutions in diverse domains.

RECOMMENDATIONS

It is recommended to use a power source for the Raspberry Pi that is not the USB micro B type cable to have sufficient voltage supply. Furthermore, use a LAN cable to have a good wifi connection.

REFERENCES

2.4 Physical Addressing – Autonomy. (n.d.). https://autonomylogic.com/docs/2-4-physical-addressing/

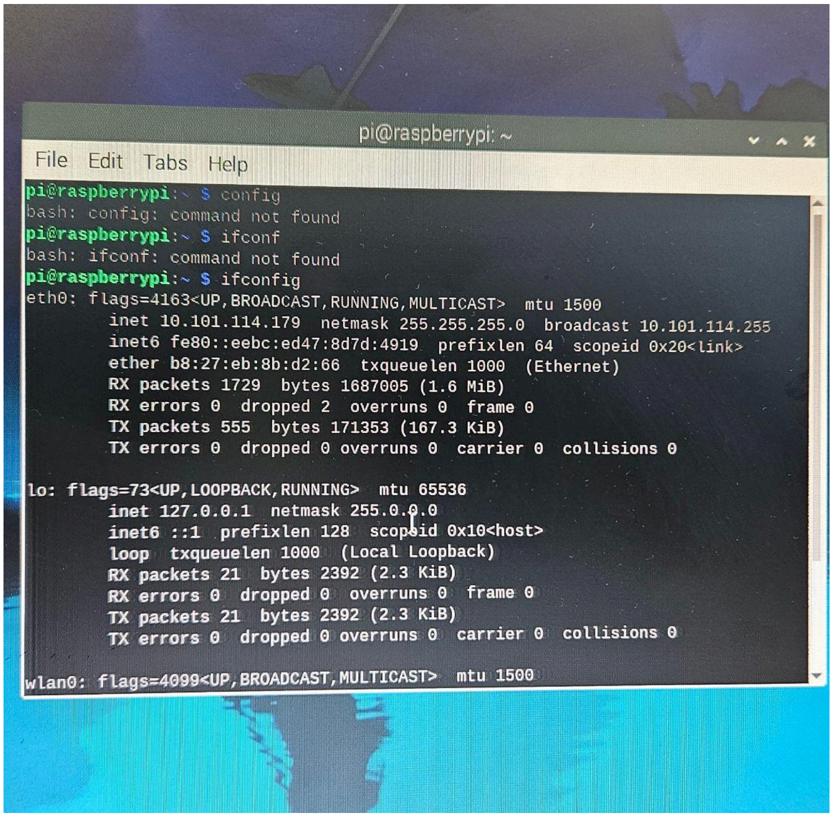
OpenPLC. (2022, October 27). Basics 04: OpenPLC on Raspberry Pi with Modbus. YouTube. https://www.youtube.com/watch?v=iC5s4CEiOB4&list=WL&index=29&t=980s

Brighterside Tech. (2023, July 22). Install OpenPLC on Raspberry Pi - Turn Your Raspberry Pi into a PLC. YouTube. https://www.youtube.com/watch?v=fRXaeerjasY&list=WL&index=28&t=193s

Butts, J. (2021, September 28). What Is the Raspberry Pi? How-to Geek. https://www.howtogeek.com/754492/what-is-raspberry-pi/

1.1 OpenPLC Overview – Autonomy. (n.d.). https://autonomylogic.com/docs/openplc-overview/

APPENDICES



```
pi@raspberrypi:~ $ config
bash: config: command not found
pi@raspberrypi:~ $ ifconf
bash: ifconf: command not found
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.101.114.179  netmask 255.255.255.0  broadcast 10.101.114.255
        inet6 fe80::eebc:ed47:8d7d:4919  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:8b:d2:66  txqueuelen 1000  (Ethernet)
        RX packets 1729  bytes 1687005 (1.6 MiB)
        RX errors 0  dropped 2  overruns 0  frame 0
        TX packets 555  bytes 171353 (167.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 21  bytes 2392 (2.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 2392 (2.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
```

ACKNOWLEDGEMENT

I would like to extend my sincere thanks to DR. Ali for his guidance and understanding. Additionally, I would like to express my sincere gratitude to our senior for her invaluable guidance and support throughout the execution of this experiment. Her expertise and insightful feedback greatly enhanced my understanding of the principles underlying Arduino-based electronics Furthermore, I am thankful to my peers for their collaborative spirit and exchange of ideas, which enriched the overall learning experience and fostered a collaborative learning environment.

STUDENT'S DECLARATION

This is to certify that we are responsible for the work submitted in this report, that the original work is our own except as specified in the references and acknowledgement, and that the original work contained herein have not been untaken or done by unspecified sources or persons.

We hereby certify that this report has not been done by only one individual and all of us have contributed to the report. The length of contribution to the reports by each individual is noted within this certificate.

We also hereby certify that we have read and understand the content of the total report and no further improvement on the reports is needed from any of the individual's contributors to the report.

We, therefore, agreed unanimously that this report shall be submitted for marking and this final printed report has been verified by us.

Name: Syeda Samia Matric Number: 2123536 Contribution: Abstract, Introduction, Procedure, Results, Discussion, Conclusion, Recommendations	Read	✓
	Understand	✓
	Agree	✓
Name: Athirah Haziqah bt Baderulhisham Matric Number: 2210240 Contribution:	Read	✓
	Understand	✓
	Agree	✓
Name: Siti Heidi Amira Binti Azrul Matric Number: 2111970 Contribution:	Read	✓
	Understand	✓
	Agree	✓