INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

Garden of Knowledge and Virtue

LAB REPORT

MCTA 3203

SECTION 1

GROUP D

EXPERIMENT 10

Systems Integration (Microcontroller, PLC and Computer Systems)

MATIC NO.	NAME
2111633	MUHAMMAD FARID BIN JAFRI
2110765	MUHAMMAD ARIF SYAZWAN BIN MOHD RIZAL
2112969	MUHAMMAD HAZIQ HAIKAL BIN SUAIB
2115207	MUHAMMAD AMIRUL SYAFIQ BIN RUSLANI
2116281	MUHAMMAD FADHLUL WAFI BIN AHMAD NAIM

SUBMISSION DATE: 10/01/2024

INSTRUCTOR: DR. NADZRIL BIN SULAIMAN

Abstract

The goal of this project is to use Modbus transmission to link sensors, microcontrollers, and computers into a single system for industrial automation and control. There are three kinds of sensors in the system: an infrared (IR) sensor, a touch sensor, and a light-sensitive resistor (LDR). These are all linked to an Arduino, which acts as a Modbus slave. The Raspberry Pi is set up as a Modbus Master and is in charge of getting sensor data from the Arduino. Additionally, OpenPLC is used as an extra Modbus Slave to receive data from the Raspberry Pi and manage an output based on that data. This paper describes how to set up, configure, and show off the system, showing how it can use Modbus transmission to watch sensor data in real time.

Table of Contents

Introduction

A cohesive system's ability to integrate disparate components has become essential for effective and intelligent operations in the rapidly changing field of industrial automation. Using the Modbus communication protocol, this experiment demonstrates how an OpenPLC, three sensors, three outputs, and a Raspberry Pi can seamlessly integrate into a single system.

This project utilizes the Raspberry Pi as the Modbus Master. It is in charge of coordinating with an Arduino that is set up as a Modbus slave, acting as a master device. Three sensors whose data the Raspberry Pi wants to read are hosted by the Arduino, which in turn serves as a Modbus Slave. Data from the sensors is sent to the Raspberry Pi using the Modbus protocol, which is essential for obtaining information from the actual world.

The Raspberry Pi and an OpenPLC system that is set up as a Modbus Slave converse simultaneously. After receiving data from the Raspberry Pi, the OpenPLC takes over the responsibility of controlling three outputs. Furthermore, one of the outputs is controlled by the OpenPLC in response to sensor data that are received from the Arduino. The Modbus protocol allows for bidirectional communication between the Raspberry Pi and the OpenPLC, which makes it easier to exchange data which is essential for the system's dynamic control.

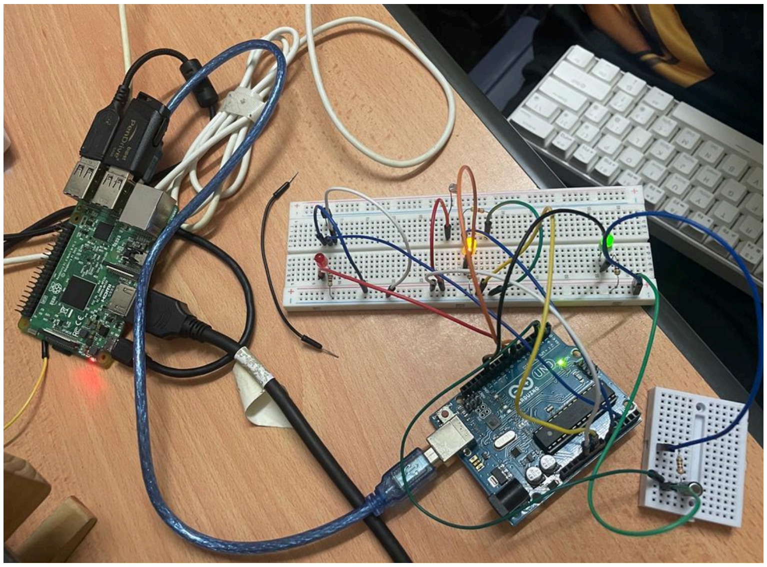
This experiment highlights the value of using established communication protocols like Modbus to promote interoperability and easy integration among various industrial components in addition to showcasing the real-world use of a distributed control system. The combination of Raspberry Pi, Arduino, and OpenPLC demonstrates the possibility of creating responsive, intelligent systems with real-time data processing, control, and acquisition capabilities.

Materials and Equipment

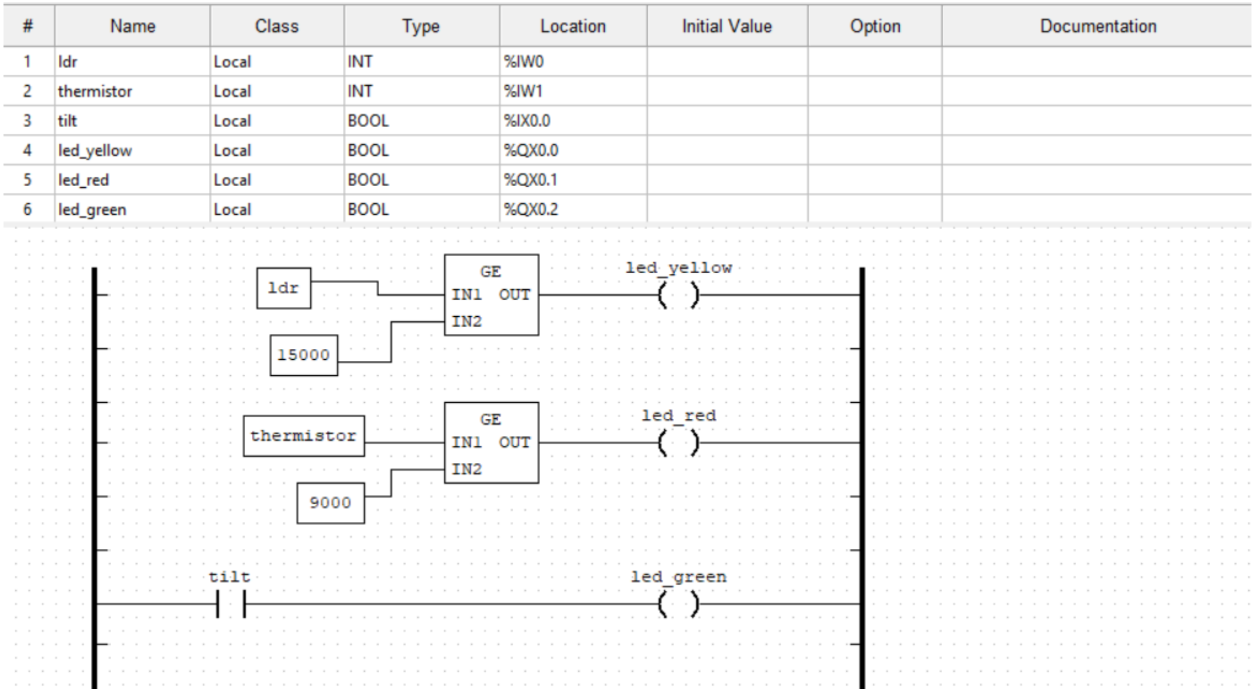
1) Raspberry Pi
2) Arduino Uno R3
3) Jumper Wires
4) LED
5) Resistor
6) Thermistor
7) LDR sensor
8) Tilt sensor
9) Breadboard
10) USB cable for Arduino Uno R3
11) HDMI cable

Experimental Setup

1. Raspberry Pi was connected to the Arduino as well as a monitor for display.
2. All the sensors and the output were connected to the Arduino as the diagram below:



3. A ladder diagram has been constructed as diagram below:



#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	ldr	Local	INT	%IW0			
2	thermistor	Local	INT	%IW1			
3	tilt	Local	BOOL	%IX0.0			
4	led_yellow	Local	BOOL	%QX0.0			
5	led_red	Local	BOOL	%QX0.1			
6	led_green	Local	BOOL	%QX0.2			

Methodology

1. Sensors (name of the sensor that we used) are selected, and Modbus RTU serial protocol has been chosen for communication to establish a master-slave connection
2. Sensors are attached to the breadboard and connected to Arduino and Arduino-Raspberry Pi communication is established.
3. Arduino is programmed to read the sensors' data and Modbus communication is implemented for data transmission to Raspberry Pi.
4. The ladder diagram is developed with OpenPLC by using graphical symbols to depict logical control sequences for operating and managing the system components (sensors, Arduino, Raspberry Pi) based on input conditions and defined actions.
5. OpenPLC Runtime is installed in the system to demonstrate the system's ability to monitor all the sensors’ data in real-time and validate the data accuracy.

Results (Observation)

We have effectively developed a system integrating a thermistor, LDR sensor, tilt sensor, Raspberry Pi, and Arduino, utilizing the Modbus communication protocol. Additionally, we have implemented real-time monitoring of all sensor data through the OpenPLC Runtime.

As you can refer in the videos:

1. When a higher temperature is applied to the thermistor, the red LED illuminates indicate that it is true and the corresponding value in the OpenPLC Runtime increases. As the temperature returns to normal, the LED turns off, signifying a false status.
2. When LDR is exposed to the light, the yellow LED illuminates indicate true status and the corresponding value increases in the OpenPLC Runtime. As the LDR is covered or not exposed to the light, the LED turns off, signifying a false status and the value becomes negative.
3. When the tilt sensor is in the normal position (upright) the green LED illuminates and it appears true in the OpenPLC Runtime, then when the sensor is tilted, the LED turns off, signifying false status as shown in the OpenPLC Runtime.

Link for the videos:

1. https://drive.google.com/file/d/1wXh5W7td3R3ZwCpPnKKQtIwd-61d0UzN/view?usp=drivesdk
2. https://drive.google.com/file/d/1visa5vuOJfaVhvHt__49zMC6S7ULFKMi/view?usp=drivesdk

Discussion

In this experiment, we aim to develop a comprehensive sensor system using a combination of sensors, a microcontroller, and a computer. We used simple sensors as inputs such as a thermistor, a LDR and a tilt sensor. For the outputs, we just used simple LEDs with various colors where each input will give a different color of outputs. When we set up the Ladder diagrams using OpenPLC Editor to provide the command to the microcontroller - Arduino UNO, we expected that we would just need to add merely input with *contact* but somehow the sensors cannot give the value to activate the outputs. But then after some findings, for the sensors to read and provide the data, we only need to use *variable* instead of *contact*.

For the thermistor, we used a comparison Greater Than *block* to activate the Yellow LED once the reading reached a certain point. But, the problem here is that we could not find a way to make OpenPLC Runtime show the true temperature value, instead it shows the raw reading of the thermistor. We tried to use an arithmetic *block* to change the value into the true temperature, but somehow it won't work. For the LDR, we used a comparison Greater Than *block* as well to activate the Green LED and we have no problem with the value since we can set the value for dark and bright. Lastly, for the tilt sensor, we just directly use normal *contact* as an input to activate the Red LED and it has no problem as well since the type of value is just BOOL compared to other sensors which are INT type.

Conclusion

This experiment can be said to be a successful integration between Raspberry Pi as Modbus Master, microcontroller as Modbus Slave and sensors - thermistor, LDR and tilt sensor. The system demonstrates the efficiency of Modbus in industrial applications by showcasing how it facilitates smooth communication between devices for control and monitoring. As students, we gain valuable insights into sensor systems through Ladder diagrams as well as communication protocols. This skill definitely will help us pursue careers in automation and control engineering.

Recommendations

Based on our experience experimenting in class and laboratory, we believe that code optimization is one of the most important things to focus on. Optimizing the Raspberry Pi and Arduino Uno R3 will directly make the entire system run smoothly. Adding comments to each code helps in terms of understanding the code much better and avoiding mistakes during runs of the experiment especially for students who just started learning programming.

Next, the recommendations that will help improve these lab projects are fault tolerance and error handling. Improvising the system's robustness by incorporating fault-tolerant mechanisms and appropriate error handling. Also, provide simple and clear wrong parts and feedback to the user when the system is incorrect or the sensor malfunctions.

Last but not least, integration with industrial components is one of the good ways to introduce students to basic components that are usually used in the industry. There are various types of components which include actuators, additional controllers, or specialized sensors to broaden the project's scope.

References

1. O. (2022, October 22). *Basics 03: Programming an Arduino Board with OpenPLC*. YouTube. https://www.youtube.com/watch?v=d39Al4w1sL0
2. E. S. (2020, August 21). Tilt Sensor with Arduino. YouTube. https://www.youtube.com/watch?v=Uve-nEE1M-4
3. E. M. E. I. M. (2023, June 20). DIY Arduino PLC - How to Use the Analog Input in OpenPLC. YouTube. https://www.youtube.com/watch?v=c1HVic0NWMU
4. O. (2022, October 28). *Basics 04: OpenPLC on Raspberry Pi with Modbus*. YouTube. https://www.youtube.com/watch?v=iC5s4CEiOB4
5. O. (2022, November 5). *Basics 05: Expanding PLC IOs with OpenPLC Slave Devices*. YouTube. https://www.youtube.com/watch?v=dQ8GcfpGDuA

Student's Declaration

Certificate of Originality and Authenticity

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 acknowledgment, and that the original work contained herein has 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 contributor 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

Signature: arifsyazwan	Read✓
Name: Muhammad Arif Syazwan Bin Mohd Rizal	Understand ✓
Matric Number:2110765	Agree ✓
Signature: haziqhaikal	Read ✓
Name: Muhammad Haziq Haikal bin Suaib	Understand ✓
Matric Number: 2112969	Agree ✓
Signature: amirul	Read ✓
Name: Muhammad Amirul Syafiq Bin Ruslani	Understand✓
Matric Number: 2115207	Agree✓
Signature: Wafi	Read ✓
Name: Muhammad Fadhlul Wafi Bin Ahmad Naim	Understand ✓
Matric Number: 2116281	Agree ✓
Signature: faridjafri	Read ✓
Name: Muhammad Farid Bin Jafri	Understand ✓
Matric Number: 2111633	Agree ✓