

**LAB 10:**

**Systems Integration**

**(Microcontroller, PLC, Computer Systems)**

**MCTA 3202**

GROUP F

GROUP MEMBERS MATRIC NO

1. Muhammad Naufal bin Mohammad Bakri 2110333
2. Muhammad Khairul Aiman bin Mohamad Basari 2113571
3. Muhammad Nabil Ikhmal Bin Khairil Rizal 2114577
4. Muhammad Zakwan Zikri bin Irwan Affandi 2111033
5. Che Muhammad Danial Bin Che Suhaimi 2119075

DATE OF SUBMISSION :

3rd January 2024

**Table of Contents**

[**Abstract 2**](#_ppngj06qmhf9)

[**Introduction 2**](#_mv8p81ag3gkh)

[**Materials and Equipment 2**](#_g941gudcypw5)

[**Experimental Setup 3**](#_j3ljcad4ojh)

[**Data Analysis 4**](#_pto93aji3shy)

[**Results 5**](#_aq7lie7o1qii)

[**Discussion 5**](#_89b52bjz24m3)

[**Conclusion 5**](#_2ywjb614l6vp)

[**Recommendations 6**](#_iu1zm3qs31z9)

[**References 6**](#_o4vgsbxvmw6s)

[**Appendices 6**](#_t1ytbocvmdah)

[**Student's Declaration 7**](#_9efyym5v108d)

# **Abstract**

This project aims to develop an integrated system for industrial automation and control, utilizing Modbus communication to connect sensors, microcontrollers, and computers. The system includes three types of sensors—Infrared (IR) sensor, Touch sensor, and Light-Dependent Resistor (LDR)—connected to an Arduino, serving as a Modbus Slave. A Raspberry Pi is configured as a Modbus Master, responsible for collecting sensor data from the Arduino. Additionally, OpenPLC is employed as another Modbus Slave, receiving data from the Raspberry Pi and controlling an output based on the received sensor data. This paper outlines the setup, configuration, and demonstration of the system, showcasing its ability to monitor sensor data in real-time through Modbus communication.

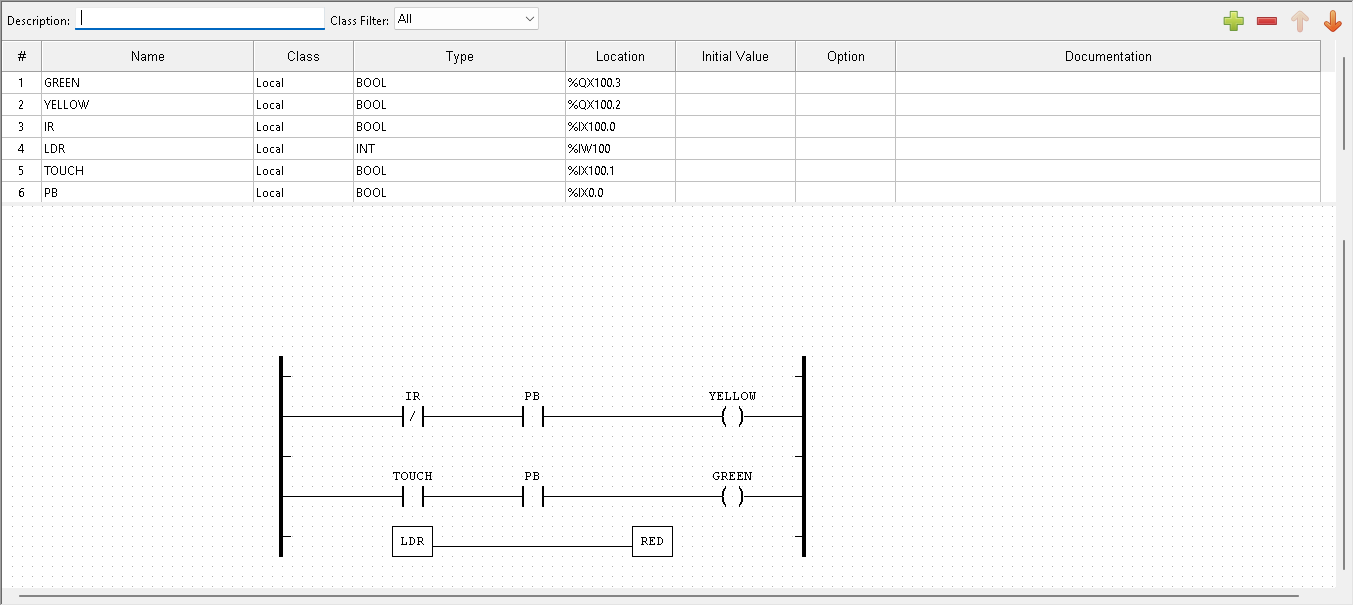
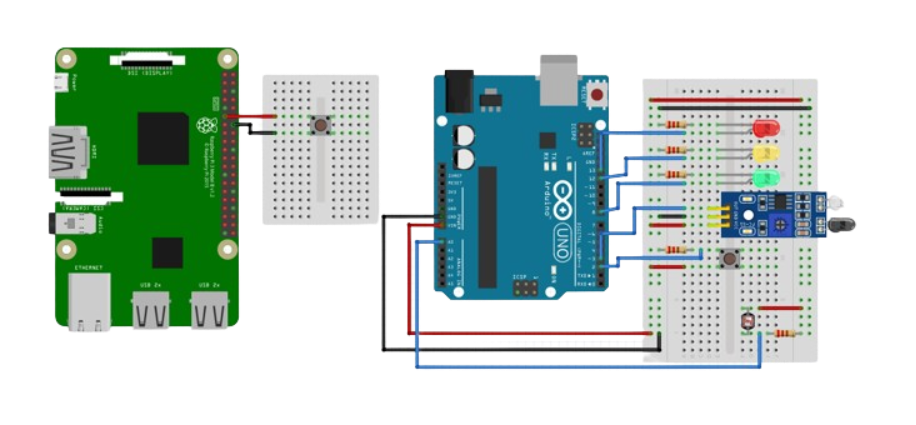
# **Introduction**

In industrial automation and control systems, efficient communication between devices is crucial. Modbus, a widely adopted communication protocol, proves instrumental in achieving seamless integration between programmable logic controllers (PLCs), microcontrollers, sensors, and computers. This project explores the implementation of a system consisting of three sensors—Ir sensor, Touch sensor, and LDR—interfaced with an Arduino, which acts as a Modbus Slave. The communication backbone is formed by a Raspberry Pi, configured as a Modbus Master, orchestrating the data exchange with the Arduino. Furthermore, OpenPLC, configured as a Modbus Slave, receives data from the Raspberry Pi and demonstrates the ability to control an output based on the incoming sensor data. The simplicity, flexibility, and reliability of Modbus make it an ideal choice for facilitating communication in various industrial applications.

# **Materials and Equipment**

* Arduino Uno
* Raspberry Pi
* IR Sensor
* Touch Sensor (replaced by the pushbutton)
* Light-Dependent Resistor (LDR)
* Jumper Wire
* Computer with OpenPLC application

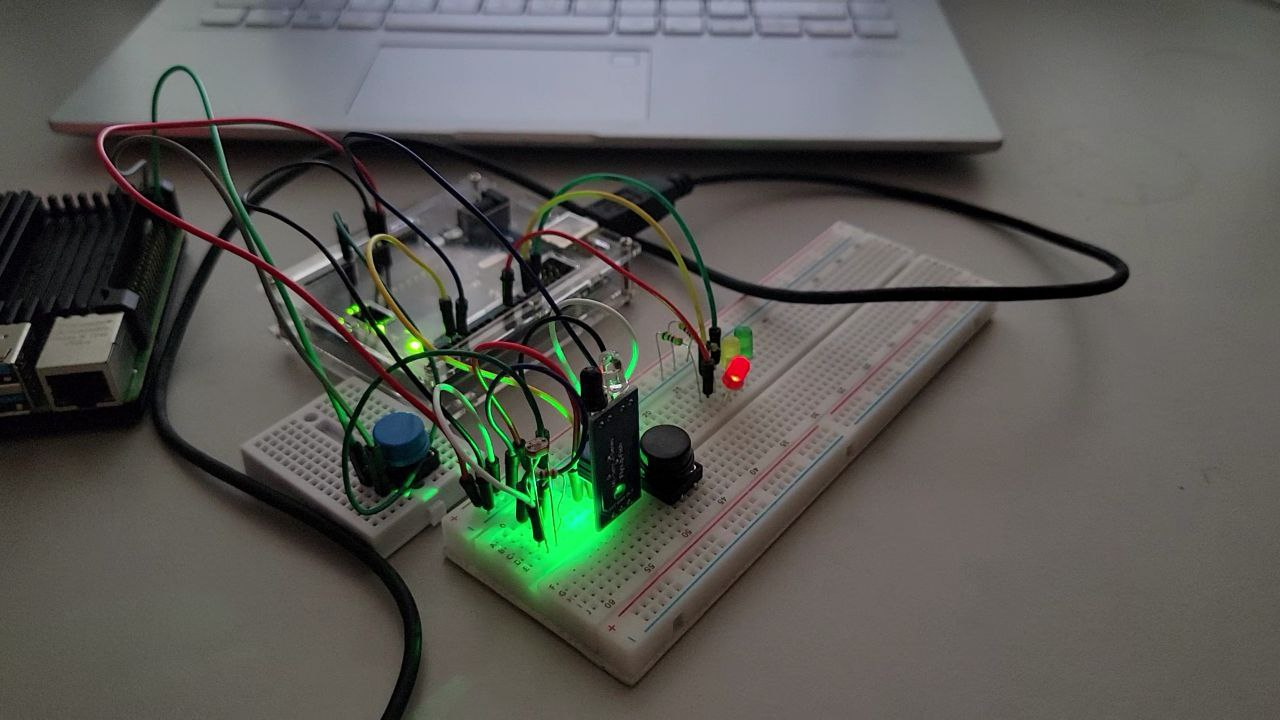
# **Experimental Setup**



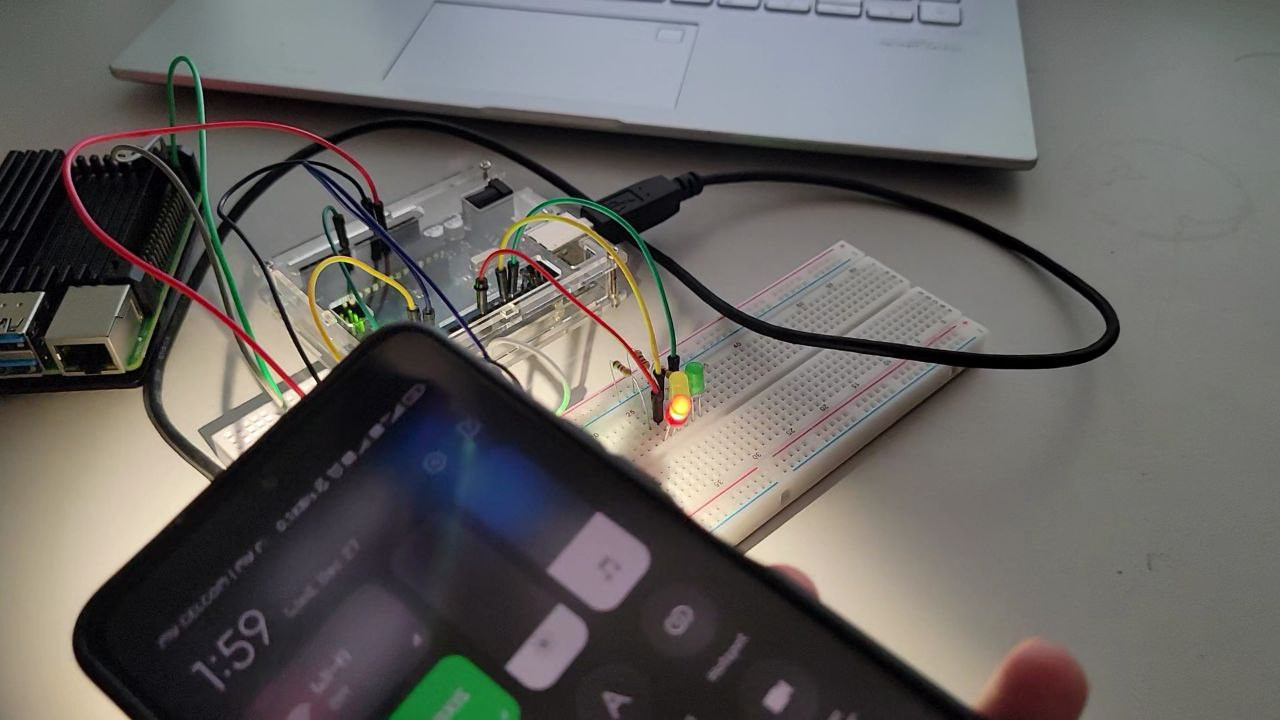
**Methodology**

1. Connect the Raspberry Pi to the Arduino via Modbus for master-slave relationship
2. Connect the LDR, IR sensor and touch sensors to the breadboard
3. LED is connected to indicate the feedback from the input
4. Design a ladder diagram using OpenPLC to execute the program
5. Test out each inputs to verify the connection from master to slave are successful

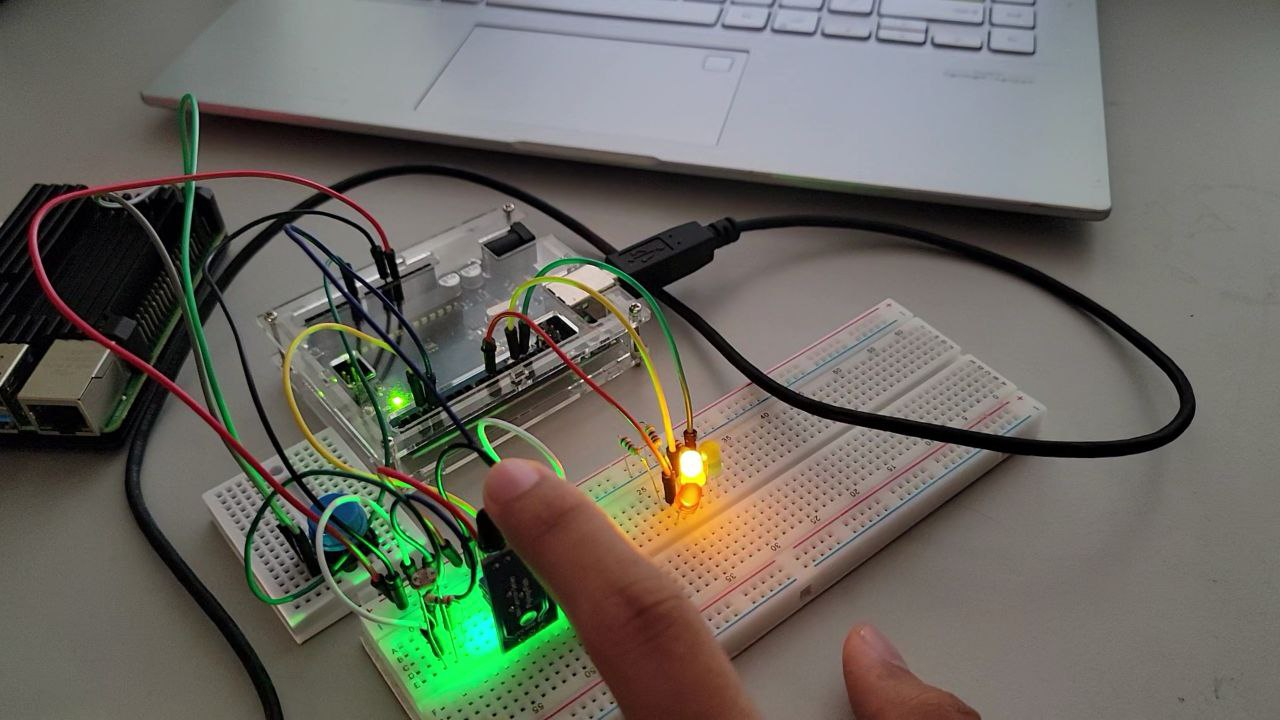
# **Data Analysis**



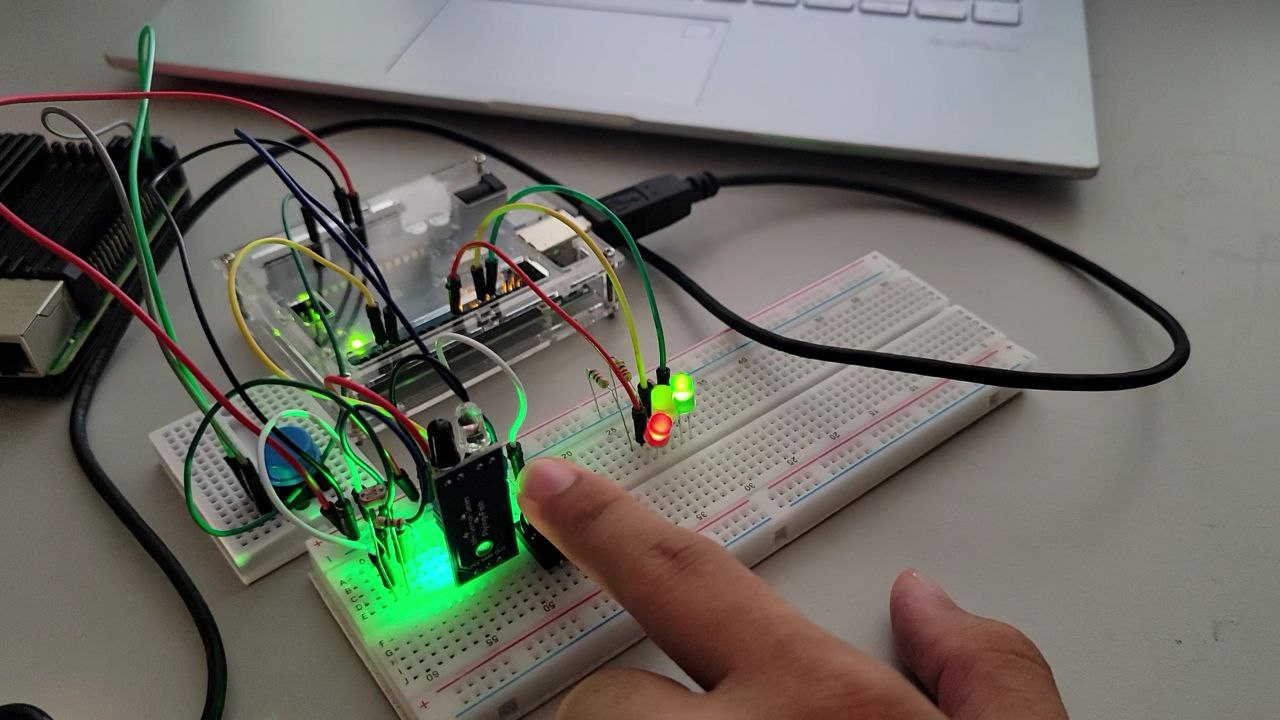
Completed Circuit



Red LED turns on when light is shine at LDR



Yellow LED turns on when finger is blocking the IR sensor



Green LED turns on when touch sensor is pressed

# **Results**

<https://github.com/NotLafuan/GROUP-F-MCTA-3203/raw/main/Week%2010/Group%20F%20Week%2010%201.mp4>

<https://github.com/NotLafuan/GROUP-F-MCTA-3203/raw/main/Week%2010/Group%20F%20Week%2010%202.mp4>

# **Discussion**

During our experiment we stumbled upon some problem while operating the components. While we were testing the sensors, the IR sensors and light-dependent resistor, it would sometimes remain unresponsive when tested or would not provide the desired output of green, yellow or red LED. After troubleshooting, we discovered that it might be caused by faulty connections and sensor degradation.

Moreover, the IR sensor is initially quite faulty in the beginning such as it would not detect when an object is placed in its sensor view, however after some fixtures, this problem might be caused by uncalibrated sensor and connection issues which finally is resolved.

Other than that, the Raspberry Pi had some serial communication issues but it was not unsolvable since we had to only adjust the value of baud rates for it to operate as we wanted. Other than that we also had to make sure the serial port chosen matches with the current serial port used in operation.

Moreover, there were some latency issues while we were executing the operation where there were some delays in the data transmission between the components hence affecting the real-time responsiveness of the system. In order to mitigate the issues that existed, we decided to check thoroughly for errors in coding so that it would operate as we wanted it.

# **Conclusion**

In conclusion, integrating multiple electronic components such as IR sensor, touch sensor, light-dependent resistor (LDR), Arduino, Raspberry Pi, and openPLC can be quite challenging to manage and organize however on the upside it could provide various benefits in the developing our skills to survive in the engineering industry in the future.

From this experiment, we learn that Raspberry Pi could act as a central hub for control and monitoring as its function is to create a streamlined control, data processing as well as reducing the complexity of individual components. The Modbus communication protocol has proven to be successful in creating a seamless data exchange and has proven its efficiency in creating a Master-Slave Architecture.

This experiment has also helped us to simulate real-time monitoring and response with inputs provided from the sensor data and how it is quick in response to changes in environment, minimizing downtime as well as other potential issues.

# **Recommendations**

We are doing the experiment of Modbus as a communication protocol where Raspberry Pi is a master and Arduino is a slave on the OpenPLC. For a further implementation in a networked environment, the security factor might be the thing that we want to consider. For instance, if using Modbus TCP, ensure that your network is secure, and you might want to explore encryption and authentication methods.

# **References**

<https://www.youtube.com/watch?v=_QETdyeMyZw>

<https://www.youtube.com/watch?v=yCcfrDmbZmM&t=339s>

<https://www.youtube.com/watch?v=QmNhdl1WREc&t=2s>

# **Appendices**

# 

# **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 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.

Signature: Naufal Read

Name: Naufal Understand

Matric Number: 2110333 Agree

Contribution: Testing and Debugging

Signature: Aiman Read

Name: Aiman Understand

Matric Number: 2113571 Agree

Contribution : Programming

Signature: Nabil Read

Name: Nabil Understand

Matric Number: 2114577 Agree

Contribution: Circuit Design

Signature: Zakwan Read

Name: Zakwan Understand

Matric Number: 2111033 Agree

Contribution: Data Analysis

Signature: Che Read

Name: Che Understand

Matric Number: 21119075 Agree

Contribution : Discussion