EVENT ANALYSIS ON CONVEYOR SORTER SYSTEM WITH IOT GATEWAY APPLICATION

LAU CHUNG CHING

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

EVENT ANALYSIS ON CONVEYOR SORTER SYSTEM WITH IOT GATEWAY APPLICATION

LAU CHUNG CHING

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled "EVENT ANALYSIS ON CONVEYOR SORTER SYSTEM WITH IOT GATEWAY APPLICATION is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ....

Name : ....

Date : ....

APPROVAL

I hereby declare that I have checked this report entitled "Event Analysis On Conveyor Sorter System With IoT Gateway Application" and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

First of all, I would like to thanks to Universiti Teknikal Malaysia Melaka (UTeM) for giving the opportunity to me to undertake my Final Year Project in partial fulfilment for Bachelor of Mechatronics Engineering. I am also grateful for having a chance to meet a lot of great people and professionals who led me through this period.

For the first person I want to highlight in my Final Year Project period is my research supervisor, Dr. Saifulza bin Alwi@Suhaimi. I am using this opportunity to express my deepest gratitude and special thanks to him because he continuous support and encouraged me along this semester of this research. During my research period, he also brings passion to me on learning new things and led me to explore a lot of knowledge that I never had in my class.

Last but not least, I would like to thanks my family. With their love, support and encouragement, I am tough to go through the challenges face by me. My friends, especially who has helped me in every possible way to complete this report also very appreciated by me.

ABSTRACT

Nowadays, the words “smart factories” are very popular in this information age due to the presence of forth industry. In Malaysia, many industry rather than investing in automation and IT, they prefer to keep their foreign workers. Hence, many manufacturing systems is not able to handle big data integration. The problem statement of this project is to verify the reliability of sensing control program, engineer needs to actually run of the manufacturing process to avoid the occurrence of a program error. The objective is to obtain the status of inputs and outputs (I/O) condition from conveyor sorter system by using Arduino Uno, to study the integration between Arduino Uno and raspberry pi for sending the data to the cloud and to compare between two different cloud data base in term of efficiency on timing. The first part of the report was an introduction of the project. The detail of the motivation, problem statement, objective and scope is mentioned in detail. In chapter 2, the literature review has conclude the previous and latest work that related to the project. Python programming language is studied to conduct the IoT gateway application using Raspberry Pi. In chapter 3, the method used and experiment was discussed in detail to achieve the objective. There was two programming platform that involve in this project which are Arduino IDE and Python language. Furtermore, some experiment is carried out to test the reliability of the data transmit between physical device with cloud server. Then, the analysis of the result is conducted by using different method of communication. Furthermore, the suitability of each method used in different condition is also discussed.

ABSTRAK

Pada masa kini, kata-kata "kilang-kilang pintar" sangat popular di zaman maklumat ini kerana kehadiran industri. Di Malaysia, banyak industri daripada melabur dalam automasi dan IT, mereka lebih suka mengupsh pekerja asing mereka. Oleh itu, banyak sistem perkilangan tidak dapat mengendalikan integrasi data yang besar. Pernyataan masalah projek ini adalah untuk mengesahkan kebolehpercayaan program kawalan sensing, jurutera perlu menjalankan proses pembuatan untuk mengelakkan berlakunya kesilapan program. Objektifnya adalah untuk mendapatkan status input dan output (I / O) dari sistem pengikat penghantar dengan menggunakan Arduino Uno, untuk mengkaji integrasi antara Arduino Uno dan raspberry pi untuk menghantar data ke atas talian dan membandingkan antara dua perkhidmatan talian yang berbeza pangkalan data dalam jangka masa kecekapan. Bahagian pertama laporan itu adalah pengenalan projek. Perincian motivasi, pernyataan masalah, objektif dan skop dinyatakan secara terperinci. Dalam bab 2, kajian literatur telah menyimpulkan kerja sebelumnya dan terkini yang berkaitan dengan projek. Bahasa pengaturcaraan Python dikaji untuk menjalankan aplikasi gateway IoT menggunakan Raspberry Pi. Dalam bab 3, kaedah yang digunakan dan percubaan dibincangkan secara terperinci untuk mencapai matlamat. Terdapat dua platform pengaturcaraan yang melibatkan projek ini iaitu bahasa Arduino IDE dan Python. Tambahan pula, beberapa uji kaji dijalankan untuk menguji kebolehpercayaan pemindahan data antara peranti fizikal dengan pelayan awan. Kemudian, analisis keputusan dijalankan dengan menggunakan kaedah komunikasi yang berbeza. Selain itu, kesesuaian setiap kaedah yang digunakan dalam keadaan yang berbeza juga dibincangkan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IoT	-	Internet of Things
I/O	-	Inputs and Outtputs
GPIO	-	General Purpose Input Output
PLC	-	Programmable Logic Controller
CPU	-	Central Processing Units
ROM	-	Read Only Memory
RAM	-	Random Access Memory
LD	-	Ladder Diagram
FBD	-	Function Block Diagram
IR	-	Infrared

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In an information age, internet playing an important role in our lifestyle, education, commercial hub, industrial zone, military, seafaring district, aerodrome, spaceport, and so on. Today, so call "smart factories" are mainly focus on monitor centered optimization and intelligence via the internet. Moreover, greater intelligence can be achieved by interacting with completely different encompassing systems, which have an instantaneous impact on machine performance [1].

In today's information society, the quality and the correctness of the systems is one of the challenges to manage big data integration in industry field. The big data platform is to handle the data in new ways in which as compared to the normal online database, hence many industries face the challenges of big data integration. For example, big data talent gap, data in a big data structure, data synchronization across data sources, data extracting in big data integration, uncertainty of data management and so on. This all about the integration of big data and some of the challenges facing through Industry 4.0 during implementation. Therefore, formal verification and validation tools are needed to improve the consistency checker which produces detailed feedback about the detected error.

The key milestones in this evolution and the use of formal verification methods have enhanced the dependability of manufacturing software. This is very important to all production and manufacturing industry because when the delivery systems go wrong: delayed progresses, item transferred to the wrong department and end up in the remote place, there will be big trouble for the industries. Over the years, there is a number of researchers have done to increase the level of dependence of manufacturers on automation systems that are expected to be safe and reliable which is increase the dependability of the management systems.

1.2 Motivation

Nowadays, computerized decision making and huge data that proliferated by the internet are the challenges to the companies and competitive business environment. Many manufacturing systems don’t seem to be able to handle big data integration due to the lack of intelligent management tools. For this reason, the German government has promoted the methods of computerization of manufacturing which is 4th Generation Industrial Revolution (Industry 4.0) base on Cyber-Physical System (CPS) to improve overall performance and maintenance management.

In Malaysia, many company and industry rather than investing in automation and IT, they prefer to keep their foreign workers. As a result, Malaysia is considered to be stuck at Industry 3.0 level in terms of manufacturing technology. In order to achieve flexible and efficient production, industry 4.0 is obtainable a brand new incentive which is Internet of Things (IoT).

The implementation of IoT nowadays is important in our lifestyle, education, commercial hub, industrial zone, military, seafaring district, aerodrome, spaceport, and so on. In industrial field, 60% of global manufacturers can use analytics data tracked using a connected device to analyze and optimize processes. With IoT, machinery can transmit the operation data information to the partners like original equipment manufacturers or field engineers to remotely manage the factory units and take advantage of process automation and optimization[2]. Besides that, the facility management which enables condition-based maintenance alert also can achieve with the implementation of IoT.

1.3 Problem Statement

In a massive automation control system or production line, there always face a similar problem which is when the error occurs in the system or machine, the operator should investigate the problem in the systems. If this happens frequently obviously will affect productivity. That’s because of the physical system is not sufficient to do the analysis for the data from the systems. Therefore, Implementation of IoT to the physical systems is needed of industrial field use for investigation and authentication at any part of complex systems. The implementation of IoT not only benefit the industrial field in term of efficiency in the production flow, accuracy facility management, inventory management, plant safety and security, quality control, packaging optimization and logistics, and supply chain optimization, it also benefits to others professional field. For example, architecture design, gathering necessities, blueprint, maintenance, implementation, examination, and development

Therefore, to verify the reliability of the sensing control program prior to an actual run of the manufacturing process is very crucial in order to avoid the occurrence of control program errors. Engineers of the manufacturing lines must ensure that the control program meets the predetermined requirements and specifications for any particular system.

1.4 Objective

The objective of the project are:

1. To obtain the status of inputs and outputs (I/O) condition from conveyor sorter system by using Arduino Uno.
2. To study the integration between Arduino Uno and raspberry pi for sending the data to the cloud.
3. To compare between two different cloud database in term of efficiency on timing.

1.5 Scope

The scopes of the project are:

1. A conveyor sorting system with several sensors which is a capacitive proximity sensor, a fiber optic sensor, inductive proximity switch, magnetic sensor, and an infrared sensor is used as the model of the system.
2. An Omron CP1E Programmable Logic Controller (PLC) is used as the control unit in the system.
3. Ladder Diagram is used as the programming tool for the PLC.
4. Software used in this project is C++ programming and python.

1.6 Summary

In overall, 4 subtopics which are the motivation of the project, the problem statement of the project, objectives and scope of the research were discussed. The objective of this project is to design a conveyor sorter system with IoT that can determine the behavior of the conveyor sorter system by using online model checking. Besides, this project also aims to develop a communication system between the conveyor sorter system and cloud by using the Arduino Uno and raspberry pi for real-time data transmission. Last but not least, to analyze the behavior of the conveyor sorter system. The next chapter will discuss the journal that related to the project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, the background of the project will be explained thoroughly for a more robust understanding of the project. A review of some connected works are going mentioned to get some fundamental data by synthesizing their project to make this research successful.

2.2 Programmable Logic Controller

The phrase programmable logic controller (PLC) also known as a programmable controller. PLC can be defined as an industrial solid-state computer that controls inputs and outputs and also makes a logic-based decision for automation processes [3]. Actually, this PLC is not the latest technology but it provides the fundamental learning of control system. Since the late 1960s, PLCs were introduced by inventor Richard Morley to provide the same functions as a relay logic systems [3]. Basically, the PLC itself is using programmable memory to store the coding command and execute the instructions through the logic control, sequencing, timing, counting and arithmetic [4]. PLC hardware components consist of a central processing unit (CPU) which function as the brain of the PLC and memory which provide the permanent storage to the operating system for the CPU [3]. PLC contains two types of memory, one is a read-only memory (ROM) which stores the data permanently for the operating system, and another one is random access memory (RAM) which is to stores status information for input and output devices [3]. Besides this, to operate the PLC, a program or language is needed upload to the CPU, either using a console or computer. Figure 2.1 shows the basic requirement to operate a PLC.

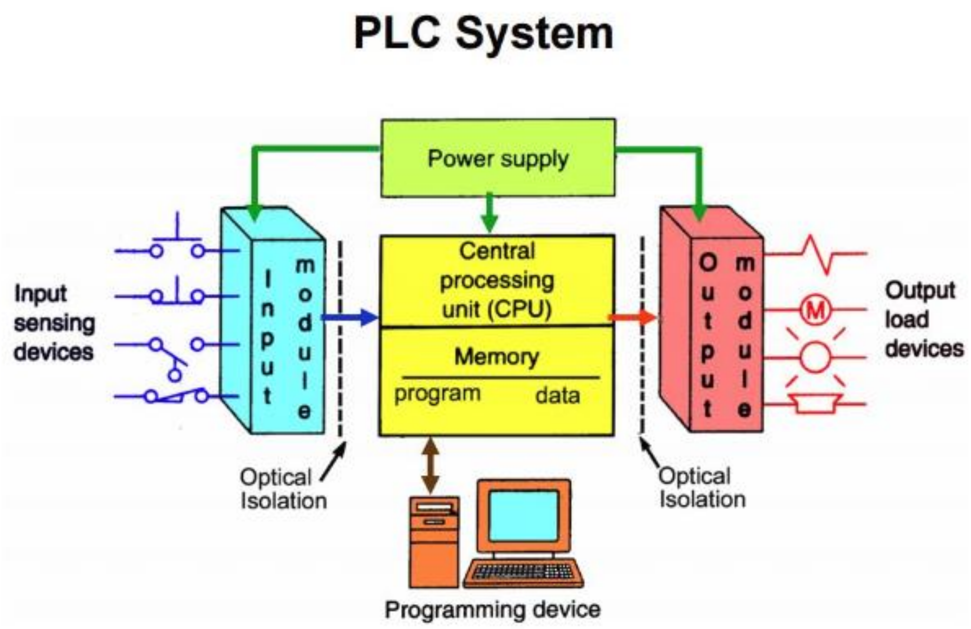


Figure 2.1 Basic Requirement to Operate a PLC [3]

In this project, PLC Omron CP1E is used to control the conveyor sorter systems with the programming method by using CX-Programmer with Ladder Diagram languages. For PLC Omron CP1E, it consists of a total 20 of input and output points (I/O), which is 12 inputs and 8 outputs control [4].

2.3 PLC Programming Language

The programming language, the elfevident is a formal language that use to create a standard form of commands that machine can be understood. In PLC, there are 4 popular languages for implementation. First, also the most popular use of language for programming PLCs which is Ladder Diagram (LD). The phrases ladder diagram (LD) also known as line diagrams or elementary diagrams. LD is used to show the function of the relay circuit and the association of the device but they don’t show the type of components used in the circuit and actual physical position [5]. The LD programs are transferred to the PLC, according to the program and connect the input and output device to I/O module and execute the system [6]. Figure 2.2 shows a simple ladder logic diagram.

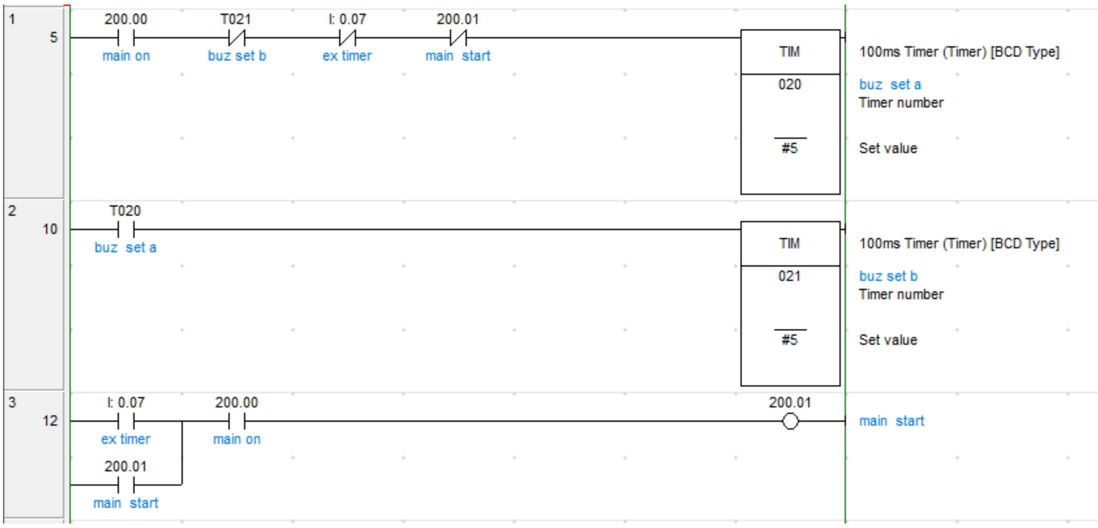


Figure 2.2 A Simple Ladder Logic Diagram

Second, function block diagram FBD is used for PLC programs express in term of graphical block [6]. In other words, FBD also known as a graphical language for depicting signal and data flows through inputs blocks. The logic gates, counter, timer or those function that defined by the user all can call a standard function in FBD [6]. FBD is also known as a set of an elementary block in which the input and output variable is connected to blocks by connection lines [7]. Figure2.3 shows a basic function block diagram.

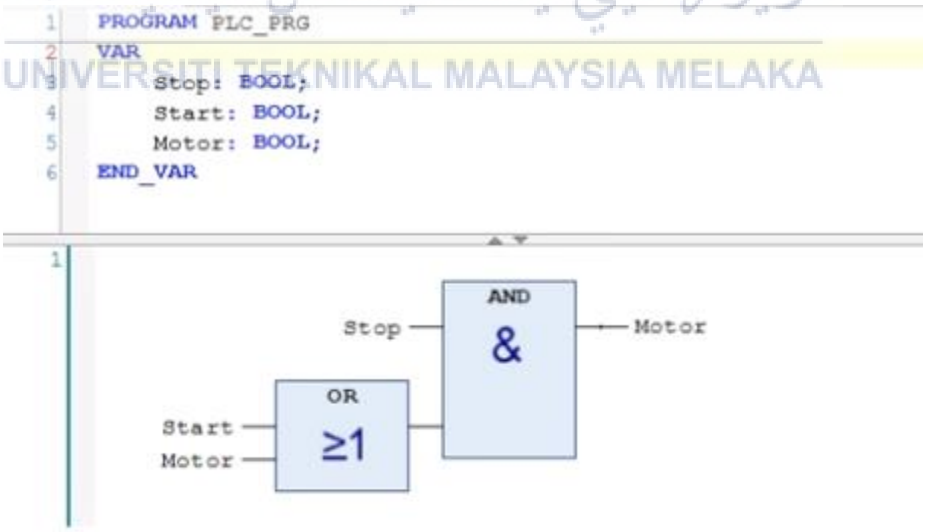
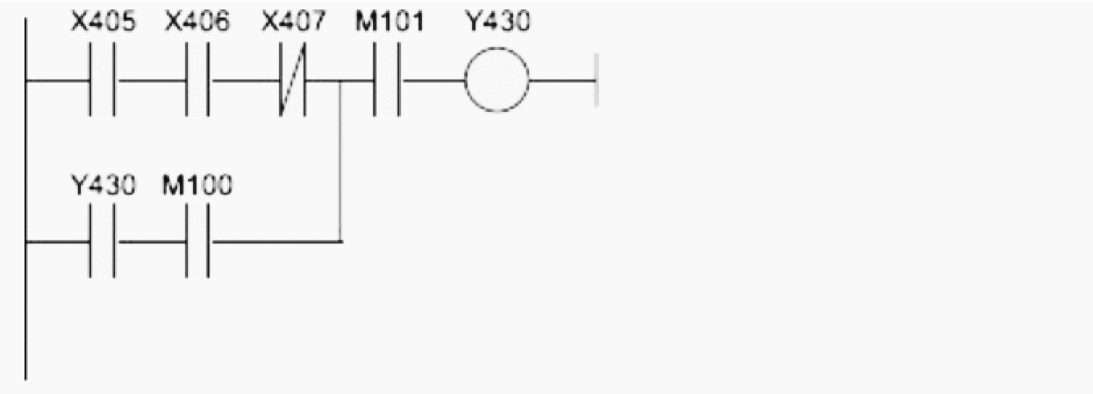


Figure 2.3 A Simple Function Block Diagram

Third, instruction list (IL) is one of the 4 languages supported by PLC. It is a low-level language and resembles assembly, and only available on a few brands of

PLC [6]. The variable and function in the circuit diagram are defined by the common elements, hence the advantage is different language can be used in the same program. Figure 2.4 shows the simple operation in ladder diagram form a PLC with an equivalent instruction list.



Line	Instruction	Comment
0	LD X405	LD starts rung or branch
1	AND X406	*Xnnn* are inputs (AND-ing is for series connection)
2	ANI X407	ANI is AND with not
3	LD Y430	LD starts a new branch leg
4	ANI M100	*Mnnn* are internal storage
5	ORB	OR the two branch legs (OR-ing for parallel connection)
6	AND M101	
7	OUT Y430	End of rung with output

Figure 2.4 Simple Operation in Ladder Diagram form a PLC with Equivalent Instruction List [6]

Fourth, logic functions also known as one of the languages supported by PLC. The logic function itself is a program that implementing a Boolean Algebra. Boolean Algebra deals with the theory which in the values of the variables are either true or false or in other words 1 and 0 respectively [8]. In logic function, 1 represent On or closed circuit, 0 represent Off or open circuit. Figure 2.5 shows a different logic function with its symbol, truth table, FBD, and LD.

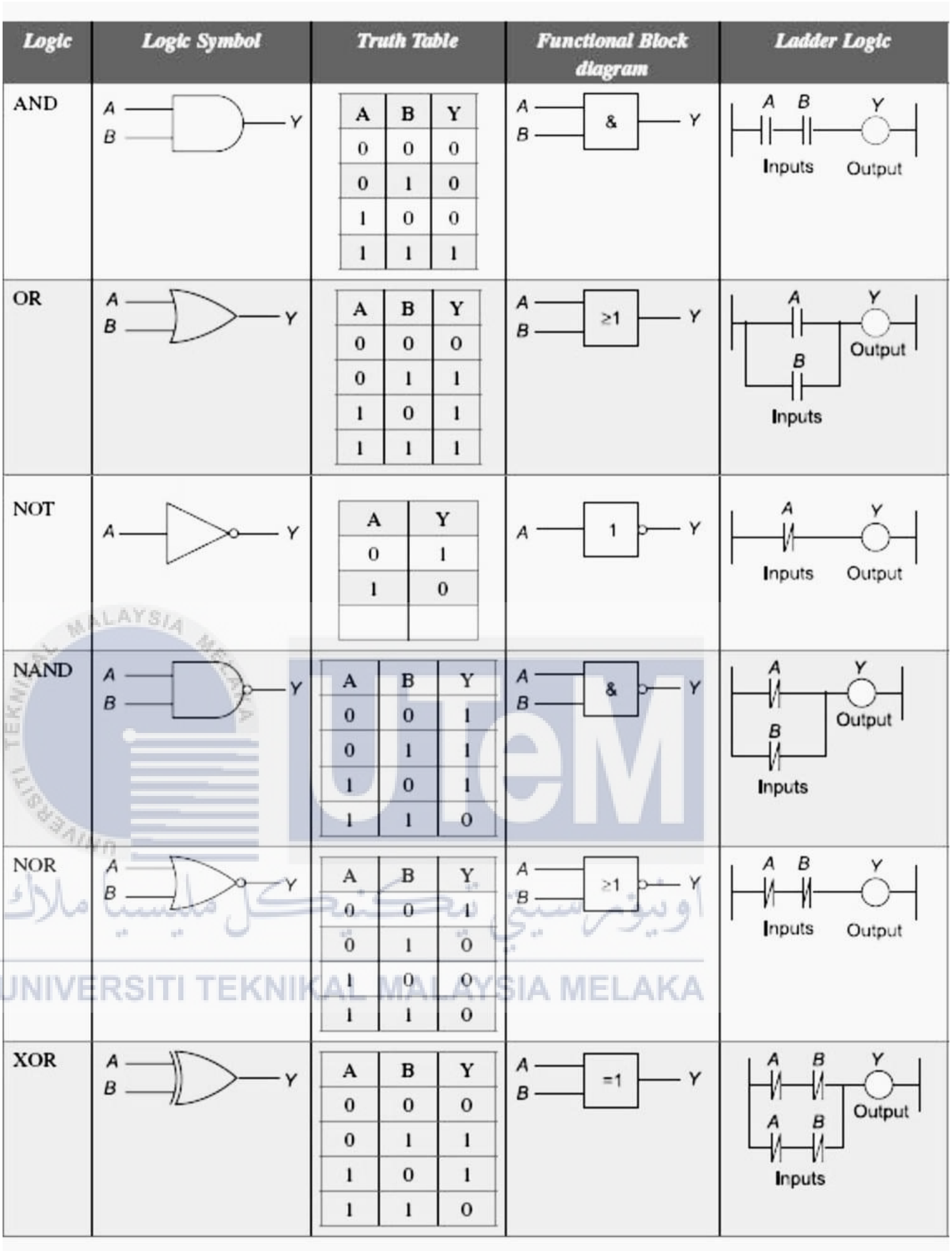


Figure 2.5 A Different Logic Function with its Symbol, Truth Table, FBD, and LD [6]

2.4 Conveyor Sorter System

The growing of the population make the huge demand in daily supplies, hence the sortation systems play an important role in any type of industry such as manufacturing industry to improve the efficiency of manufacturing processes and help increased the shipping accuracy while improving distribution efficiency [9]. The description of the sorter system is summarized in Table 2.1.

Table 2.1 The Description of the Sorter System

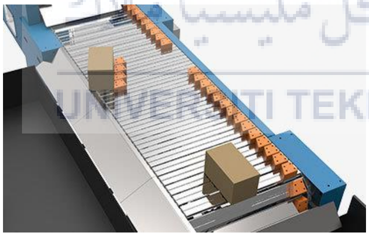
Type of sorter system	Description
Divider sorter system 	For this sorter system, the divider sorter moves the item on the conveyor, then the sorting process is using an electro-pneumatic or electric device to execute, is activated when the sensor detected and push it toward the indicated output [10].
Shoe sorter system 	This sorter system using the sliding shoe instead of normal push-pull sorting. The controller activate or deactivate the diverter will directly affect the sliding shoes toward one or more receiver and the diversion of the sliding shoes can result in the product away from the conveyor [10] [11].
Cross belt sorter system 	This kind of conveyor sorter system is widely used in the shipping industry. It is an apparatus that can convey the parcels on a plurality of transport units and providing the ability to adjust the parcel discharge to a selected receiving port [10] [11].

Table 2.2 The Description of the Sorter System (cont.)

Type of sorter system	Description
Tilt tray sorter system 	The tilt tray sorter system works similar to a cross-belt sorter system. It’s operated by a tray that will lean forward and drop the parcel when the tray reaches the correct unloading station position [10].
Vertical sorter system 	Vertical sorter system, the elfevident is diverting parcel into a different plane and usually top and bottom. It operates similar to a normal conveyor sorter system just it can divert into two different plane [10].

2.5 Internet of Things

IoT, also known as the Internet of Things, is a new technology paradigm visualized as a global network of machines and devices that interact with each other [12]. In the future technology, IoT is recognized as a key component of transfer legacy and next-generation devices to the IoT. They integrate networking protocols, facilitate manage storage and edge analysis of on information and facilitate information flow firmly between edge devices and the cloud. IHS technology is the world’s leading source for research, analysis and strategic guidance in the technology, media and telecommunication industries. HIS predict that the IoT market will grow up to 30.7 billion devices in 2020 from an installed base of 0.9 billion devices in 2009 [13] [14]. It shows the dramatics grow in the IoT field. Besides this, IoT also brings a huge benefit to human society in term of safety, comfort and efficiency to different professional field such as medical field [15] [16], smart city [17] [18], industrial field [19] and so on. The comparison of the IoT device in different criteria is summarized in Table 2.2.

Generally, IoT is a bridge that can connect between physical devices with the internet to collecting and sharing the data which enables them to communicate without the involvement of a human being and merge the digital and physical world.

In this project, the IoT gateway will be applied to the conveyor sorter system. The device that I used to connect the conveyor sorter and the cloud is raspberry pi with Arduino Uno.

In other words, to start up or developing an IoT device requires four stages of progress which are the assembly of the physical hardware. This is important because we need to fully understand the flow of the device works and this requires engineering skills. Next, is programming the device. This requires some fundamental knowledge of programming skills to read the data from the sensors that connected to the device and send the data to the server. After that, programming the server. This stage is important because all the data from the sensor are sent to the server and store it to the database and this requirement to use the server-side programming language. The last stage is to display the data to the device user. This step involving an application developed to collect the data from the sensor device to the user [20].

2.6 Table comparison of IoT device with the criteria

Table 2.3 The Comparison of IoT Device in Different Criteria

Criteria \ Journal	**Healthcare based on loT using Raspberry Pi** [15]	**A Smart Home Automation Technique with Raspberry Pi using IoT** [17]	**IoT based Monitoring and Control System for Home Automation** [18]	**Security Analysis on Consumer and Industrial IoT Devices** [19]	**An IoT based Patient Monitoring System Using Raspberry Pi** [16]
Type of central processing unit (CPU)	Raspberry Pi B+ model	Raspberry Pi B+ model	Raspberry Pi B+ model	SmartCare PCB	Raspberry Pi B+ model
CPU operating system	Raspbian	Raspbian	Debian	Linux	Raspbian
Type of communication module	GSM module	GSM module	WIFI module	ZigBee protocol	Microcontroller AT Mega 32
Type of programming language	Python Language	Python Language	Python Language	Python Language	Python Language
Sensor used	Electrocardiogram	Simple CV module, camera	IR sensor, light dependent resistor, camera	Haire smart care (design to control and read the sensor)	Heartbeat sensor, temperature sensor, respiration sensor, accelerometer sensor

In this section, comparison of IoT device with the criteria from previous journal are summarized. According to Table 2.2, all journal consists of central processing unit (CPU) to monitor the IoT gateway application. For journal [15]–[18] use Raspberry Pi B+ model while [19] uses their own CPU which is SmartCare PCB for the research. For the CPU operating system, [15]–[17] applied Raspbian as the operating system. While [18] and [19] are using Debian and Linux respectively.

Furthermore, all of the paper shown in Table 2.2 consist of communication module for their own research used. For [15], [17], GSM module are used for their communication system. While [18] using WIFI module, [19] using ZigBee protocol and [16] using microcontroller AT Mega 32 as their own communication system.

Beside this, for the programming language use to program the system to apply the IoT gateway application, it shown [15]–[19] are using python language. After this the sensor used for [15] is electrocardiogram, for [17] is simple CV module and camera, for [18] is using IR sensor, light dependent resistor and camera, for [19] is using their own technology which is Haire SmartCare, and for [16] is using heartbeat sensor, temperature sensor, respiration sensor and accelerometer sensor.

In overall, it shows that most of the research are using raspberry pi with python language to achieve the IoT gateway application.

2.7 Summary

In overall, this chapter presents all the theoretical background related to this project. For the PLC module, CP1E and LD programming will be used in this research project. The type of conveyor sorter system is divider sorter system and the raspberry pi will use as the central unit processing link with the internet. In the next chapter, the setup and experiment will be discussed in order to achieve the objective of the project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, all the method that will achieve the objective of this research project will be discussed including the method used and experiment. Thus this chapter will divide into hardware understanding, program development, IoT getaway setup and experiment setup.

3.2 Project Overview

In figure 3.1 shows the flow chart for the overall project.

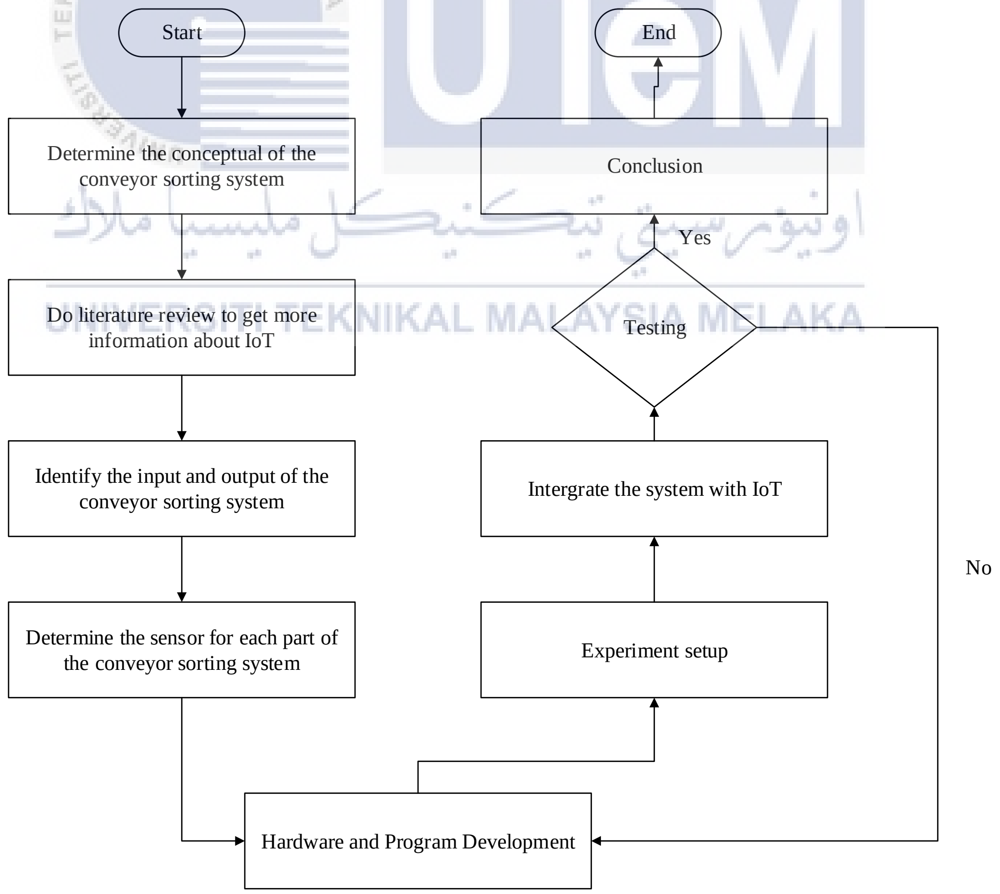


Figure 3.1 The Flow Chart for the Overall Project

3.3 Conveyor Sorter System with IoT Design

The hardware and software selection is done after the previously studied research journal that discusses in chapter 2. Figure 3.2 shows the conveyor sorter system with IoT.

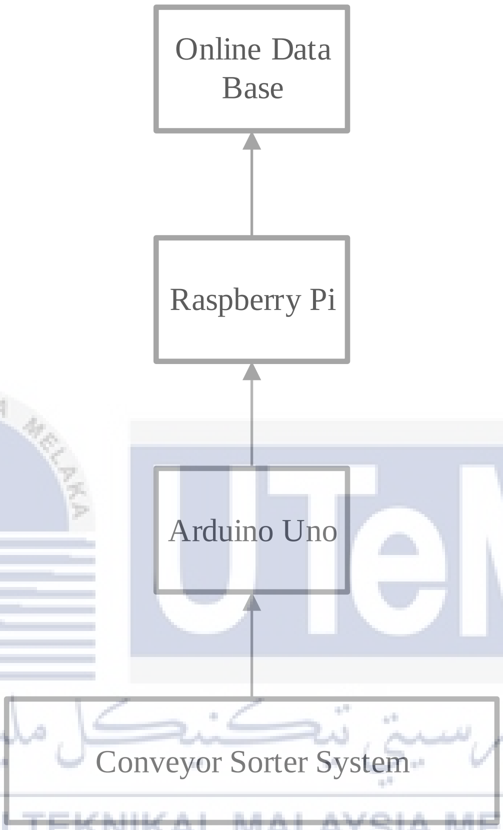


Figure 3.2 The Conveyor Sorter System with IoT

According to Figure 3.2, the devices that choose to communicate between conveyor sorter systems with internet are Arduino Uno and raspberry pi. Arduino Uno is used to collecting the sensor data from the conveyor sorter system. Then, Arduino Uno will connect to the raspberry pi to transfer the sensor data from Arduino to cloud. For the cloud server, thingspeak and firebase are used as the server to receive the sensor data.

3.4 Hardware connection and development

In this project, the hardware connection and development for conveyor sorter systems with IoT involve three parts which are controller, communication and sensors system.

3.4.1 Controller

The controller used in this project is Arduino Uno which to conduct in experiment.

3.4.1.1 Arduino Uno

Generally, Arduino is an open source electronics platform that provide easy-to-use hard ware and software. Basically Arduino Uno are used to read the input which is the signal given by other device such as sensors or button, and turn the signal to an output to control or program the other physical device. To do so, Arduino programming language is needed to monitor the Arduino and the Arduino software (IDE) is the platform to write the coding.

There are few reason that I choose Arduino as the microcontroller for this project. First, Arduino boards are relatively inexpensive compare to others. Second, Arduino Uno is available to cross platform. It is very convenience because is easy for my project when switching the operating system from Window to Linux. Last but not least, the Arduino software (IDE) is suitable for beginners to learn. Figure 3.3 shows the Arduino Uno model use in this project.

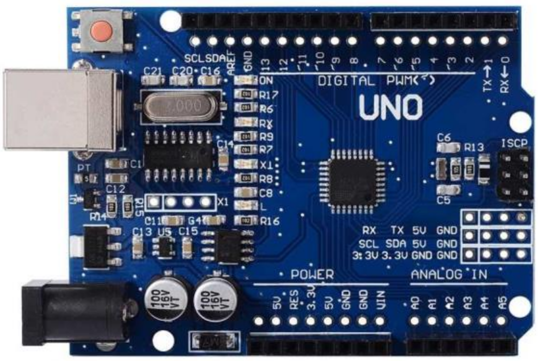


Figure 3.3 Arduino Uno

3.4.2 Communication

The communication used in this project is Raspberry Pi which used to conduct in experiment.

3.4.2.1 Raspberry Pi

Basically, Raspberry Pi is a credit card sized computer and lunched in 2012. Activate the Raspberry Pi just need to plugs into a computer monitor or TV, and used a standard mouse and keyboard then a budget desktop is ready to used. The latest model which is Raspberry Pi 3 Model B Vi 2 consists of built in WIFI module. Bluetooth will built in the board and the powered by a Quad Core Broadcom BCM2837 64bit ARMv8 processor. The most important things is the operating system for Raspberry Pi is Linux and also provides a set of GPIO (general purpose input/output) pins to control electronic components for physical computing and explore the Internet of Things (IoT). Figure 3.4 shows the Raspberry Pi model used in the project.

Figure 3.4 Raspberry Pi model

3.4.3 Sensors System

The sensors used in the conveyor sorter system are capacitive proximity sensor, fiber optic sensor, magnetic sensor, and IR sensor.

3.4.3.1 Capacitive Proximity Sensor

Capacitive proximity sensor LJC18A3-B-Z/BX 18mm diameter, 10mm detective distance DC6-36V 3 wires, no sensor switch. This sensor used to detect the metal and non-metallic such as plastic, glass, water, oil and so on. Detection distance was detected with the body of conductivity, dielectric constant, water absorption, the volume varies, the maximum for the detection of a grounded metal distance. Figure 3.5 shows the capacitive proximity sensor that used in conveyor sorter system.

Figure 3.5 Capacitive Proximity Sensor

3.4.3.2 Fiber Optic Sensor

Amplifier, fiber optic, dual-display, auto tune, NPN output, 12-24 VDC. The fiber optic sensor has an optical fiber connected to a light source to allow for detection in tight spaces or where a small profile is beneficial. Figure 3.5 shows the fiber optic sensor that used in conveyor sorter system.

Figure 3.6 Fiber Optic Sensor

3.4.3.3 Magnectic Sensor

The magnetic sensor used in this project call reed switch also known as the simplest magnetic field sensor. This kind of sensor require no standly power and can activate with both AC and DC electrical load. Hence, normally the double acting cylinder will used this type of sensor as a switch. Figure 3.6 shows the magnetic sensor used in conveyer sorter system.

Figure 3.7 Magnetic Sensor

3.4.3.4 IR Sensor

The implementations of modulated IR signal immune the sensor to the interferences caused by the normal light if a light bulb or the sun light. The sensing distance can be adjusted manually. Figure 3.7 shows the IR sensor used in conveyor sorter system.

Figure 3.8 IR Sensor

3.5 Program Development

In this project, total two programming platform are involve when develop the conveyor sorter system with IoT gateway application. The two programming platform are Aduino IDE and Python.

3.5.1 Arduino IDE

Arduino IDE so call Arduino integrated development environment is an open-source software to write code and upload to the board. It can runs on many operation platform such as Windows, Mac OS X and Linux. Arduino IDE supports language C and C++. It also provide planty of software library to allow user to create their own personal project. The Figure 3.8 shows the Arduino IDE programming software.

Figure 3.9 Arduino IDE Programming Software

3.5.2 Python

In technical term, Python is a high-level programming language with integrated dynamic sementics primarily for web and application development. It is a graet language for beginners because of its realiability and other structural elemants designed to make it easy to understand. It is an interpreted language which means that the program written in Python no need to complied in order to run, making it easy to move between multible platforms. Hence in this project, Python is used to transmit the data from Arduino to cloud data base. Figure 3.9 shows the Python software.

Figure 3.10 Python Software

3.6 Experiment Setup

In this path, all the experiment are listed.

3.6.1 Experiment 1: Understand the operation of the conveyor sorter system

The Conveyor Sorter with PLC Trainer CS-101 is an ideal training system makes it possible to get to know and compare different type of sensor used in systems. The sorting process sorts the work pieces onto the sorting compartment which place at the end of the conveyor belt. The IR sensor will detected the work pieces when the work pieces pass through the sensor through the conveyor belt. Few sensor are used to characterize the work piece features which shows in Figure 3.11. The apparatus is self-sufficient and is mounted on a wheeled frame. Its main components are conveyor belt, sensors (magnetic, inductive, capacitive, IR, Fiber-optic), double acting cylinder 5/2 wave control valve, Omron CP1E PLC, counter and timer. Figure 3.10 shows the conveyor sorter system with PLC Trainer CS-101.

The procedure to start-up the conveyor sorter system are proceed as follow:

1. The bench is connected to the power mains.
2. The fuse is checked before switch on the main switch.
3. All the sensor wiring, pressure regulating valve are in order.
4. All the sensor are mounted on the conveyor frame.
5. All the cylinder are mounted on the conveyor frame.
6. The connector is connected into pressure regulator adjustable the pressure area 4~6 bar.
7. The ladder diagram is run and simulated by using the CX-Programmer.
8. The Program is connected to the conveyor sorter with PLC trainer CS-101.
9. The flow of the conveyor is recorded and the flowchart of the conveyor sorter is plot as shown in Figure 3.12.
10. The input and output of the conveyor sorter system is list down in table as shown in chapter 4.

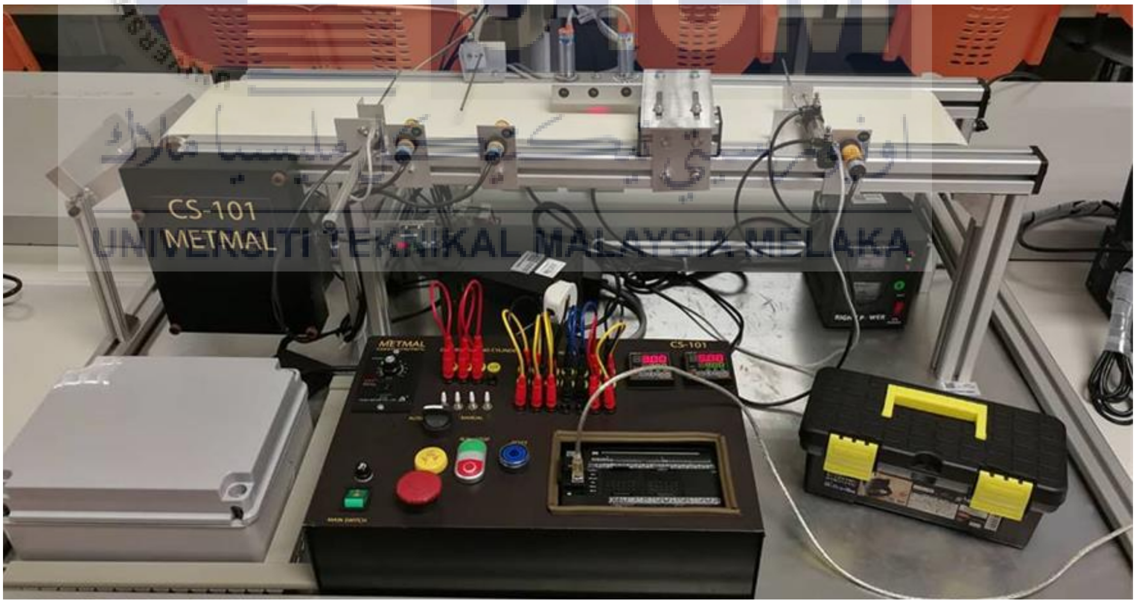


Figure 3.11 Conveyor Sorter System with PLC Trainer CS-101

Figure 3.12 Work Piece Features

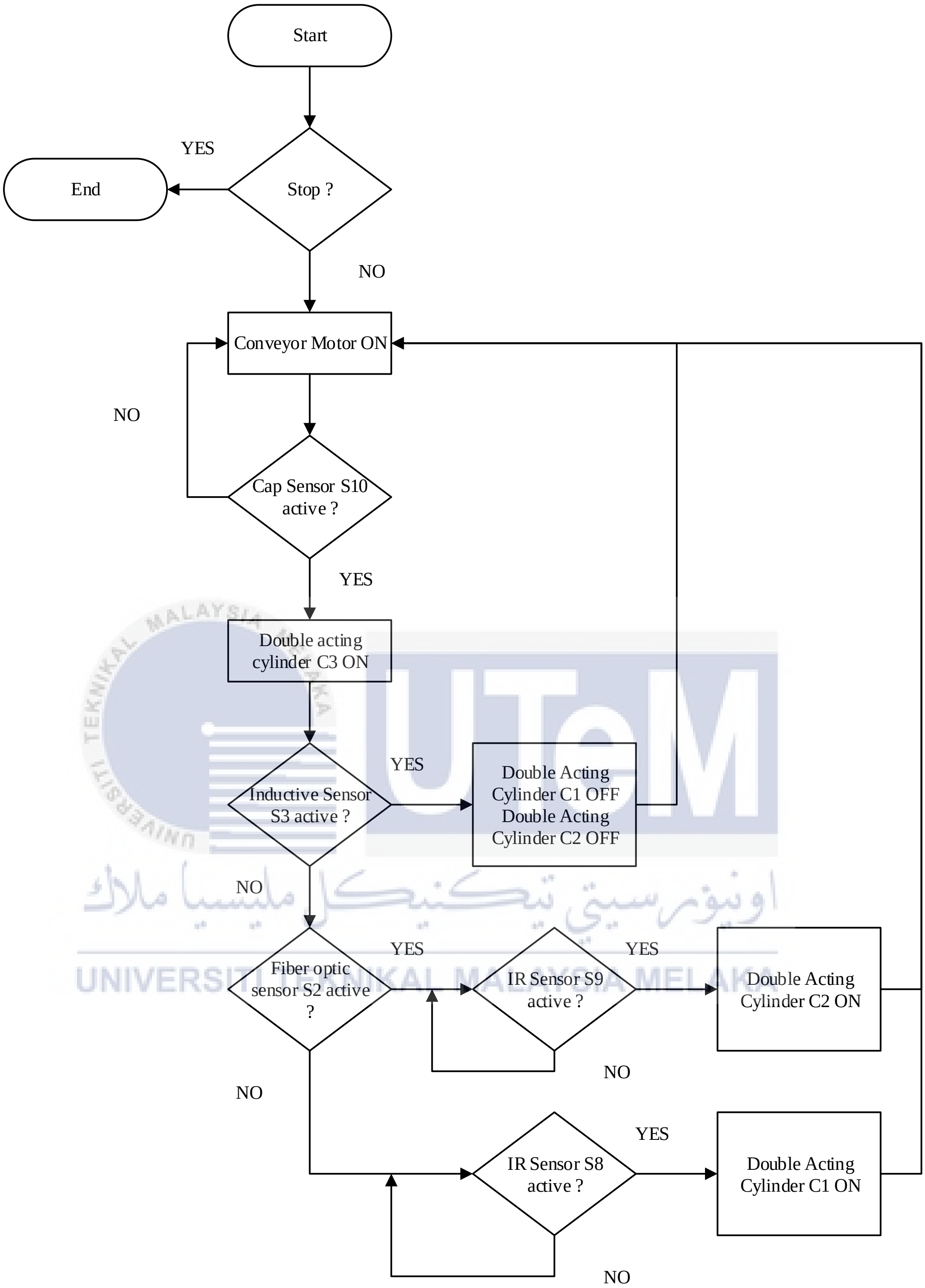


Figure 3.13 Flowchart for the Conveyor Sorter System

The flowchart for the conveyor sorter system as shown in Figure 3.12. First, when the conveyor sorter system is activated, the conveyor motor is turned on. After that, the workpiece on the conveyor belt is moved, until the cap sensor S10 active. When the cap sensor active, the double acting cylinder C3 is on and allow the workpiece to enter the sorting path. In the sorting path, if the inductive sensor S3 is active, the rest of the double acting cylinder which is C1 and C2 is off. Else, if the fiber optic sensor S2 is active, the double acting cylinder C2 will active after the IR sensor S9 is active. If both conditions which is an inductive sensor and fiber optic sensor is passive, then the double acting cylinder C1 will active after the IR sensor S8 is active. Then the process will loop again.

3.6.2 Experiment 2: Connecting the physical device with Arduino Uno

This experiment is to obtain the status of inputs and outputs (I/O) condition from conveyor sorter system by using Arduino Uno. To achieve this, few electronic components are used to connect the physical device with the Arduino Uno. The program design for Arduino IDE needs to test after the wire connection is connected. The idea of this experiment is let the Arduino Uno to receive the signal from PLC Omron CP1E and show the status of inputs and outputs condition for the sensors used in the conveyor sorter system.

3.6.3 Experiment 3: Connect the Arduino with Raspberry Pi

This experiment is to study the integration between Arduino Uno and Raspberry Pi. The purpose of conduct this experiment is to transfer the data that collected from Arduino Uno to Raspberry Pi. The program design for python is needed to test after the Arduino Uno is successfully received the signal from the conveyor sorter system. After the Raspberry Pi successfully receive the data from Arduino Uno, then can proceed to the next experiment.

3.6.4 Experiment 4: Transfer the data to cloud server

This experiment is to analyze the data transfer from Raspberry Pi to the cloud server. The objective to conduct this experiment is to analyze the data transfer from Raspberry is successful and reliable. There are few servers can be used for real-time data transfer. Hence, this experiment also will test the suitability of the cloud server with this project.

3.7 Summary

As a conclusion, this chapter shows a detail plan to conduct the experiment to achieve the objective of the project. The functionality of the conveyor sorter and the sensor will be tested before starting the experiment.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

In this chapter, all the experimental result are presented.

4.2 The flow of the conveyor sorter system

According to experiment 1 the flowchart as shown in Figure 3.11, the conveyor sorting system after activates, the conveyor belt is starting to move. After this, the sorting material is put on the center of the conveyor belt as shown in Figure 4.1.

Figure 4.1 Material before Enter the Sorting Path

The cap sensor (s10) is activated when the material passes through the sensor and the double acting cylinder is retract (open) to allow the material to enter the sorting process. The sorting process only allows one material to pass through, hence the double acting cylinder is extended (close) to ensure only one material can go in the sorting process.

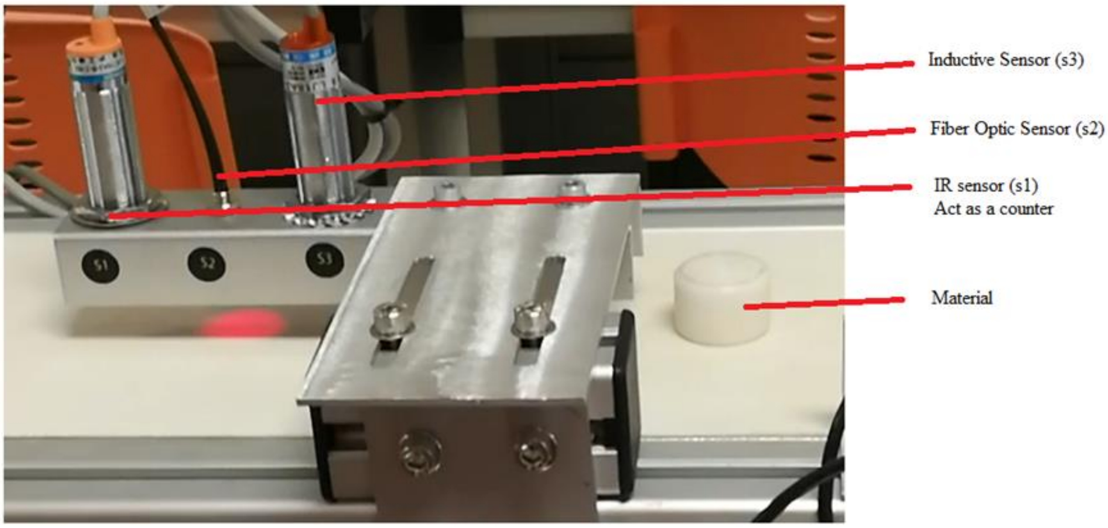


Figure 4.2 Sorting Path of the Conveyor Sorter System

Figure 4.2, the sorting process of the conveyor sorter system which contains two sensors to classify the material which is the inductive sensor (to identify the metal sample) and fiber optic sensor (to identify the black and white sample). The IR sensor (s1) acts as the counter in the conveyor sorter system.

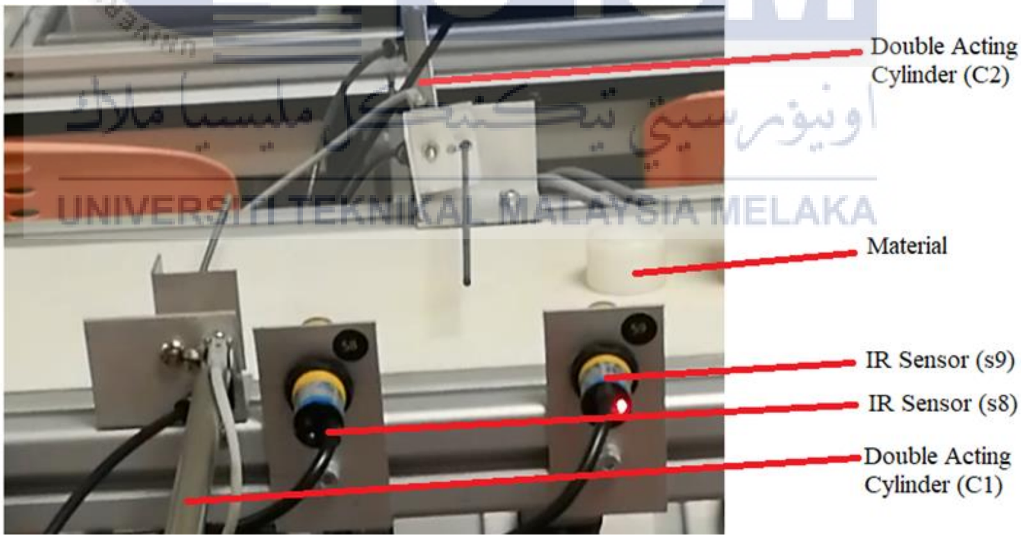


Figure 4.3 Distribution Process of the Conveyor Sorter System

Figure 4.3 shows the distribution processes of the conveyor sorter system. The outcomes of the sorting system are as shown in Table 4.1.

Table 4.1 Outcomes for the distribution processes

Material	The process of the cylinder
Aluminum	Both C1 and C2 remain passive
White	C2 activate
Black	C1 activate

Figure 4.4 Sorting Compartment of the Conveyor Sorter System

Figure 11 shows the final process for the conveyor sorter system which is distributing the material to its own compartment. The conveyor belt will keep running until the cap sensor (s10) detect the material, the process will run again to sort the sample. The ladder diagram can refers to Appendix A.

In conclusion, this experiment is important for me to know the flow of the conveyor sorter system and the input/output condition to the purpose of connecting the physical device to Arduino Uno which is stated in the first objective.

4.3 Connecting the physical device with Arduino Uno

The aim of this experiment is to connect all the sensors from PLC Omron CP1E to Arduino Uno to achieve the first objective. The corresponding sensors in the conveyor sorter system can get from PLC Omron CP1E that show in Figure 4.5.

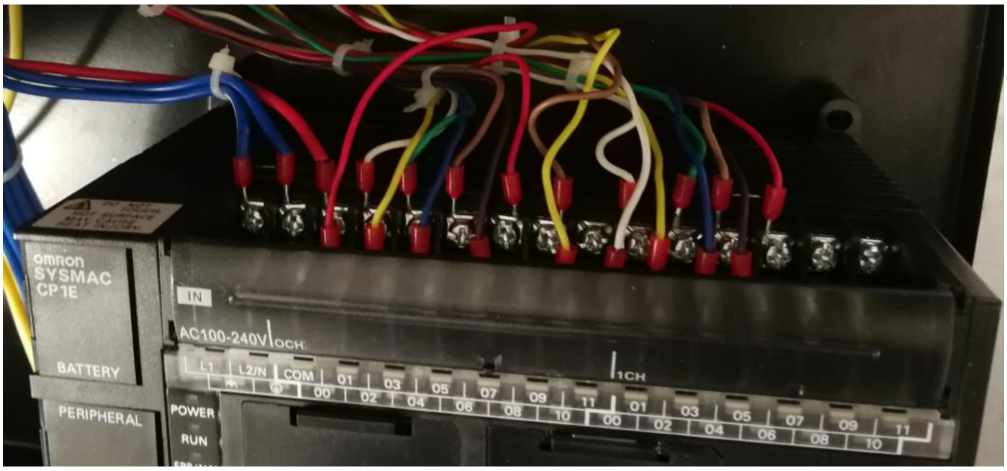


Figure 4.5 Input Signal from Conveyor Sorter System

Figure 4.5 shows all the input connection from the conveyor sorter system. Hence, the signal of the sensors can get from here then connect them to Arduino Uno. To identify all the sensors used in the conveyor sorter system which are a capacitive proximity sensor, fiber optic sensor, magnetic sensor, and IR sensor is to activate the sensor manually and observe the change in PLC Omron CP1E. After this, connect the activate place with a wire which shown in Figure 4.6.

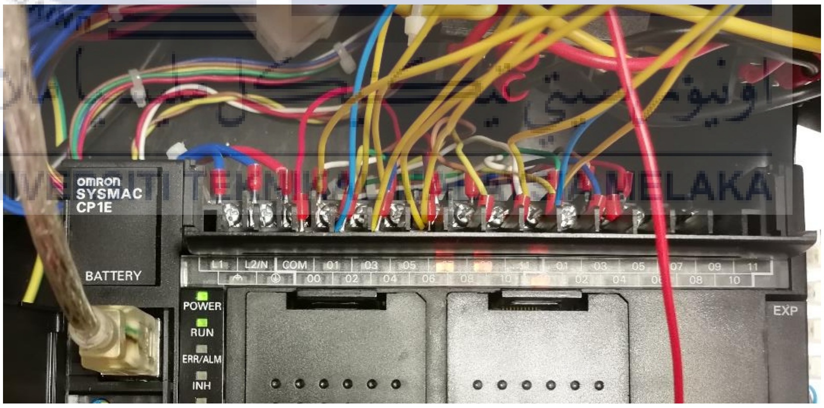


Figure 4.6 Input Signal with External Wire Connection

After that, arrange the sequence of the wire then connect it to Arduino Uno with a connector which as shows as Figure 4.7. This arrangement sequence can simplify future work and troubleshooting for other experiments.

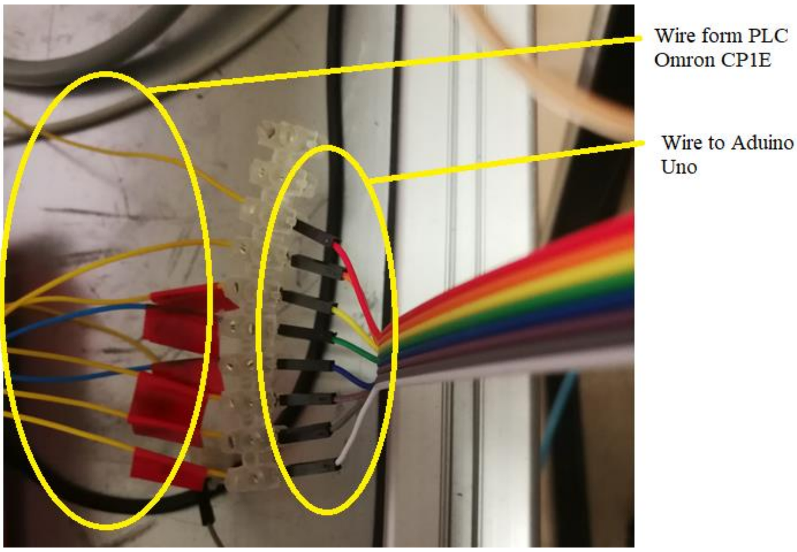


Figure 4.7 Input Signal with External Wire Connection

The next phase is to connect the jumper to Arduino Uno to collect the analog signal from sensors. Unfortunately, this experiment is failed. This is because the result shows from Arduino Uno do not have any changes shown in Figure 4.8.

Figure 4.8 Result Shows in Arduino Uno

In the end, this is because not all the sensors can provide analog reading and lead to this problem happened. Hence, the experiment is changed to collect the digital signal from the sensors. The 8 relay module is added to the connection before connecting to Arduino Uno and shows in Figure 4.9.

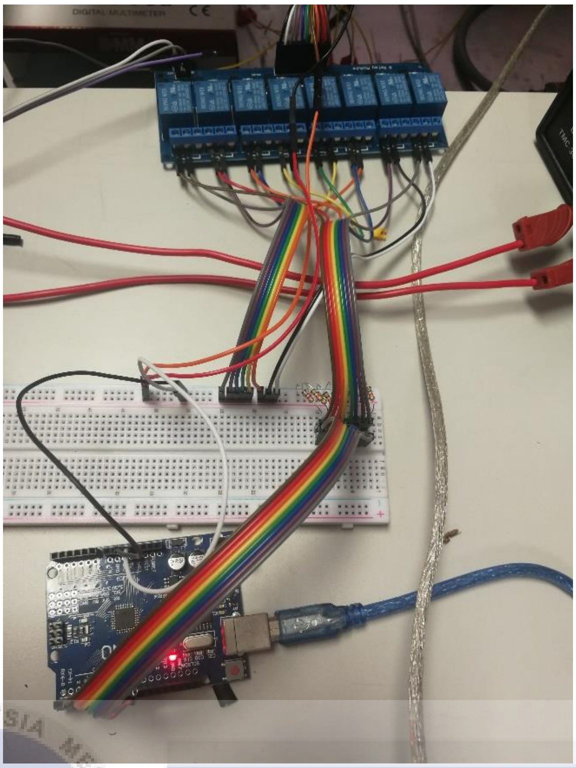


Figure 4.9 Connection between Physical Device and Arduino Uno with 8 Relay Module

After this, all connection and device are in good condition and ready to receive the signal from the sensor. Finally, the digital signal is successfully received and the result shows in Figure 4.10, 4.11 and 4.12. The result showing the sensing information when sorting the three different type of workpiece feature. Sensor 1, 4, and 5 represent IR sensor, used to activate the Actuator 1, 2 and 3 respectively. Sensor 2 represent capacitive proximity sensor and sensor 3 represent the fiber optic sensor.

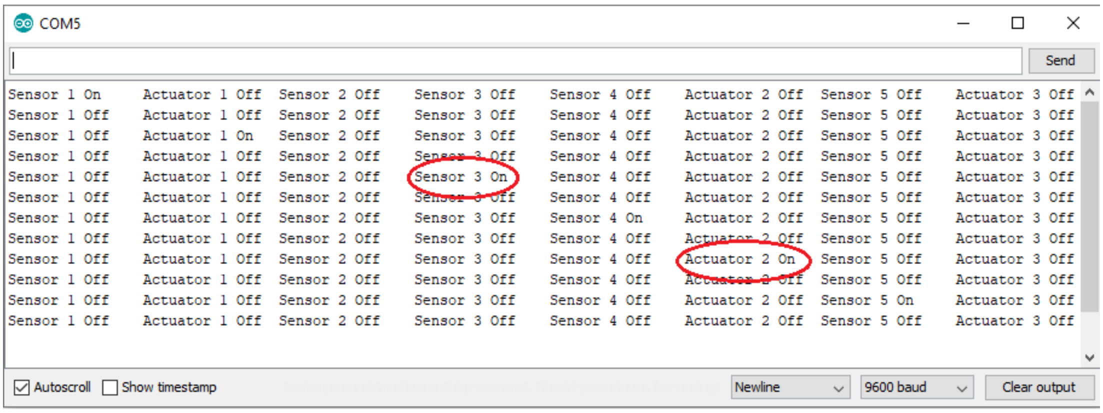


Figure 4.10 Result of Sorting Black Colour Work Piece Feature

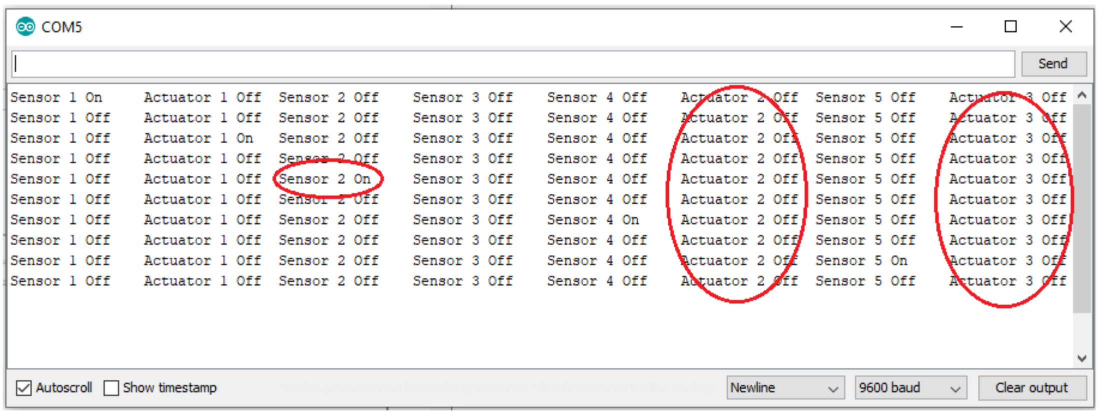


Figure 4.11 Result of Sorting Metal Type Work Piece Feature

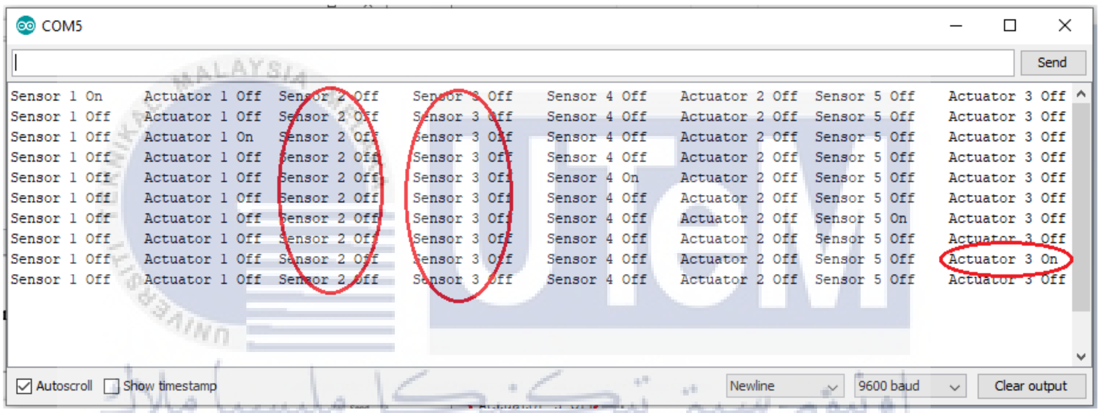


Figure 4.12 Result of Sorting White Colour Work Piece Feature

Figure 4.10, 4.11 and 4.12 shows the data collected by Arduino Uno. From the result above, the important part is circled by the red circle. Figure 4.10 shows when the sensor 3 is activated, then the material is moved to the left side by actuator 2, which mean this type of data shows black colour workpiece feature. For Figure 4.11, it shows when the sensor 2 is activated, the material stays on the middle line which means this is the metal type workpiece feature. Figure 4.12 shows when sensor 2 and 3 remain passive, then the workpiece is moved to the right side by actuator 3, which mean this data shows white colour workpiece feature. Coding refers to Appendix B.

4.4 Study the Integration between Arduino Uno and Raspberry Pi

The aim of this experiment is to connect the Arduino Uno with Raspberry Pi. The purpose of this experiment is to send the data from Arduino Uno to the cloud server, and Raspberry Pi is the platform that transfers the data from Arduino Uno represent the physical device to the cloud server represent IoT gateway.

The Debian is installed to the Raspberry Pi as the operating system. After that run Raspberry Pi as a normal budget desktop to operate the Arduino Uno. First, connect the Arduino Uno to Raspberry Pi with the type B cable, used Arduino IDE to run the Arduino Uno. To integrate between Arduino and Raspberry Pi, Python programming language is used. Besides this, the cloud server used for this experiment is ThingSpeak. After that, the Arduino coding and python program are designed which shows in Appendix C and Appendix D.

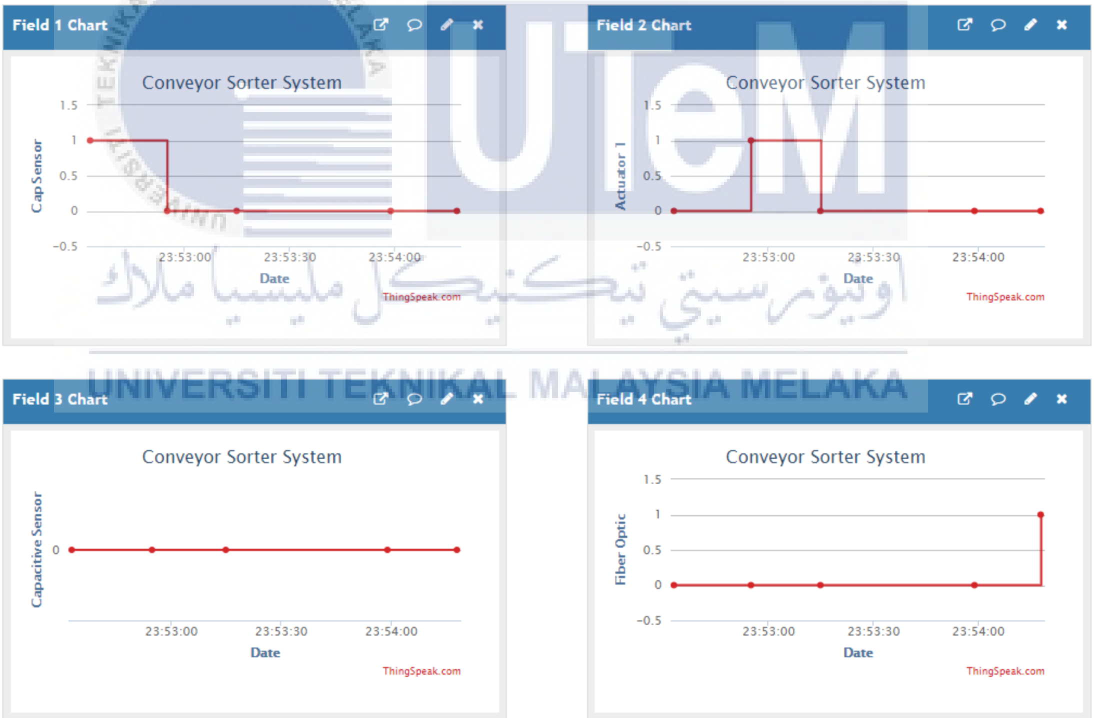


Figure 4.13 Result for IoT Gateway with ThinSpeak Server

Figure 4.13 shows the four sensing result that present on ThingSpeak cloud server. The data from the result shows only 1 and 0 which represent sensor on and off respectively. For real-time situation, ThinkSpeak cloud server couldn’t get the proper result from the conveyor sorter system due to the message update interval limit remains limited at 15 seconds. Hence the result shows almost all the time in 0 (off) condition.

In overall, the integration between Arduino Uno and raspberry pi for sending the data to the cloud is successful achieve. The data is successfully transferred and presented on the ThinsSpeak cloud server but because of the limitation of the server, this server is not suitable for instant data transfer. Hence in the next experiment will show the new method of IoT gateway application for data transfer.

4.5 Transfer the Data to Cloud Server

The aim of this experiment is to transfer the data from a physical device to cloud server by using Raspberry Pi without using Arduino Uno, the coding refer to appendix E. The connection between conveyor sorter system with Raspberry Pi shows in Figure 4.14.

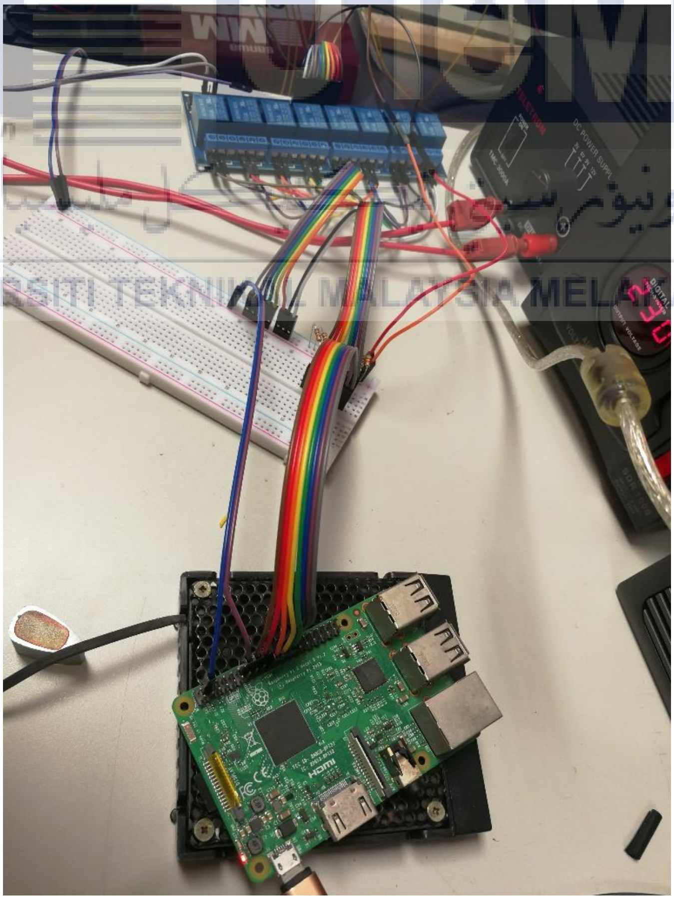


Figure 4.14 Connection between Conveyor Sorter System with Raspberry Pi

After this, all the general purpose input/output (GPIO) of the Raspberry Pi are tested and well function. The cloud server that used to transfer the data in Firebase. The Firebase cloud server able to receive the data from Raspberry Pi and presented the data in real time. Figure 4.15 shows the interface of the Firebase real-time database.

Figure 4.15Firebase for Real-Time Database Interface

Figure 4.15 shows the initial condition of the Firebase real-time database. When the Raspberry Pi receives the signal from the conveyor sorter system, the real-time database will update the on/off condition and change it. Figure 4.16 shows an example when the sensor 1 is triggered and the change of the Firebase.

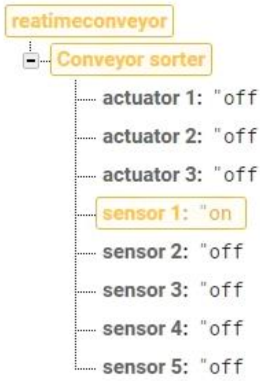


Figure 4.16Change of Firebase when the Signal is Receive

Besides that, the real-time data information is recorded by using the python program and the type of workpiece feature can be analyzed base on the data that save in the txt format. The sensor information in actual time for black colour workpiece feature shown in Table 4.2 with the line graph which shows in Figure 4.17.

Table 4.2 Sensor Information in Actual Time for Black Colour Work Piece Feature

Time	Sensor 1	Actuator 1	Sensor 2	Sensor 3	Sensor 4	Actuator 2	Sensor 5	Actuator 3
32:26.8	1	2	4	6	8	10	12	14
32:27.7	1	2	4	6	8	10	12	14
32:28.7	0	2	4	6	8	10	12	14
32:36.5	0	3	4	6	8	10	12	14
32:37.5	0	3	4	6	8	10	12	14
32:38.4	0	3	4	6	8	10	12	14
32:39.3	0	2	4	6	8	10	12	14
32:46.6	0	2	4	6	8	10	12	14
32:53.8	0	2	4	7	8	10	12	14
32:54.8	0	2	4	7	8	10	12	14
32:55.7	0	2	4	7	8	10	12	14
32:56.6	0	2	4	6	8	10	12	14
33:03.9	0	2	4	6	9	10	12	14
33:04.8	0	2	4	6	9	10	12	14
33:05.7	0	2	4	6	9	10	12	14
33:06.6	0	2	4	6	9	10	12	14
33:07.5	0	2	4	6	8	10	12	14
33:14.7	0	2	4	6	8	11	12	14
33:15.7	0	2	4	6	8	11	12	14
33:16.6	0	2	4	6	8	11	12	14
33:17.5	0	2	4	6	8	11	12	14
33:18.4	0	2	4	6	8	10	12	14
33:25.9	0	2	4	6	8	10	13	14

33:26.8	0	2	4	6	8	10	13	14
33:27.7	0	2	4	6	8	10	13	14
33:28.6	0	2	4	6	8	10	12	14

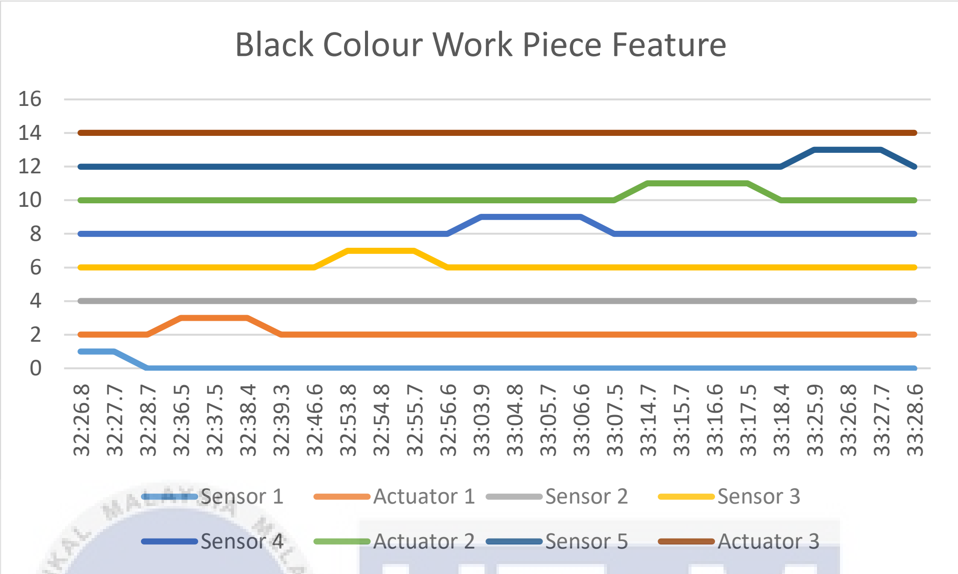


Figure 4.17I/O Analysis of Black Colour Work Piece Feature

Figure 4.17 shows the data analysis from the Firebase server. The analysis of the sorting material can be analyzed from the line graft. In this black colour workpiece feature, the important data is sensor 3 which represent the fiber optic sensor and actuator 2 which is placed the material to the left side sorting compartment.

Table 4.3 Sensor Information in Actual Time for Metal Type Work Piece Feature

Time	Sensor 1	Actuator 1	Sensor 2	Sensor 3	Sensor 4	Actuator 2	Sensor 5	Actuator 3
47:10.2	1	2	4	6	8	10	12	14
47:11.2	1	2	4	6	8	10	12	14
47:12.1	0	2	4	6	8	10	12	14
47:19.5	0	3	4	6	8	10	12	14
47:20.4	0	3	4	6	8	10	12	14
47:21.3	0	3	4	6	8	10	12	14
47:22.2	0	2	4	6	8	10	12	14
47:29.5	0	2	4	6	8	10	12	14
47:36.9	0	2	5	6	8	10	12	14
47:37.8	0	2	5	6	8	10	12	14

47:38.7	0	2	5	6	8	10	12	14
47:39.6	0	2	4	6	8	10	12	14
47:46.9	0	2	4	6	8	10	12	14
47:54.3	0	2	4	6	9	10	12	14
47:55.2	0	2	4	6	9	10	12	14
47:56.1	0	2	4	6	9	10	12	14
47:57.0	0	2	4	6	8	10	12	14
48:06.2	0	2	4	6	8	10	13	14
48:07.1	0	2	4	6	8	10	13	14
48:08.0	0	2	4	6	8	10	13	14
48:08.9	0	2	4	6	8	10	13	14
48:09.8	0	2	4	6	8	10	13	14
48:10.7	0	2	4	6	8	10	12	14

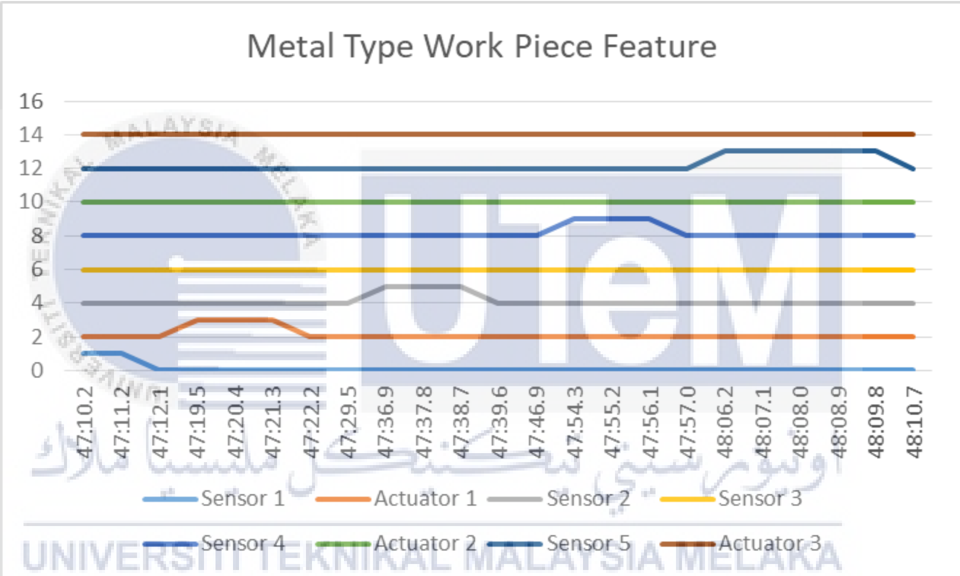


Figure 4.18I/O Analysis of Metal Type Work Piece Feature

Figure 4.18 shows the second data analysis from the Firebase server. The type of sorting material can be analyzed through the line graph. In this metal type workpiece feature, the important data is sensor 2 represent capacitive proximity sensor and both actuator 2 and 3 are off which place the material to the middle of the sorting compartment.

Table 4.4 Sensor Information in Actual Time for White Colour Work Piece Feature

Time	Sensor 1	Actuator 1	Sensor 2	Sensor 3	Sensor 4	Actuator 2	Sensor 5	Actuator 3

00:36.6	1	2	4	6	8	10	12	14
00:37.6	1	2	4	6	8	10	12	14
00:38.5	0	2	4	6	8	10	12	14
00:45.8	0	3	4	6	8	10	12	14
00:46.7	0	3	4	6	8	10	12	14
00:47.6	0	2	4	6	8	10	12	14
00:54.9	0	2	4	6	8	10	12	14
01:03.3	0	2	4	6	8	10	12	14
01:10.6	0	2	4	6	9	10	12	14
01:11.5	0	2	4	6	9	10	12	14
01:12.4	0	2	4	6	9	10	12	14
01:13.3	0	2	4	6	8	10	12	14
01:20.7	0	2	4	6	8	10	12	14
01:28.0	0	2	4	6	8	10	13	14
01:28.9	0	2	4	6	8	10	13	14
01:29.8	0	2	4	6	8	10	12	14
01:37.7	0	2	4	6	8	10	12	15
01:38.6	0	2	4	6	8	10	12	15
01:39.5	0	2	4	6	8	10	12	15
01:40.4	0	2	4	6	8	10	12	14

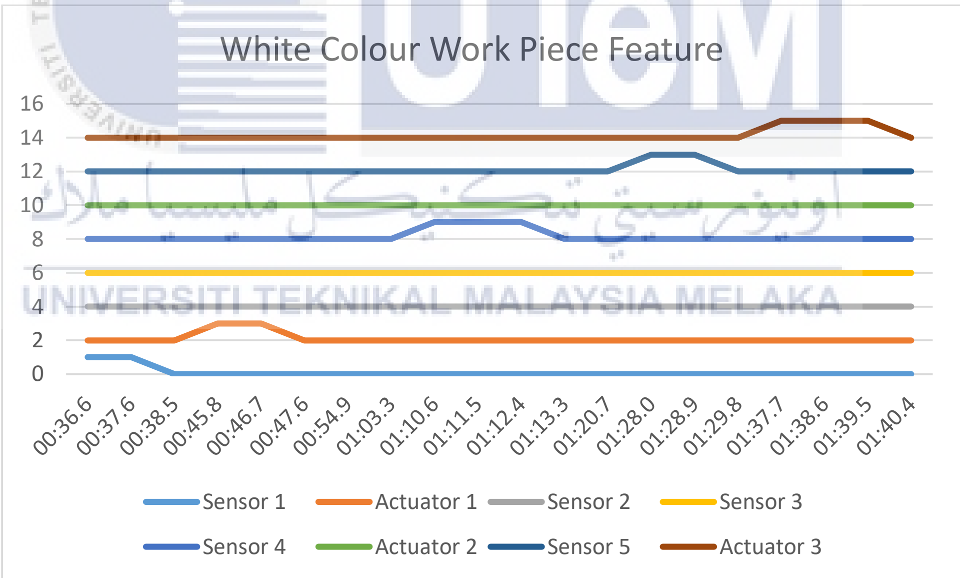


Figure 4.19I/O Analysis of White Colour Work Piece Feature

Figure 4.19 shows the third data analysis from the Firebase server. The analysis of the sorting material can be analyzed from the line graft. In this white colour workpiece feature, the important data both sensor 2 and 3 remain passive and actuator 3 which is place the material to right side sorting compartment.

Besides this, the time taken from the conveyor sorter system and the Firebase server is tested and the results shown in Table

Table 4.5 The Time Taken from The Conveyor Sorter System and The Firebase Server

Experiment Testing for Sensor 1	Time taken (second)
1	4
2	5
3	5
4	6
5	6

Table 4.5 shows the time taken that transfer the data from a physical device to the cloud database. In this testing, only one sensor takes to measure the transfer time from a hardware device to an online database. The result shows the time taken in the range of 4 to 6 second. This may due to internet latency and the configuration of the transferring device. If this experiment is tested in high-speed internet area and good configuration transferring device, the result for transferring the data from the hardware device to an online database will be shorter.

4.6 Comparison between Two Different Data Base

In this path, the main concern is to compare the cloud server that used for IoT gateway in this project which is the result from chapter 4.4 and 4.5. The reason to have this comparison is both have a different method of hardware connection, timing result, suitability, and their own advantages.

First, the hardware connection. In chapter 4.4, Arduino is used to collecting the data from the physical device which is conveyor sorter system then transfer to cloud server by using Raspberry Pi compared to chapter 4.5 which only using the Raspberry Pi. Basically, this type of connection mainly for collecting the analog sensing result because Arduino Uno has this advantage to read the analog signal compare to Raspberry Pi itself which only can read the digital signal. In this project, getting the digital signal need to be prioritized. Hence, Arduino Uno and Raspberry Pi both fulfill the requirement.

Second, the timing result. In chapter 4.4, the Figure 4.13 shows the timing result which automatically generates when the ThingSpeak IoT server receives the data from Raspberry Pi compare to chapter 4.5, Firebase IoT server need to collect and plot the time graph manually. Moreover, both methods require time to transmit the data to the cloud. In term of efficiency on timing, Firebase IoT server behaves better which need average 4 to 6 seconds refresh rate compare to ThingSpeak IoT server which the message update interval limit remains limited at 15 seconds. In this project, the faster the refresh rate of the IoT server is prioritized. Hence, Firebase IoT server is more suitable for this project.

Third, the suitability and own advantages. For ThingSpeak IoT server, it suitable to analyze the analog sensing result such as humidity or temperature because the changes of the surrounding do not instantly change. Hence it's more suitable to record and analyze the change of surrounding. The advantages are easy to configure the device to send the data to the server by using simple hardware, able to visualize the sensor data in real-time and able to work with MATLAB. On the other hand, Firebase IoT server is suitable for the real-time database because it can be synchronized continuously to every related element and support an instant change of data. The advantage is high-quality app development, simple serialization of app state and able to work with android studio. In this project, the ability to support the instant change of data is prioritized. Hence, Firebase IoT server is more suitable.

4.7 Summary

As a conclusion, the inputs and outputs of the conveyor sorter system are tested. The flow of the conveyor sorter system is fully understood. The ladder diagram of the conveyor sorter system is modified due to the original program is not matched with the flow of the system. Besides, all the sensors are tested to make sure the conveyor sorter system can perform well in the design experiment. The next chapter will conclude the finding and results of the project and the recommendation of the project.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

As a conclusion, the first experiment is to know the flow of the conveyor sorter system and the input/output condition to the purpose of connecting the physical device to Arduino Uno. Furthermore, the second experiment is to design the connection between the conveyor sorter system and Arduino Uno, and design the program to receive the signal from conveyor sorter system and the first objective is achieved which is to obtain the status of inputs and outputs (I/O) condition from conveyor sorter system by using Arduino Uno. The third experiment is to test the suitability of the IoT gateway using Raspberry Pi and Arduino Uno. The second objective is to achieve which is to study the integration between Arduino Uno and raspberry pi for sending the data to the cloud. Next, the fourth experiment has used another method to connect the sorter system with IoT gateway and test again the suitability. Finally is to compare the suitability of two different database and the Firebase IoT server is more suitable for this project due to it can receive and update the data faster than ThingSpeak. In the end, the third objective also successfully achieve which to compare between two different cloud database in term of efficiency on timing.

5.2 Future Works

For future improvements, more about event analysis such as to reconfigurable the conveyor sorter system can be carried out after this because the fundamental of the research is done in this paper. Hence, can proceed to a higher level of research. This reconfigurable system is to recognize the error of the system then run another program to continue working without fixing the error. If this can be done, it will raise the industrial zone in the future.

REFERENCES

[1] J. Lee, H. A. Kao, and S. Yang, "Service innovation and smart analytics for Industry 4.0 and big data environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[2] "8 Uses, Applications, and Benefits of Industrial IoT in Manufacturing." [Online]. Available: https://www.newgenapps.com/blog/8-uses-applications-and-benefits-of-industrial-iot-in-manufacturing. [Accessed: 21-May-2019].

[3] "Engineering Essentials: What Is a Programmable Logic Controller?" [Online]. Available: https://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller. [Accessed: 18-May-2019].

[4] M. Mahadi, N. A. Mohd Amin, M. Ab Rahim, and M. S. Abdul Majid, "PLC Trainer Kit Simulator: An Improvement for Automation Study in Polimas," *Appl. Mech. Mater.*, vol. 786, pp. 367–371, 2015.

[5] D. F. Bender, B. Combemale, X. Crégut, J. M. Farines, B. Berthomieu, and F. Vernadat, "Ladder metamodeling and plc program validation through time petri nets," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5095 LNCS, pp. 121–136, 2008.

[6] "4 most popular PLC programming languages nowdays." [Online]. Available: https://electrical-engineering-portal.com/most-popular-plc-programming-languages. [Accessed: 18-May-2019].

[7] O. Pavlovíc and H. D. Ehrich, "Model checking PLC software written in function block diagram," *ICST 2010 - 3rd Int. Conf. Softw. Testing, Verif. Valid.*, pp. 439–448, 2010.

[8] "Logic AND Function - Digital Logic GatesBasic Electronics Tutorials." [Online]. Available: https://www.electronics-tutorials.ws/boolean/bool_1.html. [Accessed: 18-May-2019].

[9] S. P. Dabade, "Research in Engineering Automatic Sorting Machine Using Conveyor Belt," no. June 2015, 2016.

[10] "Types of sorters or classifiers you should be familiar with." [Online]. Available: https://blog.gieicom.com/english/types-of-sorters-or-classifiers-you-should-be-familiar-with. [Accessed: 19-May-2019].

[11] P. Examiner and P. Marcantoni, "United States Patent(再生纤维混凝土制造专利）," vol. 2, no. 12, 2002.

[12] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015.

[13] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Bus. Horiz.*, vol. 58, no. 4, pp. 431–440, 2015.

[14] IHS Technology, "IoT platforms: enabling the Internet of Things," *IHS Technol.*, vol. Whitepaper, no. March, pp. 1–19, 2016.

[15] M. S. D. Gupta, V. Patchava, and V. Menezes, "Healthcare based on IoT using Raspberry Pi," *Proc. 2015 Int. Conf. Green Comput. Internet Things, ICGCIoT 2015*, pp. 796–799, 2016.

[16] R. Kumar and M. Pallikonda Rajasekaran, "An IoT based patient monitoring system using raspberry Pi," *2016 Int. Conf. Comput. Technol. Intell. Data Eng. ICCTIDE 2016*, 2016.

[17] V. Patchava, H. B. Kandala, and P. R. Babu, "A Smart Home Automation technique with Raspberry Pi using IoT," *2015 Int. Conf. Smart Sensors Syst. IC-SSS 2015*, pp. 3–6, 2017.

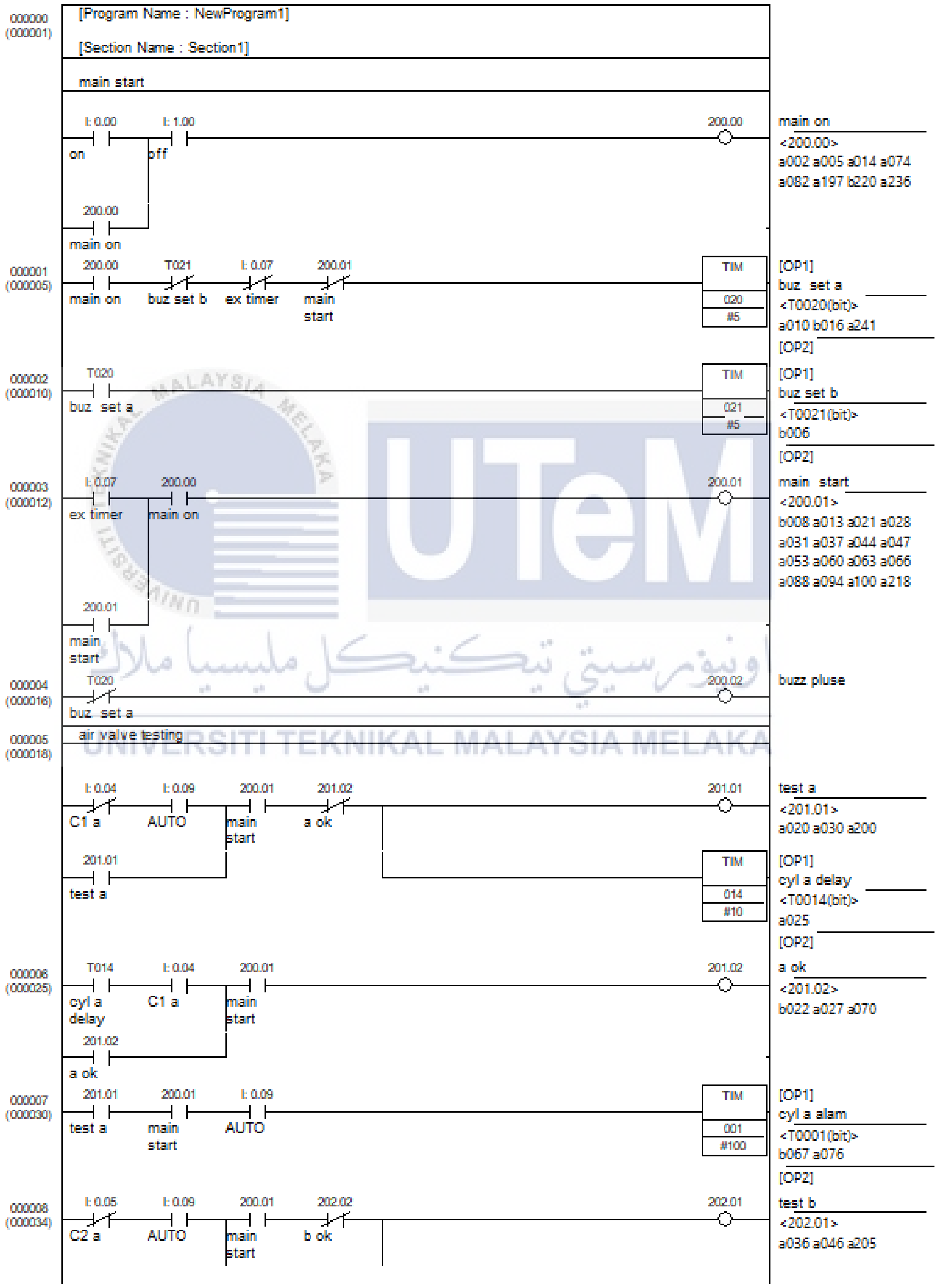
[18] D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," *Glob. Conf. Commun. Technol. GCCT 2015*, no. Gcct, pp. 169–173, 2015.

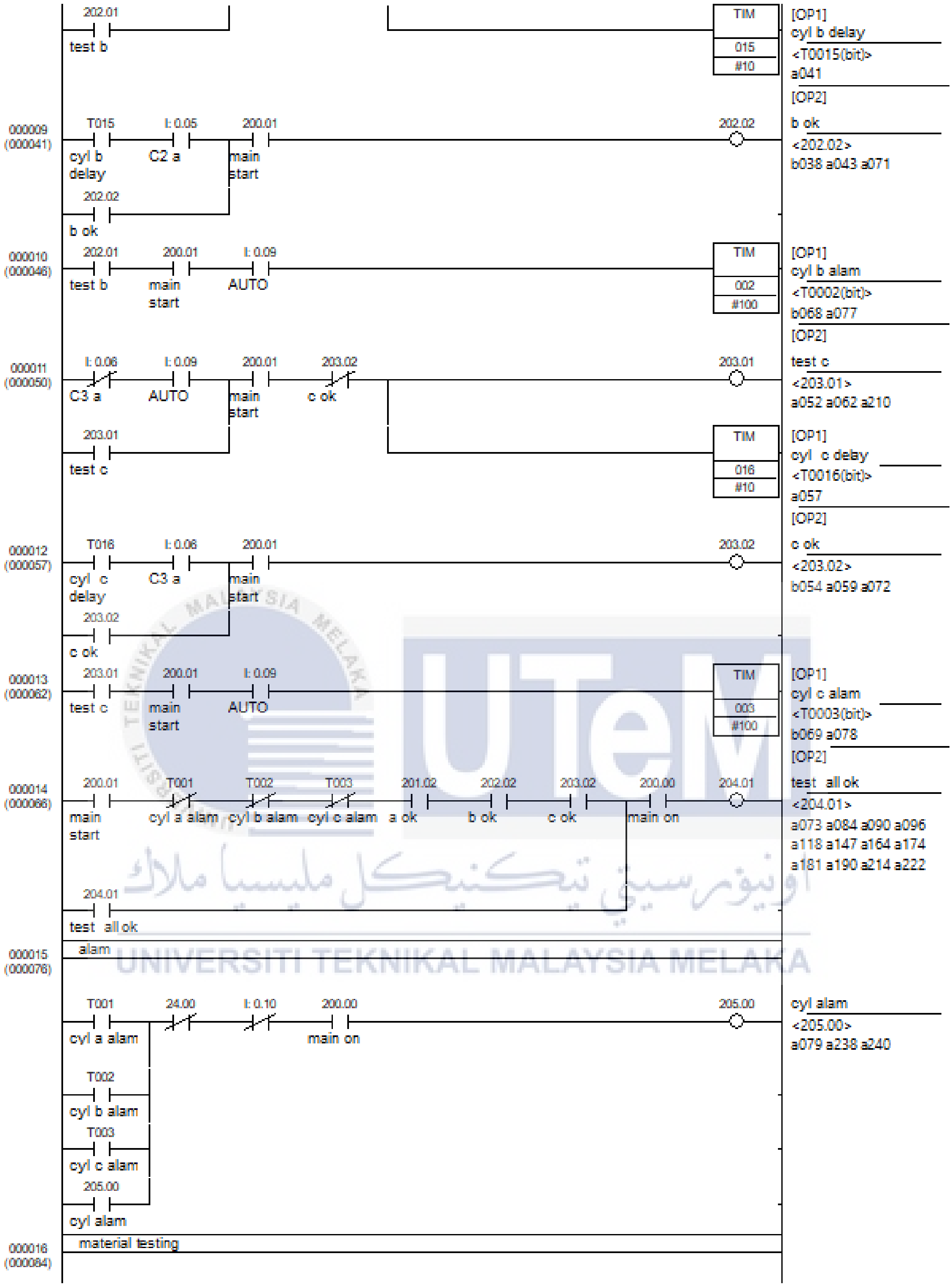
[19] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," *Proc. Asia South Pacific Des. Autom. Conf. ASP-DAC*, vol. 25-28-Janu, pp. 519–524, 2016.

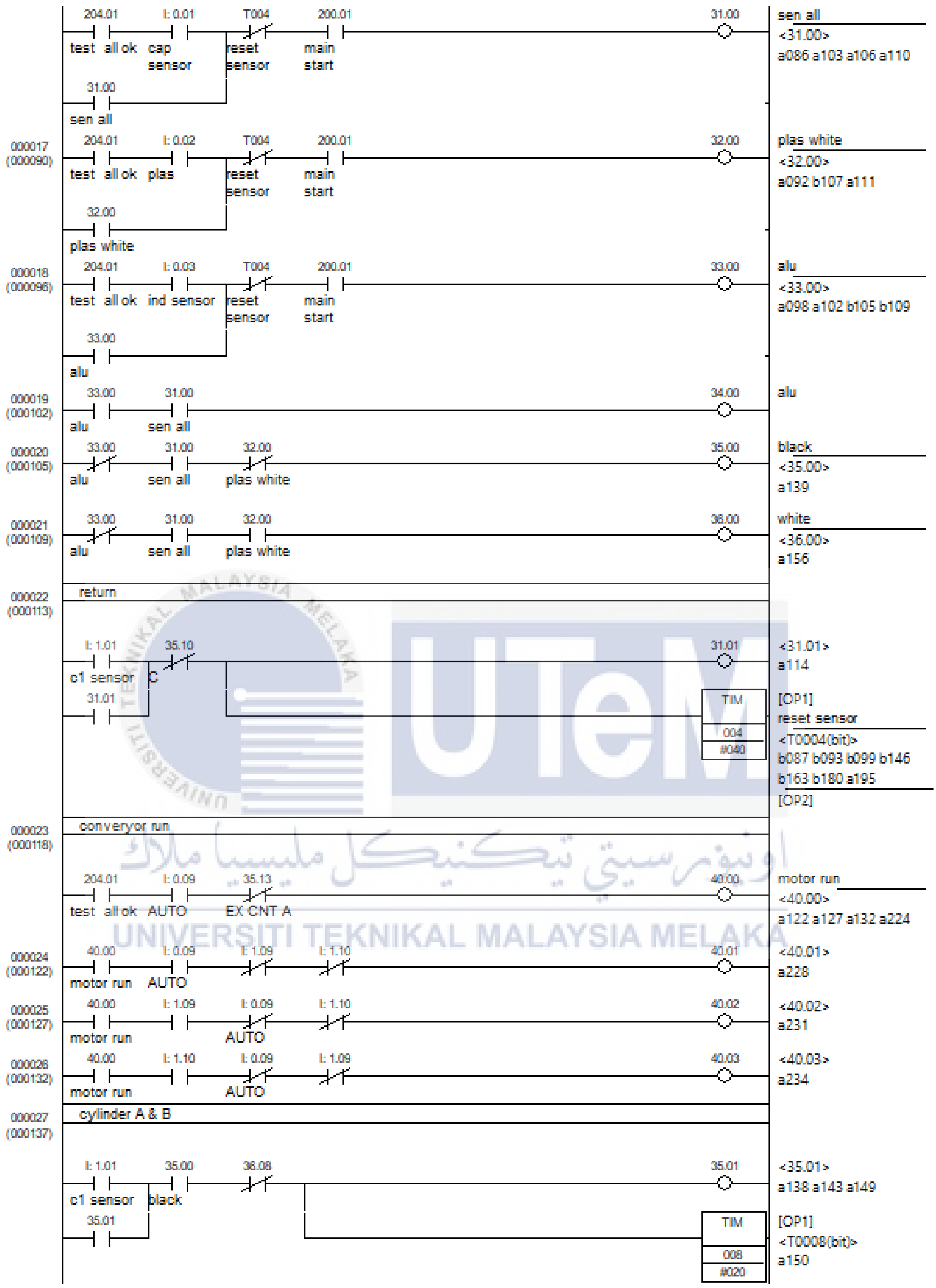
[20] "How to become an IoT developer: 6 tips - TechRepublic." [Online]. Available: https://www.techrepublic.com/article/how-to-become-an-iot-developer-6-tips/. [Accessed: 19-May-2019].

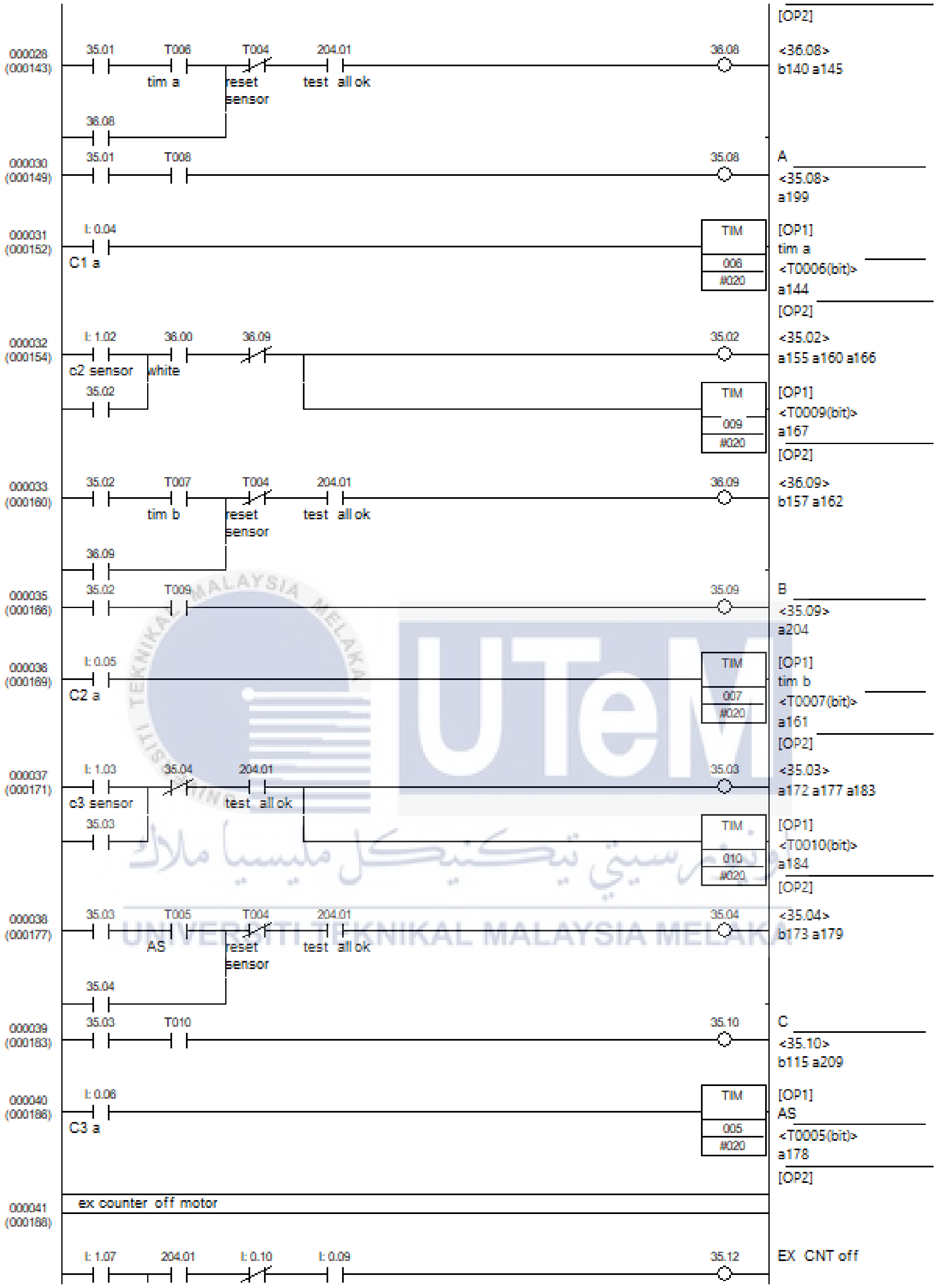
APPENDICES

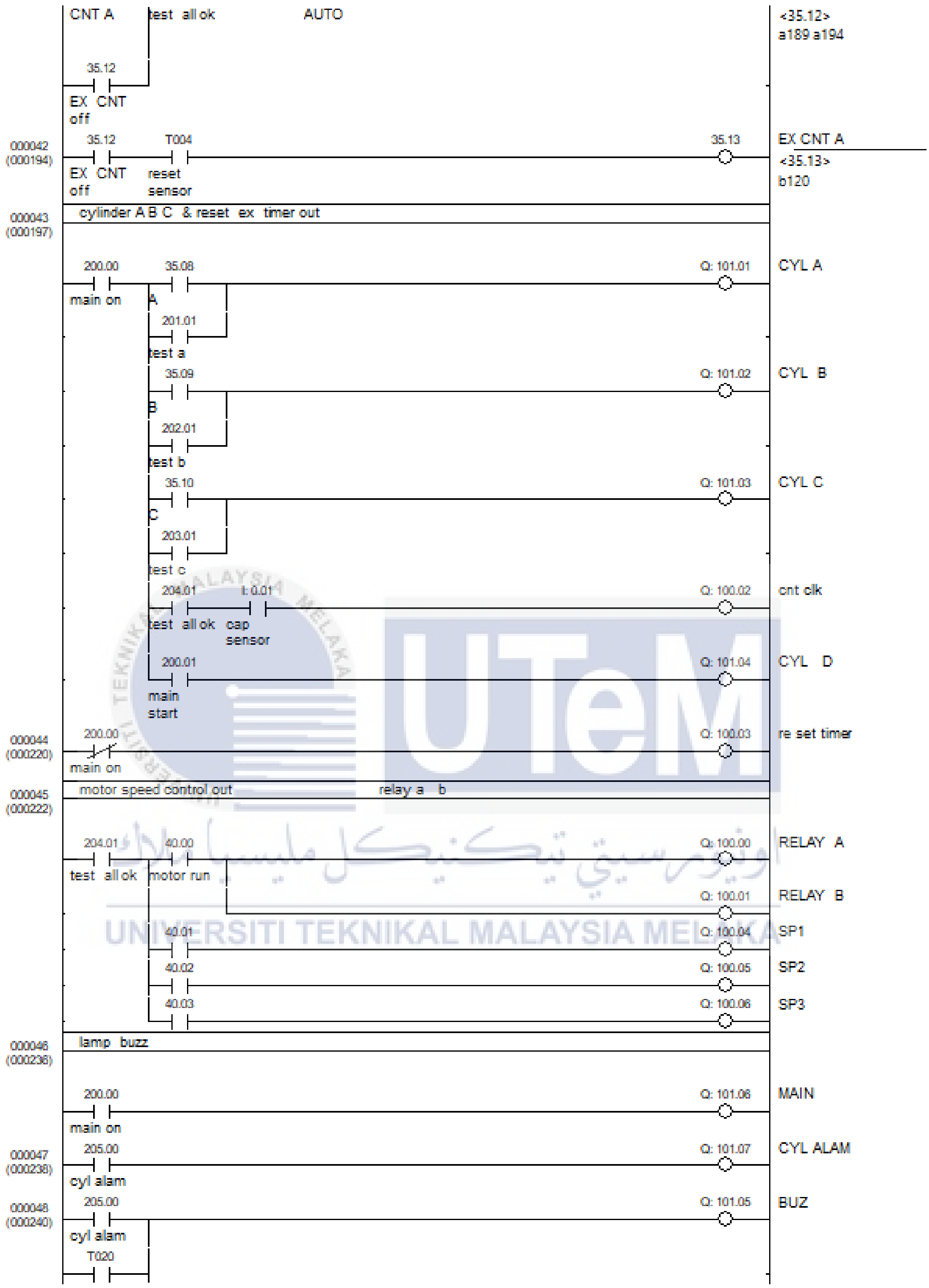
APPENDIX A LADER DIAGRAM FOR THE CONVEYOR SORTER SYSTEM




202.01
test b
TIM
015
#10
[OP1]
cyl b delay
<T0015(bit)>
a041
[OP2]
000009
(000041)
T015
cyl b delay
I: 0.05
C2 a
200.01
main start
202.02
b ok
202.02
b ok
<202.02>
b038 a043 a071
000010
(000046)
202.01
test b
200.01
main start
I: 0.09
AUTO
TIM
002
#100
[OP1]
cyl b alam
<T0002(bit)>
b068 a077
[OP2]
000011
(000050)
I: 0.06
C3 a
I: 0.09
AUTO
200.01
main start
203.02
c ok
203.01
test c
<203.01>
a052 a062 a210
203.01
test c
TIM
016
#10
[OP1]
cyl c delay
<T0016(bit)>
a057
[OP2]
000012
(000057)
T016
cyl c delay
I: 0.06
C3 a
200.01
main start
203.02
c ok
<203.02>
b054 a059 a072
203.02
c ok
000013
(000062)
203.01
test c
200.01
main start
I: 0.09
AUTO
TIM
003
#100
[OP1]
cyl c alam
<T0003(bit)>
b069 a078
[OP2]
000014
(000066)
200.01
main start
T001
cyl a alam
T002
cyl b alam
T003
cyl c alam
201.02
a ok
202.02
b ok
203.02
c ok
200.00
main on
204.01
test all ok
<204.01>
a073 a084 a090 a096
a118 a147 a164 a174
a181 a190 a214 a222
204.01
test all ok
000015
(000076)
alam
T001
cyl a alam
24.00
I: 0.10
200.00
main on
205.00
cyl alam
<205.00>
a079 a238 a240
T002
cyl b alam
T003
cyl c alam
205.00
cyl alam
000016
(000084)
material testing


204.01 I: 0.01 T004 200.01 31.00
test all ok cap sensor reset sensor main start
sen all <31.00> a086 a103 a106 a110
31.00 sen all
000017 (000090)
204.01 I: 0.02 T004 200.01 32.00
test all ok plas reset sensor main start
plas white <32.00> a092 b107 a111
32.00 plas white
000018 (000096)
204.01 I: 0.03 T004 200.01 33.00
test all ok ind sensor reset sensor main start
alu <33.00> a098 a102 b105 b109
33.00 alu
000019 (000102)
33.00 31.00 34.00
alu sen all
alu
000020 (000105)
33.00 31.00 32.00 35.00
alu sen all plas white
black <35.00> a139
000021 (000109)
33.00 31.00 32.00 36.00
alu sen all plas white
white <36.00> a156
000022 (000113)
return
I: 1.01 35.10 31.01
c1 sensor C
31.01
<31.01> a114
TIM 004 #040
[OP1] reset sensor <T0004(bit)> b087 b093 b099 b146 b163 b180 a195 [OP2]
000023 (000118)
converyor run
204.01 I: 0.09 35.13 40.00
test all ok AUTO EX CNT A
motor run <40.00> a122 a127 a132 a224
000024 (000122)
40.00 I: 0.09 I: 1.09 I: 1.10 40.01
motor run AUTO
<40.01> a228
000025 (000127)
40.00 I: 1.09 I: 0.09 I: 1.10 40.02
motor run AUTO
<40.02> a231
000026 (000132)
40.00 I: 1.10 I: 0.09 I: 1.09 40.03
motor run AUTO
<40.03> a234
000027 (000137)
cylinder A & B
I: 1.01 35.00 36.08 35.01
c1 sensor black
35.01
<35.01> a138 a143 a149
TIM 008 #020
[OP1] <T0008(bit)> a150


[OP2]
000028
(000143)
35.01
T006
T004
204.01
36.08
tim a
reset
sensor
test all ok
36.08
<36.08>
b140 a145
000030
(000149)
35.01
T008
35.08
A
<35.08>
a199
000031
(000152)
I: 0.04
C1 a
TIM
006
#020
[OP1]
tim a
<T0006(bit)>
a144
[OP2]
000032
(000154)
I: 1.02
36.00
36.09
35.02
c2 sensor
white
35.02
<35.02>
a155 a160 a166
TIM
009
#020
[OP1]
<T0009(bit)>
a167
[OP2]
000033
(000160)
35.02
T007
T004
204.01
36.09
tim b
reset
sensor
test all ok
36.09
<36.09>
b157 a162
000035
(000166)
35.02
T009
35.09
B
<35.09>
a204
000036
(000169)
I: 0.05
C2 a
TIM
007
#020
[OP1]
tim b
<T0007(bit)>
a161
[OP2]
000037
(000171)
I: 1.03
35.04
204.01
35.03
c3 sensor
test all ok
35.03
<35.03>
a172 a177 a183
TIM
010
#020
[OP1]
<T0010(bit)>
a184
[OP2]
000038
(000177)
35.03
T005
T004
204.01
35.04
AS
reset
sensor
test all ok
35.04
<35.04>
b173 a179
000039
(000183)
35.03
T010
35.10
C
<35.10>
b115 a209
000040
(000186)
I: 0.06
C3 a
TIM
005
#020
[OP1]
AS
<T0005(bit)>
a178
[OP2]
000041
(000188)
ex counter off motor
I: 1.07
204.01
I: 0.10
I: 0.09
35.12
EX CNT off


CNT A
test all ok
AUTO
<35.12>
a189 a194
35.12
EX CNT
off
000042
(000194)
35.12
T004
35.13
EX CNT A
<35.13>
b120
EX CNT
off
reset
sensor
000043
(000197)
cylinder A B C & reset ex timer out
200.00
35.08
Q: 101.01
CYL A
main on
A
201.01
test a
35.09
Q: 101.02
CYL B
B
202.01
test b
35.10
Q: 101.03
CYL C
C
203.01
test c
204.01
I: 0.01
Q: 100.02
cnt clk
test all ok
cap
sensor
200.01
Q: 101.04
CYL D
main
start
000044
(000220)
200.00
Q: 100.03
re set timer
main on
000045
(000222)
motor speed control out
relay a b
204.01
40.00
Q: 100.00
RELAY A
test all ok
motor run
Q: 100.01
RELAY B
40.01
Q: 100.04
SP1
40.02
Q: 100.05
SP2
40.03
Q: 100.06
SP3
000046
(000236)
lamp buzz
200.00
Q: 101.06
MAIN
main on
000047
(000238)
205.00
Q: 101.07
CYL ALAM
cyl alam
000048
(000240)
205.00
Q: 101.05
BUZ
cyl alam
T020


buz set a
35.14
EX CNT B
000049
(000244)
END
(001)

APPENDIX B ARDUINO CODING WITH I/O STATUS

```
int sensorPin1 = 2;int sensorPin2 = 3;
int sensorPin3 = 4;int sensorPin4 = 5;
int sensorPin5 = 6;int sensorPin6 = 7;
int sensorPin7 = 8;int sensorPin8 = 9;

int sensorState1 = 0;int sensorState2 = 0;
int sensorState3 = 0;int sensorState4 = 0;
int sensorState5 = 0;int sensorState6 = 0;
int sensorState7 = 0;int sensorState8 = 0;

int state1 = 0;int state2 = 0;
int state3 = 0;int state4 = 0;
int state5 = 0;int state6 = 0;
int state7 = 0;int state8 = 0;

void setup() {
  Serial.begin(9600);
  pinMode(sensorPin1, INPUT);pinMode(sensorPin2, INPUT);
  pinMode(sensorPin3, INPUT);pinMode(sensorPin4, INPUT);
  pinMode(sensorPin5, INPUT);pinMode(sensorPin6, INPUT);
  pinMode(sensorPin7, INPUT);pinMode(sensorPin8, INPUT);

}

void loop() {
  sensorState1 = digitalRead(sensorPin1);
  sensorState2 = digitalRead(sensorPin2);
  sensorState3 = digitalRead(sensorPin3);
  sensorState4 = digitalRead(sensorPin4);
  sensorState5 = digitalRead(sensorPin5);
  sensorState6 = digitalRead(sensorPin6);
  sensorState7 = digitalRead(sensorPin7);
  sensorState8 = digitalRead(sensorPin8);

  //Cap sensor
 if (sensorState1 == HIGH && state1 == 0) {
    Serial.print("Sensor 1 On\t Atuator 1 Off\t");
    Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
    Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
    Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
    state1 = 1;
  }

  else if (sensorState1 == LOW && state1 == 1){
    Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
    Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
    Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
    Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
    state1 = 0;
  }

  //Actuator 1
  if (sensorState2 == HIGH && state2 == 0) {
    Serial.print("Sensor 1 Off\t Atuator 1 On\t");
    Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
    Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
    Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
    state2 = 1;
  }

  else if (sensorState2 == LOW && state2 == 1){
    Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
    Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state2 = 0;
}

//capacitive proximity sensor
if (sensorState3 == HIGH && state3 == 0) {
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 On\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state3 = 1;
}

else if (sensorState3 == LOW && state3 == 1){
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state3 = 0;
}

//fiber optic
if (sensorState4 == HIGH && state4 == 0) {
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 On\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state4 = 1;
}

else if (sensorState4 == LOW && state4 == 1){
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state4 = 0;
}

//IR sensor 1
if (sensorState5 == HIGH && state5 == 0) {
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 On\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state5 = 1;
}

else if (sensorState5 == LOW && state5 == 1){
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state5 = 0;
}

//Actuator 2
if (sensorState6 == HIGH && state6 == 0) {
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 On\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state6 = 1;
```

```
}

else if (sensorState6 == LOW && state6 == 1){
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state6 = 0;
}

//IR sensor
if (sensorState7 == HIGH && state7 == 0) {
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 On\t Actuator 3 Off\t");
  state7 = 1;
}

else if (sensorState7 == LOW && state7 == 1){
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state7 = 0;
}

//Actuator 3
if (sensorState8 == HIGH && state8 == 0) {
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 On\t");
  state8 = 1;
}

else if (sensorState8 == LOW && state8 == 1){
  Serial.print("Sensor 1 Off\t Atuator 1 Off\t");
  Serial.print("Sensor 2 Off\t Sensor 3 Off\t");
  Serial.print("Sensor 4 Off\t Actuator 2 Off\t");
  Serial.println("Sensor 5 Off\t Actuator 3 Off\t");
  state8 = 0;
} }
```

APPENDIX C ARDUINO CODING WITH I/O STATUS IN THINGSPEAK

```
int sensorPin1 = 2; int sensorPin2 = 3;
int sensorPin3 = 4; int sensorPin4 = 5;
int sensorPin5 = 6; int sensorPin6 = 7;
int sensorPin7 = 8; int sensorPin8 = 9;

int sensorState1 = 0; int sensorState2 = 0;
int sensorState3 = 0; int sensorState4 = 0;
int sensorState5 = 0; int sensorState6 = 0;
int sensorState7 = 0; int sensorState8 = 0;

int state1 = 0; int state2 = 0;
int state3 = 0; int state4 = 0;
int state5 = 0; int state6 = 0;
int state7 = 0; int state8 = 0;

void setup() {
 Serial.begin(9600);
 pinMode(sensorPin1, INPUT);  pinMode(sensorPin2, INPUT);
 pinMode(sensorPin3, INPUT);  pinMode(sensorPin4, INPUT);
 pinMode(sensorPin5, INPUT);  pinMode(sensorPin6, INPUT);
 pinMode(sensorPin7, INPUT);  pinMode(sensorPin8, INPUT);
}
void loop() {
 sensorState1 = digitalRead(sensorPin1);
 sensorState2 = digitalRead(sensorPin2);
 sensorState3 = digitalRead(sensorPin3);
 sensorState4 = digitalRead(sensorPin4);
 sensorState5 = digitalRead(sensorPin5);
 sensorState6 = digitalRead(sensorPin6);
 sensorState7 = digitalRead(sensorPin7);
 sensorState8 = digitalRead(sensorPin8);

 //Cap sensor
 if (sensorState1 == HIGH && state1 == 0) {
  Serial.println("1\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
  state1 = 1;
 }

 else if (sensorState1 == LOW && state1 == 1){
  Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
  state1 = 0;
 }

 //Actuator 1
 if (sensorState2 == HIGH && state2 == 0) {
  Serial.println("0\t 1\t 0\t 0\t 0\t 0\t 0\t 0\t");
  state2 = 1;
 }

 else if (sensorState2 == LOW && state2 == 1){
  Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
  state2 = 0;
 }

 //capacitive proximity sensor
 if (sensorState3 == HIGH && state3 == 0) {
  Serial.println("0\t 0\t 1\t 0\t 0\t 0\t 0\t 0\t");
  state3 = 1;
 }
else if (sensorState3 == LOW && state3 == 1){
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
 state3 = 0;
}

//fiber optic
if (sensorState4 == HIGH && state4 == 0) {
 Serial.println("0\t 0\t 0\t 1\t 0\t 0\t 0\t 0\t");
 state4 = 1;
}

else if (sensorState4 == LOW && state4 == 1){
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
 state4 = 0;
}

//IR sensor 1
if (sensorState5 == HIGH && state5 == 0) {
 Serial.println("0\t 0\t 0\t 0\t 1\t 0\t 0\t 0\t");
 state5 = 1;
}

else if (sensorState5 == LOW && state5 == 1){
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
 state5 = 0;
}

//Actuator 2
if (sensorState6 == HIGH && state6 == 0) {
 Serial.println("0\t 0\t 0\t 0\t 0\t 1\t 0\t 0\t");
 state6 = 1;
}

else if (sensorState6 == LOW && state6 == 1){
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
 state6 = 0;
}

//IR sensor
if (sensorState7 == HIGH && state7 == 0) {
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 1\t 0\t");
 state7 = 1;
}

else if (sensorState7 == LOW && state7 == 1){
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
 state7 = 0;
}

//Actuator 3
if (sensorState8 == HIGH && state8 == 0) {
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 1\t");
 state8 = 1;
}

else if (sensorState8 == LOW && state8 == 1){
 Serial.println("0\t 0\t 0\t 0\t 0\t 0\t 0\t 0\t");
 state8 = 0;
}}
```

APPENDIX D PYTHON CODING FOR THINGSPEAK CONNECTION

```
import time
import serial
import sys
import os
from time import sleep
import urllib

myAPI = "W8S9E5GI7NKKKXRR"
baseURL = 'https://api.thingspeak.com/update?api_key=%s'%myAPI



while 1:
  ser =serial.Serial('/dev/ttyUSB0',9600)
  data = ser.readline()


  print(data)
  fields = data.split('\t')
  print(len(fields))


  if(len(fields) is 9):

    f = urllib.urlopen(baseURL + "&field1=" + str(fields[0]) + "&field2=" + str(fields[1]) +
    "&field3=" + str(fields[2]) + "&field4=" + str(fields[3]) + "&field5=" + str(fields[4]) +
    "&field6=" + str(fields[5]) + "&field7=" + str(fields[6]) + "&field8=" + str(fields[7]))
    print(fields[0]+fields[1]+fields[2]+fields[3]+fields[4]+fields[5]+fields[6]+fields[7])
```

APPENDIX E PYTHON CODING FOR FIREBASE CONNECTION

```
import RPi.GPIO as GPIO
from time import sleep
import time
from datetime import datetime
from time import localtime, strftime
from firebase import firebase

firebase = firebase.FirebaseApplication('https://reatimeconveyor.firebaseio.com/')

firebase.put('Conveyor sorter','sensor 1','off')
firebase.put('Conveyor sorter','actuator 1','off')
firebase.put('Conveyor sorter','sensor 2','off')
firebase.put('Conveyor sorter','sensor 3','off')
firebase.put('Conveyor sorter','sensor 4','off')
firebase.put('Conveyor sorter','actuator 2','off')
firebase.put('Conveyor sorter','sensor 5','off')
firebase.put('Conveyor sorter','actuator 3','off')

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)
GPIO.setup(24, GPIO.IN)
GPIO.setup(10, GPIO.IN)
GPIO.setup(9, GPIO.IN)
GPIO.setup(11, GPIO.IN)
GPIO.setup(25, GPIO.IN)
GPIO.setup(8, GPIO.IN)
GPIO.setup(7, GPIO.IN)


while True:
  t=datetime.now()
    if GPIO.input(23): # if port 25 == 1
      firebase.put('Conveyor sorter','sensor 1','on')
      s1=str(1)
      s2=str(2)
      s3=str(4)
      s4=str(6)
      s5=str(8)
      s6=str(10)
      s7=str(12)
      s8=str(14)
      file = open("data.txt","a+")
      file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
      file.close()



    elif GPIO.input(24): # if port 25 == 1
      firebase.put('Conveyor sorter','actuator 1','on')
      s1=str(0)
      s2=str(3)
      s3=str(4)
      s4=str(6)
      s5=str(8)
      s6=str(10)
```

```
    s7=str(12)
    s8=str(14)
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()

elif GPIO.input(10): # if port 25 == 1
    firebase.put('Conveyor sorter','sensor 2','on')
    s1=str(0)
    s2=str(2)
    s3=str(5)
    s4=str(6)
    s5=str(8)
    s6=str(10)
    s7=str(12)
    s8=str(14)
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()
elif GPIO.input(9): # if port 25 == 1
    firebase.put('Conveyor sorter','sensor 3','on')
    s1=str(0)
    s2=str(2)
    s3=str(4)
    s4=str(7)
    s5=str(8)
    s6=str(10)
    s7=str(12)
    s8=str(14)
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()

elif GPIO.input(11): # if port 25 == 1
    firebase.put('Conveyor sorter','sensor 4','on')
    s1=str(0)
    s2=str(2)
    s3=str(4)
    s4=str(6)
    s5=str(9)
    s6=str(10)
    s7=str(12)
    s8=str(14)
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()

elif GPIO.input(25): # if port 25 == 1
    firebase.put('Conveyor sorter','actuator 2','on')
    s1=str(0)
    s2=str(2)
    s3=str(4)
    s4=str(6)
    s5=str(8)
    s6=str(11)
    s7=str(12)
    s8=str(14)
```

```
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()

elif GPIO.input(8): # if port 25 == 1
    firebase.put('Conveyor sorter','sensor 5','on')
    s1=str(0)
    s2=str(2)
    s3=str(4)
    s4=str(6)
    s5=str(8)
    s6=str(10)
    s7=str(13)
    s8=str(14)
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()
elif GPIO.input(7): # if port 25 == 1
    firebase.put('Conveyor sorter','actuator 3','on')
    s1=str(0)
    s2=str(2)
    s3=str(4)
    s4=str(6)
    s5=str(8)
    s6=str(10)
    s7=str(12)
    s8=str(15)
    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()


else:
    firebase.put('Conveyor sorter','sensor 1','off')
    firebase.put('Conveyor sorter','actuator 1','off')
    firebase.put('Conveyor sorter','sensor 2','off')
    firebase.put('Conveyor sorter','sensor 3','off')
    firebase.put('Conveyor sorter','sensor 4','off')
    firebase.put('Conveyor sorter','actuator 2','off')
    firebase.put('Conveyor sorter','sensor 5','off')
    firebase.put('Conveyor sorter','actuator 3','off')

    s1=str(0)
    s2=str(2)
    s3=str(4)
    s4=str(6)
    s5=str(8)
    s6=str(10)
    s7=str(12)
    s8=str(14)

    file = open("data.txt","a+")
    file.write(str(t)+"\t"+s1+"\t"+s2+"\t"+s3+"\t"+s4+"\t"+s5+"\t"+s6+"\t"+s7+"\t"+s8+"\n")
    file.close()
```