INTRUSION DETECTION SECURITY SYSTEM WITH IOT SENSORS

BY

VERNON CHEAN LYNN CHII

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2021

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: **INTRUSION DETECTION SECURITY SYSTEM WITH IOT SENSORS**

Academic Session: MAY 2021

I VERNON CHEAN LYNN CHII

(CAPITAL LETTER)

declare that I allow this Final Year Project to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor Signature)

Address:

113 PERSIARAN BERCHAM TIMUR 1

TAMAN BERCHAM BARU

31400 IPOH PERAK

Dr. Goh Hock Guan

Supervisor's name

Date: 17th August 2021

Date: 20/8/2021

INTRUSION DETECTION SECURITY SYSTEM WITH IOT SENSORS

BY

VERNON CHEAN LYNN CHII

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2021

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 17th August 2021

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that ___VERNON CHEAN LYNN CHII___ (ID No: ___18ACB06319___) has completed this final year project entitled "**INTRUSION DETECTION SECURITY SYSTEM WITH IOT SENSORS**" under supervision of ___Ts Dr Goh Hock Guan___ (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

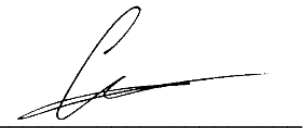
I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(VERNON CHEAN LYNN CHII)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**INTRUSION DETECTION SECURITY SYSTEM WITH IOT SENSORS**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Vernon Chean Lynn Chii

Date : 17th August 2021

ACKNOWLEGDEMENT

I would like to express my deep and sincere gratitude to my supervisor, Ts Dr Goh Hock Guan who has provided me the opportunity and guidance in this IoT-based project titled as Intrusion Detection Security System with IoT sensors. Without his advice and support, I would not reach the altitude of the projects as this is the first IoT-based project I involved in. His valuable advice and motivation are much appreciated.

Other than my supervisor, I would like to extend the gratitude to my family for their love and support throughout the project. Thank you for the understanding and freedom to allow me to focus on the project during the development of the project.

ABSTRACT

This project is an IoT sensor-based security project that detect intrusion for indoor area. Usually, security systems were installed at outdoor area which vulnerable to physical attack as the systems are visible. However, if the system was installed indoor, the system will be protected and prevent from physical damage. The intrusion detection security system is using the IoT sensors to detect the surrounding especially there is abnormal activities happened in the indoor area. However, IoT sensors do not recognise the abnormal activities, when the magnetic sensor of the door is disconnected, the door is opened; when the IR sensor detect blockage, there is something pass by. The sensors do not know what object is passed by, is it a person or others object. Therefore, object detection is introduced in the project as a guidance. Object detection using the camera to scan the area and know whether that is a person or not. This can reduce the false alarm that trigger by the system hence increase the creditability of the system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IoT	Internet of Things
IR	Infrared
HDMI	High-Definition Multimedia Interface
PoE	Power-over-Ethernet
GUI	Graphical User Interface
SSL	Secure Sockets Layer
PIR	Passive Infrared
AWS	Amazon Web Services
SNS	Simple Notification Services
CSI	Camera Serial Interface
SDLC	System Development Life Cycle
PC	Personal Computer

CHAPTER 1 INTRODUCTION

1.1 Motivation and Problem Statement

Based on the statistic from Department of Statistic Malaysia, property crime in year 2019, there are a total of 66,967 cases happened in Malaysia. Property crimes consist of house break-in and theft, vehicle theft, snatch theft and other theft. However, for house break-in and theft consist of 16,497 cases out of the total of property crime. (Department of Statistic Malaysia Official Portal, 2020) It is a 24.6% of the total property crime in Malaysia in year 2019. House break-in theft or known as burglary is an unlawful entry into a home of closed structured with intend of stealing property from other. (Property Crime - FindLaw n.d.) During stealing happening in the property, there is chances for the owner of the property to get injured when they are in the closed area.

Security is a state of being freedom from any danger or threat. Based on Dictionary.com, security means precautions taken to guard against crime, attack, sabotage, espionage. (Definition of security | Dictionary.com, n.d.) A security system is important for our life no matter we are in cyber world or the physical world, as it blocks unauthorized intrusion into our area or properties.

The motivation of the project is to create a security system that using IoT sensor. There are a lot of smart devices in the market but using IoT sensor in smart security system is very less as security is not attractive enough compare to Smart Car and other Smart Home Appliances. Security system was created to make sure human and property away from danger or threats. People do not trust the Smart Security System and IoT devices to protect them or their property because the false alarm happened. This decreases the confidential of the IoT sensor and the confidence of the users. Some of the security systems are easily be destroyed as it exposed outside of the room. When system is fatal, the intruder can enter to the room. We can see in the market right now, there are many systems that were installed and applied outside of the area, if it was disabled by the robbers or thief and they successfully enter the private area, it will very dangerous to the property in the area.

Moreover, the security system that install outside area will bring inconvenience to the users if they wish to monitor their area. Users who wish to know about the situation inside will be harder if there is no surveillance system installed. It also quite

danger for the user if the user does not know is there any intruder inside their area. If the owner enters inside the property without any information the owner might injured by the intruder.

Security system can be implemented inside the property to ensure the safety of the system and the property, as it can become our eyes to monitor the situation inside when there is someone successfully pass through the security system outside of the property. An intrusion detection security system can be built by implementing IoT sensors inside of the area. Intrusion detection security system can detect whenever there is someone enter the area that the owner kept private. Whenever there are people that trigger the system, the owner or person in charge will be alert and can monitor the situation in it at a certain range, without themselves to present at the location, hence their safety is protected. This will also decrease the chances of false alarm happened that triggered by the IoT sensor. It will annoy the user if the security system keeps on reporting false alarm to them. Sooner, the user will not trust the system anymore.

1.2 Project Objectives

The objectives of the project are:

- To implement an intrusion detection security system by using Raspberry Pi together with various IoT sensors.
- To carry out a performance study for the connectivity between the sensor, Raspberry Pi and communication platform.
- To enhance the system robustness by reducing the false alarm issue on system that using IoT sensor.

1.3 Project Scope

The plan of the project is to create an intrusion detection security system by implementing IoT sensor. Through the project, the false alarm that is trigger by the sensor will be reduced. False alarm will create unnecessary panic to the users, and it will cause the user lost confidence on the system. Usually, most of the security system will focus on the door security system which the system is exposed to the intruder. For intrusion detection security system, the system is hidden inside the area that is more secured. The hardware system will be easier to protect from destroying by the intruder easily.

To make this intrusion detection security system, some of the computing devices and IoT sensor are used in the project to implement inside a room. The components that are using is Raspberry Pi, infrared (IR) sensor, magnetic sensor, and camera for the hardware part. In the proposed idea, Raspberry Pi will be the computing devices where all the sensor signal that received will be sent into here. IR sensor and the magnetic sensor will be the activity checker, where if there is any inactivity is happening, it will send the signal to the computing device. Then the computing devices will turn on the camera to detect the intrusion. We can refer to Figure 1.3-1 on hardware configuration.

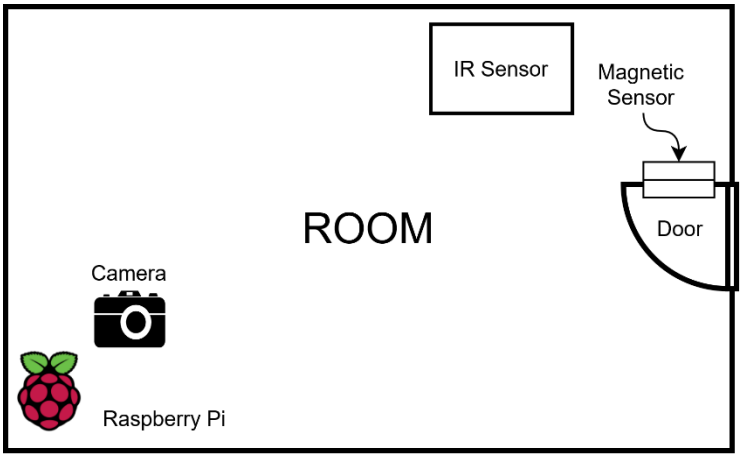


Figure1.3-1: Hardware Configuration of Proposed System

To decrease the false alarm, object detection algorithm will be added into the system. After the Raspberry Pi received signal from the sensor and turn on the camera, the camera will use the algorithm to detect whether the object is a person or not. If the object is a person, the system will send a notification to the owner to allow the user to

choose whether the user want to trigger the alarm or define the detection is false if the system make a mistake. Figure 1.3-2 shows the scenario of the proposed system works.

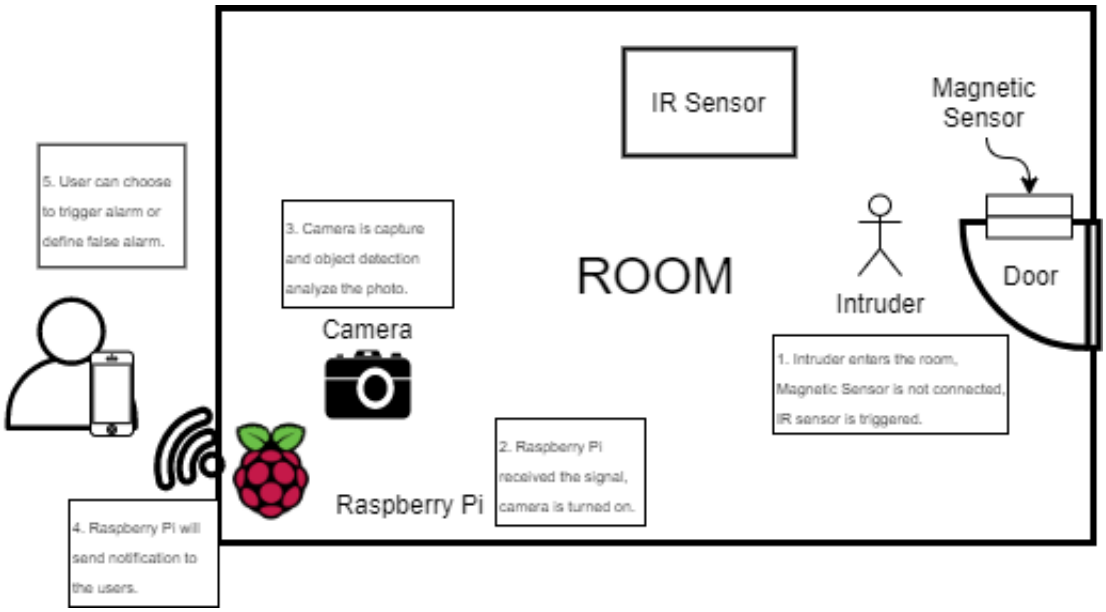


Figure 1.3-2: Scenario of Proposed system works.

To allow the object detection algorithm to work, the programming language that the system using is Python. The Python script will be used to control the sensor and receive the signal from the sensor and make some decisions with it. Python language is also used for other purpose such as sending and communication with the user. For communication platform between the user and the Raspberry Pi, cloud and Telegram will be the communication platform and the notification platform. The data will be stored in the cloud and notification and photo evidence will send into Telegram.

1.4 Main Contributions from the Project

In this project, we increased the usage of the IoT sensor in the home security and ensure the safety of the property and especially human or owner of the property. We can know that a lot of safety issues such as burglary which involve the owner will cause injuries when they try to stop the incidents, even sometimes they accidentally appear on the situation.

From this project, we want to increase the usage of IoT sensor in the intrusion detection security system to act as the senses of human with motion detection sensor and camera sensor. This can decrease the injuries that might possibly happened when or before the checking of human physically. The owner or the security can base on the data that given, make sure that the location is safe then only enter to the location physically.

We also know that security is a concern to many of the users, however if the system is too sensitive and keep on give false alarm will annoy the users and decrease the confident level to the system. Hence, the system does not reach the requirement when the users do not believe it. Throughout this project, we want people to have more confidence on the system.

1.5 Organisation of the Report

The report consists of 7 chapter which is Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Methodology, Chapter 4 System Design, Chapter 5 System Implementation, Chapter 6 System Evaluation and Discussion and Chapter 7 Conclusion and Recommendation. Chapter 2 is discussing the paper and things that we studied, Chapter 3 is discussing the methodology and the requirements for the system. While for Chapter 4 shows the system architecture, functional modules, system flow and designs. For Chapter 5, it guides the setup of hardware and software for the system and Chapter 6 performing the testing and study of performance of the system. Lastly, for Chapter 7, it is the discussion and recommendation for the system.

CHAPTER 2 LITERATURE REVIEW

2.1 Review of the Technologies

2.1.1 Hardware platform

We need a controller or microcomputer that can receive and process the data that received from our sensor such as magnetic sensor and Infrared (IR) sensor to involve in this project. The device needs to have the ability to connect with the camera as we will need to camera sensor to capture or record the situation of the location. To enhance the security, user must able to look into the location through far distance. Hence, we need to have a hardware to have ability to communicate with other devices through Internet.

There is some microcontroller and microcomputer that available in the market such as Raspberry Pi, Arduino, and micro:bit.

2.1.1.1 Raspberry Pi3 B+

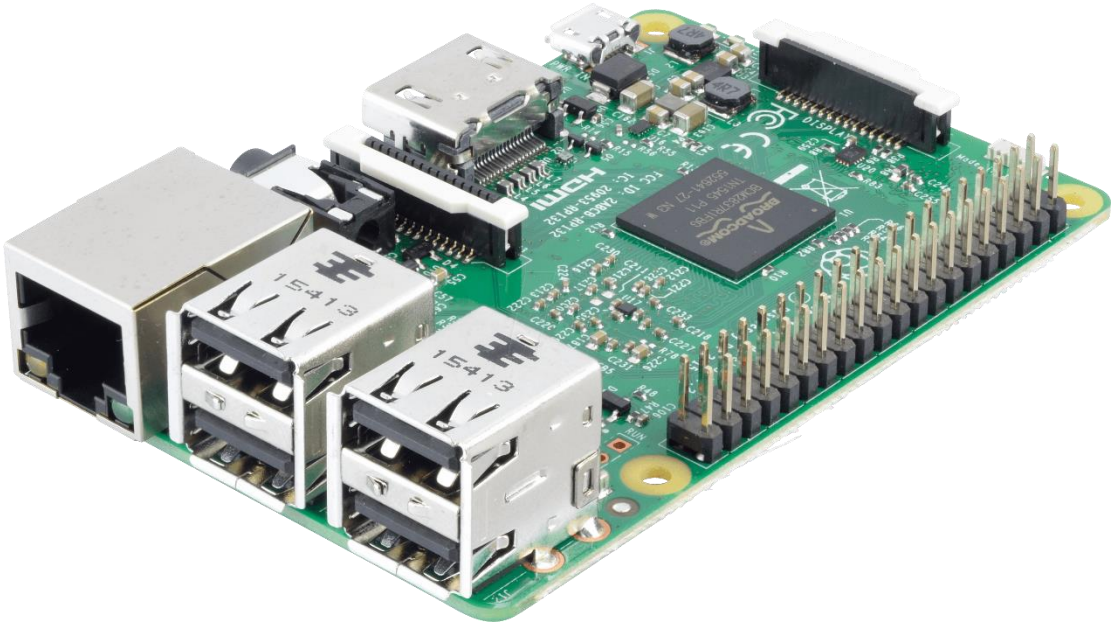


Figure 2.1.1.1-1: Raspberry Pi3 B+

Raspberry Pi3 B+ is developed by the Raspberry Pi Foundation from United Kingdom and it was released in year 2018. It is a small and light weight device which can install various GUI in this microcomputer.

Following are the brief specification of the Raspberry Pi3 B+:

- Dimensions: 85.6mm x 56mm x 21mm
- 1.4GHz 64-bit quad-core Broadcom Arm Cortex A53-architecture
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN
- 40-pin extended GPIO
- CSI camera port for Raspberry Pi Camera
- Power-over-Ethernet (PoE) support
- Full Size HDMI
- MicroSD Card Slot
- 4 USB2.0 ports
- 3.5mm 4-pole composite video and audio output jack
- DSI display port for Raspberry Pi touchscreen display

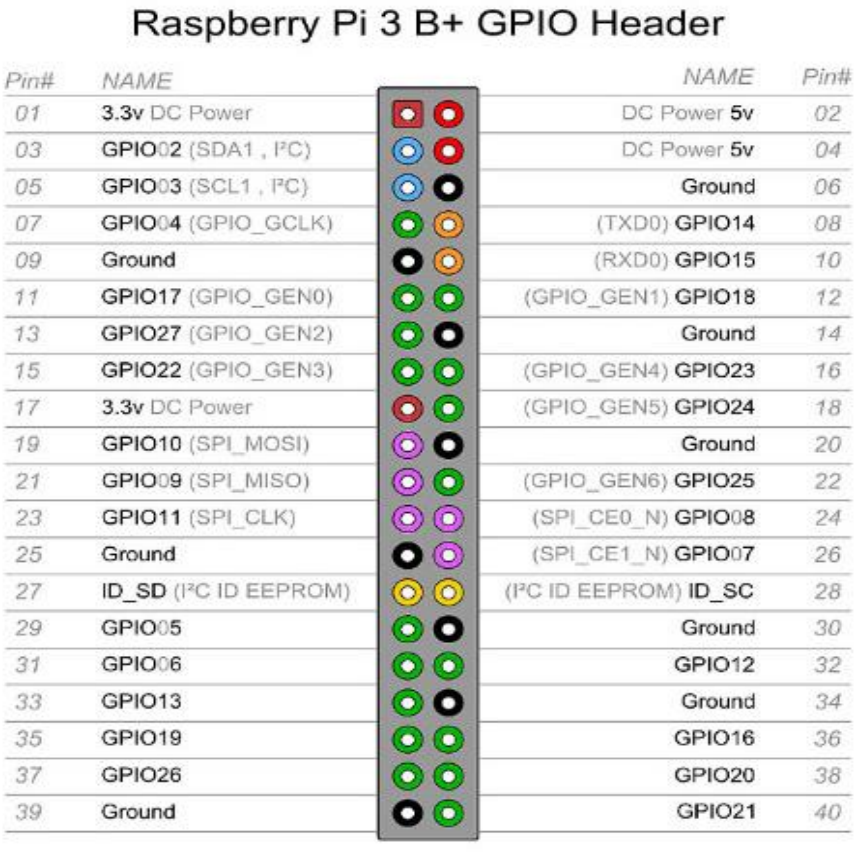


Figure 2.1.1-2: Raspberry Pi3 B+ GPIO Header

There are 40-pin extended GPIO available in Raspberry Pi3 B+ which allow us to connect with external. Moreover, Raspberry Pi3 B+ have CSI camera port to connect a Raspberry Pi Camera. Lastly, there is onboard wireless LAN connection for our Raspberry Pi3 B+ to communicate wirelessly through the Internet.

2.1.1.2 Arduino Uno

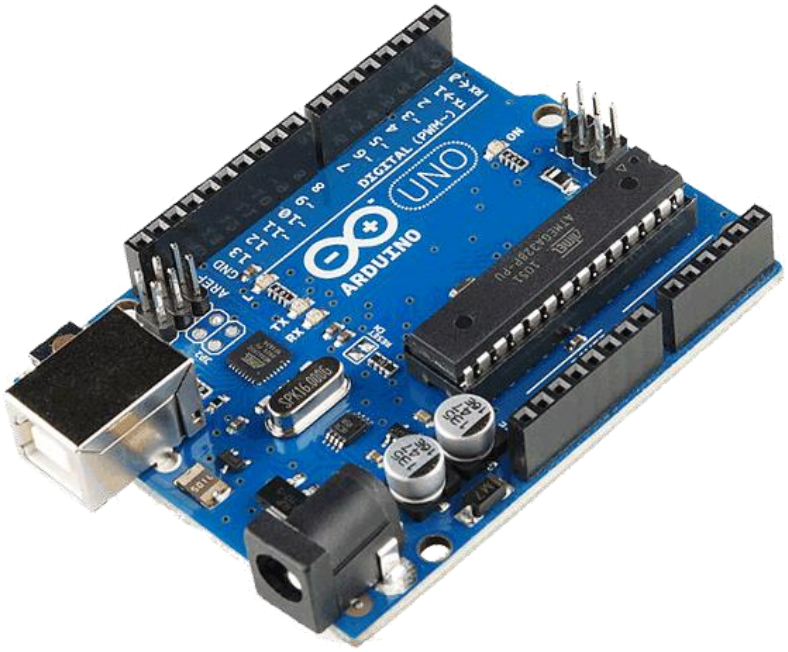


Figure 2.1.1.2-1: Arduino Uno

Arduino Uno was developed by Arduino.cc. Arduino Uno is a microcontroller board that design for simple hands-on experience. It is also small with various ways to power up the microcontroller.

Following are the brief specification of the Arduino Uno:

- Dimensions: 68.6mm x 53.4mm
- Weight: 25g
- Microcontroller on-board: ATmega328P
- 14 digital input output pins (6 of it provide pulse-width modulation (PWM))
- 6 analog input pins
- USB Type-B port
- USB, battery, and alternative current to direct current adapter to power up the board.

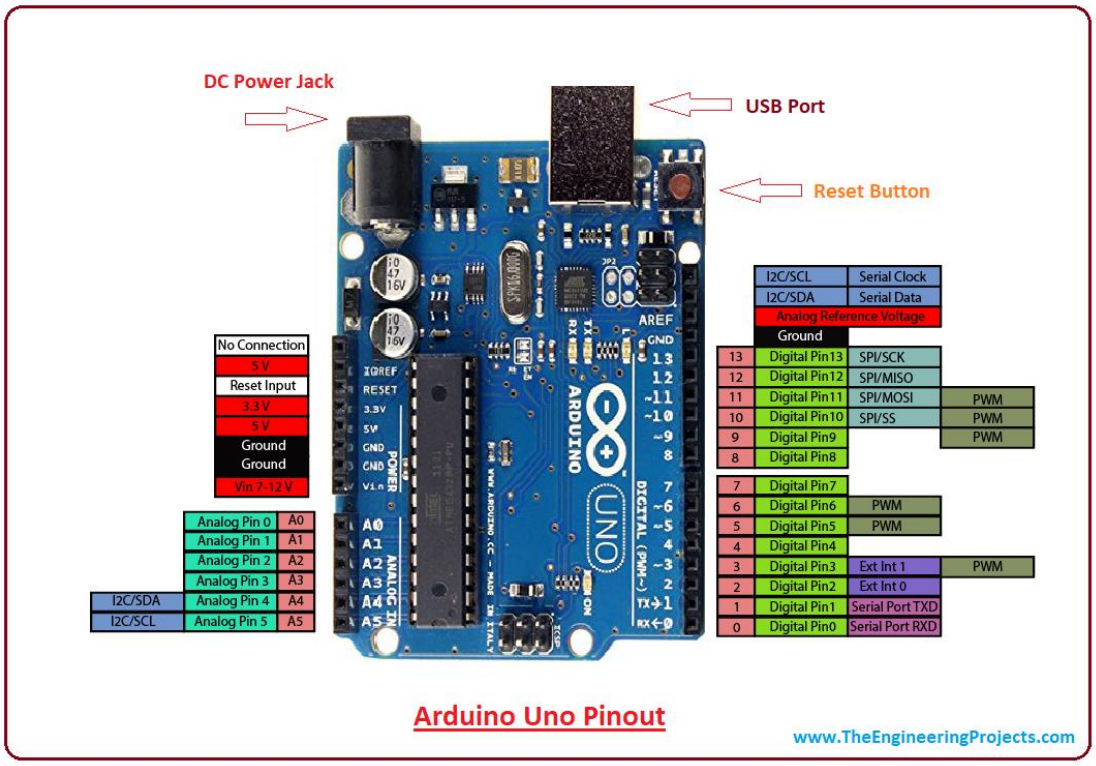


Figure 2.2.1.2-2: Arduino Uno Pinout

(Aqeel, 2021)

Arduino Uno has a total of 14 digital pins and 6 analog pins. All the pins are operating at 5V. By coding the pin using the Arduino Software (IDE), the pins that available can used for reading inputs and output to the sensor or actuator. Since Arduino Uno does not support any OS, it does not have any user interface, which means that the coding will need to be done by connecting to the serial interface.

For Arduino Uno, it is not capable of internet connectivity, it requires other modules such as ESP8266 to make this works. ESP8266 is a module that belongs to ESP's family which has a build-in microcontroller and 1MB flash to allow it to connect to WiFi. To support more ideal IoT projects, there are some component boards such as NodeMCU ESP8266 contain the ESP8266 chip. With the Arduino IDE, the chip and board that related with Arduino can be programmed through it.

2.1.1.3 Mirco:bit

Figure 2.1.1.3-1: micro:bit

Micro:bit was designed by British Broadcast Corporation (BBC). It is a small-sized computer that almost a size of a credit card. It was different with other microcomputer or microcontroller; it is using edge connection as external connection.

Following are the brief specification of the micro:bit:

- Dimension: 51.6mm x 42.0mm
- 32-bit ARM Cortex-M0 CPU
- 5x5 LED light display
- BLE Smart Antenna
- Motion Sensor on-board with accelerometer and magnetometer
- Temperature sensor on board
- 20 assignable GPIO pins with edge connector

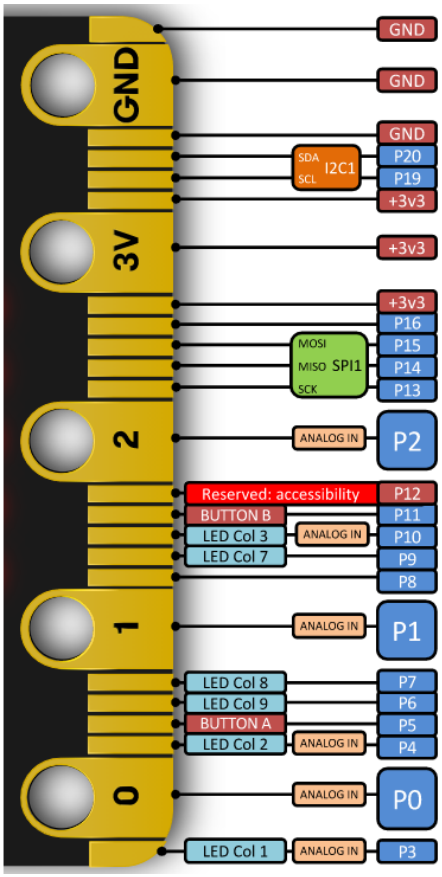


Figure 2.1.1.3-2: micro:bit pinout

Mirco:bit's GPIO is using edge connector for the GPIO input and output like other microcomputer and microprocessor did. It is the same function that allow micro:bit to connect with other components. It has 20 strips and 5 rings, where the 0, 1 and 2 rings are for GPIO that can perform analog and pulse-width modulation.

2.1.2 Firmware/OS

2.1.2.1 Raspberry Pi OS

Figure 2.1.2: The logo of Raspberry Pi OS

Raspberry Pi OS, known as the Raspbian OS is the official operating system from Raspberry Pi Foundation. Raspberry Pi OS is based on Debian and it is optimized for the Raspberry Pi hardware. Raspberry Pi OS has their own Graphical User Interface (GUI). The GUI in the OS allow us to have more easier access to manipulate and control the function in the OS. In Raspberry Pi OS, it allows the user to get or install various of software provided by Raspberry for ease of use.

In Raspberry Pi OS, they also support some GUI application with integrated development environment for user to write code or script in the OS. In the OS, they also have the terminal, so that the user can see through the OS.

Moreover, since Raspberry Pi OS is the official operating system from Raspberry Pi, it is more convenience to use the GPIO pin that available on the Raspberry Pi. The user do not need to install Rpi.GPIO again since the library is provided with the OS.

2.1.2.2 Ubuntu

Ubuntu is an open-source operating system which based on Debian or Linux Distribution. It is a popular OS which it has a highly customizable GUI. The GUI also allow the user to interact with the system itself. So, user can install any applications that they want into the system.

Other than that, Ubuntu has huge number of free software available in the software center. User can install the application by just click the install button. However, for some users that wish to use command lines to install, they are allowed to do so. Ubuntu also has a huge community which can find for solutions in the community support.

Lastly, Ubuntu has the latest version of Linux kernel in their latest version. This also mean that older hardware or newest chips can run with the OS. (Verma, 2018)

2.1.2.3 Ubuntu MATE

Ubuntu MATE is an open-source operating system based on Ubuntu. It was designed to run on most of the computer hardware in the market. Raspberry Pi are also supporting this OS. They have a configurable desktop environment, same as Ubuntu, they have their own GUI so that user can interact with the OS.

The MATE Desktop, which is the desktop environment provide the tradition metaphors like Windows OS and macOS does, which means for user that is new to Linux or Ubuntu, they can familiarize with it easier. (Ubuntu Mate, n.d.)

2.1.3 Programming Language

2.1.3.1 Python

Python is a programming language with extensive support libraries that we need during the writing of the python script. It is very easy to use and develop. Python is designed to be easier to understand as the python is much more like English language.

Python is a general-purpose coding language. Python is widely used at software development and web development. There are a lot of open source algorithm are using python language as their primary language.

2.1.3.2 C++

C++ is a middle-level programming language based on tutorialspoint article. (C++ Tutorial - Tutorialspoint, n.d.) C++ is widely used in games, operating systems, and some applications. It is the basic of the programming language to allow developer or programmer to handle other language easier such as Java and Python.

C++ is closer to the hardware, where the language can also control the hardware such as memory management. C++ can run the code faster and with higher computational efficiencies.

2.1.3.3 Java

Java was developed in year 1995 and own by Oracle Corporation. It is a class based object-oriented programming language. Java is a platform independent language where the code can run on different platform without making any chance to the code. Java's code can be reused because it is an object-oriented based programming language. It was widely used in desktop and mobile application, big data processing and embedded systems. (Learn Java Programming, n.d.)

2.1.4 Algorithm

2.1.4.1 TensorFlow

TensorFlow is an open-source library that developed by Google Brain Team. It is an algorithm focusing on deep learning and machine learning applications. It can run a large numerical computation. TensorFlow collect the data in the form of multi-dimensional arrays which are better at controlling large amount of data. TensorFlow based on the data flow graphs that have nodes and edges. It is easier to execute TensorFlow as the execution method is in graph form. (What is Tensorflow: Deep Learning Libraries and Program Elements Explained, 2021)

Figure 2.1.4: Logo of TensorFlow

TensorFlow were applied on different use cases in real life such as image recognition, video detection, recommendation, and text-based applications. It is widely used in the industrial and companies such as Carousell and Airbus. For Carousell uses TensorFlow to improve buy and seller experience by introduce image search so that buyer can get better recommendation from them. While for Airbus, they are using TensorFlow to extract information from satellite images. TensorFlow is used for monitoring Earth's surface changes that cause by the natural disaster to their client. (Case Studies and Mentions | TensorFlow, 2021)

TensorFlow is widely used on image recognition and video detection. It can be trained by deep learning. Deep learning takes the advantages of TensorFlow by analyzing thousands of samples so that the algorithm can learn and identify the sample with general features of the objects. Since TensorFlow is widely used, there are a lot of well-trained model available and the library and community that they provided is much richer than another object detection algorithm.

One of the weaknesses that TensorFlow has is it is very not understandable easily, might face difficulty for the beginner.

2.1.4.2 You Only Look Once (YOLO)

You Only Look Once (YOLO) is specialized in object detection algorithm with still image and real-time data. YOLO is different with another object detection algorithm. They are having a different approach on object detection; they redeploy the classifier to allow the object detection to perform better. YOLO is looking like single-stage detector, which can perform faster obviously than two-stage detector, but its accuracy is yet to be improved. YOLO frame their object detection as regression problem. The bounding boxes are separately with the class probability maps. Bounding Boxes, confidence and class probability maps are predicted directly from images under a single neural network. Hence, end-to-end detection will be optimized. (Redmon, Divvala, Girshick and Farhadi, 2016)

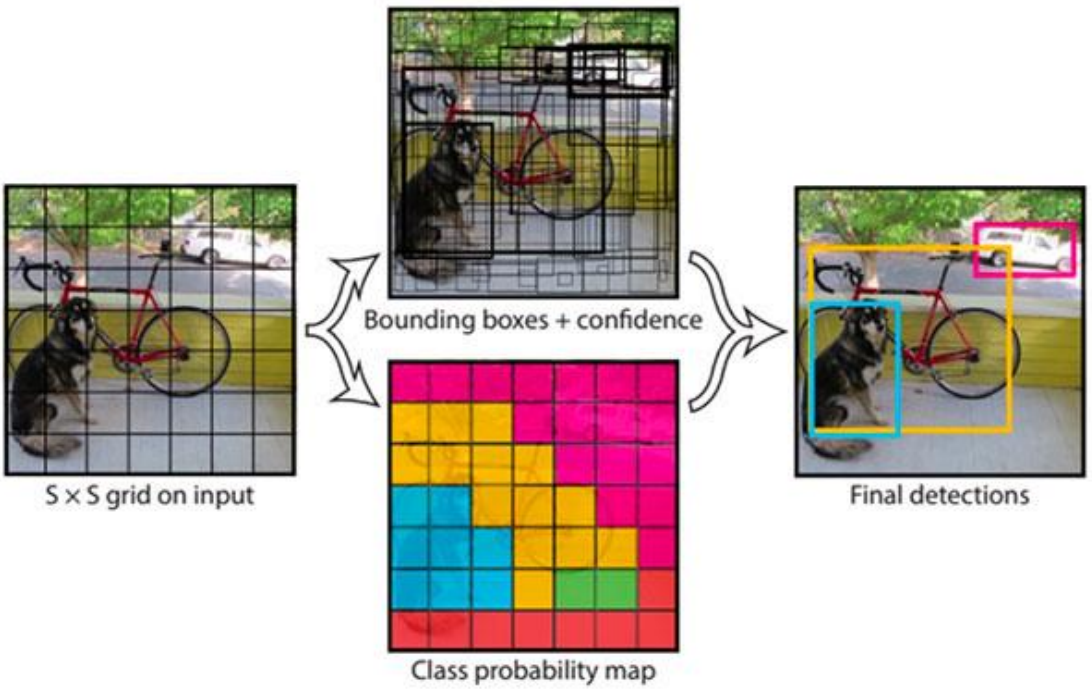


Figure2.1.4.2: Object Detection of YOLO.

Since, the nature of the YOLO algorithm is where each SxS grid can detect one object, therefore YOLO does not handle the object that is small in the photo or in the stream. They also have difficultly to detect the object when there is object stick closed together. (Rosebrock, 2018)

2.1.4.3 Keras

Keras is an open source neural network library that based on Python. It can work on top of some backends algorithm such as TensorFlow, Microsoft Cognitive Toolkit and Theano. (TensorFlow Alternatives, n.d.)

Keras is an API which is design for much more simple and easier to learn. It builds on top of TensorFlow. Keras can have easier model building and training and it is easier to understand since Keras is python-based. It is working in a simpler network, hence for maintaining or debugging the algorithm will be easier. (Keras vs Tensorflow vs Pytorch: Popular Deep Learning Frameworks, 2021)

However, compare with another algorithm, Keras has a smaller dataset. Keras is also slower and low performance than some other algorithm such as TensorFlow.

2.1.5 Summary of the Technologies Review

After comparing with the hardware, firmware, programming language and algorithm, the summary is shown on the Table 2.1.5 below.

Table 2.1.5: Summary of Technologies Review

Hardware			
Devices	**Raspberry Pi3 B+**	**Arduino Uno**	**Micro:bit**
Processor/Controller	1.4GHz 64-bit quad-core Broadcom Arm Cortex A53-architecture	ATmega328P	32-bit ARM Cortex-M0 CPU
Bluetooth	BLE	N/A	BLE Smart Antenna
GPIO Pin	40-pin extended GPIO	14 digital input output pins 6 analog input pins	20 assignable GPIO with edge connector
Operating System	Linux (Raspberry Pi OS)	N/A	N/A
Price	RM 156.00	RM 105.00	RM 85.00
Firmware/OS			
OS	Raspberry Pi OS	Ubuntu	Ubuntu MATE
Based on	Debian-based	Debian /Linux Distribution	Ubuntu Based.
GUI	Yes	Yes	Yes
Special	Official OS from Raspberry Pi Foundation	Open-source OS, with huge supported community.	Desktop Environment, Windows OS or macOS like

Programming Language			
Language	Python	C++	Java
Programming Language Type	High level with extensive support libraries	Middle level	Class based Object Oriented
Usage	- software development - web development	- games - operating systems - applications	- desktop and mobile application - big data processing - embedded systems
Algorithm			
Object Detection	TensorFlow	YOLO	Keras
Background	Open-source library focusing in Deep Learning and Machine Learning.	Specialize in Object Detection	Open-Source Neural Network Library
Advantages	Huge amount of well-trained model and huge community.	Detection is faster	- Build on top of TensorFlow, easier to understand. - Easier to debug.
Disadvantages	Hard to understand.	Detection does not work well with small objects or object that sticked together.	Smaller Datasets

2.2 Review of the Existing Systems/Applications

2.2.1 Door Security System for Home Monitoring Based on ESP32 (Andreas et al., 2019)

According to the thesis, they are using of the hardware in the system such as, Microcontroller ESP32, PIR Sensor, Magnetic Sensor, Internal Touch Sensor, Alarm Buzzer and Electric Strike for this project. While on the software part, they are using an Android-based mobile application with Cloud MQTT.

Microcontroller ESP32 was implement together with the Lolin D32 board which the board is equipped with the wireless LAN module. It was used to control all the devices in one environment since ESP32 has two cores. The cores can have different function, one can use for running WiFi functions while the other one can use to execute uploaded programs. ESP32 has WiFi and Bluetooth module. It has internal touch sensor and large memory. Motion sensor, PIR sensor was implemented to detect motion and magnetic sensor was used to detect the door state. To response the unauthorize entrance, alarm buzzer was added to warn the people and electric buzzer is to lock and unlock the door.

The microcontroller took control of the message sending function to the MQTT broker. They were using a service that provide by a company, MQTT Cloud that provide MQTT Broker service over the Internet. MQTT Broker in charge of receiving message from the microcontroller or a device and deliver the received message to any client that subscribe to the topic. In the design, a smartphone device is used to send message to the broker and as a client, that subscribe to the topic. The topic subscribed will then processed as a notification message. And for the last part will a subscriber with a java-based program that use framework spring as listener for all topic exist on MQTT broker. The listener is to update the Firebase Database and the updated data will be the status of the online device and some other commands.

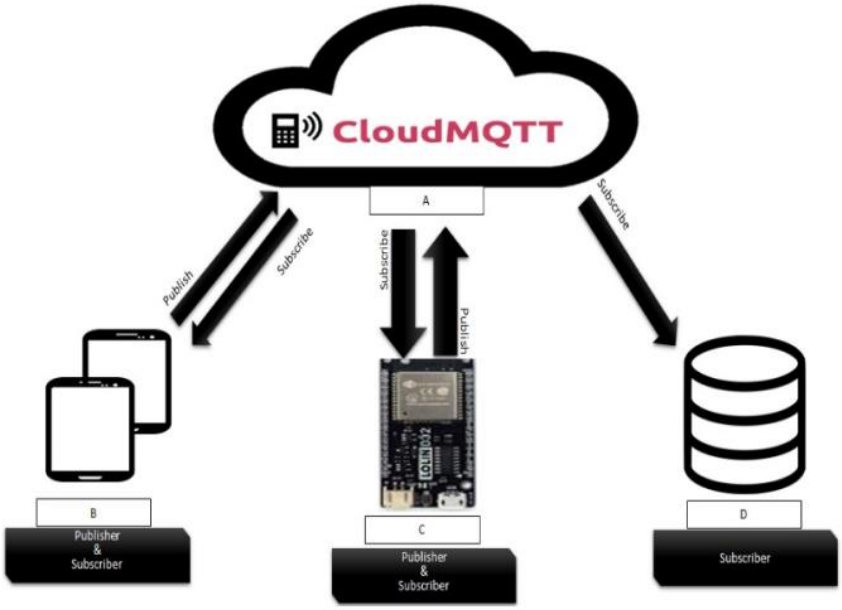


Figure 2.2.1-1: System Architecture Design

For the hardware part, programming language, Arduino is used to compile the ESP32. There is two LED that indicate the power, red LED indicate the power supply; green LED indicate as WiFi indicator. For PIR Sensor and Magnetic Sensor, it will send the signal back to the ESP32 for motion detection and the state of the door. Buzzer alarm module is used to warn the people that control by the ESP32 too. ESP32 can also control the electric strike module to lock and unlock the door.

On the software application, it can separate into two parts, one is the Door Security Mobile application and the door lock software. For the door lock system, we can understand it by looking at the door lock system flow analysis as shown in the Figure 2.2.1-2 below.

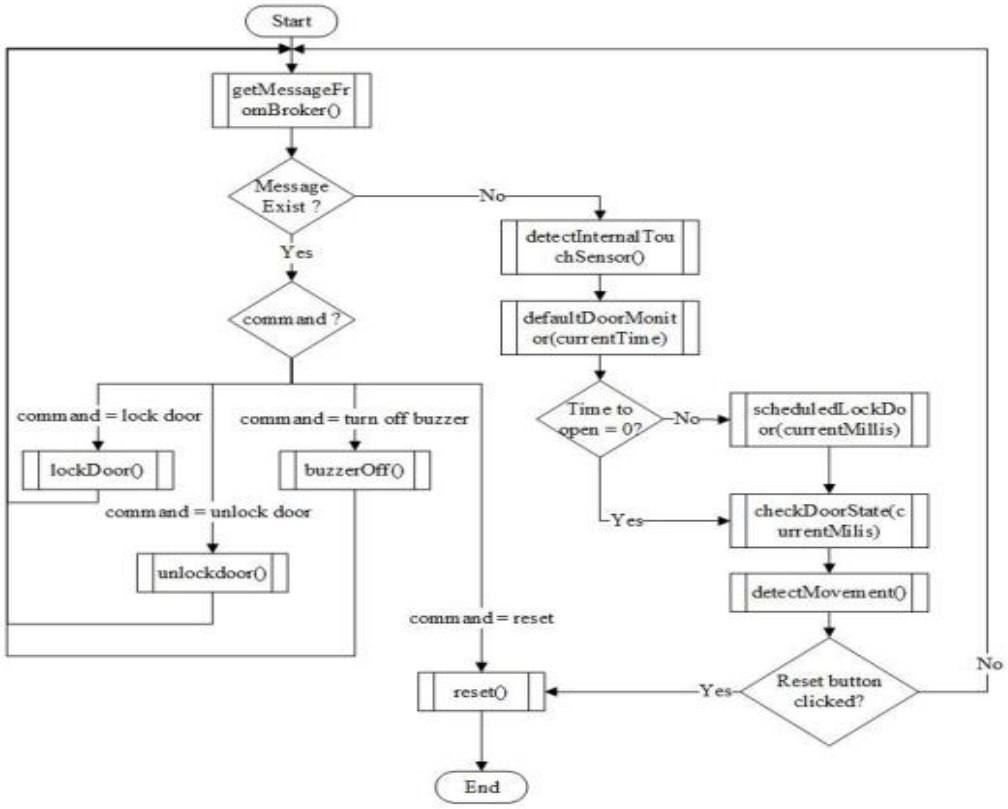


Figure 2.2.1-2: System Flow of Door Lock System

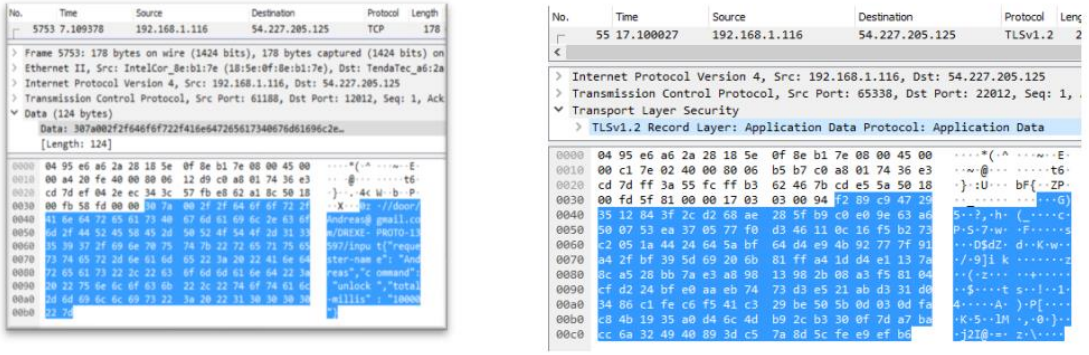
Based on the System flow, we can see that the system is subscribing to the MQTT Broker. The MQTT broker is the one that controlling the door lock. Then the system will break into two paths, where the system is receiving the message or not. If message is received, it will proceed to 3 command, which is “lock door”, “unlock door” and “turn off buzzer”. When “unlock door” command is received, door lock will be opened. If door lock is unlocked but the door is not opening, the door will be locked automatically; if the door is opened, the requester name will be saved. For “lock door” command, door status will be checked first. A message will be published if the door is not closed. If the door status is closed, door lock will be triggered, and the door is locked. For the buzzer part, if it receives command on turn off buzzer, it will be turned off, if the buzzer alarm is on. The system will end when the “reset” command is received.

If no message is received from the broker, the internal touch sensor in ESP32 will be checked. If the sensor is sending signal and the specific time given is out, door lock will proceed to the system based on time interval. The system will then check the door status, the buzzer status, if door is not locked then will reset it. When the input time of door more than 0, door lock will be scheduled. The last requester name will be

recorded in the database. The buzzer alarm will be turned on when the requester name is unknown. For the motion sensor (PIR sensor), if it detect there is movement, it will publish a message that movement is detected. Same as message received side, if reset button is pressed, the system will be reset and end the process. If not, it will loop back to the beginning.

For the Door Security Mobile application int this system, they allow user to check the door status and unlock or lock the door if they sign in with the authorize account. With the application, they can check the log history in it and who accessed the door. They can also check the door status, door lock status and buzzer alarm status. In this system user are allowed to lock the door automatically or not. If they want it to lock automatically, user need to input a certain number od second once the door is unlocked; if they do not want to lock automatically, they can simply insert a 0. The door can only unlock with the mobile application, the system will trigger the buzzer when the door is unlocked without the application because the system will assume that the door is force opened. However, if the touch sensor that located in the room is touched, the buzzer will not trigger if the door is opened. This is because the system will assume the door is open from inside.

Since MQTT is implemented, it is vulnerable to network security issues because the information will be capture by the attackers. In this project, they also implemented Secure Sockets Layer (SSL) encryption when sending message by using MQTT. A testing was brought out in their project. They have tested sending and receiving message without SSL encryption and sending message and receiving message using SSL.

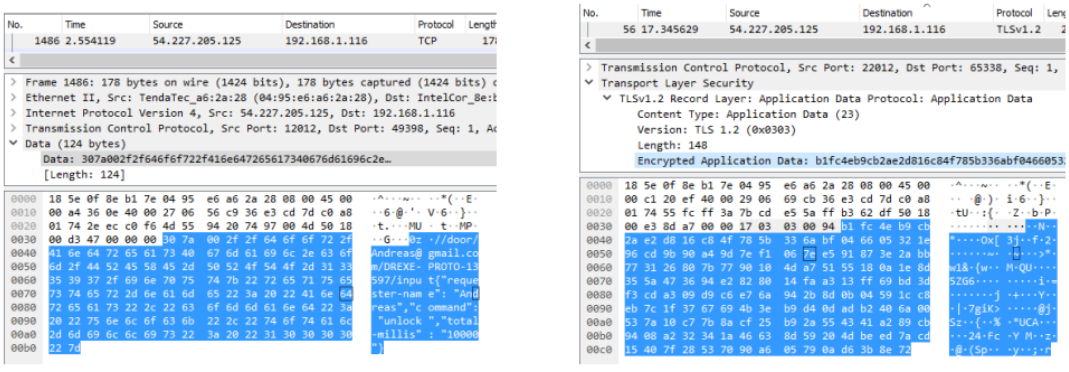


(a) (b)

Figure 2.2.1-3.a: Data of Publish MQTT Without SSL Encryption

Figure 2.2.1-3.b: Data of Publish MQTT with SSL Encryption

Based on the data that used and demonstrate in the project, we can refer to Figure 2.2.1-3.a, which is publish the MQTT without SSL Encryption. We can see some information like: */Door/Andreas@gmail.com/DREXE-PROTO-13597/input* that was capture with Wireshark. The information that was publish was clearly seen and it is vulnerable to sniffing. However, when SSL encryption is implemented, we can see that there were symbols appearing in Figure 2.2.1-3.b, which the message was encrypted. Hence, sniffing data is not allowed.



(a) (b)

Figure 2.2.1-4.a: Data of Subscribe MQTT Without SSL Encryption

Figure 2.2.1-4.b: Data of Subscribe MQTT with SSL Encryption

For Figure 2.2.1-4.a, we can also see the message that was received: */Door/Andreas@gmail.com/DREXE-PROTO-13597/input* captured in the Wireshark. It was same as the publish, when SSL encryption is implemented, the message is encrypted hence it was protected.

2.2.2 Home Security System using IOT and AWS Cloud Services (Mehra at el., 2019)

Based on the title, we can know the following project implemented IoT and Amazon Web Services (AWS) cloud services into home security system. For the system, they are using Amazon Rekognition Algorithm on the face detection. The system will collect the image of the user and store the important features of the image into S3 Bucket. It will use the face that has the largest area in each of the picture. Face details will be recognized such as bounding box around the face, facial landmarks, gender, and the confidence value to know the detection result. Then Amazon Rekognition will use the data and compare with the person standing in front of the door.

If the face match with the data that stored in the S3 bucket, door will be unlocked by the system. The owner will be notified as there is someone enter or wish to enter via the Amazon Simple Notification Services (SNS). If the person is not match with the images in S3 bucket, the user can unlock the door automatically in the future too with the system. On the other hand, if the untrusted person was trying to enter the door, the owner can raise the alarm, which also trigger the system into an emergency state, which all the windows and doors will be locked.

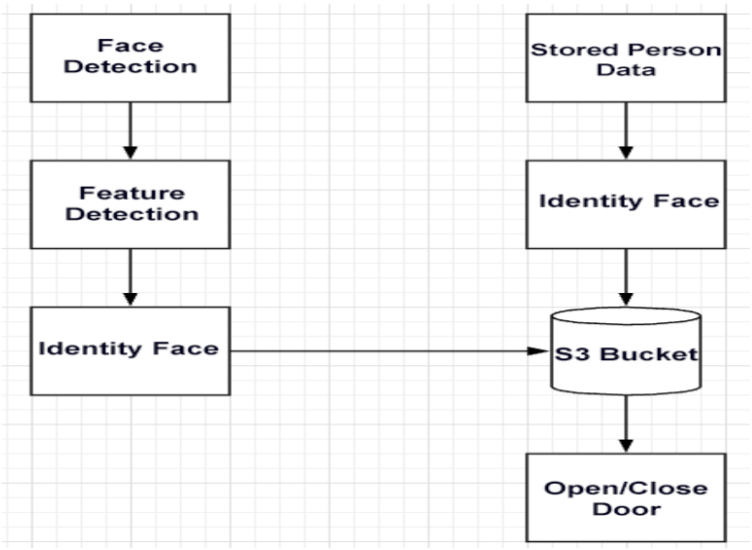


Figure 2.2.2-1: System Flow of the software

For the overall system flow based on Figure 2.2.2-1, an input of image with face is given to the system. Then the system will detect the image and recognize the face details and store in the database. The faces information will then be compared with

existing data in the database. If the person's face does not match the images, notification service will alert the owner. Owner can know the situation from live streaming. Then, the owner can then choose to open the door or choose to turn on the alarm.

For the software part, they use Python as their programming language, Amazon Rekognition API, Amazon S3 Database and Telegram Bot API for mobile/web application.

In the face recognition part, they are based on the Amazon Rekognition, It can analyse any photo that is store in Amazon S3 Bucket. Discovery API is used together with the Amazon Rekognition, as analyse separately the identified appearances which is the Detect Faces and IndexFaces. They will return the metadata that consist of face areas, present point, top-notch appraisals, and highlights. While the IndexFaces will then stores the face trademarks accurately on AWS. Then CompareFaces will try to attempt the 1:1 examination. For each situation, the API will look at the most noticeable data inside the source and test the appearance that appears inside an objective photo. The image of the person standing in front of the door will be captured and uploaded to the S3 Bucket via Internet. S3 Bucket is stored with the images that the person that is recognize, which allow them to unlock the door. The image that just capture will store in the bucket database and if the owner wishes to add them as trusted person they are allowed. The system will check any of the movement of the surrounding area when the owner activates the security system. Any motion that is unexpected, a photo will be captured by the camera. The picture will then send to the owner. The PIR sensor and camera will monitor the change with software-based technology which mention by the author. They mention that using PIR sensor and camera will have a more precise data to the environment of the house.

For the hardware part, they use Raspberry Pi, Raspberry Pi Camera, Passive Infrared (PIR) Sensor, Web Camera and Solenoid Lock. Raspberry Pi is cheap, light weight device that can connect to a display, mouse, and keyboard. It is the command centre to the whole system. They are using an 8-megapixel Raspberry Pi Camera Module. PIR sensor is used to detect an object motion pass through the system. Then for the Web Camera, they use it to broadcast video onto the Web. Lastly, the solenoid lock is a remote door locking mechanism to lock and unlock the door.

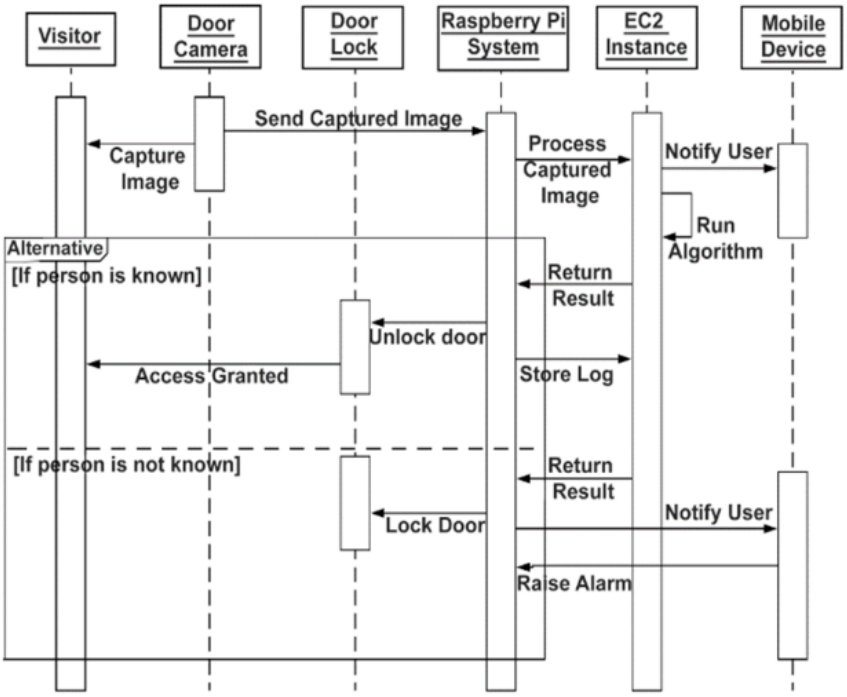


Figure 2.2.2-2: System Analysis Diagram

Raspberry Pi is the bridge in between the hardware sensor and the software. Based on Figure 2.2.2-2, When a Visitor is at the footstep, the Door Camera will capture an image of the visitor and send the image to the EC2 Instance through Raspberry Pi. The Owner will be notified. After EC2 process the image, it will send back the result to the Raspberry Pi. If the Visitor is recognised, the door will be unlocked, the visitor will be allowed to enter through the door. A log will be recorded in the S3 Bucket. If the visitor is not recognised by the system, the door will be locked. Owner will be notified, and the owner can choose to make the visitor as trusted face, the record will be saved in the S3 Bucket and allowed to enter. The owner can also choose to raise to alarm.

2.2.3 An Advanced Internet of Things based Security Alert System for Smart Home (Tanwar et al., 2017)

According to this project based on the report, the approach of the system is an intrusion detection system that detect inactivity inside an area, and it will notify the owner whenever there is any intruder enter.

In this project, the hardware they are using is Raspberry Pi Model B, PIR Sensor and security camera. Raspberry Pi Model P is the overall computing devices and the PIR Sensor and security camera were connected to the Raspberry Pi. The sensor and the camera will send signal to the Raspberry Pi when intrusion is detected. Raspberry Pi not only receiving the signal from the sensor and the camera, it also processes the data from the sensor and the image from the camera and send an email to the homeowner by using email. The connector that used in the Raspberry Pi will be the USB port and the GPIO pins as connector. For the security camera part, they used a Web camera as their camera which capture the image of inactivity happened in the home. It works together with the PIR Sensor instantly, they will send signal and captured image to the Raspberry Pi. Other than that, PIR sensor is a sensor that using infrared to detect object at their range. It can be a motion detection sensor as it detects object over the sensor range; whenever there is any inactivity, it will be trigger and send signal. The benefits of PIR sensor is it can also work in darkness which the camera does not work the same way. It increases the security of the system instead using the camera for detection.

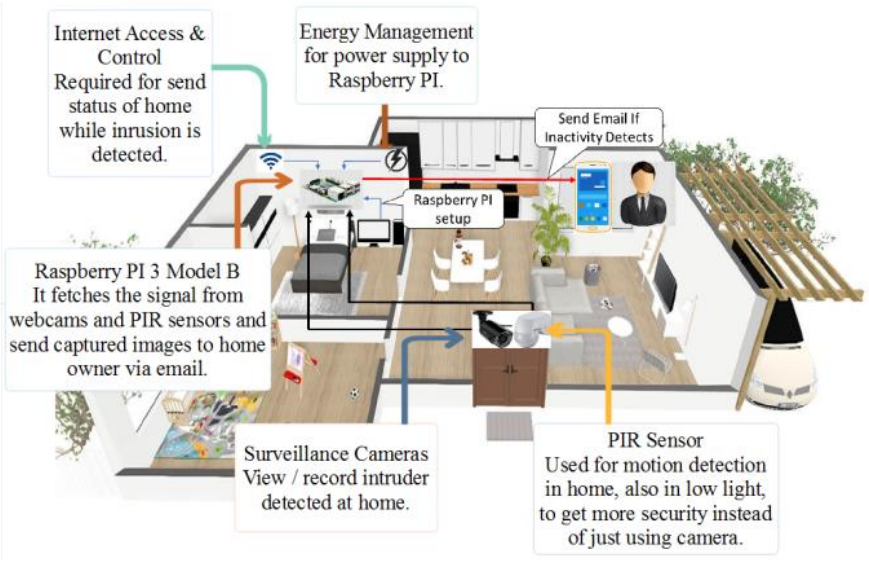


Figure 2.2.3-1: Overall System Configuration

Based on Figure 2.2.3-1, the overall system configuration, the Raspberry Pi will receive the signal from the PIR sensor and surveillance camera. If the sensor detects intrusion, signal and image will send to Raspberry Pi, Internet Access & Control will be required to the send the status of home then an email will be sent to the user. The following scenario can refer to Figure 2.2.3-2 below. When the intrusion was detected by the sensor and the camera, the inactivity signal will then send to the Raspberry Pi. Then the Raspberry Pi will send an email to the user.

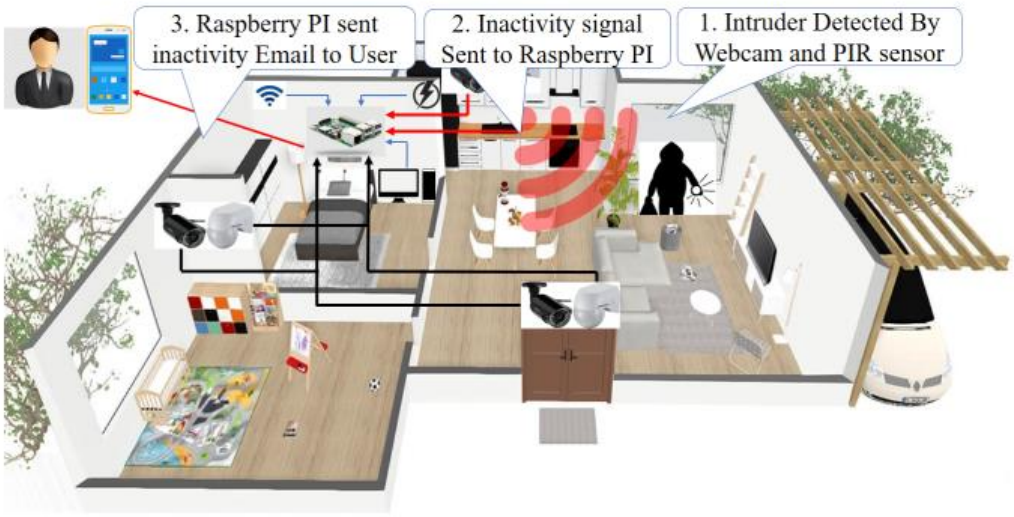


Figure 2.2.3-2: Intrusion Detection system works

For the algorithm part, the first signal that received is the signal from the PIR sensor from the GPIO pin that connected to the Raspberry Pi. If the previous signal and the current signal are same then there is no interruption hence, it will exit from algorithm, If the previous signal and current signal is different, the camera that connected will be started this is because the different signal indicates that there is intruder inside the home. The camera will capture the image and store in the temporary storage available in the system. Lastly an email will be sent to the homeowner.

Figure 2.2.3-3: Intruder detected by the system

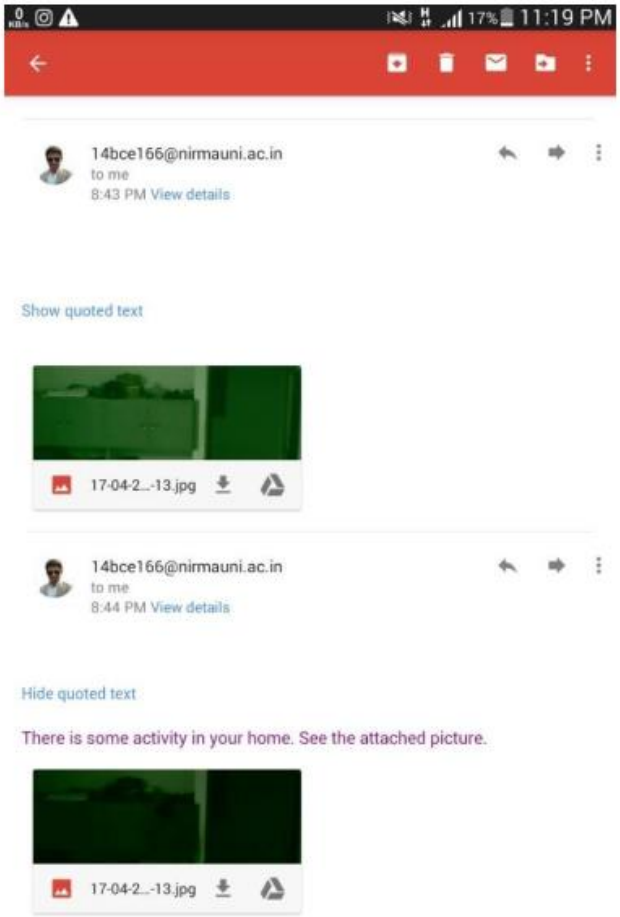


Figure 2.2.3-4: Email sent by the system in the project

2.2.4 Summary of the Existing Systems

For the exiting systems for this project, there are 3 paper have been reviewed. The following are the title of the Paper and it summarize in the Table 2.2.4.

Paper 1: Door Security System for Home Monitoring Based on ESP32 (Andreas et al., 2019)

Paper 2: Home Security System using IOT and AWS Cloud Services (Mehra at el., 2019)

Paper 3: An Advanced Internet of Things based Security Alert System for Smart Home (Tanwar et al., 2017)

Table 2.2.4: Summarization of the existing systems

Paper	Advantages	Disadvantages	Critical Comments
1	- Secure by using SSL Encryption while publishing and subscribing MQTT - User friendly by having their own mobile application.	- False alarm might triggered when door intrusion is detected.	- Will cause unnecessary panic to the user when the intrusion is detected.
2	- User able to view the situation through live streaming. - Using face detection and face recognition system. - User allow to choose untrusted user pass through or choose to trigger the alarm.	- Expensive, need to have subscription with AWS. - Untrusted user who can enter once, their face data will save in the database. They can pass through next time.	- Face recognition through normal camera can tricked by just using 2D photo.

3	- Intrusion detection system which capture the intruder silently. - Camera equipped with night mode.	- Notification is sent to the email. - No alarm for warning the intruder which can trigger by the user.	- Mainly dependent on the PIR sensor. If it is blocked, the sensor will be unable to detect the inactivity.

From Table 2.2.4 above, we can see that most of the security system are implemented on the door security. Door security system is used at the door which it is unable to detect other intrusion if the intrusion is not entering from the door. Moreover, door security system is much easier to be destroyed by the intruder. When the door security is not working. There is a chance for the intruder to enter to the area.

In Door Security System for Home Monitoring Based on ESP32, it implemented SSL to secure the data that are publishing and subscribing to the MQTT which is a good for data credibility. However, the disadvantage of the system is the alarm system will be triggered whenever there is intrusion happened. It will cause unnecessary panic to the users.

For Home Security System using IOT and AWS Cloud Services, it allows the user to have a live view of the situation through streaming. They also let the user choose to allow untrusted face to enter. It is connected to the AWS, which the face recognition and face detection will be happened in the AWS cloud. But on the other hand, AWS is much more expensive compare with other free source system available in the market right now. Moreover, face detection and face recognition will be tricked by the 2D photos. The system has another disadvantage which the untrusted users' face data will be stored, and they can enter next time.

Lastly, in An Advanced Internet of Things based Security Alert System for Smart Home which is having different approach with the other two system, is using a Intrusion Detection Security System which installed inside instead of outside. In this system, they are using camera to silently capture the intruder if intruder detected. It is using a camera with night mode turn on so that it can be working fine in the dark. However, the system is mainly dependent on PIR sensor, which means if the sensor is blocked or malfunction, the whole system will not be working. Other than that, the system is using email to send notification to user.

CHAPTER 3 SYSTEM METHODOLOGY

3.1 System Development Models

3.1.1 System Development Model 1: Waterfall

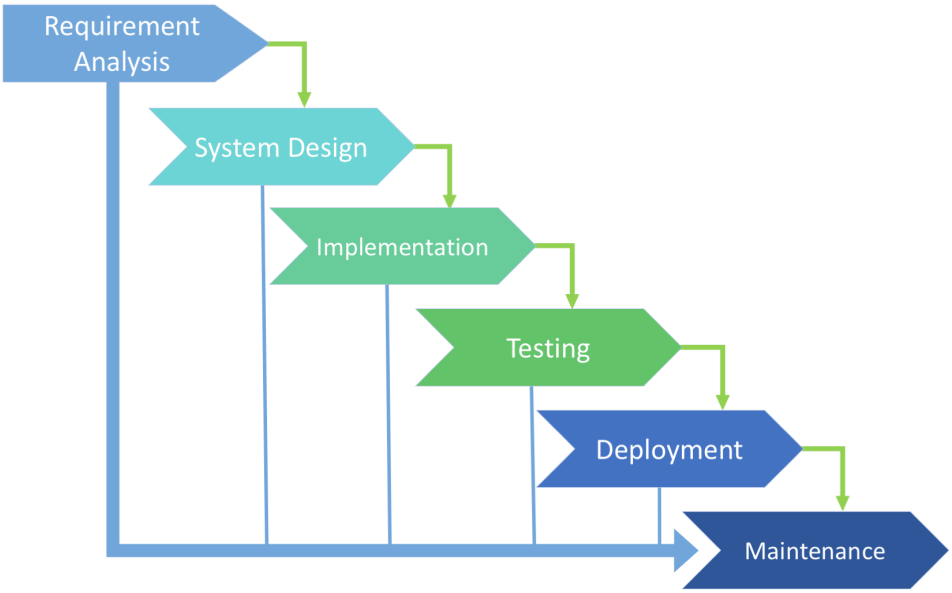


Figure 3.1.1: Waterfall System Development Life Cycle (SDLC) Model.
(SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral, 2017)

Waterfall model is a cascade SDLC model which emphasizes logical progression throughout the SDLC. In waterfall model, it is moving steps by steps like an incremental waterfall through the phases of analysis, design, implement, testing, establish and supporting. Each phase will be separated, while the outcome of the phase will be the progress of next process.

Waterfall model is easy to manage because every phase of the process has a define result and review. Moreover, since the model is simple, it is easier to determine the key points during the cycle. It is easier to assign task and more suitable for small to mid-size project when the requirements of the project are clear.

However, waterfall model is following steps by steps. To start with a new process, we need to finish the last phase. This also means that the software or the system will ready only after the last phase is over. We cannot identify the problem in advance as the integration is done at the very last end, so it is higher risk.

3.1.2 System Development Model 2: Agile

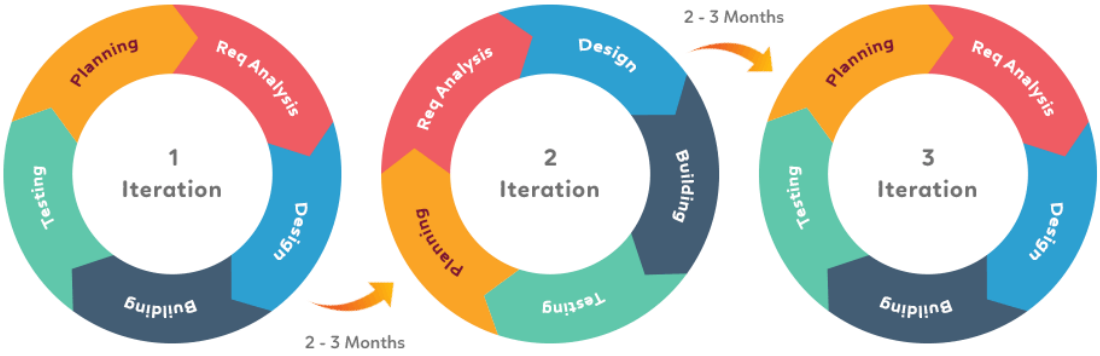


Figure 3.1.2-1: Agile SDLC Model

(agile vs waterfall | Hiteshi, n.d.)

Agile SDLC is based on adaptive software development methods. In Agile development life cycle, tasks are separated into small time frames to deliver small incremental build for a release. Each time frames will be fix with certain duration (weeks or months) with a list that need to be completed in this period.

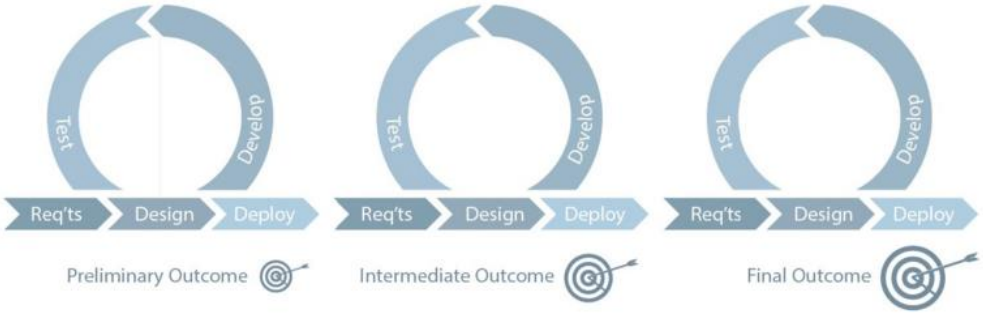


Figure 3.1.2-2: Each iteration with one outcome.

Agile SDLC break task into smaller iterations, it will get an outcome after each iteration. Short iteration can minimize the risk and functionality can be developed more rapidly. Clients can also learn product at the early stages and provide changes to the system. Other than that, Agile SDLC only requires initial planning to start a project.

For disadvantages of agile, it might have chance to exceed the expect time with all the changes that need to make. It is having a bad resource planning, as changes happen frequently.

3.1.3 System Development Model 3: Spiral

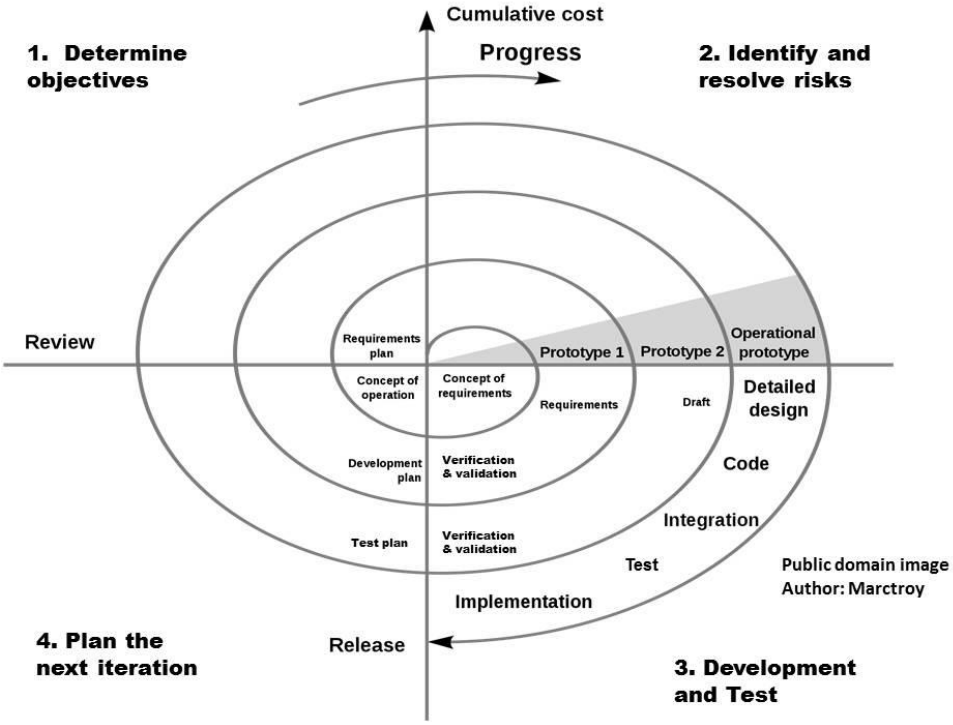


Figure 3.1.3: Spiral SDLC Model

(Jr, 2021)

In spiral SDLC Model, it will be separated into 4 part, which is:

1. Determine objectives
2. Identify and resolve risks
3. Development and Test
4. Plan the next iteration

Based on the Figure 3.1.3 above, we can see that the spiral model is starting from determine objective then proceed to identify and resolve risks, continue with development, and test and lastly plan the next iteration. When the project went through 4 phases that mention above, it will proceed back to determines objective part. It shows the characteristic of iterative model and waterfall model as the project is break into small pieces only.

With the spiral model, the lifecycle of the project is separated into a smaller part. It also allows to make changes or add new function in the system as every new phase can plan a new iteration. Moreover, the prototype of the project can be shown in earlier stage as it has development and test in each phase. However, due to the model with huge number of stages, it is not effective to small projects and it requires of huge amount of documentation.

3.1.4 Selected Model

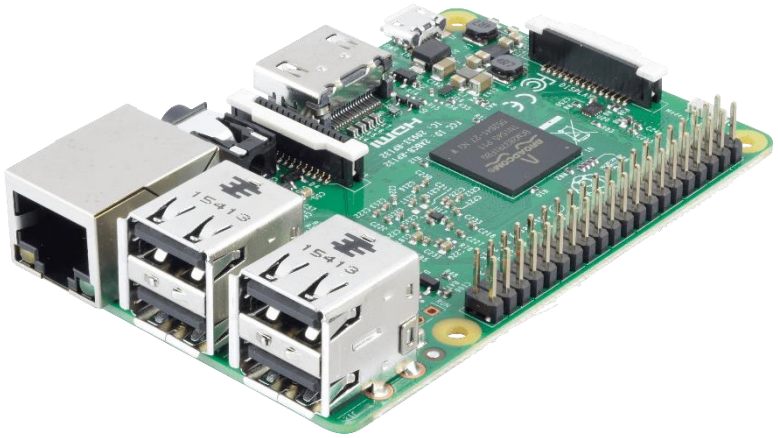
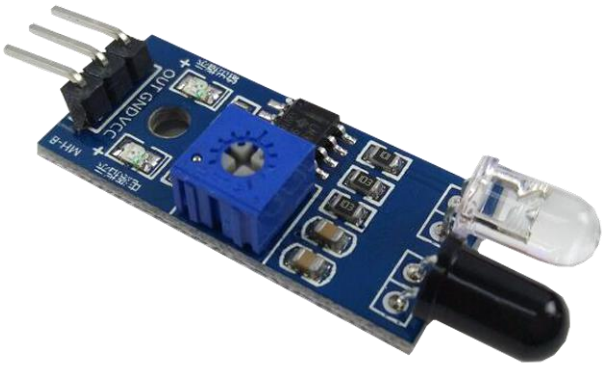
After comparing with both waterfall model, agile model, and spiral model agile model is much more suitable, and it will be the selected SDLC model for this project.

In this project, the risk of failure of the system need to be minimize. So, for every iteration, the system will be tested. Other than that, in this project, the system can break into small incremental build by building, deploy and testing each of the small build one by one. It allows the system to make changes if the feature that developed not reaching the expectations. Moreover, module and more functionality can be added in later if there is needed to make the system to works more fluently.

3.2 System Requirement (Technologies Involved)

To complete the project there are some hardware and software component involved. Following Table 3.2-1 and table 3.2-2 indicates the Hardware Components and Software Components involved in the system.

Table 3.2-1: Hardware Components

Hardware Components	
Raspberry Pi3 B+ Quantity: 1	
Infrared Sensor Module Quantity: 2	
Door Window Sensor Magnetic Switch Quantity: 1	

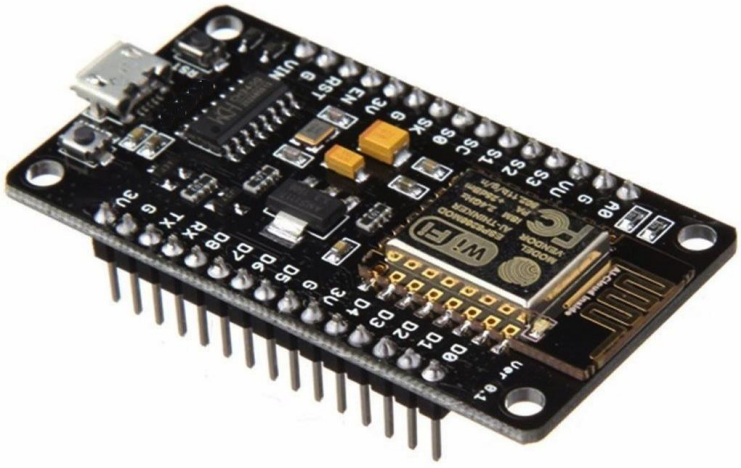
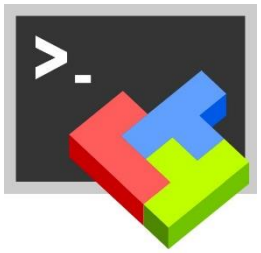
Raspberry Pi Camera Module Quantity: 1	
NodeMCU Lua V3 ESP8266 WIFI with CH340C Quantity: 1	

Table 3.2-2: Software Components

Software Components	
Raspberry Pi OS	

MobaXterm	
Ubidots	
Telegram	

Based on the Table 3.2-1 and Table 3.2-2, which is the hardware and software component that will be involved in the project. Raspberry Pi3 B+ is the central control unit for the Raspberry Pi Camera Module, 2 Infrared Sensor Module and the Magnetic Switch Sensor (Door Window Sensor Magnetic Switch). The Magnetic Switch Sensor is connected to the NodeMCU Lua V3 ESP8266, it will send the signal to Raspberry Pi3 B+. For Raspberry Pi3 B+, it is the power supply for all the IR sensor and acts as the communication tools with other devices through cloud and Telegram.

In the Raspberry Pi OS, it allows use to run python script to control the sensor and read the reading from the sensor. For MobaXterm application, it is installed in the Personal Computer (PC) to allow us to view the GUI from Raspberry Pi3 B+ via a SSH connection without any HDMI capable monitor. Ubidots is used to store the data that received from the Raspberry Pi3B+. While for Telegram, it is used to receive the data from the Raspberry Pi3+ and notify the users when there is data coming in.

3.3 Functional Requirement

3.3.1 Detect abnormal activities

The function of the magnetic switch and the IR sensor is to detect the abnormal activity in the indoor area. The magnetic switch will use to detect whether the door is open or not. When the magnetic switch is disconnected, it means the door is opened. Hence, the signal that detected by Raspberry Pi is different and the camera will be turned on. For the IR sensors, they are using to detect whether the object is coming in or going out so that the user will know whether there are people coming in or out.

3.3.2 Photo Capturing

Whenever there is an abnormal activity happens in the room, camera must be captured. Photo is an evidence or a trace back for the user when there is intrusion happened in the room. Photo that captured can be used for other purpose such as object detection or for user to look back.

3.3.3 Reduce false alarm

To reduce false alarm, object detection is added into the system. Object detection is using the camera sensor to monitor the area. When there is a person pass by, the object detection algorithm will identify whether it is a person or not a person. If it is a person, it will only notify the user that a person is detected; else user will not be notified.

3.4 Project Milestone

Following is the project timeline for the development of the system shown in Table 3.4-1 and project milestone in Table 3.4-2.

Table3.6-1: Project Timeline for FYP 1

Project Task	Project Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Determine System Requirements														
Preparing Hardware														
Preparing Prototype														
System Testing														
Finalize Prototype and Report														
FYP 1 Documentation														
FYP 1 Presentation														
Meeting with Supervisor (*)														

Table3.4-2: Project Timeline for FYP 2

Project Task	Project Week													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Knowledge Research														
Purchasing Extra Equipment														
Implement ESP8266 Node														
Implement Telegram														
Implement Object Detection Module														
Preparing Prototype														
System Testing														
Finalize Prototype and Report														
FYP 2 Documentation														
FYP 2 Presentation														
Meeting with Supervisor (*)														

* Meeting with Supervisor to update the progress and discussion on the project direction.

3.5 Estimated Cost

Items	Quantity	For FYP Development	Quantity	For Commercialization
Raspberry Pi3 B+	1	RM 156.00	5	RM 156.00
MicroSD Card (16GB)	1	RM 23.00	5	RM 23.00
Raspberry Pi Camera Module	1	RM 40.00	5	RM 40.00
Infrared Sensor Module	2	RM 3.80	10	RM 3.80
Door Sensor Magnetic Switch	2	RM 4.50	10	RM 4.50
NodeMCU Lua V3 ESP8266	1	RM14.90	2	RM29.80
Raspberry Pi OS	-	FOC	-	FOC
MobaXterm	-	FOC	-	FOC
Ubidots	-	FOC	-	FOC
Telegram	-	FOC	-	FOC
		RM 2461.70		RM 1207.80

Table 3.5: Estimated Cost for FYP Development and commercialization

The estimated cost for commercialize which include so many components is for home system which include 1 Living Room, 1 Kitchen and 3 Bedroom which follow a normal apartment setting. While for Door Sensor Magnetic Switch in "For Commercialize" are also including the Magnetic Switch for the window each for every room area.

3.6 Concluding Remark

In summary, this chapter discussed the system development model that chosen, the technology involved in the system, functional requirement, testing on the component, milestone of the project and the estimated price of the project. System testing was carried out to test the IR sensor, camera and object detection that will be used in the system. Gantt chart shows the project timeline and project milestone which the time that used during the FYP 1 and for FYP 2.

CHAPTER 4 SYSTEM DESIGN

4.1 System Architecture

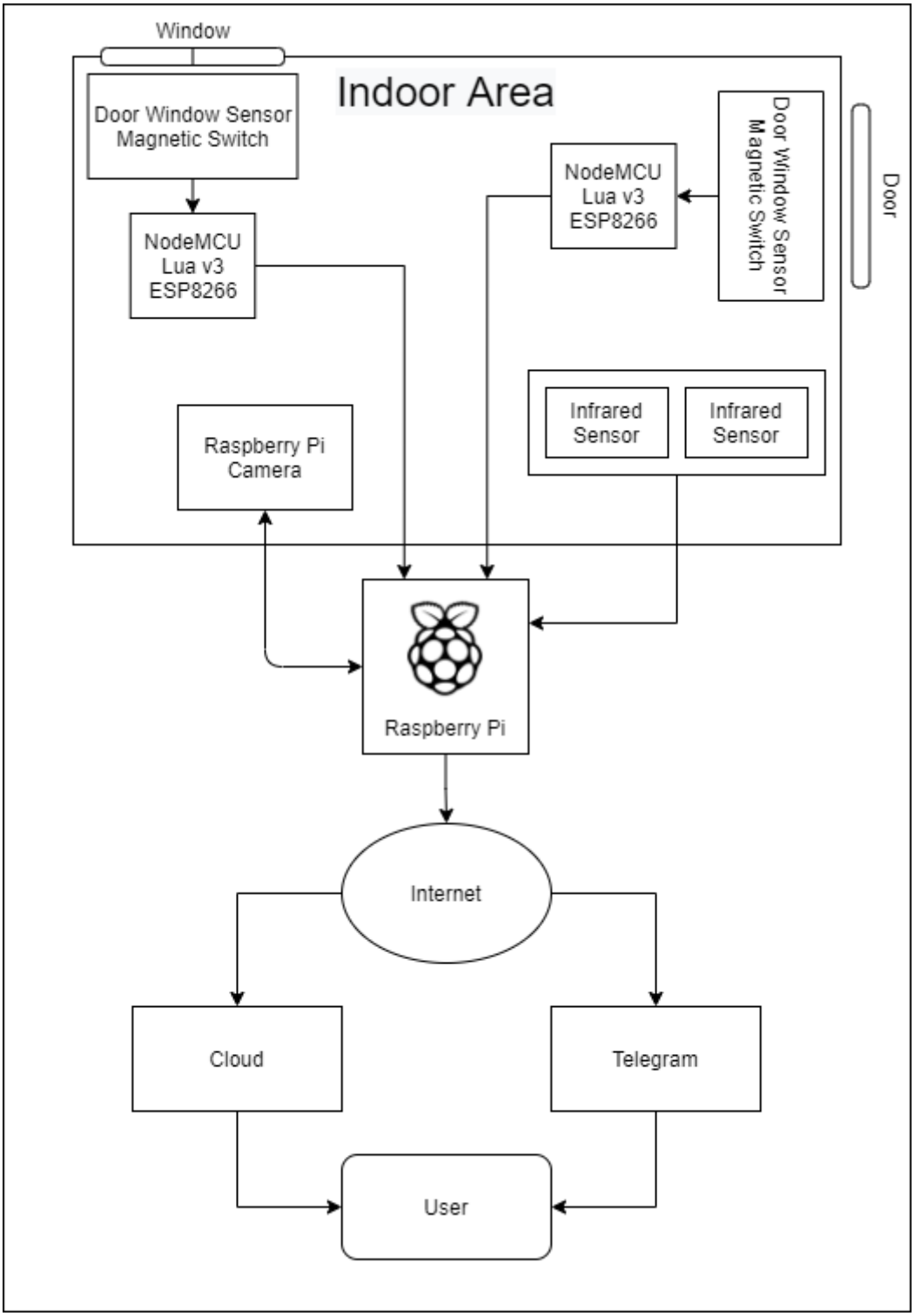


Figure 4.1: System Architecture of the System

In the system shown in Figure 4.1, there are some of the sensors used such as door window sensor magnetic switch, infrared sensors, and raspberry pi camera. Door Magnetic Sensor is connected to the NodeMCU Lua v3 ESP8266, while the infrared sensors and raspberry pi camera is directly connected to the Raspberry Pi. NodeMCU Lua v3 ESP8266 is wirelessly connected to the Raspberry Pi. The data that collected by the sensor will then transfer to Raspberry Pi which is the computing devices of the system.

When the door window sensor magnetic switch is triggered, the NodeMCU Lua v3 ESP8266 will send different readings to the Raspberry Pi. The Raspberry Pi will be turned on the camera and capture a photo. Then the photo that captured will then analyze by the Object Detection Algorithm (TensorFlow) to detect whether there is “person” detected before sending the information and notification to the user through Telegram. All the data that collected from the IR sensor and the door window sensor magnetic switch will also sent to the cloud (Ubidots) to allow the user to monitor the data and stored the data in the cloud.

4.2 Functional Modules in the System

4.2.1 Door Window Sensor Magnetic Switch Module

In this module, it is responsible of sending the signal from NodeMCU Lua v3 ESP8266 to Raspberry Pi. Raspberry Pi will keep on read and receiving the signal from the Door Window Sensor Magnetic Switch. When the receiving different value from the NodeMCU Lua v3 ESP8266, it will trigger the camera to capture a photo of the situation.

4.2.2 Infrared Sensor Module

Two infrared sensors were used in this module to detect the movement of the object moving by. Raspberry Pi will keep on reading from the sensor, when there is an object pass by the sensors, it will record the time and compare the time to indicate whether the object is “Coming In” or “Going Out”. This module is allowing the system and user to understand the movement of the person that is going in or going out.

4.2.3 Object Detection Module

Object Detection is using TensorFlow to identify the object that appears in the photo that captured. This module is to prevent the system to raise alert to the user although there is door accidentally opened. When there is a “person” detected, the object detection will then print out “Person is detected.”

4.2.4 Notification Module

In this module, it will collect the data from the Door Window Sensor Magnetic Switch Module. When the reading is different, the module will trigger the camera and capture a photo for the object detection. Notification will then send to the user’s Telegram mobile application.

4.2.5 Cloud Module

In this module, the user can view the data at the dashboard from Ubidots. through the web browser. The data that collected from the two sensors, which is the Infrared sensors, and the Door Window Sensor Magnetic Switch will be uploaded to the dashboard. The data will be stored in Ubidots for future reference when the user wish to look back the record.

4.3 System Flow

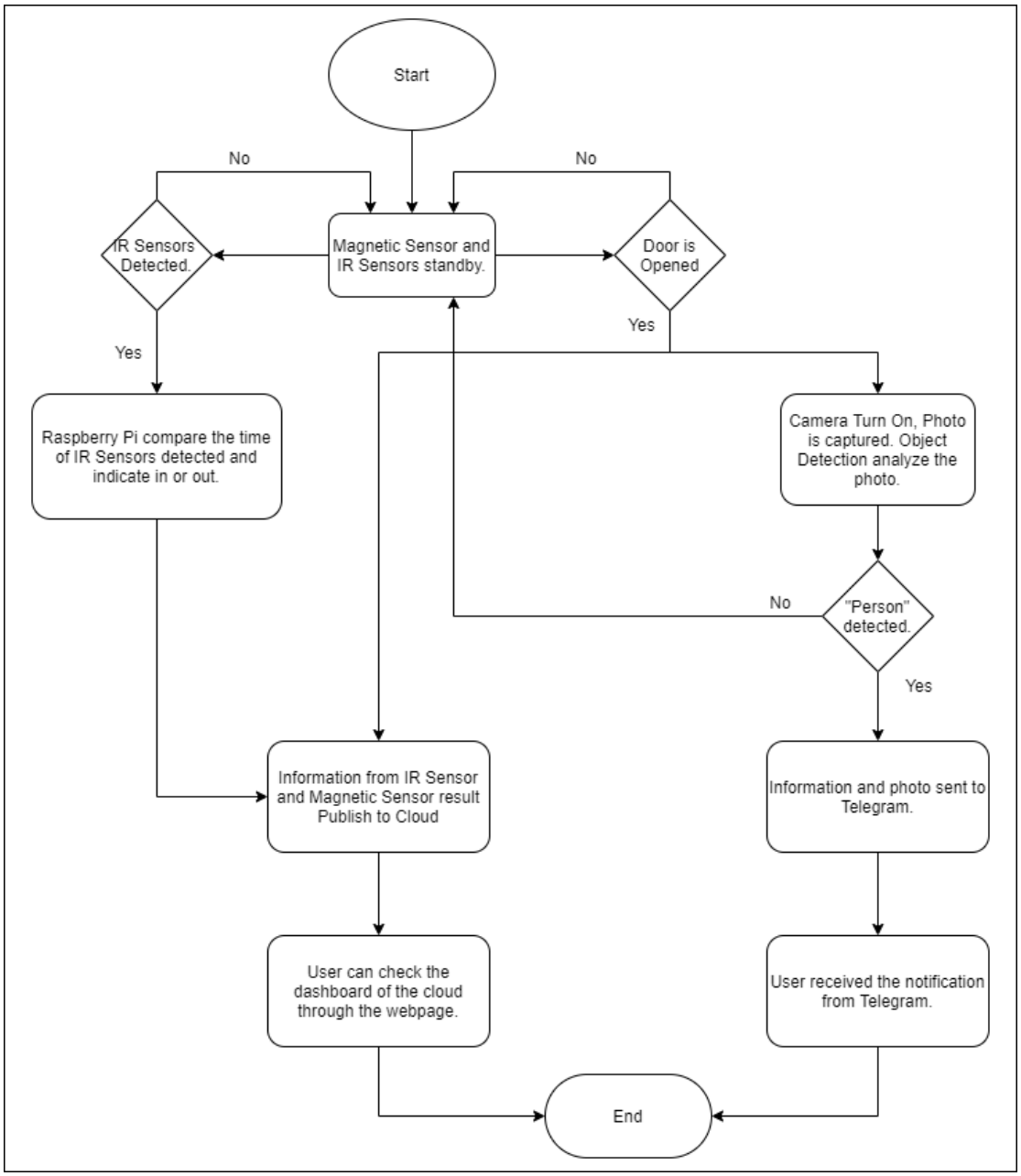


Figure 4.3: Flow Chart of the System

When the system is started, the Door Window Sensor Magnetic Switch and the IR Sensor will be standby to detect the surrounding. When both IR Sensors is detected, Raspberry Pi will compare the time recorded that the IR Sensor detected. It will indicate whether there is object going in or out. Then it will publish the information to the cloud.

For Door Window Sensor Magnetic Switch, if the sensor is separated, the door is opened. The NodeMCU Lua v3 ESP8266 will send the signal to the Raspberry Pi wirelessly to indicate that the door is opened. When the door is opened, the camera will

be started, a photo will be captured, and object detection will analyze the photo. At the same time, the data will also be uploaded to the cloud.

When the object detection analyzes the photo and detects the surrounding from the camera view with a “Person”, the information and the photo will be sent to the Telegram. The user will receive the notification and the photo at the Telegram application. User can use the browser to login into the Ubidots to view the data that uploaded before.

4.4 GUI Design

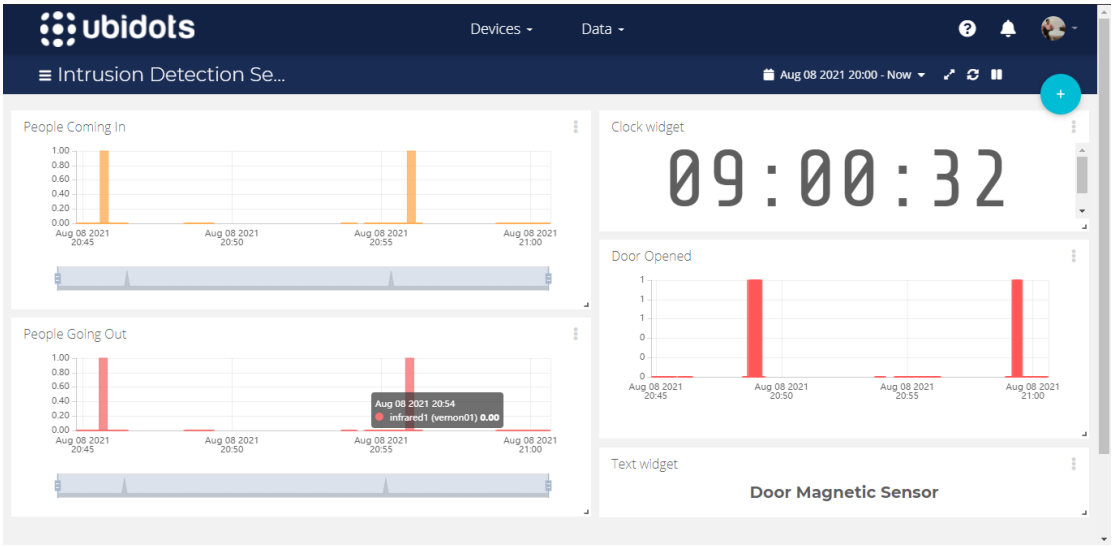


Figure 4.4: GUI Design for Cloud

For the Cloud (Ubidots) Graphic User Interface, it will display 4 information which is the time, “People Coming In”, “People Going Out” and “Door is Opened”. The user can login into the cloud dashboard, which is the Ubidots to view the data uploaded. These data allow the user to understand that whether there is intrusion or not.

4.5 Concluding Remark

In this chapter, it shows the system architecture of the system, determine the functional modules of the system, system flow and the GUI design for the system. System Architecture shows how the system is linked, while the functional modules are the function that broke down from the system. System flow indicates how the system goes and what is the following steps in the system. GUI designs show the design that related to the data presentation to the users.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Hardware Setup

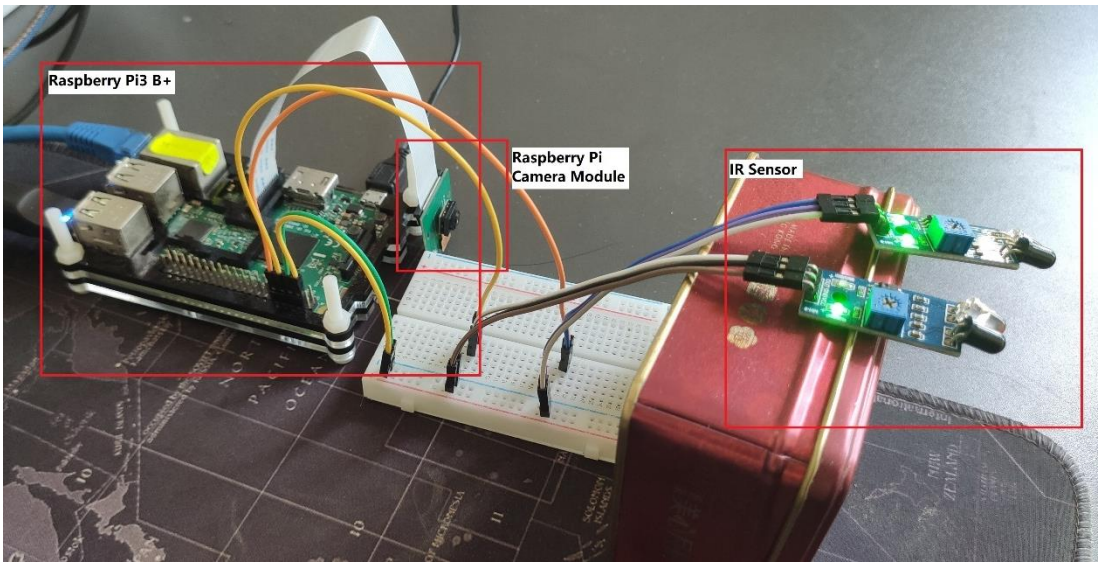


Figure 5.1-1: System Protype on Raspberry Pi3 B+

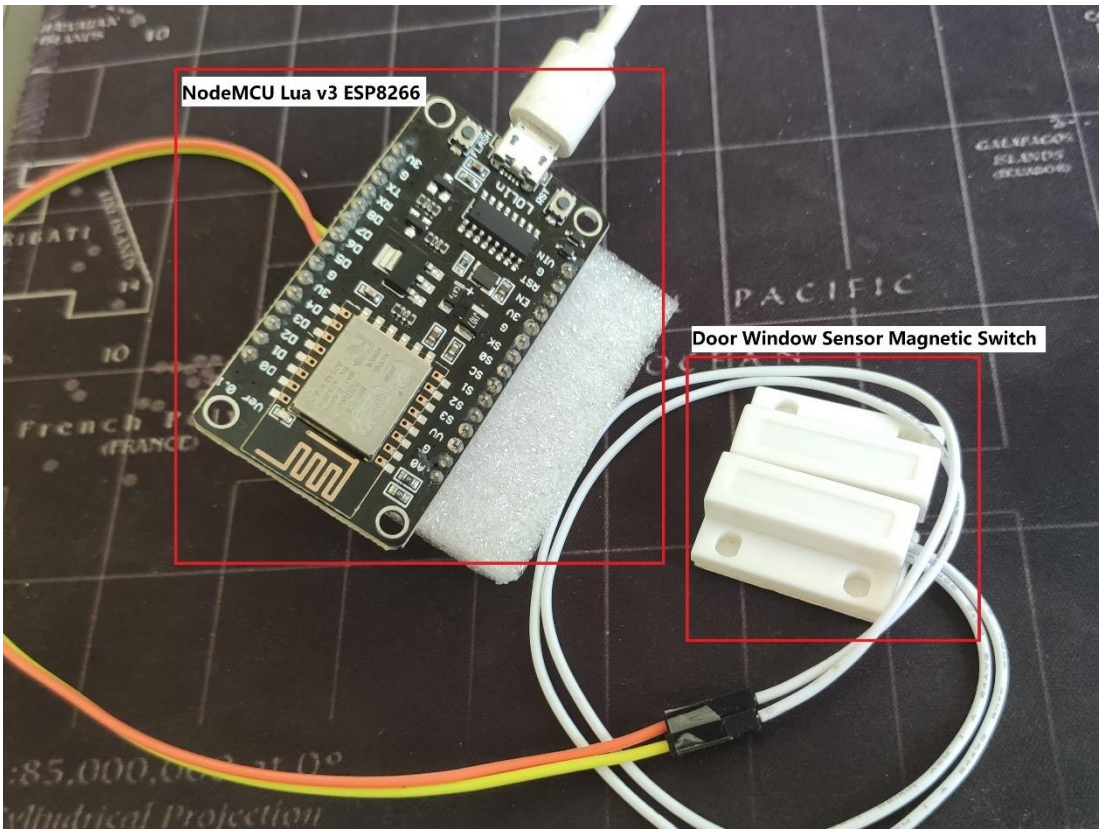


Figure 5.1-2: System Protype on NodeMCU Lua v3 ESP8266

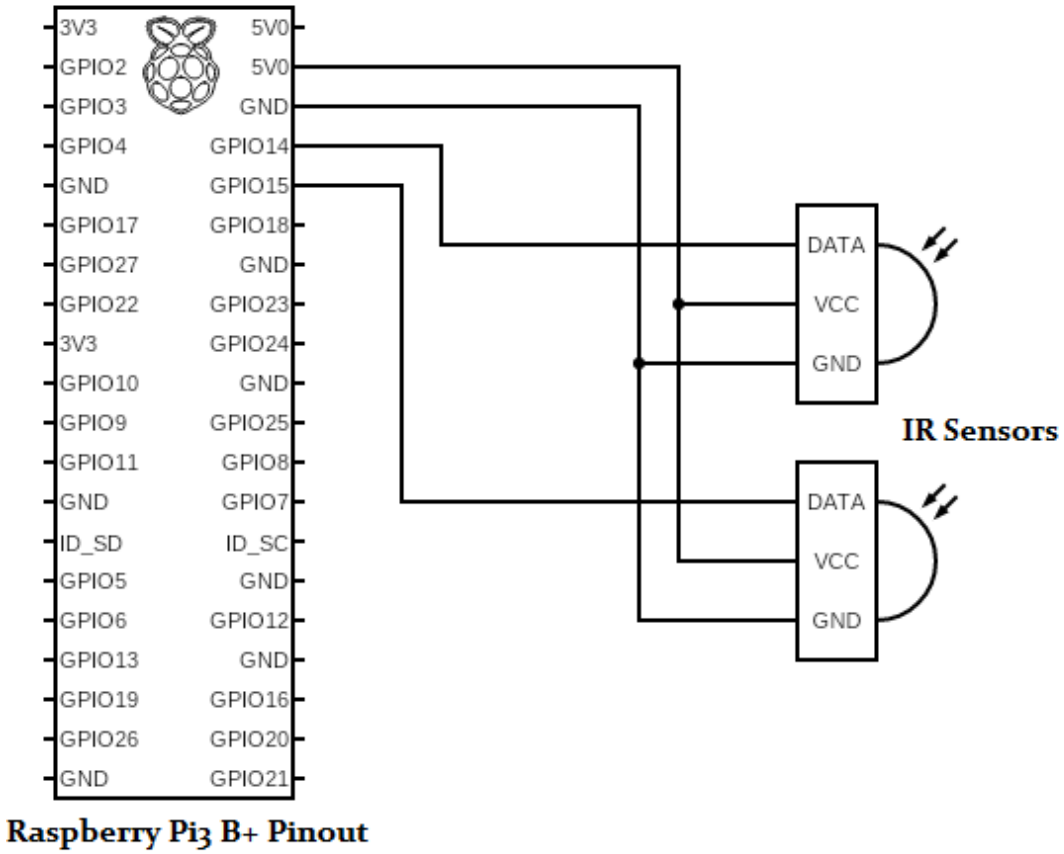


Figure 5.1-3: Wiring Diagram for Raspberry Pi3 B+

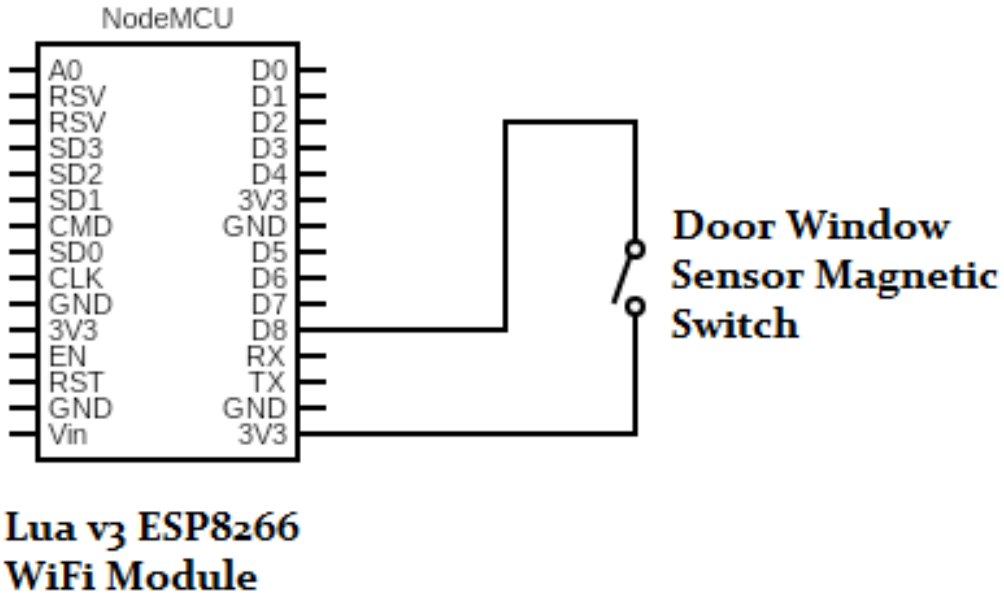


Figure 5.1-4: Wiring Diagram for NodeMCU Lua v3 ESP8266

Table 5.1: Pin connection between the sensor and components.

Component	Pin	Sensors	Pin
Raspberry Pi3 B+	04 (5V)	IR Sensor 1 & 2	VCC
	06 (Ground)		Ground
	08 (GPIO 14)	IR Sensor 1	Data Out
	10 (GPIO 15)	IR Sensor 2	Data Out
NodeMCU Lua v3 ESP8266 WiFi Module	3V	Door Window Sensor Magnetic Switch	-
	D8		

The system prototype is separated into 2 part, one is the main control unit which is the Raspberry Pi3 B+ and the node which is the NodeMCU Lua v3 ESP8266. Based on Figure 5.1-1, we can see that the Raspberry Pi3 B+ is connected with two IR Sensor and one Raspberry Pi Camera Module. Raspberry Pi Camera Module is attached in the Camera Serial Interface (CSI). It is the interface that allow the Raspberry Pi camera to be installed in the Raspberry Pi itself.

For the IR sensor, it is plug into the breadboard. The power supply, ground and GPIO pin from Raspberry Pi3 B+ is connected to the breadboard to supply power and retrieve data from the sensor. From Figure 5.1-3, Raspberry Pi3 B+ is supplying 5V power to both IR sensor and both of the IR sensor is grounded back to the Raspberry Pi3 B+. For the first IR sensor, the data pin is connected with pin 08, which is GPIO14 from Raspberry Pi, while the second IR sensor, its data pin is connected to pin 10, which is GPIO15. When the is object pass by, the IR sensor will have changes in the reading, and it is sent through the pin 08 and pin 10 to the Raspberry Pi.

For the NodeMCU Lua v3 ESP8266, we can refer to Figure 5.1-2. It is connected with the Door Window Magnetic Switch only. It sends the data to the Raspberry Pi wirelessly through the build-in WiFi. The Door Window Sensor Magnetic Switch is a simple switch which connected to the power supply of the NodeMCU Lua v3 ESP8266 and read through the data pin. Based on the Wire Diagram in Figure 5.1-4, one of the wire from the switch connected to the 3V of the Node MCU Lua v3 ESP8266 while another wire is connected to the D8 pin.

5.2 Software Setup

5.2.1 Raspberry Pi OS

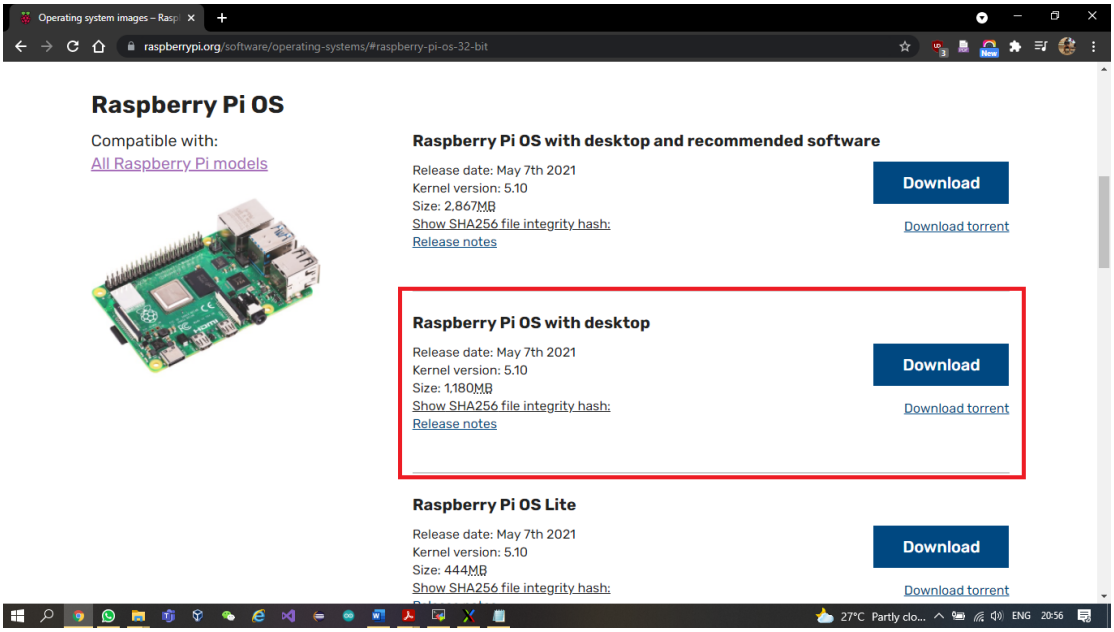


Figure 5.2.1-1: Raspberry Pi OS download page

Raspberry Pi OS is the operating system that needed to flash into the SD so that there is operating system and user interface in the Raspberry Pi hardware. The second option that boxed with the red box was chosen in the Figure 5.2.1-1. After downloaded, then flash the file into the SD card by using the Raspberry Pi Imager as shown as Figure 5.2.1-2.

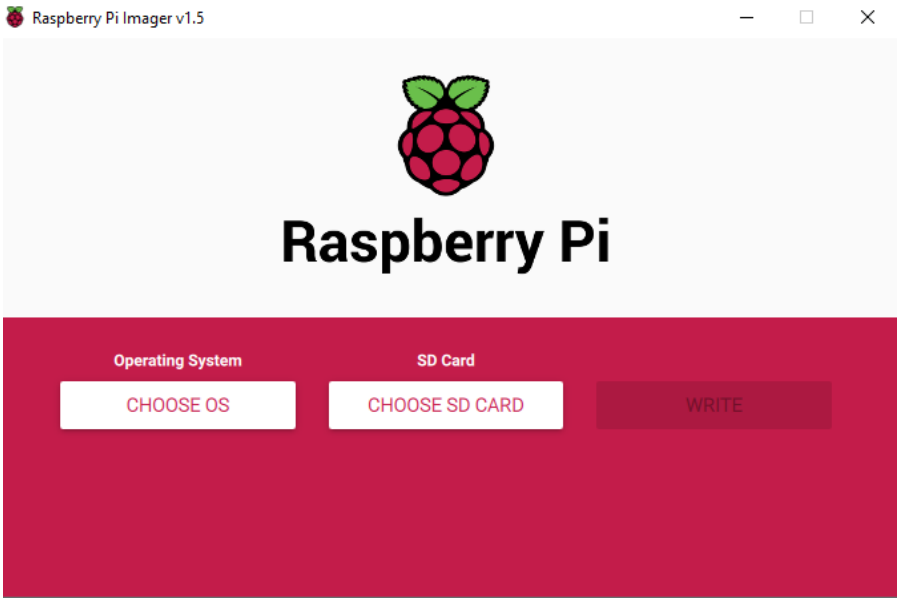


Figure 5.2.1-2: Raspberry Pi Imager

5.2.2 MobaXterm

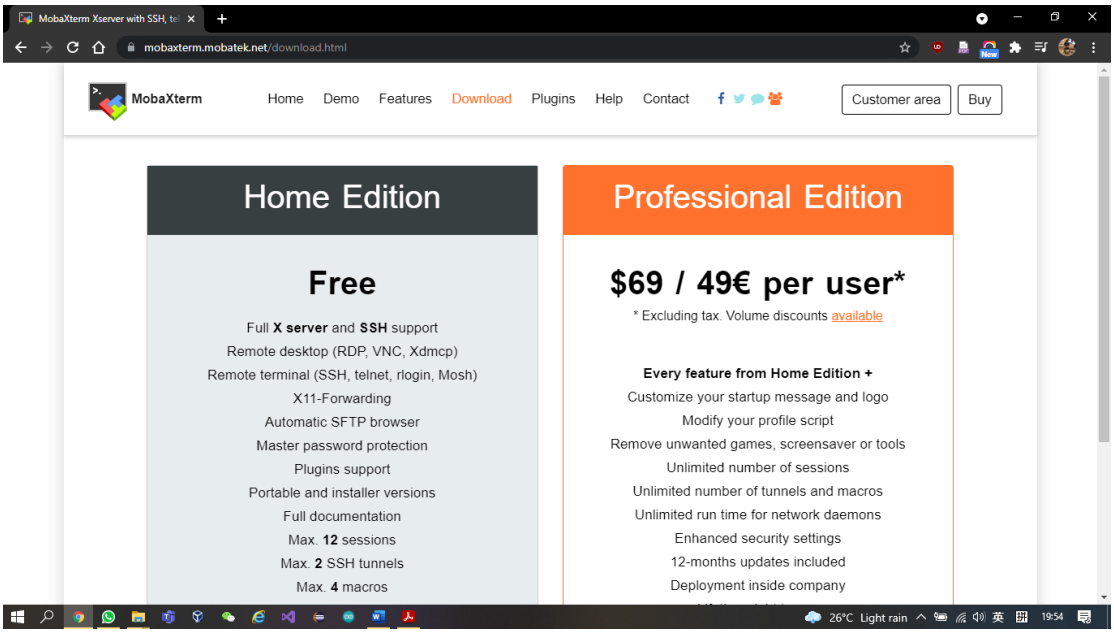


Figure 5.2.2-1: MobaXterm download page

MobaXterm is an alternative way for user to connect to the Raspberry Pi other than using HDMI cable to a monitor. MobaXterm is using SSH connection to allow user to access the Raspberry Pi by just using an ethernet cable and a laptop. In this project, free version of MobaXterm is used. The installation file can get from their official website shown in Figure 5.2.2-1.

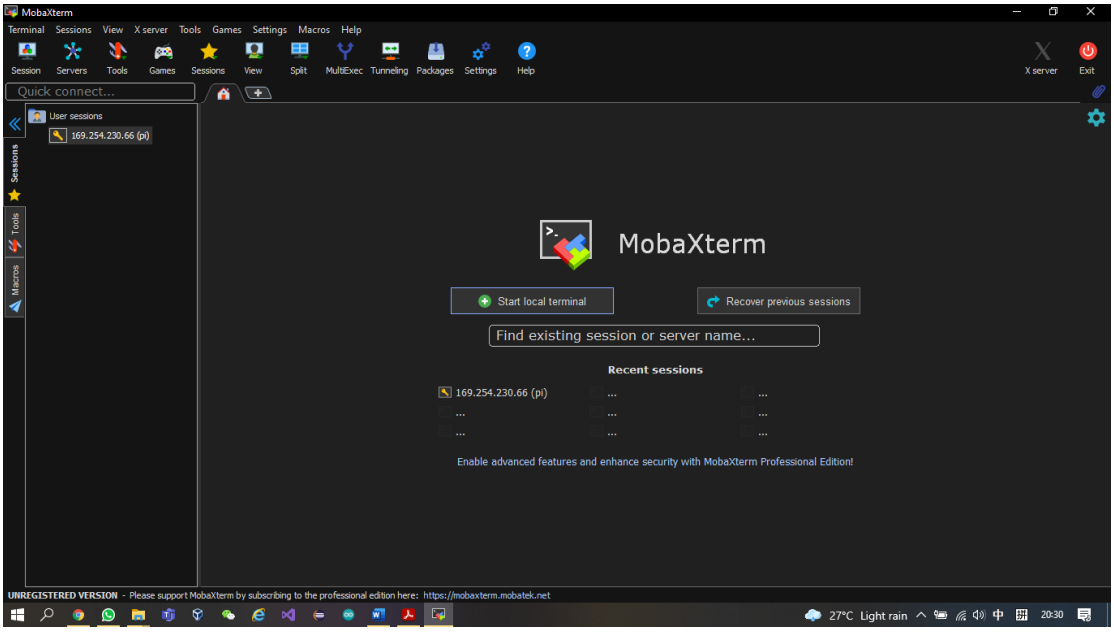


Figure 5.2.2-2: MobaXterm Application Main Page

After the installation, to configure SSH session with Raspberry Pi, set the host with the Raspberry Pi local IP address which can found from the Raspberry Pi itself, username as “pi” and port as 22. When all the setting is configured, connect the Raspberry Pi to the laptop with the ethernet cable, click on the “169.254.230.66(pi)” shown in the Figure 5.2.2.2. After key in the password of Raspberry Pi, type the command “/etc/X11/Xsession”. Then the Raspberry Pi user interface will then be shown on the window.

5.2.3 TensorFlow Lite

TensorFlow Lite is an Object Detection which detect the photo that capture from the Raspberry Pi Camera. TensorFlow Lite that are using follow the script, configuration, and guide from Edge Electronics. To allow the TensorFlow Lite to work in the Raspberry Pi, follow the following steps:

1) Download from GitHub with the following command in the Terminal:
    - *git clone https://github.com/EdgeElectronics/TensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi.git*
2) Rename the file as tflite1
3) Install TensorFlow and OpenCV with the shell script provided by typing the command:
    - *bash get_pi_requirements.sh*
4) Download the Sample Model with the command:
    - *wget {coco_ssd_mobilenet model website}*
5) Unzip the downloaded file into a new folder named Sample_TFLite_Model:
    - *unzip {Model Folder Name} -d Sample_TFLite_Model*

5.2.4 Telegram

To allow the Raspberry Pi to send notification to Telegram, Raspberry Pi need to install Teleport and the mobile devices need to download the mobile application. Teleport allow Raspberry Pi send messages, photos, and videos to the Telegram platform.

To install Teleport in Raspberry Pi, follow the following command:

- *sudo apt-get install python-pip*
- *sudo pip install telepot*

5.3 Setting and Configuration

5.3.1 Firmware for NodeMCU Lua v3 ESP8266

NodeMCU Lua v3 ESP8266 is a node that wirelessly connected to the Raspberry Pi with WiFi connection to send the data of the Door Window Sensor Magnetic Switch. NodeMCU Lua v3 ESP8266 require its firmware before writing the code into it. NodeMCU Flasher Master, NodeMCU Firmware and Arduino IDE has to be downloaded to proceed the configuration for the NodeMCU Lua v3 ESP8266.

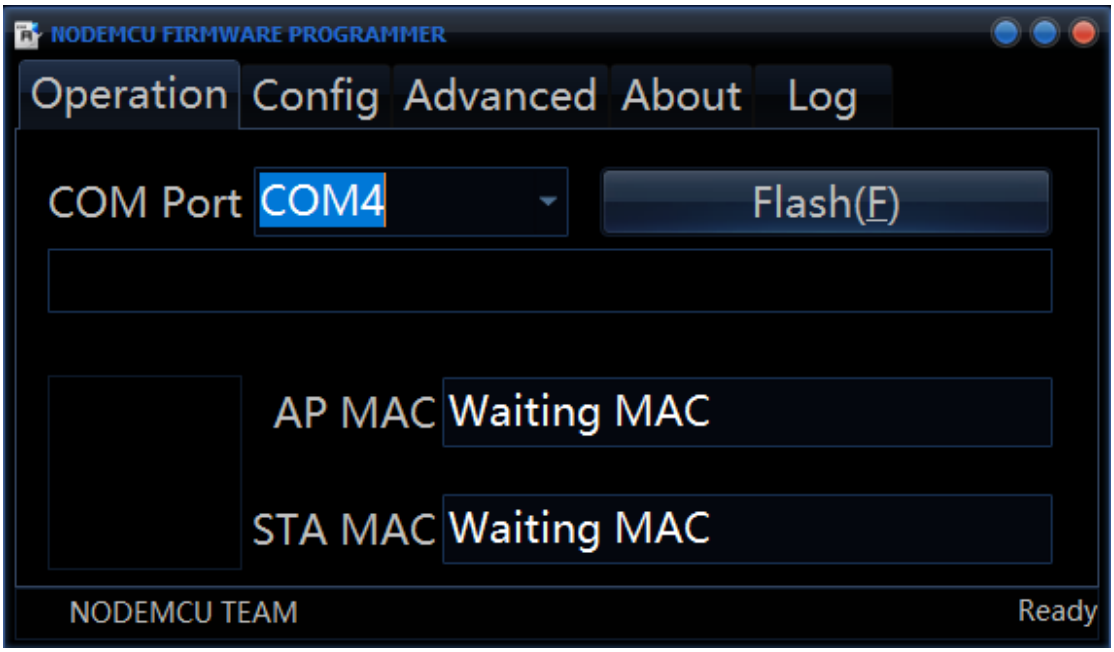


Figure 5.3.1-1: NodeMCU Flasher Master

Firstly, plug in the NodeMCU Lua v3 ESP8266 into the computer with the MicroUSB port. Then check the COM port that the component connected from Device Manager. Then choose the COM port that related as shown in Figure 5.3.1-1. Then go to config tab shown in Figure 5.3.1-2, insert the download NodeMCU firmware into the first line. After that, press the flash button and the flashing process will be started.

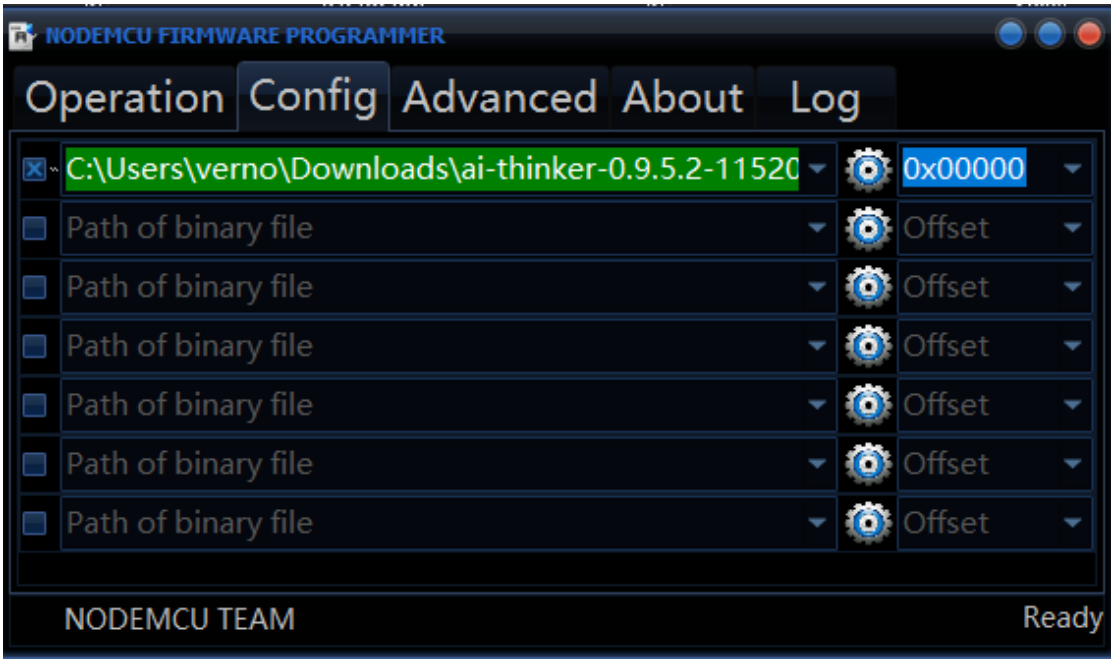


Figure 5.3.1-2: NodeMCU Firmware

After done flashing, open Arduino IDE, click on File > Preferences and insert the link at Additional Boards Manager URLs shown in Figure 5.3.1-3. Then go to Tools > Boards > Boards Manager, type esp8266 in the search box and install it. Before code the NodeMCU Lua v3 ESP8266, go to Tools > Boards, then select NodeMCU 1.0 (ESP-12E Module).

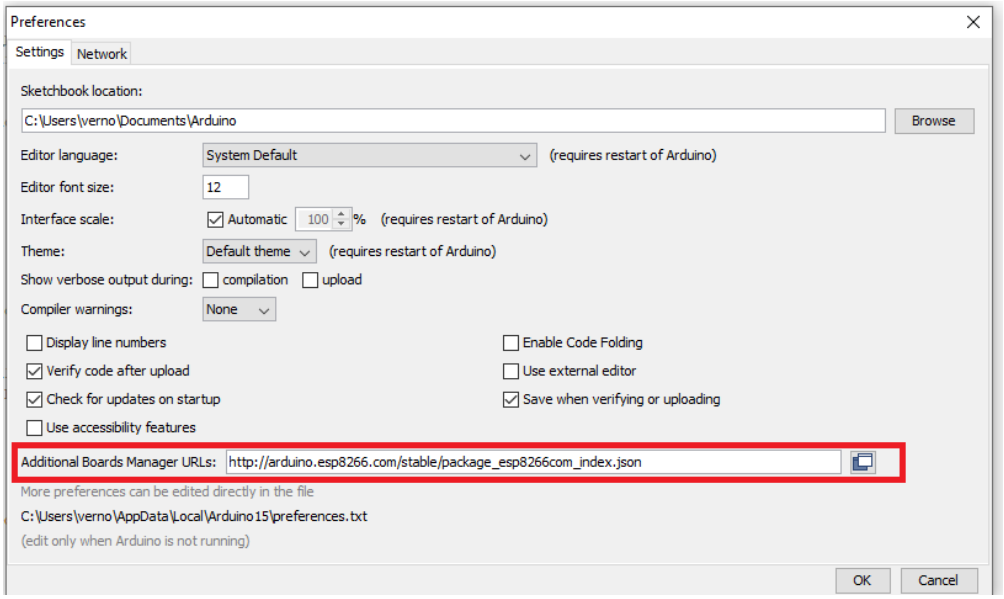


Figure 5.3.1-3: Preference Setting in Arduino IDE

5.3.2 Raspberry Pi Camera Module Configuration

To use Raspberry Pi Camera Module, it must be enabled through the Raspberry settings. This is due to the camera interface is disabled originally. Firstly, go to the Raspberry Pi Menu, Preferences and click on the Raspberry Pi Configuration shown in Figure 5.3.2-1. Then follow with Figure 5.3.2-2, go into Interfaces tab, at the Camera part, choose enable.

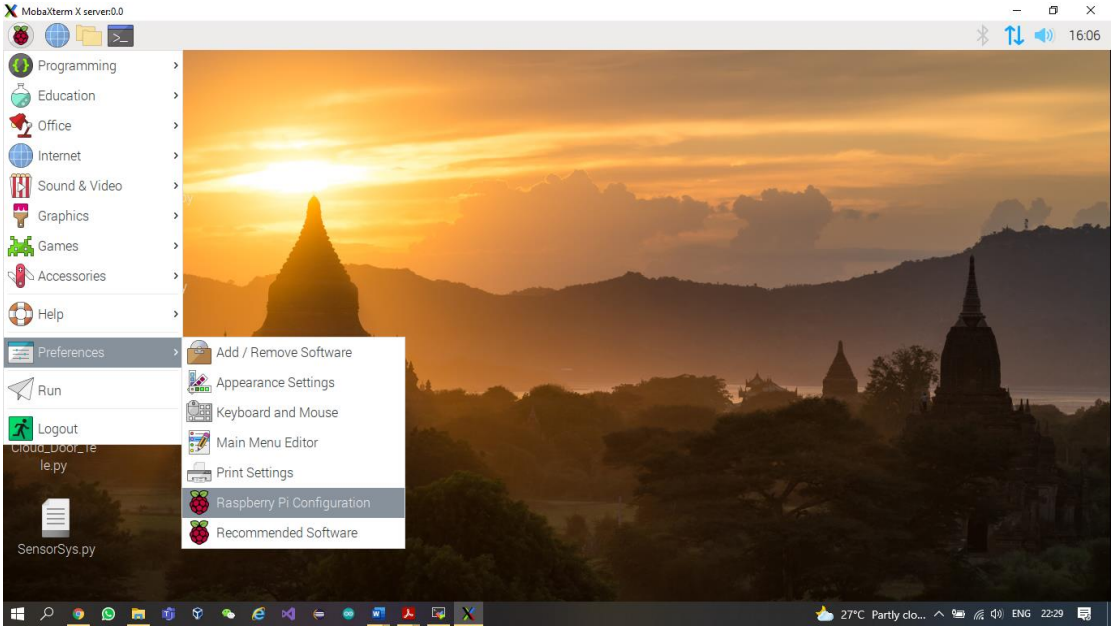


Figure 5.3.2-1: Raspberry Pi OS Preferences

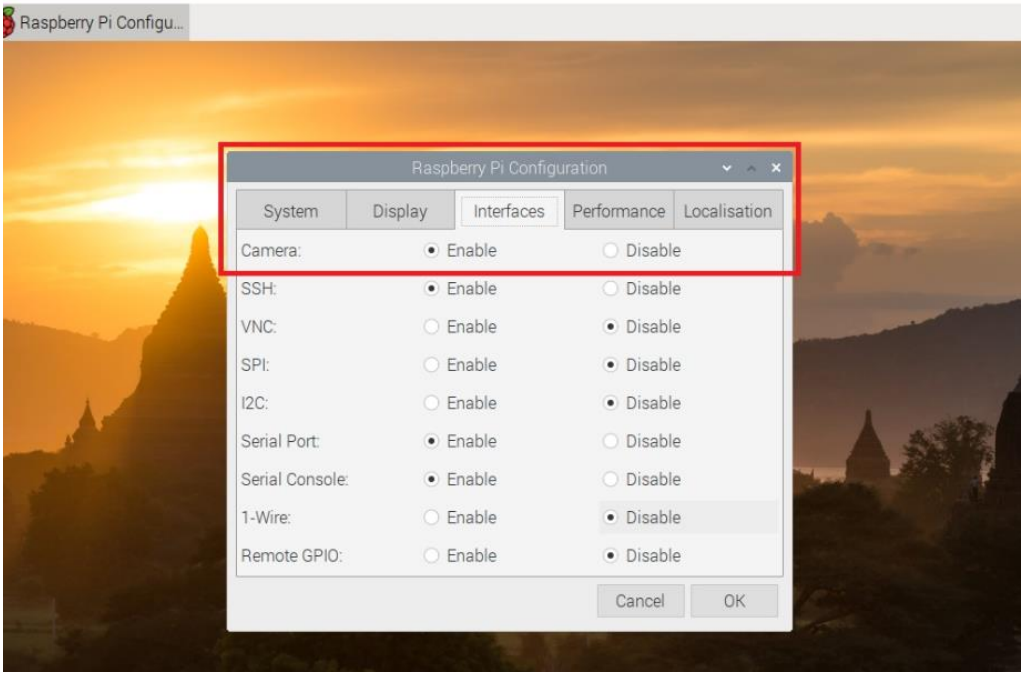


Figure 5.3.2-2: Raspberry Pi Configuration Page

5.3.3 Raspberry Pi Python Script

In figure 5.3.3-1, figure 5.3.3-2 and figure 5.3.3-3 is the python script for Raspberry Pi which contain the IR Sensor module, Object Detection module, Door Window Sensor Magnetic Switch module, Cloud module and Notification module.

```
import RPi.GPIO as GPIO
from picamera import PiCamera
from datetime import datetime
from subprocess import call
import time
from time import sleep
import datetime
import requests
import math
import telepot
from telepot.loop import MessageLoop
import urllib.request
import os
import cv2
import numpy as np
import sys
import importlib.util
import glob

url = "http://192.168.43.30/"

cam = PiCamera()
cam.resolution = (640, 480)
cam.vflip = True        #vertical flip
cam.framerate = 25

TOKEN = "BBFF-E8yZJtueCyt6cq8QJmnDf9IsytMCmP"  # Put your TOKEN here
DEVICE_LABEL = "vernon01"  # Put your device label here
Magnetic = "door"  # Put your first variable label here
Infrared = "infrared"  # Put your second variable label here
Infrared1 = "infrared1"  # Put your second variable label here

#sensor = 18
a = 0
mag = 0
magNew = 0
door = 0
infrared = 0
infrared1 = 0
t1 = 0.0
t2 = 0.0
detection = 0
```

```
def my_callback1(channel):      #sensor1 - Pin 14 (Outside)
        global t1, t, t2, infrared

        time1 = time.time()
        t1 = time1
        #infrared = 0
        #x = GPIO.input(14)
        #print("14: ", x)

        if GPIO.input(14) > 0.5:
                t = t1 - t2
                if (t2 > 0.0):
                        #print("T2 > 0.0")
                        t1 = 0.0
                        t2 = 0.0
                        if (t > 0):
                                print ("People Coming In")
                                infrared = 1

                        else:
                                #print ("Checking: ", t)
                                pass
                else:
                        #print ("T2 too small!")
                        pass

        else:
                #print("GPIO too small!")
                pass
```

```
def my_callback2(channel):      #sensor2 - Pin 15 (Inside)
        global t2, t, t1, infrared1

        time2 = time.time()
        t2 = time2
        #y = GPIO.input(15)
        #print("15: ", y)
        #infrared = 0

        if GPIO.input(15) > 0.5:
                t = t1 - t2
                if (t1 > 0.0):
                        #print("T1 > 0.0")
                        t2 = 0.0
                        t1 = 0.0
                        if (t < 0):
                                print ("People Going Out")
                                infrared1 = 1

                        else:
                                #print ("Checking: ", t)
                                pass
                else:
                        #print ("T1 too small!")
                        pass

        else:
                #print("GPIO too small!")
                pass


def get_data():
        global data

        n = urllib.request.urlopen(url).read() # get the raw html data in bytes (sends request and warn our esp8266)
        n = n.decode("utf-8") # convert raw html bytes format to string :3

        data = n
```

Figure 5.3.3-1: Raspberry Pi Python Script (1)

```
def build_payload(variable_1, variable_2, variable_3):
        global door, infrared, infrared1
        #if infrared = 1, then value2 = 1

        value_1 = door
        value_2 = infrared
        value_3 = infrared1

        payload = {variable_1: value_1, variable_2: value_2, variable_3: value_3}

        infrared = 0
        infrared1 = 0

        return payload
```

```
def post_request(payload):
    # Creates the headers for the HTTP requests
    url = "http://industrial.api.ubidots.com"
    url = "{}/api/v1.6/devices/{}".format(url, DEVICE_LABEL)
    headers = {"X-Auth-Token": TOKEN, "Content-Type": "application/json"}

    # Makes the HTTP requests
    status = 400
    attempts = 0
    while status >= 400 and attempts <= 5:
        req = requests.post(url=url, headers=headers, json=payload)
        status = req.status_code
        attempts += 1
        time.sleep(1)

    # Processes results
    if status >= 400:
        print("[ERROR] Could not send data after 5 attempts, please check \
            your token credentials and internet connection")
        return False

    print("[INFO] request made properly, your device is updated")
    return True
```

```
def handle(msg):
    global telegramText
    global chat_id

    chat_id = msg['chat']['id']
    telegramText = msg['text']

    print('Message received from ' + str(chat_id))

    if telegramText == '/start':
        bot.sendMessage(chat_id, 'Security camera is activated.')#Put your welcome note here

    while True:
        Door()

bot = telepot.Bot('1870631043:AAEb-EYUjDhNZVK9Q54MXtYFngT7Ca3KMAo')
bot.message_loop(handle)
```

```
def object_detection():
        global detection

        MODEL_NAME = 'Sample_tflite_Model'
        GRAPH_NAME = 'detect.tflite'
        LABELMAP_NAME = 'labelmap.txt'
        min_conf_threshold = 0.5

        # Parse input image name and directory.
        IM_NAME = 'Testing.jpg'
        #IM_DIR = args.imagedir


        # Import TensorFlow libraries
        # If tflite_runtime is installed, import interpreter from tflite_runtime, else import from regular tensorflow
        # If using Coral Edge TPU, import the load_delegate library
        pkg = importlib.util.find_spec('tflite_runtime')
        if pkg:
                from tflite_runtime.interpreter import Interpreter

        else:
                from tensorflow.lite.python.interpreter import Interpreter



        # Get path to current working directory
        CWD_PATH = os.getcwd()

        # Define path to images and grab all image filenames
        PATH_TO_IMAGES = os.path.join(CWD_PATH,IM_NAME)
        images = glob.glob(PATH_TO_IMAGES)

        # Path to .tflite file, which contains the model that is used for object detection
        PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)

        # Path to label map file
        PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)
```

```
        # Load the label map
        with open(PATH_TO_LABELS, 'r') as f:
                labels = [line.strip() for line in f.readlines()]

        # Have to do a weird fix for label map if using the COCO "starter model" from
        # https://www.tensorflow.org/lite/models/object_detection/overview
        # First label is '???', which has to be removed.
        if labels[0] == '???':
                del(labels[0])

        # Load the Tensorflow Lite model.

        interpreter = Interpreter(model_path=PATH_TO_CKPT)

        interpreter.allocate_tensors()

        # Get model details
        input_details = interpreter.get_input_details()
        output_details = interpreter.get_output_details()
        height = input_details[0]['shape'][1]
        width = input_details[0]['shape'][2]

        floating_model = (input_details[0]['dtype'] == np.float32)

        input_mean = 127.5
        input_std = 127.5
```

Figure 5.3.3-2: Raspberry Pi Python Script (2)

```
        # Loop over every image and perform detection
        for image_path in images:

                # Load image and resize to expected shape [1xHxWx3]
                image = cv2.imread(image_path)
                image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
                imH, imW, _ = image.shape
                image_resized = cv2.resize(image_rgb, (width, height))
                input_data = np.expand_dims(image_resized, axis=0)

                # Normalize pixel values if using a floating model (i.e. if model is non-quantized)
                if floating_model:
                        input_data = (np.float32(input_data) - input_mean) / input_std

                # Perform the actual detection by running the model with the image as input
                interpreter.set_tensor(input_details[0]['index'],input_data)
                interpreter.invoke()

                # Retrieve detection results
                boxes = interpreter.get_tensor(output_details[0]['index'])[0] # Bounding box coordinates of detected objects
                classes = interpreter.get_tensor(output_details[1]['index'])[0] # Class index of detected objects
                scores = interpreter.get_tensor(output_details[2]['index'])[0] # Confidence of detected objects
                #num = interpreter.get_tensor(output_details[3]['index'])[0]  # Total number of detected objects (inaccurate and not needed)

                # Loop over all detections and draw detection box if confidence is above minimum threshold
                for i in range(len(scores)):
                        if ((scores[i] > min_conf_threshold) and (scores[i] <= 1.0)):

                                # Get bounding box coordinates and draw box
                                # Interpreter can return coordinates that are outside of image dimensions, need to force them to be within image using max() and min()
                                ymin = int(max(1,(boxes[i][0] * imH)))
                                xmin = int(max(1,(boxes[i][1] * imW)))
                                ymax = int(min(imH,(boxes[i][2] * imH)))
                                xmax = int(min(imW,(boxes[i][3] * imW)))

                                cv2.rectangle(image, (xmin,ymin), (xmax,ymax), (10, 255, 0), 2)
```

```
                                # Draw label
                                object_name = labels[int(classes[i])] # Look up object name from "labels" array using class index
                                label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'
                                labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2) # Get font size
                                label_ymin = max(ymin, labelSize[1] + 10) # Make sure not to draw label too close to top of window
                                cv2.rectangle(image, (xmin, label_ymin-labelSize[1]-10), (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED) # Dr
                                cv2.putText(image, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2) # Draw label text

                                a = object_name

                                if (a =="person"):
                                        #print('Object ' + str(i) + ': ' + object_name)
                                        print('Person detected!')
                                        detection = 1

                # All the results have been drawn on the image, now display the image
                cv2.imshow('Object detector', image)


                # Press any key to continue to next image, or press 'q' to quit
                #if cv2.waitKey(0) == ord('q'):
                        #break


        # Clean up
        cv2.destroyAllWindows()
```

```
def Door():

        global chat_id
        global mag
        global magNew
        global door
        global data

        get_data()
        print("Data: " + data)
        a = data
        sleep(2)

        if (a == "1"):
                print("Door Opened")
                mag = 1
                door = 1
                if magNew != mag:
                        magNew = mag
                        sendNotification(mag)

        elif (a == "0"):
                #print("No motion detected")
                mag = 0
                door = 0

        if magNew != mag:
                magNew = mag
```

```
def sendNotification(mag):

        global chat_id, detection

        if mag == 1:
                cam.start_preview()
                time.sleep(2)
                cam.capture("Testing.jpg")
                object_detection()

        if(detection == 1):
                bot.sendPhoto(chat_id, photo = open("Testing.jpg", 'rb'))
                bot.sendMessage(chat_id, 'The door is opened.')
        else:
                pass


GPIO.setmode(GPIO.BCM)
GPIO.setup(14, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(15, GPIO.IN, pull_up_down=GPIO.PUD_UP)

GPIO.add_event_detect(14, GPIO.RISING, callback=my_callback1, bouncetime=500)
GPIO.add_event_detect(15, GPIO.RISING, callback=my_callback2, bouncetime=500)

def main():
    payload = build_payload(
        Magnetic, Infrared, Infrared1)

    print("[INFO] Attemping to send data")
    post_request(payload)
    print("[INFO] finished")

if __name__ == '__main__':
        while (True):

                main()
```

Figure 5.3.3-3: Raspberry Pi Python Script (3)

5.3.4 NodeMCU Lua v3 ESP8266 C++ Code

```
#include "ESP_MICRO.h" //Include the micro library
#include <NTPClient.h>
#include <WiFiUdp.h>

const long utcOffsetInSeconds = 28800;
int magnetic = 0;
int a = 0;

WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", utcOffsetInSeconds);

void setup(){
  Serial.begin(9600); // Starting serial port for seeing details
  start("Vernon's 9T","lynnchii");  // EnAIt will connect to your wifi with given details
  timeClient.begin();
  pinMode(D8, INPUT);
}
void loop(){
  timeClient.update();
  Serial.println(timeClient.getFormattedTime());


  a = digitalRead(D8);

  //Serial.print("A: ");
  //Serial.println(a);
  if(a == 1){
    Serial.println("Door is Closed!");
    magnetic = 0;
  }
  else{
    Serial.println("Door is Opened!");
    magnetic = 1;
  }

  waitUntilNewReq();  //Waits until a new request from python come
  /* increases index when a new request came*/
  /*testvariable += 1;*/
  returnThisInt(magnetic); //Returns the data to python

}
```

Figure 5.3.4-1: NodeMCU Lua v3 ESP8266 Code

Figure 5.3.4-1 is the main code in NodeMCU Lua v3 ESP8266 for sending the data of Door Window Sensor Magnetic Switch to Raspberry Pi. For Figure 5.3.4-2 below, it is a header for the main code that refer from KebabLord to allow the NodeMCU Lua v3 ESP8266 to send data via http.

```
#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
#include <WiFiClient.h>

// PORT
WiFiServer server(80);
WiFiClient client;
String rule;

void start(String ssid, String pass){
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid.c_str(),pass.c_str());

  Serial.println("");
// Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");

  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
// Setting up mDNS responder
  if (!MDNS.begin("esp8266")) {
    Serial.println("Error setting up MDNS responder!");
    while (1) {
      delay(1000);
    }
  }
  Serial.println("mDNS responder started");
// Start TCP (HTTP) server
  server.begin();
  Serial.println("TCP server started");
// Add service to MDNS-SD
  MDNS.addService("http", "tcp", 80);
}

bool isReqCame = false;

void CheckNewReq(){
  client = server.available();
  if (!client) {
    return;
  }
  Serial.println("");
  Serial.println("NEW REQUEST");
// Waiting client to connect
  while (client.connected() && !client.available()) {
    delay(1);
  }
// Read the first line of HTTP request
  String req = client.readStringUntil('\r');
  int addr_start = req.indexOf(' ');
  int addr_end = req.indexOf(' ', addr_start + 1);
  if (addr_start == -1 || addr_end == -1) {
    Serial.print("Invalid request: ");
    Serial.println(req);
    return;

  }
  req = req.substring(addr_start + 1, addr_end);
  Serial.print("Requested Path: ");
  Serial.println(req);

  rule = req;
  isReqCame = true;
  client.flush();
}
void waitUntilNewReq(){
  do {CheckNewReq();} while (!isReqCame);
  isReqCame = false;
}

void returnThisStr(String final_data){
  String s;
  //HTTP Protocol code.
  s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n";
  s += final_data; //Our final raw data to return
  client.print(s);

  Serial.println("Returned to client.");
}
void returnThisInt(int final_data){
  returnThisStr(String(final_data));
}

String getPath(){
  return rule;
}
```

Figure 5.3.4-2: NodeMCU Lua v3 ESP8266 Header, ESP._Micro.h

5.3.5 Ubidots

Ubidots is the cloud and dashboard to display the data that collected from the Raspberry Pi. Raspberry Pi need to configure with the token achieved from the Ubidots. After created an account from Ubidots, press on the user and API Credentials shown in Figure 5.3.5-1. The token will then be shown like in Figure 5.3.5-2.

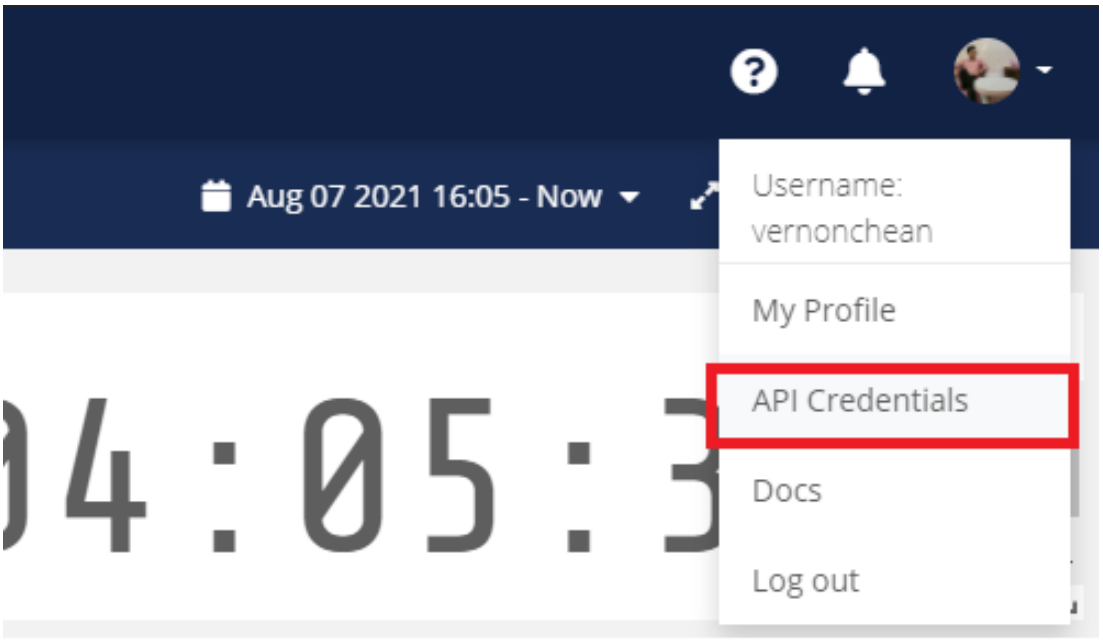


Figure 5.3.5-1: Ubidots API Credentials

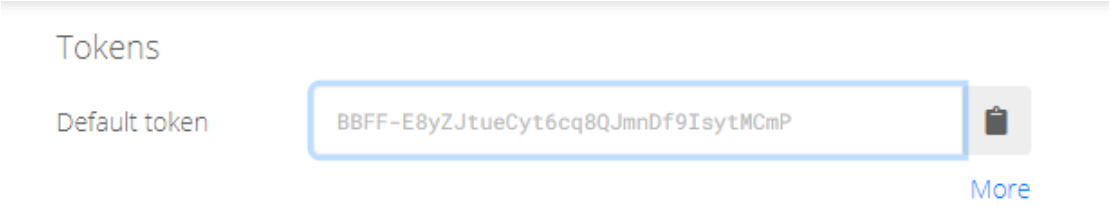


Figure 5.3.5-2: Ubidots Token

Copy the token shown at Figure 5.3.5-2 into the Python Script as shown in Figure 5.3.5-3 into the TOKEN part. After that rename the DEVICE_LABEL to a name that we want. The DEVICE_LABEL will then be shown in Ubidots as shown as Figure 5.3.5-4.

```
TOKEN = "BBFF-E8yZJtueCyt6cq8QJmnDf9IsytMCmP"  # Put your TOKEN here
DEVICE_LABEL = "vernon01"  # Put your device label here
```

```
def post_request(payload):
    # Creates the headers for the HTTP requests
    url = "http://industrial.api.ubidots.com"
    url = "{}/api/v1.6/devices/{}".format(url, DEVICE_LABEL)
    headers = {"X-Auth-Token": TOKEN, "Content-Type": "application/json"}
```

Figure 5.3.5-3: Ubidots Token in the Script

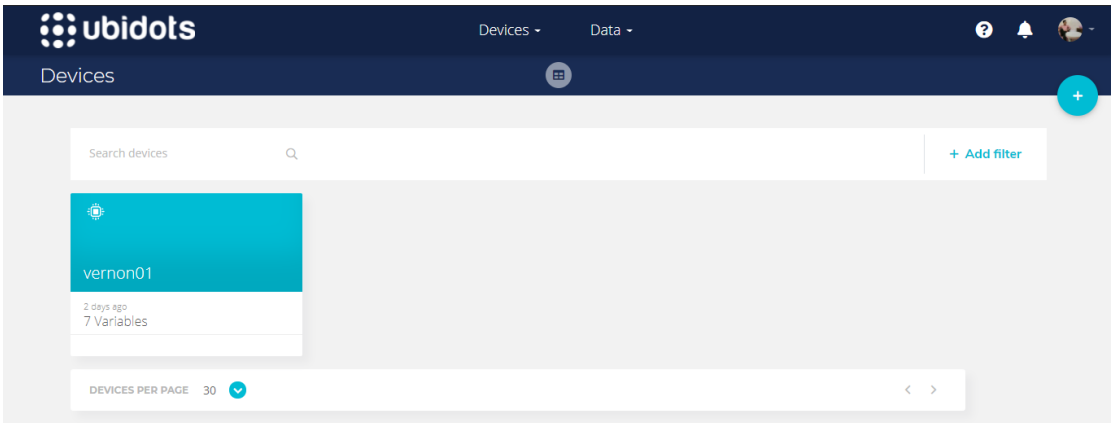


Figure 5.3.5-4: Device Label at Ubidots

Then we can configure the dashboard by tapping the "+" button at the dashboard page shown in Figure 5.3.5-5. We can configure the Dashboard based on our interest.

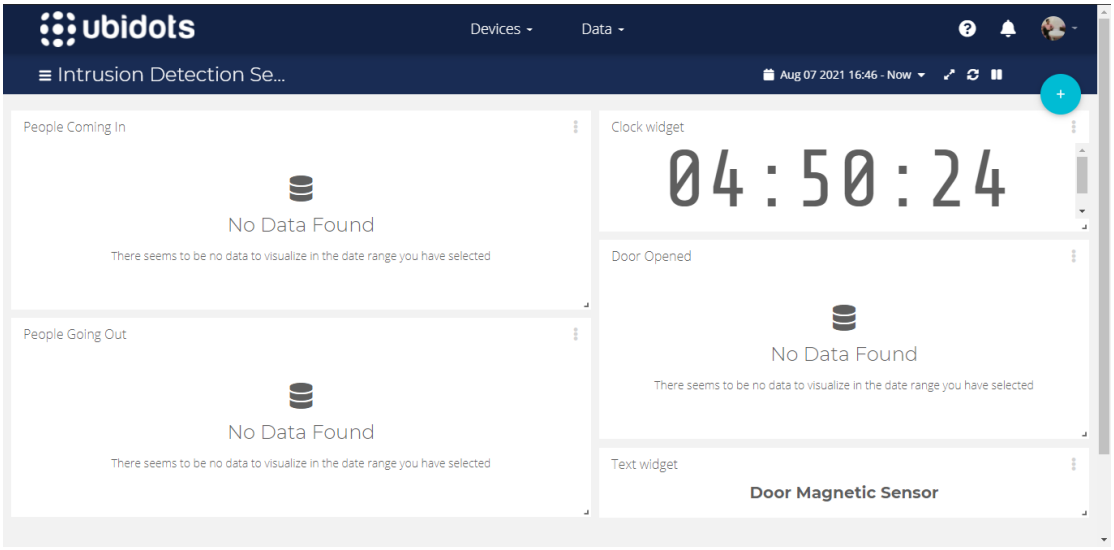


Figure 5.3.5-5: Ubidots Dashboard

5.3.6 Telegram Bot

Telegram Bot allow the Teleport to send the message, photos, and videos to the Telegram. Go to the search in Telegram Mobile Application, search “BotFather” as shown in Figure 5.3.6.1.

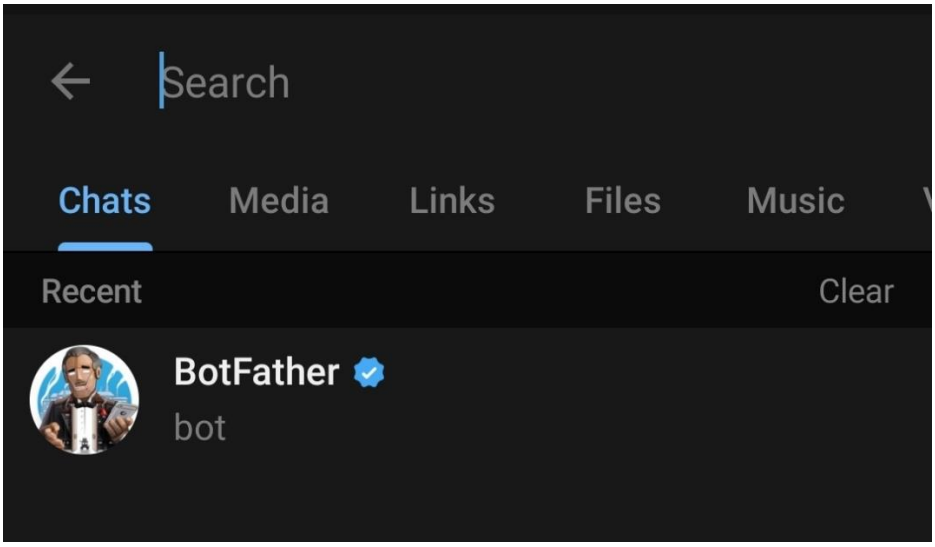


Figure 5.3.6-1: Search “BotFather” in Telegram

Press “/start” in the BotFather, then type “/newbot” to create a new bot. Enter the name for the bot and username for the bot. When it successfully created, it will send the token and a message page for the user as shown as Figure 5.3.6-2.

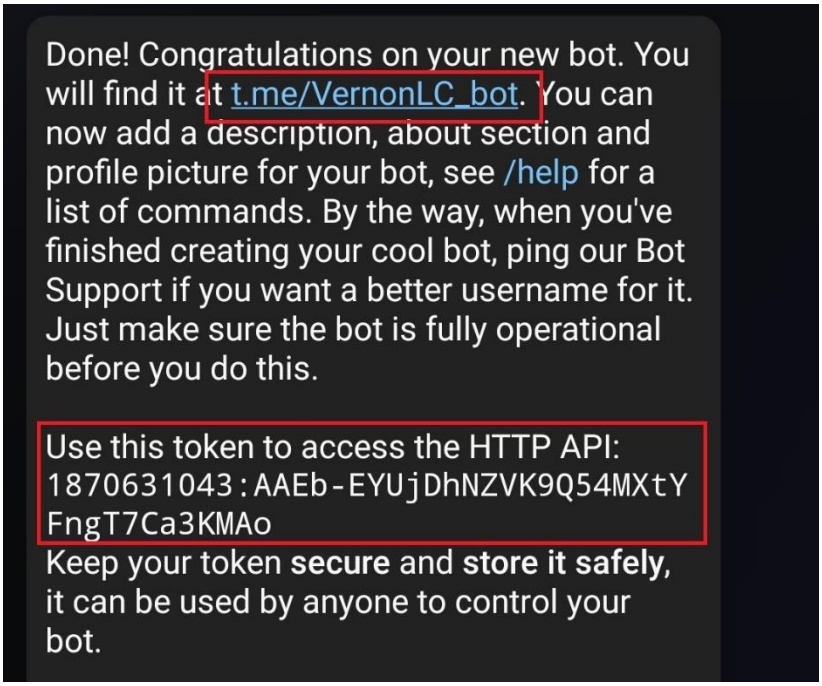


Figure 5.3.6-2: Token and message page for Telegram.

5.4 System Operation

To run make the system run, there is 3 part need to be run, which is the python script from Raspberry Pi, NodeMCU Lua v3 ESP8266 and Telegram. Based on Figure 5.4-1, type the command "python3 Sys_Img.py" in the terminal. The sentence prompted "Message received from 1033998345" when the Telegram pressed start. The "Data: 0" is the data that received from the NodeMCU Lua v3 ESP8266.

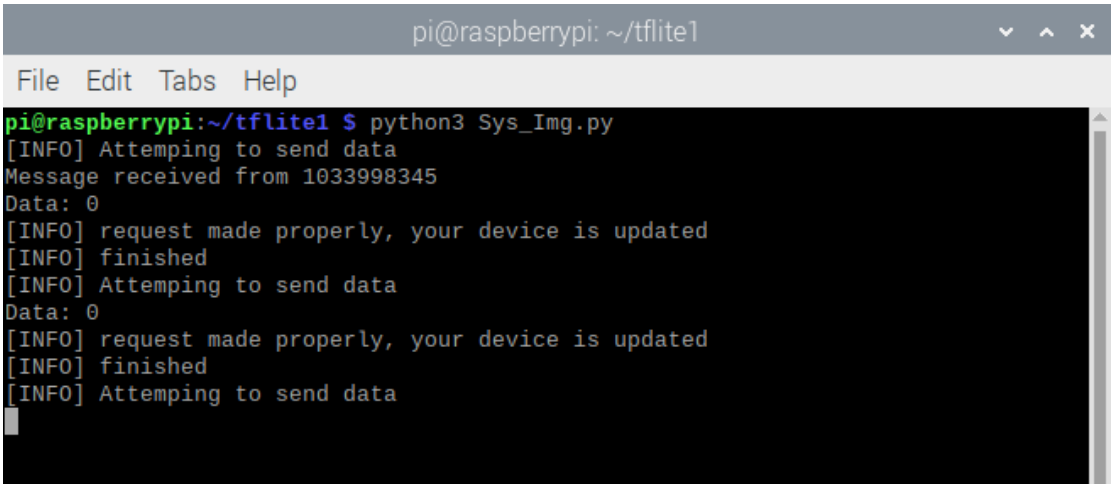


Figure 5.4-1: Raspberry Pi Terminal

When there is a person pass by the IR sensor, it will show two different line which is "People Going Out" and "People Coming In". When the person passes by the sensor inside first only pass by the sensor outside, it will prompt "People Going Out"; while the pass by the sensor outside first only pass by the sensor inside will prompt "People Coming In" based on Figure 5.4-2.

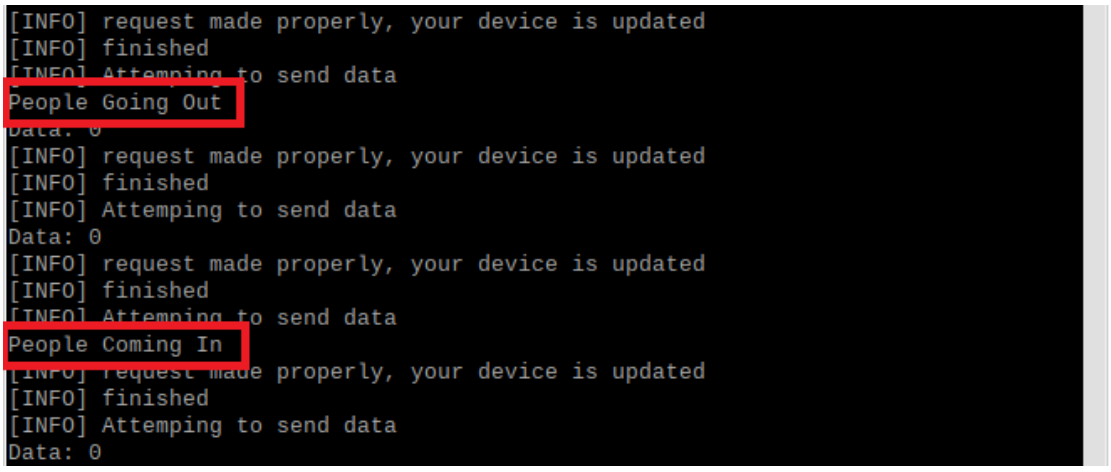


Figure 5.4-2: IR Sensor Detected and prompt line

In Figure 5.4-3, when the python script is running at Raspberry Pi, the console in Arduino IDE will show the door status. If the Door Window Sensor Magnetic Switch is disconnected, it will show “Door is Opened!”. The terminal in Raspberry Pi will show “Data:1” and prompt the line “Door Opened” which can see in Figure 5.4-4. A photo will be taken, and Object Detection will analyze the photo. When there is a person detected, it will prompt the line “Person Detected” then the photo that captured will send to Telegram.

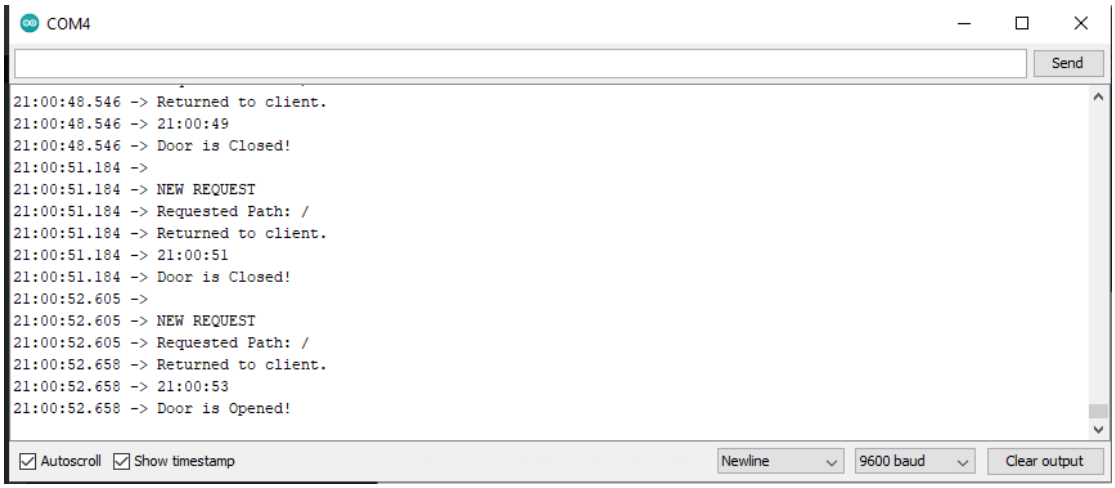


Figure 5.4-3: Console in Arduino IDE for NodeMCU Lua v3 ESp8266

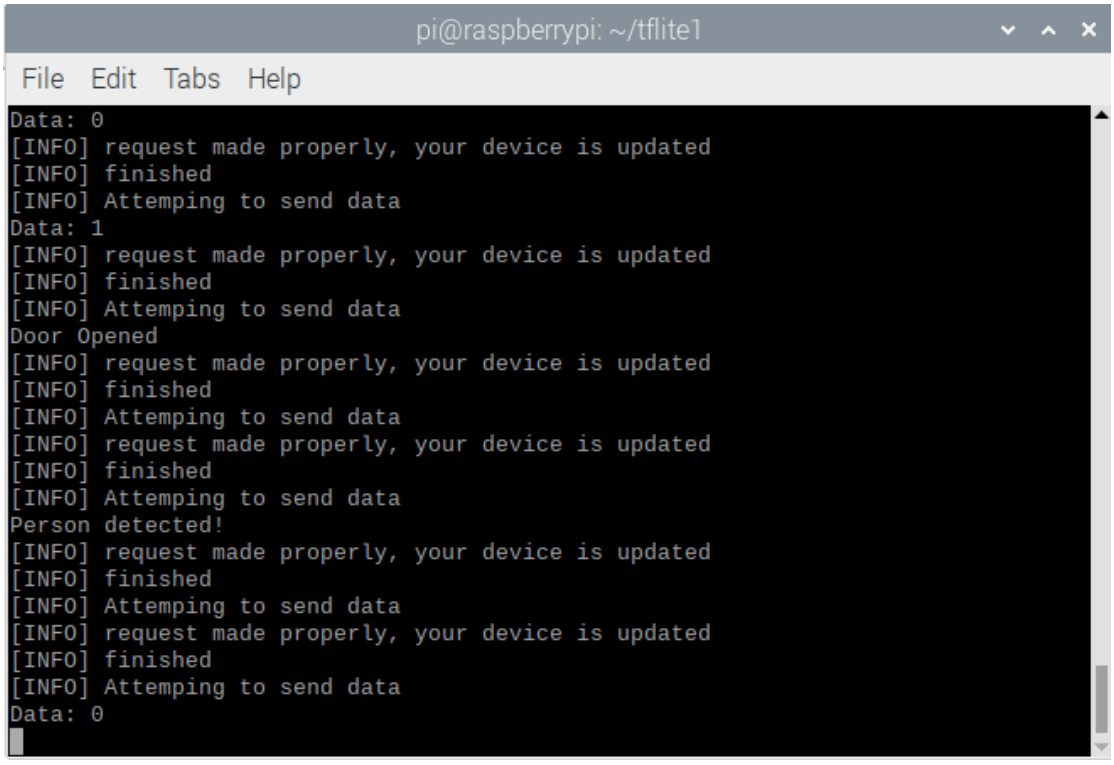


Figure 5.4-4: Terminal when received different data.

In Figure 5.4-5, "/start" is to activate the system and it will display the line "Security camera is activated". When the Door Window Sensor Magnetic Switch is disconnected and a person is detected just like the photo in Figure 5.4-5, it will be sent to Telegram.

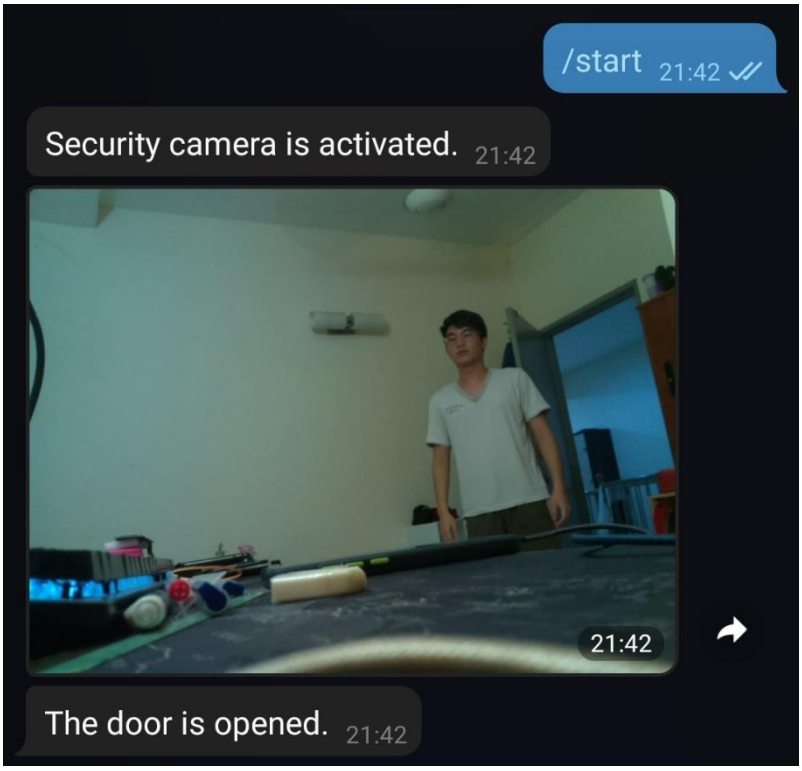


Figure 5.4-5: Telegram when receiving the notification.

All the data will be uploaded to Ubidots and it will display on the Dashboard as shown as Figure 5.4-6.

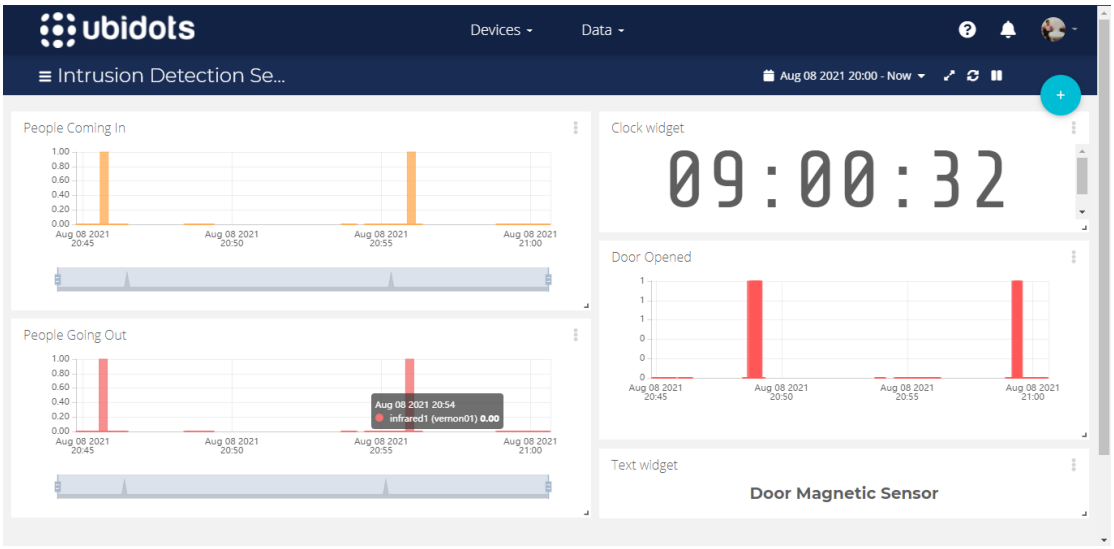


Figure 5.4-6: Ubidots Dashboard

5.5 Concluding Remark

In this chapter, it explained the setup of hardware and software that needed in this project. The setting and configuration for the things that needed for the system is shown step by steps and with the example of the script, code, and configuration. Lastly, the system operation is also explained in this chapter.

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

In the system testing, it will be conducting the accuracy testing and connectivity testing. For accuracy testing, it will be testing the accuracy of two IR sensor that indicates object going in and out and testing the accuracy of the Object Detection (TensorFlow). The system uses two IR sensors to detect whether there is object pass by and using time based to understand that whether there is an object going in or going out. In the accuracy testing of IR sensor, it tests the object passing by two sensors and detect whether the system indicates the object going out or going in correctly. For Object Detection accuracy testing, it tests the image that captured and analyzed by the object detection is accurate or not since the detection model that used in the system is not trained by us. Moreover, object detection is important as it act as the verification of "person" before sending the image and notification to the user through Telegram.

For connectivity testing, it will be testing three parts, which is from NodeMCU to Raspberry Pi, Raspberry Pi to Cloud (Ubidots) and Raspberry Pi to Telegram. From NodeMCU to Raspberry Pi, NodeMCU will be sending the data of Door Window Sensor Magnetic Switch to Raspberry Pi so that Raspberry Pi will know when the door is opened. Since it is wireless connection through WLAN, the connectivity in between them need to test to understand the connectivity. Other than that, for Raspberry Pi to Cloud (Ubidots) is for the data that collected from sensor to upload and store inside the cloud. In the testing, we will study the connectivity between them, whether there is data not uploaded to the cloud. Lastly, Telegram is to send the notification and store the photo of evidence for the user. In the testing we will study whether the notification and photo are sent successfully.

6.2 Testing Setup and Result

6.2.1 Accuracy Testing for IR sensors

There are two different ways of testing on the IR Sensor have been done, which is pass through regularly and pass through the sensor with different behavior. Due to focus testing on the IR Sensor, the code has been modified for showing results.

Table 6.2.1-1: Testing Result from pass through regularly.

Detection	Time	Success
"People Going Out"	10	10
"People Coming In"	10	10

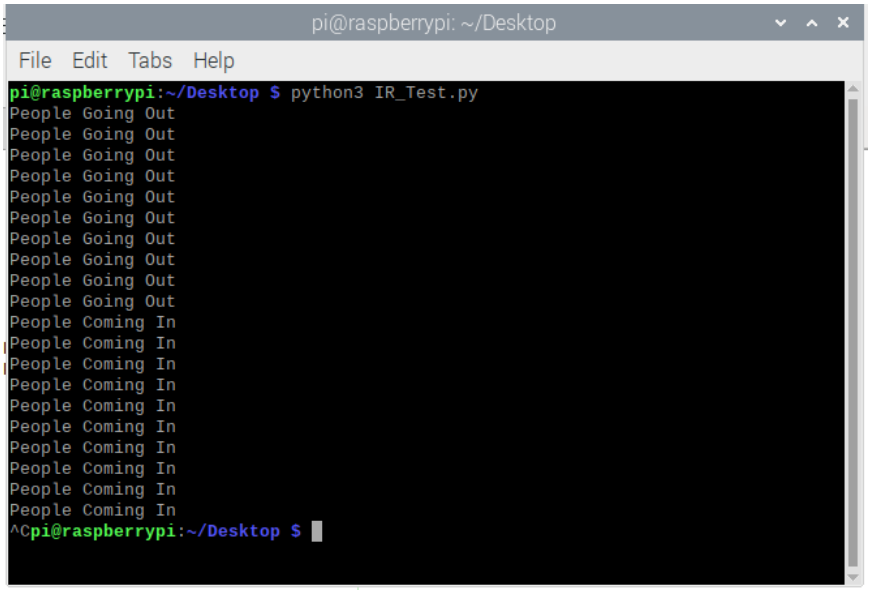


Figure 6.2.1-1: Screenshot of Result when object pass through.

Table 6.2.1-1 and Figure 6.2.1-1 shows that the testing where a person pass through the two sensors by regular speed. The word "People Going Out" and People Coming In" will be shown when both of the sensor is detected. For "People Going Out" it detects the inside sensor first, then only detect the outside sensor, where "People Coming In" is detected the outside sensor then only detect the inside sensor. Since the sensor is using time to know when it is detected, so the IR Sensor detected time will be compare and get the result. Hence, there are all success for the whole testing.

On the other hand, the second testing is carried out where the people pass by with different behavior. Table 6.2.1-2 is the result of the testing.

Table 6.2.1-2: Testing Result with Different Behavior.

Detection	**Time Interval**	**Description**
People Going Out	-1.9709	Normal Speed
People Going Out	-1.1497	Normal Speed
People Going Out	-0.1190	Run
People Going Out	-0.5133	Run
People Going Out	-7.7603	Walk, Stop and Walk
People Coming In	0.5012	Run
People Coming In	0.3441	Stop and Run
People Coming In	0.1844	Stop and Run
People Coming In	5.8193	Walk, Stop and Walk
People Coming In	7.2254	Walk, Stop and Walk

From Table 6.2.1-2, different behavior happen will affects the time interval, but it does not affect the detection of the sensor. For the first and second result, the person that pass through is using a normal speed to pass through the two sensors.

For third, fourth and sixth result, the person run pass through both sensors. However, for result seventh and eighth, the person stops in front of the sensor then run pass through the second sensor. We can see that the different between the time interval not that much. The sensor will detect the person and record down the latest time that the sensor does not detect the person.

Lastly, for the behavior of “Walk, Stop and Walk”, which indicates the result of fifth, nineth and tenth that have higher time interval. In this behavior, the person walk pass through the first sensor then stop for a while then continue to walk pass through the second sensor. We can see that there is higher time interval compare with the other behavior.

In this testing we can see that the IR sensor’s part is accuracy because the IR sensors is time-based record the detection. When the IR sensor detected there is something pass by, it will record and compare time. Hence, the detection is accurate.

6.2.2 Accuracy Testing for Object Detection

Object Detection is to validate the photo that captured to make sure that there is no false alarm happened. For this accuracy testing, it tests for 20 times to see that how the object detection will detect the object that captured in the photo correctly. All the figure that captured are attached in Appendix A. There is different scenario will be tested and it is listed in the description in Table 6.2.2-1.

Table 6.2.2-1: Object Detection Result with Different Behavior.

Figure	Description	Result	Accurate
A-1	Normal lighting with no person	No Person Detected	/
A-2		No Person Detected	/
A-3		No Person Detected	/
A-4		No Person Detected	/
A-5		No Person Detected	/
A-6	Normal lighting with person facing the camera	Person Detected	/
A-7		Person Detected	/
A-8	Normal lighting with person not facing the camera	Person Detected	/
A-9		Person Detected	/
A-10	Normal lighting with person standing near	No Person Detected	X
A-11		Person Detected	/
A-12		No Person Detected	X
A-13		Person Detected	/
A-14		Person Detected	/
A-15		Person Detected	/
A-16	Dark Situation with no person	No Person Detected	/
A-17		No Person Detected	/
A-18	Dark Situation with person	No Person Detected	X
A-19		No Person Detected	X
A-20	Dark Situation with person holding flashlight.	Person Detected	/
Number of accurate detections			16/20
Percentage of Accuracy			80%

Based on Table 6.2.2-1, there are 16 out of 20 of the detection is correct, which contribute of 80% of the accuracy of the object detection. Object Detection is working better in a brighter situation, it has higher accuracy which is 13 out of 15 or 86.7% of the accuracy compare with the dark situation where it does not able to detect well in the dark situation due to the nature of the camera sensor. Object detection does not work well when the person is too near the camera. There are 6 examples of the person standing near to the camera, 2 out of 6 or 33.33% of detection do not detected the person. However, for normal situation, the intruder will not stand that close to the camera.

For dark situation, it does not detect the surrounding well. Although the object detection does not detect any person in the dark while there is no person, the object detection does not detect any person in dark when there is person standing in front too. However, when the person that standing in the dark holding a flashlight, the object detection able to detect it. This is because the flashlight shows the body shape and the face of the person so that the detection algorithm able to detect it.

In this accuracy testing, we can see that the object detection algorithm is acceptable and accurate as it can detect most of the situation correctly. For normal situation where the intruder enters the room, the object detection able to detect the person accurately.

6.2.3 Connectivity Testing for NodeMCU to Raspberry Pi

NodeMCU Lua v3 ESP8266 is a component that allow the data that collected to be send to the Raspberry Pi wirelessly through the build-in WiFi. NodeMCU is located at the door or window to detect whether the door or window is opened or closed. For this connectivity test, we print out the time from NodeMCU that send the data out and the time that Raspberry Pi received the data as shown as Figure 6.2.3-1 below. All the data is recorded in Table 6.2.3-1.

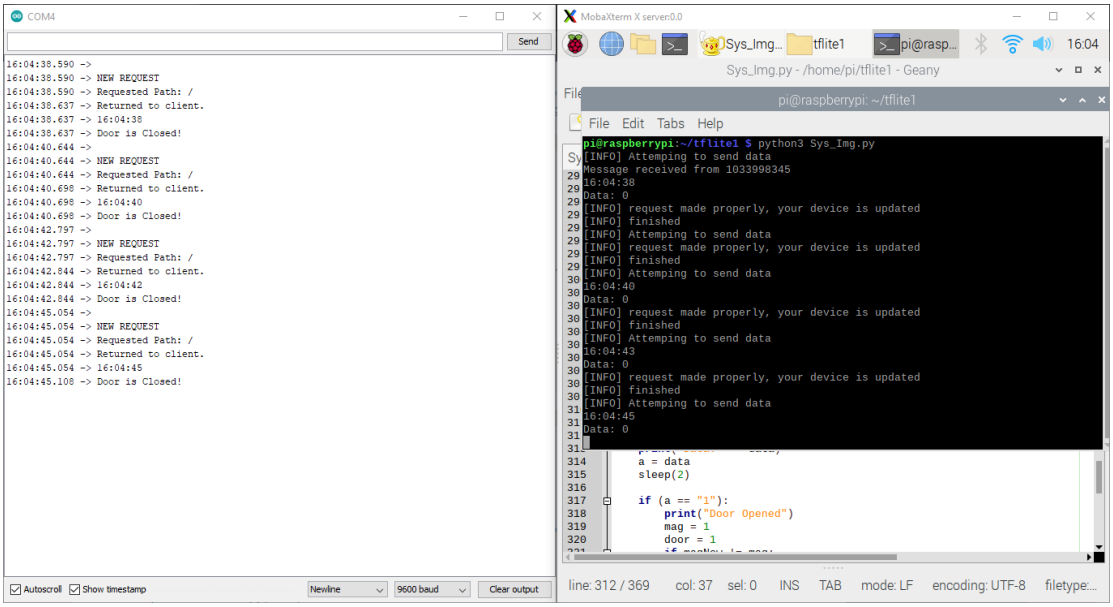


Figure 6.2.3-1: Screenshot of Time in NodeMCU and Raspberry Pi.

We have run through the testing for 20 times, and it is separated into two session. For the first 10 times, the door is closed, and it will the result is recorded. From the table, we can see that every 2 seconds the NodeMCU will release a reading and the Raspberry Pi will read the receive the reading. There is not much delay happened between the connectivity. The maximum delay that recorded is 1 second, which is from Testing 5 to Testing 8. However, it does not affect the result, Raspberry Pi still understand that the door is closed.

For the second 10 times, the door is opened and closed to collect the result. At the Testing 11 and 12 and Testing 18 and 19, the NodeMCU indicates that door is opened. From Raspberry Pi side, Testing 12 and Testing 19 shows that the door is opened as stated in NodeMCU. However, for Testing 11 and Testing 18, we can see that at Raspberry Pi, it does not show the door is opened, although NodeMCU shows it

is opened. For Testing 13, 16, 17, and 20, it indicates that the door is closed, and Raspberry Pi also does not show that the door is opened. For Testing 14 and 15, we can see that the NodeMCU shows that door is opened, and the Raspberry Pi also shows that the door is opened. Hence, for testing 13, 14, 15, 16, 17, and 20 are accurate.

In Testing 12, 14, 15 and 19, we can see that the time between the next data is more than 2 seconds. This is because when the Raspberry Pi detected that the door is opened, object detection will be turned on. In that period, the Raspberry Pi will not send request to the NodeMCU, so the NodeMCU will not sending the data back to the Raspberry Pi.

For the second 10 times testing, the result shows that 2 out of 10 that the connectivity is not accurate. This is acceptable as when the door is opened, the door will not be closed in under 2 seconds. For the first 10 times, all the data that transfer to the Raspberry Pi is accurate. Hence, the connectivity is stable.

Table 6.2.3-1: Result of connectivity between NodeMCU and Raspberry Pi.

Testing	**Description**	**Time in NodeMCU**	**Description shown in Raspberry Pi**	**Time in Raspberry Pi**	**Interval**	**Accurate**
1	Door is closed!	16:05:42	N/A	16:05:42	0 sec	/
2		16:05:44	N/A	16:05:44	0 sec	/
3		16:05:47	N/A	16:05:47	0 sec	/
4		16:05:49	N/A	16:05:49	0 sec	/
5		16:05:51	N/A	16:05:52	1 sec	/
6		16:05:53	N/A	16:05:54	1 sec	/
7		16:05:55	N/A	16:05:56	1 sec	/
8		16:05:57	N/A	16:05:58	1 sec	/
9		16:06:00	N/A	16:06:00	0 sec	/
10		16:06:02	N/A	16:06:02	0 sec	/
11	Door is opened!	16:35:06	N/A	16:35:06	0 sec	X
12		16:35:08	Door Opened	16:35:08	0 sec	/

13	Door is closed!	16:35:22	N/A	16:35:22	0 sec	/
14	Door is opened!	16:35:24	Door Opened!	16:35:24	0 sec	/
15		16:35:48	Door Opened!	16:35:48	0 sec	/
16	Door is closed!	16:36:12	N/A	16:36:12	0 sec	/
17		16:36:14	N/A	16:36:14	0 sec	/
18	Door is opened!	16:36:16	N/A	16:36:16	0 sec	X
19		16:36:18	Door Opened!	16:36:18	0 sec	/
20	Door is closed!	16:36:32	N/A	16:36:32	0 sec	/

6.2.4 Connectivity and Accuracy Testing for Raspberry Pi to Cloud

Cloud (Ubidots) is to collect the data that send from the Raspberry Pi. The data that collected is to allow the user to monitor the situation. In this testing, we need to test the connectivity and accuracy for the cloud. We have run the test for 30 minutes and there is none of the dysconnectivity issues encounter during the testing. However, we can see from Figure 6.2.4-1, there is a small gap in between the 7th bar and 8th bar of "People Coming In" Graph. We believe that it is disconnection during that short period of time. However, it does not affect the data.

For accuracy testing, we have gone through 10 times in and 10 times out on IR sensor and 10 times open the door for Door Window Magnetic Sensor. The result can refer to Figure 6.2.4-1, Figure 6.2.4-2, and Figure 6.2.4-3.

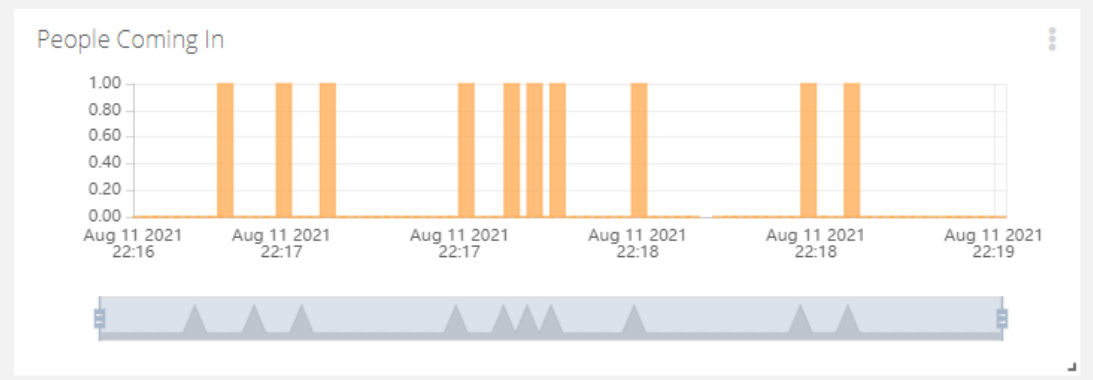


Figure 6.2.4-1: Dashboard data display of the IR sensor People Coming In

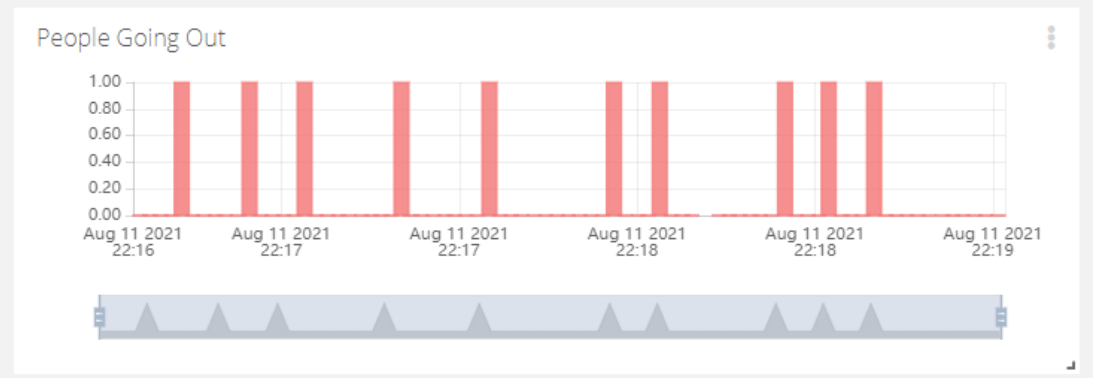


Figure 6.2.4-2: Dashboard data display of the IR sensor People Going Out

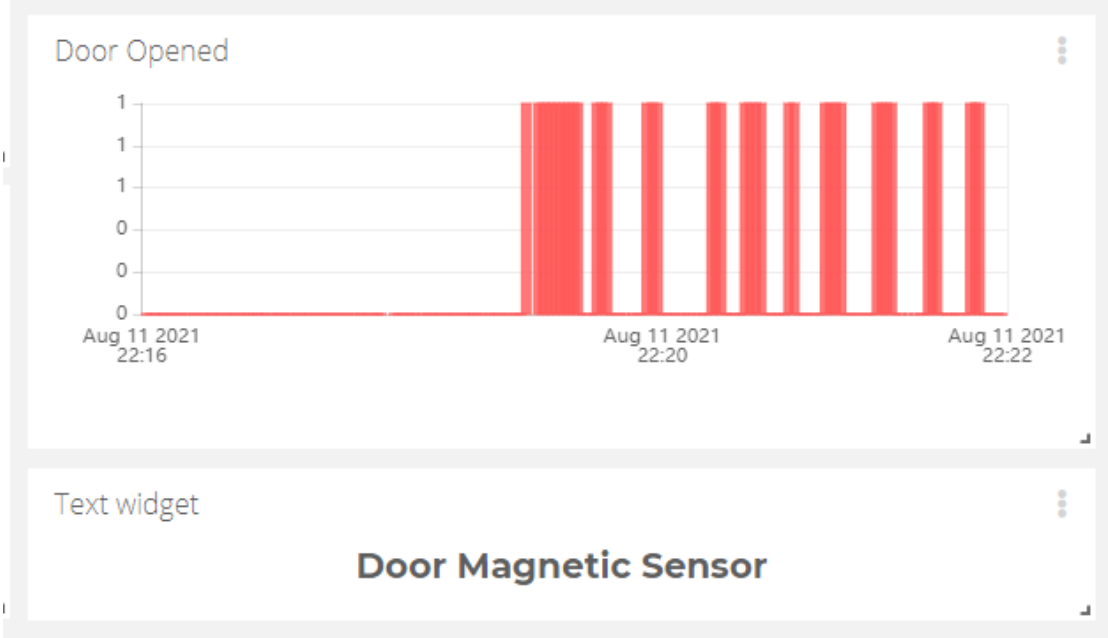


Figure 6.2.4-3: Dashboard data display of the Door Window Sensor Magnetic Switch

Based on Figure 6.2.4-1 and Figure 6.2.4-2, there are 10 data collected and display on the dashboard for each graph, which is "People Coming In" and People Going Out". Each of the data is uploaded correctly and none of the data is lost. For Figure 6.2.4-3, which is the Door Window Sensor Magnetic Switch, there are also 10 data recorded and display on the dashboard. However, for the first bar of the graph, the bar is thicker. This is due to the Magnetic sensor does not contact properly during the test and it cause the bar to be thicker. However, for the other 9 bars, there is no issue. All the data is correct and none of the data is lost.

In this testing, we can see that the connectivity and the accuracy between the Raspberry Pi to the Cloud (Ubidots) is accurate and the connection is stable. This is because there is none of the data recorded correctly or none of the data is lost when the data is uploaded from the Raspberry Pi to the Cloud, which is Ubidots.

6.2.5 Connectivity Testing for Raspberry Pi to Telegram

Telegram is the communication platform for Raspberry Pi to send the notification and captured photo with person to the users. In this system, Telegram will then be triggered when the object detection detected a person. We will be using the result from Chapter 6.2.2 on the result person detected as the testing part

Table 6.2.5-1: Telegram received period.

Figure	Result	Raspberry Pi Time	Telegram Time	Time interval	Successful
A-6	Person Detected	20:16	20:17	1 min	/
A-7	Person Detected	20:17	20:18	1 min	/
A-8	Person Detected	20:18	20:18	0 min	/
A-9	Person Detected	20:19	20:19	0 min	/
A-11	Person Detected	20:20	20:20	0 min	/
A-13	Person Detected	20:21	20:22	1 min	/
A-14	Person Detected	20:22	20:22	0 min	/
A-15	Person Detected	20:23	20:23	0 min	/
A-20	Person Detected	20:27	20:28	1 min	/

Based on Table 6.2.5-1, we are using 9 result from the Chapter 6.2.2. We have recorded the time down on the Raspberry Pi when the object detection is detected a person and record down in the column “Raspberry Pi Time”. While for the “Telegram Time” it is the time stamp that stated on the message and the image received. We can see that 4 out of 9 the time interval is a difference of 1 minutes, while 5 out of 9 is in between 1 minute. All the image that detected “Person” is successfully send to the Telegram to alert the user.

In this testing, we can see that the connectivity between the Raspberry Pi to the Telegram is stable. It does not delay the message and the image to Telegram much. Raspberry Pi is able to successfully send all the image and message to the Telegram without any other issues.

6.3 Project Challenges

In this project, there is different kind of challenges encountered which unable to solve by at this moment. The challenges might cause by the nature of the components which the efforts make on the software do not settle the problems, it need to replace with new components or event others software.

Challenge 1: IR sensor do not understand when people pass through side by side.

The IR sensor do not know how many people coming in if there are multiple people coming in together side by side. Since the nature of IR sensor is beaming out with IR blaster and receive the reflected infrared to know whether there is object in from of it or not. If there is something or someone blocking one of the sensors, then the sensor cannot work properly to indicate is there any object pass through of it.

How to overcome: There is no others way to overcome this problem yet.

Challenge 2: Normal Camera does not work well in dark.

Normal camera such as the Raspberry Pi Camera Module that are using in the system do not work well when the situation is dark. The image capture will have a lot of noise. This will decrease the accuracy of the object detection. The camera will also be failed to capture the face of the intruder more clearly compare with situation that is brighter.

How to overcome: To overcome this problem, IR camera or Night mode camera able to solve this problem since they are sensitive to light and works under a dark situation.

Challenge 3: Object detection algorithm not 100% accurate.

Object detection algorithm will detect the object wrongly, such as “People” when there is other object that looks similar in the area. This is due to the indoor area such as a room or inside a factory there are many different types of object, it will cause the algorithm to detect other object as “People” which is the things that this project wishes to trace and detect only. Moreover, for the detection model that used in this project, it is not specially trained for people detection, so it might have chance to face problems when detecting other things.

How to overcome: To overcome this problem, we can train the model by ourselves based on the situation instead of using the model that provided online. This can prevent the other object detected wrongly by the object detection.

6.4 Objectives Evaluation

The objectives of the project have mention earlier in Chapter 1 which are:

1) To implement an intrusion detection security system by using Raspberry Pi together with various IoT sensors.

   This objective is fully achieved since the project was well implemented (an intrusion detection security system) and was tested at an indoor area. A Raspberry Pi 3B+ was used as a central control unit to collect the measured data from various sensors. 2 IR sensors and 1 Door Window Sensor Magnetic Switch were interfaced to detect the surrounding. The Door Window Sensor Magnetic Switch was connected to the NodeMCU Lua v3 ESP8266 that has WiFi build-in which allows the data from it to be sent to the central control unit wirelessly.

2) To carry out a performance study for the connectivity between the sensor, Raspberry Pi, and communication platform.

   This objective is fully achieved. In Chapter 6, performance study for the connectivity between sensor and Raspberry Pi and connectivity between Raspberry Pi and communication platform have been carried out. From the testing, we could study the performance on how the sensor and Raspberry Pi work and how good is the Raspberry Pi communicate with the other communication platform.

3) To enhance the system robustness by reducing the false alarm issue on system that using IoT sensor.

   This objective is fully achieved as well. An object detection mechanism was implemented in the project. The object detection can increase the robustness of the system because the system was able to process and analyse the surrounding by taking a photo with the minimum time interval which is 2 seconds. This is to ensure if there is a person is detected then it will only send the notification to the users. Hence, the false alarm issue can be reduced.

6.5 Concluding Remark

In this chapter, it shows the performance study and system testing for the system. The project challenges that cannot be solved at this moment and the evaluation of the objectives is shown in this chapter.

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

Security is the state of being freedom from any danger. Home security system or property security system is something that we need to protect our home or workplace every day in our life. There are a lot of system are selling it at higher price which not most of the people can afford. There are many of the security system available in the market. However, they are mainly focus at Outside, which implemented on the Door Security. It is vulnerable to destroy on purpose. In this project, we implemented the system as Intrusion Detection Security System which detect whenever there is intrusion happened. It will silently detect the intruders while notify the user whenever there is intrusion happened.

From the proposed idea, we planned to use 2 different sensors which is IR sensors and Magnetic Sensor to detect the surrounding. When the door is opened, the camera will be turned on too to capture the surroundings. Then the object detection will analyze the photo. If the object detection detected a person, it will send the notification and photo to the user. For other data such as IR sensor and magnetic sensor, it will be uploaded to the cloud for user to monitor the data. At the end of the project, it working as expected and successfully solved the problem that faced during the development of the project.

The system that developed also reach the expectations of the objective that proposed at the beginning of the planning. In this project, performance study has been done to understand the connectivity of the components and module and it able to reduce the robustness of the system with the object detection to reduce false alarm happened.

7.2 Recommendation

There is some recommendation to improve this Intrusion Detection Security System to make sure that the system improved in the future:

- User can alert the intruder by press a button on the Dashboard or in Telegram and make sound in the indoor area with a buzzer.
- The camera can be replaced with the night mode camera which capable of capturing photo in the dark situation.
- Implement counter in the system with the object detection, so that the user will know there is how many people inside the room.
- Replace the IR Sensor with Ultrasonic Sensor to have a longer range and will not affected by the sunlight.
- Allow the user to check the last update on the IR sensor from Telegram.

REFERENCE

Arduino Project Hub. 2021. ESP8266 - Beginner Tutorial + Project. [online] Available at: <https://create.arduino.cc/projecthub/Niv_the_anonymous/esp8266-beginner-tutorial-project-6414c8> [Accessed 6 August 2021].

Andreas, Aldawira, C., Putra, H., Hanafiah, N., Surjarwo, S. and Wibisurya, A., 2019. Door Security System for Home Monitoring Based on ESP32. *Procedia Computer Science*, 157, pp.673-682.

Aqeel, A., 2018. *Introduction to Arduino Uno - The Engineering Projects*. [online] The Engineering Projects. Available at: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html> [Accessed 3 April 2021].

Dosm.gov.my. 2020. *Department of Statistics Malaysia Official Portal*. [online] Available at: <https://www.dosm.gov.my/v1/index.php?r=column/cthemeByCat&cat=455&bul_id=UFZxVnpONEJqUU5pckJIbzlXeEJ1UT09&menu_id=U3VPMldoYUxzVzFaYmNkWXZteGduZz09> [Accessed 3 April 2021].

Existek.com. 2017. *SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral*. [online] Available at: <https://existek.com/blog/sdlc-models/> [Accessed 3 April 2021].

Findlaw. n.d. *Property Crimes - FindLaw*. [online] Available at: <https://www.findlaw.com/criminal/criminal-charges/property-crimes.html> [Accessed 3 April 2021].

Flores, A., 2018. *TOP FIVE USE CASES OF TENSORFLOW (DEEP LEARNING)*. [online] Linkedin.com. Available at: <https://www.linkedin.com/pulse/top-five-use-cases-tensorflow-deep-learning-anna-flores> [Accessed 3 April 2021].

GeeksforGeeks. 2021. *Software Engineering | Spiral Model - GeeksforGeeks*. [online] Available at: <https://www.geeksforgeeks.org/software-engineering-spiral-model/> [Accessed 3 April 2021].

Hiteshi. n.d. *agile vs waterfall | Hiteshi*. [online] Available at: <https://www.hiteshi.com/agile-vs-waterfall/> [Accessed 3 April 2021].

Jr, V., n.d. *The Spiral Model - The Ultimate Guide to the SDLC*. [online] The Ultimate Guide to the SDLC. Available at: <https://ultimatesdlc.com/spiral-model/> [Accessed 3 April 2021].

M. Mehra, V. Sahai, P. Chowdhury and E. Dsouza, "Home Security System using IOT and AWS Cloud Services," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 2019, pp. 1-6, doi: 10.1109/ICAC347590.2019.9089839.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. *You Only Look Once: Unified, Real-Time Object Detection*. [online] arXiv.org. Available at: <https://arxiv.org/abs/1506.02640> [Accessed 3 April 2021].

Simplilearn.com. 2021. *What is Tensorflow: Deep Learning Libraries and Program Elements Explained*. [online] Available at: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow> [Accessed 3 April 2021].

S. Tanwar, P. Patel, K. Patel, S. Tyagi, N. Kumar and M. S. Obaidat, "An advanced Internet of Thing based Security Alert System for Smart Home," 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, 2017, pp. 25-29, doi: 10.1109/CITS.2017.8035326.

Team, U., n.d. About | Ubuntu MATE. [online] Ubuntu MATE. Available at: <https://ubuntu-mate.org/about/> [Accessed 3 April 2021].

TensorFlow. n.d. *Case Studies and Mentions | TensorFlow*. [online] Available at: <https://www.tensorflow.org/about/case-studies> [Accessed 3 April 2021].

www.dictionary.com. n.d. *Definition of security | Dictionary.com*. [online] Available at: <https://www.dictionary.com/browse/security> [Accessed 3 April 2021].

APPENDIX A – IMAGE OF OBJECT DETECTION

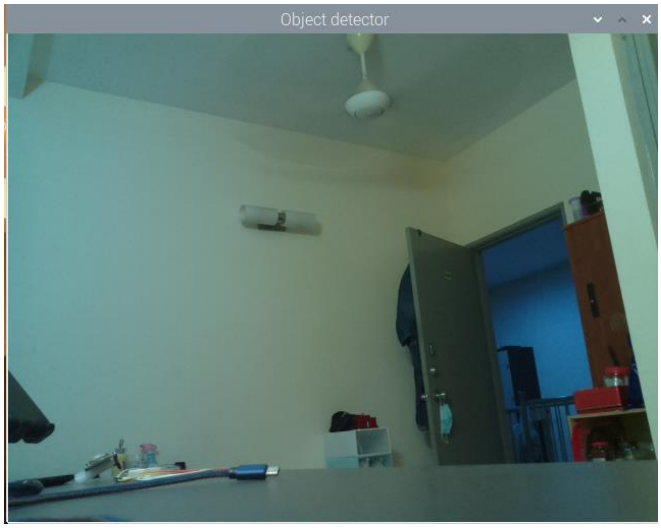


Figure A-1: Detection Result 01

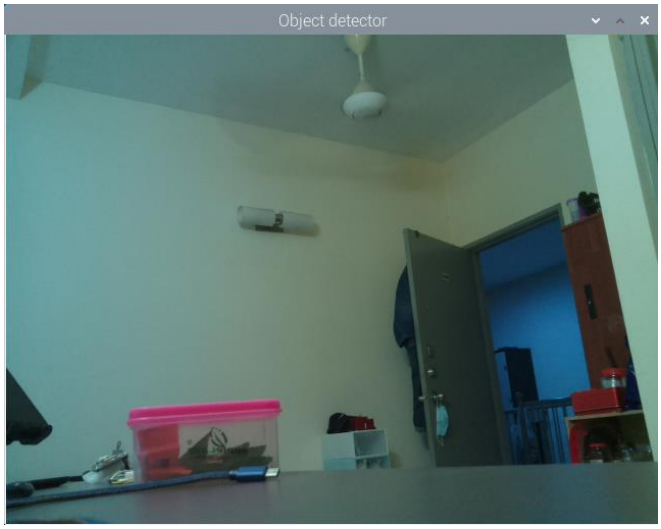


Figure A-2: Detection Result 02

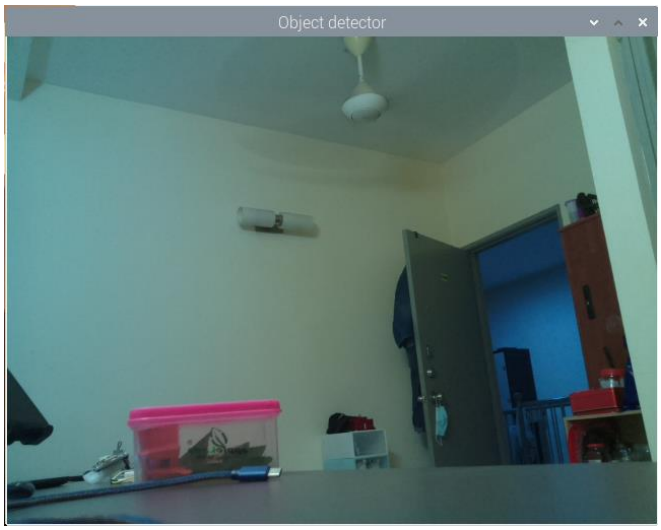


Figure A-3: Detection Result 03

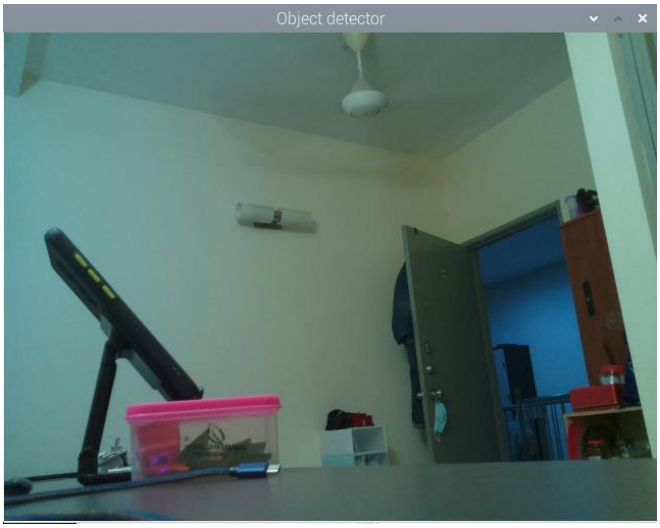


Figure A-4: Detection Result 04

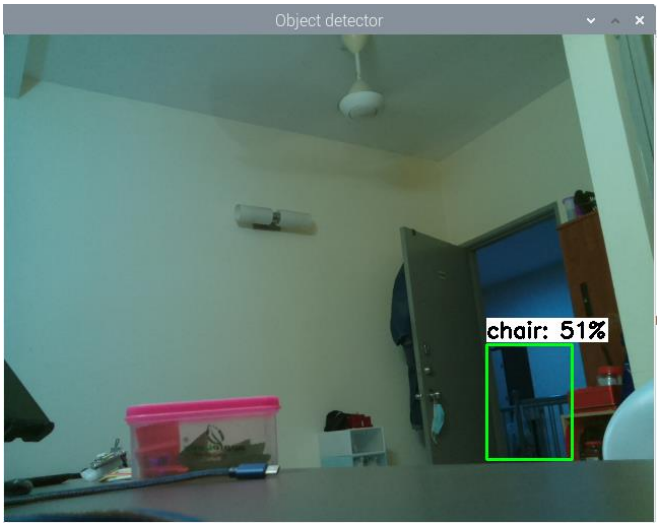


Figure A-5: Detection Result 05

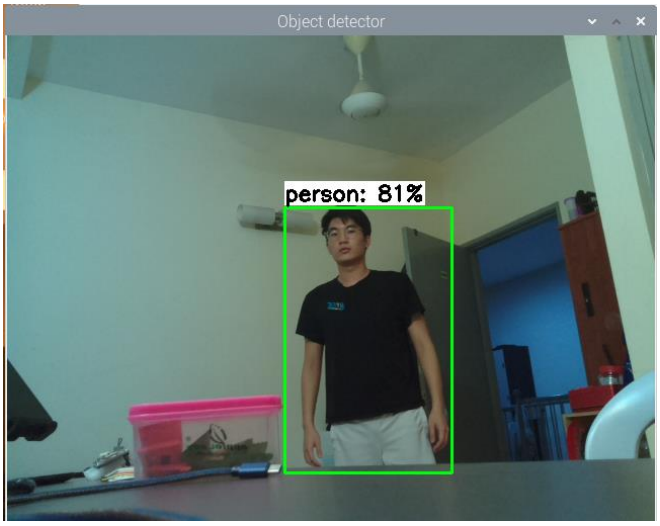


Figure A-6: Detection Result 06

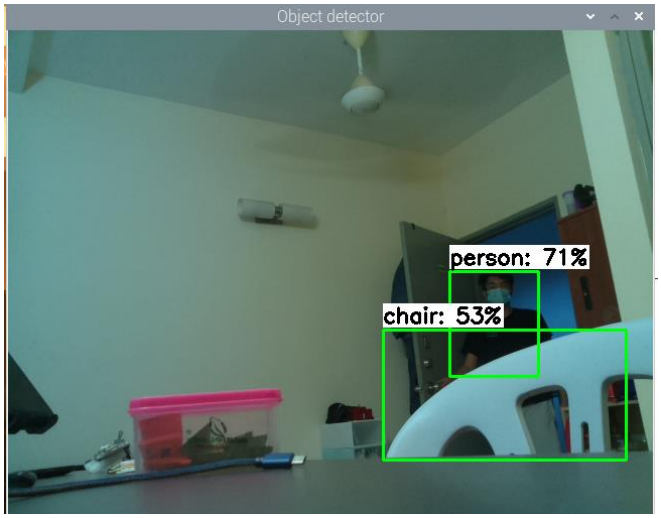


Figure A-7: Detection Result 07

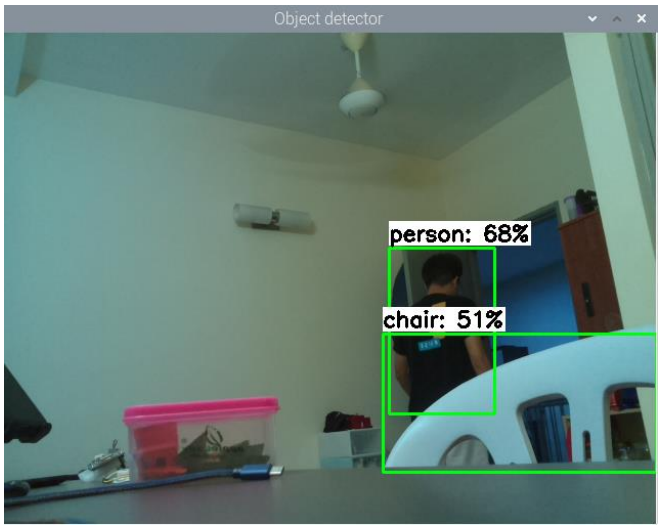


Figure A-8: Detection Result 08

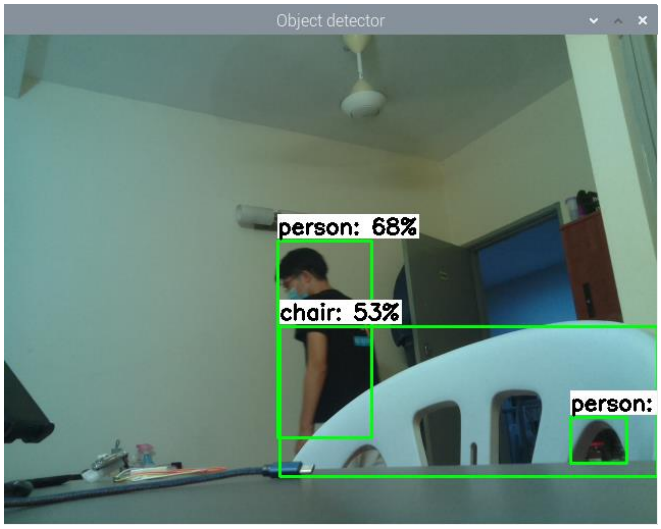


Figure A-9: Detection Result 09

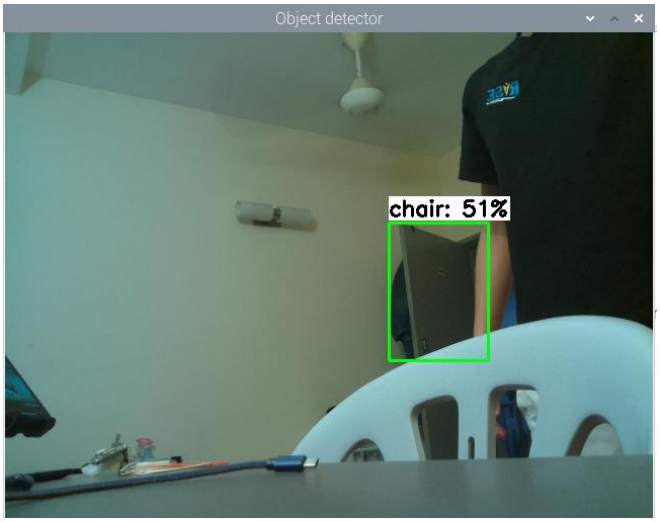


Figure A-10: Detection Result 10

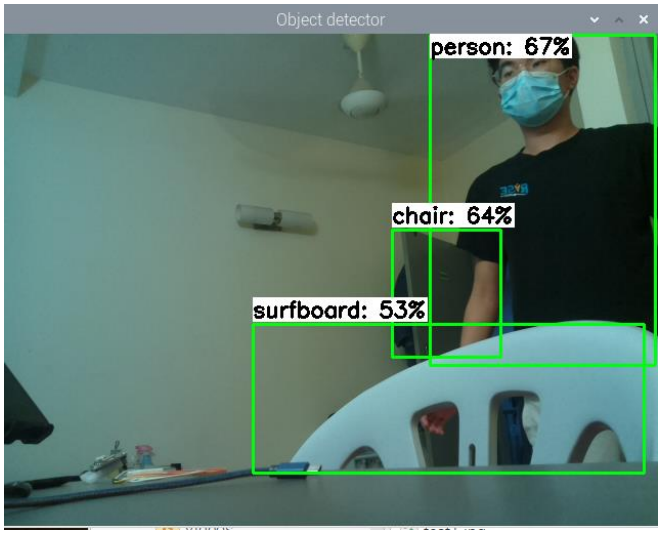


Figure A-11: Detection Result 11

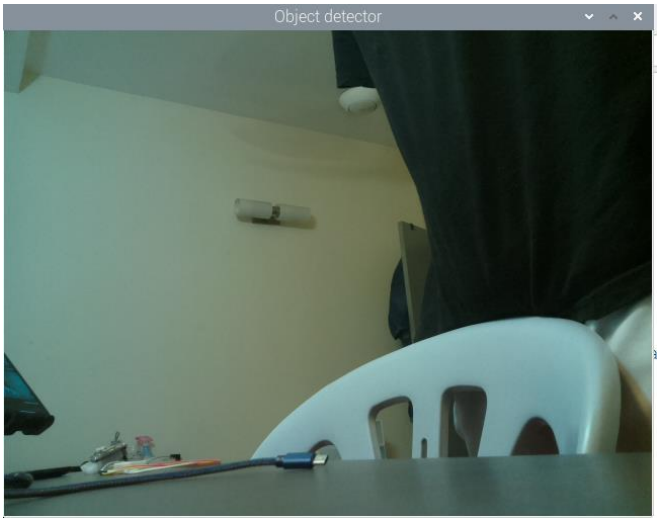


Figure A-12: Detection Result 12

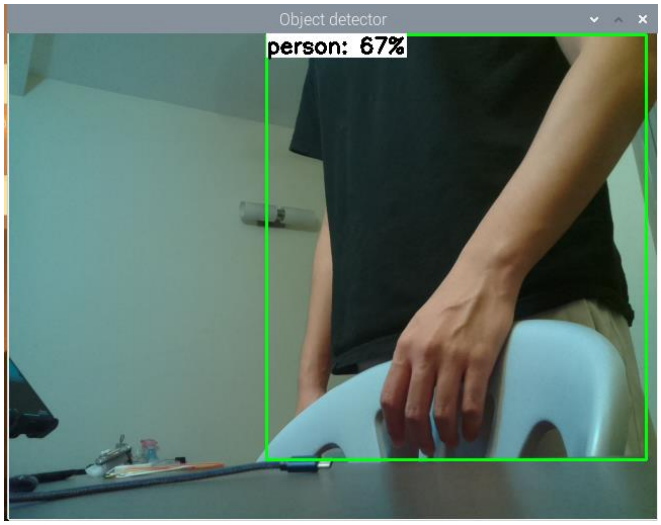


Figure A-13: Detection Result 13

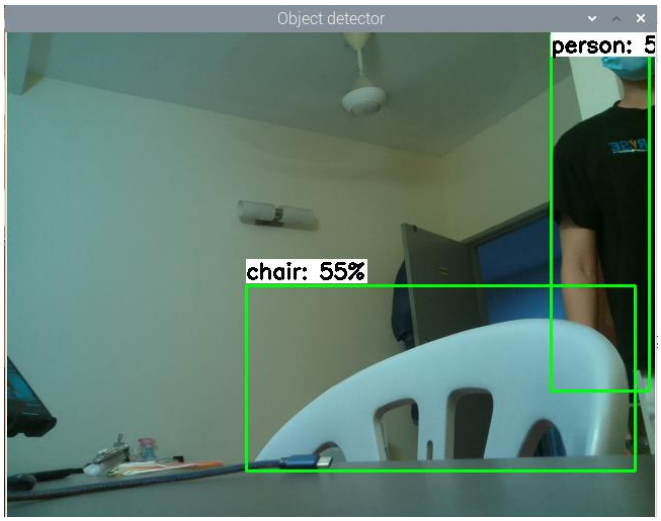


Figure A-14: Detection Result 14

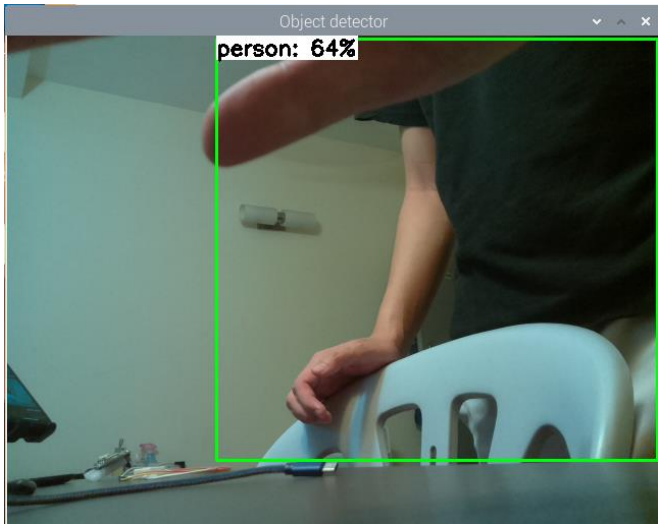


Figure A-15: Detection Result 15

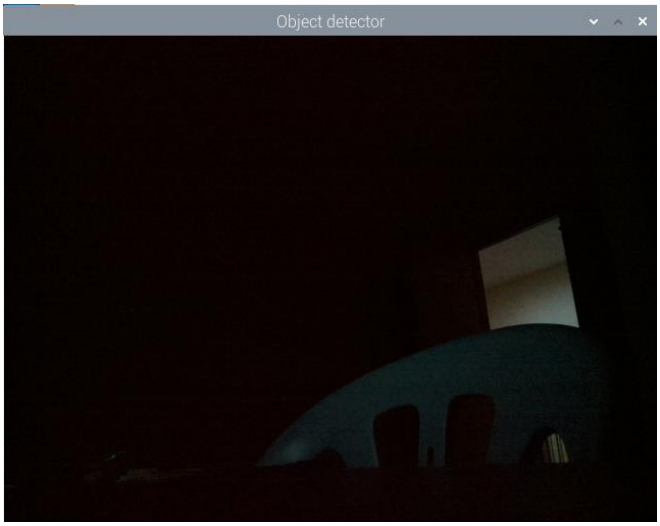


Figure A-16: Detection Result 16

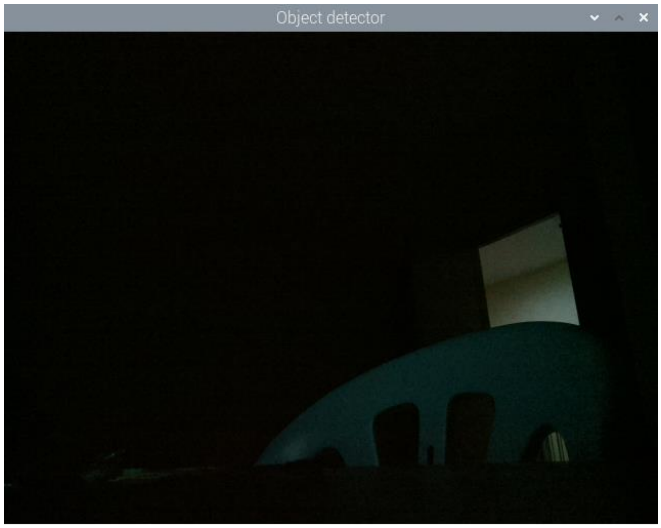


Figure A-17: Detection Result 17

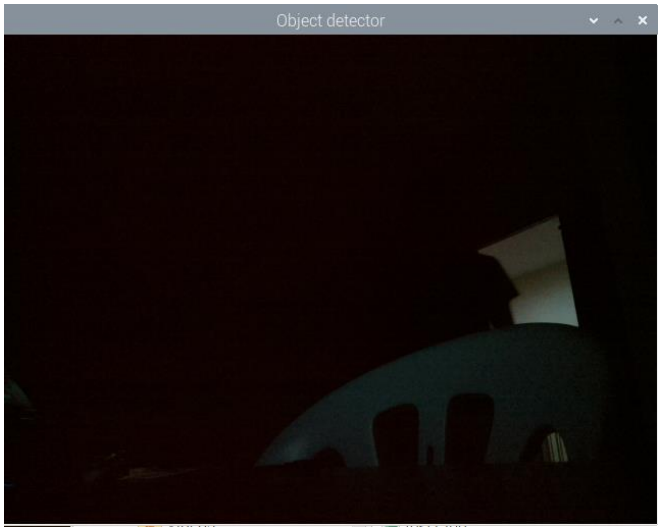


Figure A-18: Detection Result 18

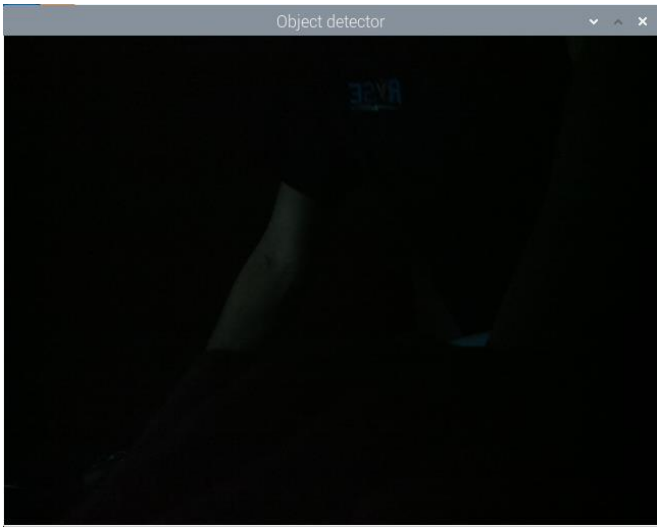


Figure A-19: Detection Result 19

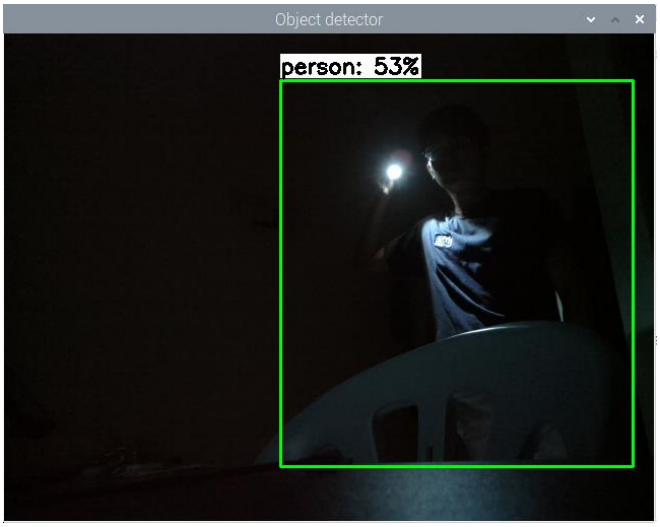


Figure A-20: Detection Result 20

APPENDIX B - BI-WEEKLY REPORT

APPENDIX B

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 1
Student Name & ID: Vernon Chean Lynn Chii 18ACB06319	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Intrusion Detection Security System with IoT Sensors	

1. WORK DONE

- Read through the format of the report format and presentation.
- Revise the code that done from FYP 1.
- Done the report for Chapter 1 to Chapter 4

2. WORK TO BE DONE

- Test with the hardware that purchase.

3. PROBLEMS ENCOUNTERED

- Camera resource overload as there is 3 modules are using the camera.

4. SELF EVALUATION OF THE PROGRESS

- Starting with progress.

Supervisor's signature

Student's signature

APPENDIX B

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 3
Student Name & ID: Vernon Chean Lynn Chii 18ACB06319	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Intrusion Detection Security System with IoT Sensors	

1. WORK DONE

- Test with the hardware that purchased.

- Revise and modify the code for FYP 2.

2. WORK TO BE DONE

- Switch the MQTT module to Cloud and Telegram notification.

- Setup WiFi access point on Raspberry Pi to collect data from ESP8266.

- Setup ESP8266 to send data to Raspberry Pi.

3. PROBLEMS ENCOUNTERED

- Communications between ESP8266 and Raspberry Pi does not work.

4. SELF EVALUATION OF THE PROGRESS

- 7/10. Hardware is tested and tried, know how it works. However, there is modification on the system that takes times, require more time and resources to make it done.

Supervisor's signature

Student's signature

APPENDIX B

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 5
Student Name & ID: Vernon Chean Lynn Chii 18ACB06319	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Intrusion Detection Security System with IoT Sensors	

1. WORK DONE

- Switched the MQTT to Cloud and Telegram Notification.

- ESP8266 able to send data to Raspberry Pi.

2. WORK TO BE DONE

- Setup the cloud for IR sensor to receive the data.

- Implement object detection into the main code.

3. PROBLEMS ENCOUNTERED

- Communications between ESP8266 and Raspberry Pi does not work.

4. SELF EVALUATION OF THE PROGRESS

- 6/10. Progress is slow, as there is too much trial and error during implementing the code. Spend too much time on learning how to communicate the ESP8266 with Raspberry Pi.

Supervisor's signature

Student's signature

APPENDIX B

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 7
Student Name & ID: Vernon Chean Lynn Chii 18ACB06319	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Intrusion Detection Security System with IoT Sensors	

1. WORK DONE

- Cloud able to receive the data of the IR sensor.

2. WORK TO BE DONE

- Planning for Testing.
- Implement Object Detection into the main code.
- Make the cloud to receive the data more accurate from the IR.

3. PROBLEMS ENCOUNTERED

- Cloud does not receive the data as planned.
- Object Detection part still unable to implement into the main code.

4. SELF EVALUATION OF THE PROGRESS

- 5/10. Spend too much time one learning the Object Detection to allow it to implement into the main code. The progress is slow, too dependent on trial and error to the script.

Supervisor's signature

Student's signature

APPENDIX B

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 9
Student Name & ID: Vernon Chean Lynn Chii 18ACB06319	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Intrusion Detection Security System with IoT Sensors	

1. WORK DONE

- Object Detection is implemented into the main code
- Cloud able to receive accurate data.
- Drafted the plan for testing.

2. WORK TO BE DONE

- Continue with the report for Chapter 5 and Chapter 6
- Start with the testing that proposed to Dr Goh.

3. PROBLEMS ENCOUNTERED

- NodeMCU Lua v3 ESP8266 firmware faced problem where it cannot work every time it is powering up.

4. SELF EVALUATION OF THE PROGRESS

- 7/10. The prototype is working fine, while the report is not having progress.

Supervisor's signature

Student's signature

APPENDIX B

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 11
Student Name & ID: Vernon Chean Lynn Chii 18ACB06319	
Supervisor: Ts Dr Goh Hock Guan	
Project Title: Intrusion Detection Security System with IoT Sensors	

1. WORK DONE

- Report for Chapter 5 and Chapter 6 is done

- Testing has been carried out

2. WORK TO BE DONE

- Presentation and Demonstration for the FYP II

3. PROBLEMS ENCOUNTERED

- Currently no problems encounter for this two week.

4. SELF EVALUATION OF THE PROGRESS

- 7/10. All things follow the progress, and it is working fine. Things that planned for this two-week done. Still haven't encounter any problem for this period.

Supervisor's signature

Student's signature

POSTER

UNIVERSITI TUNKU ABDUL RAHMAN

Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UTAR KAMPAR CAMPUS

INTRUSION DETECTION SECURITY SYSTEM USING IOT SENSORS

By: Vernon Chean Lynn Chii
Supervisor: Ts Dr Goh Hock Guan

PROJECT OVERVIEW

This project is an IoT sensor-based security that detect intrusion for indoor area. The IoT sensors are used to detect the abnormal activities in the indoor area.

However, the IoT sensors do not recognize the object that passed by is a human or not.

Hence, object detection is added as a guidance in the system to decrease false alarm trigger by the system.

PROBLEM STATEMENT

-Security System that installed at the outdoor can experience physical damage.

-People do not trust the system as false alarm happened frequently.

OBJECTIVE

- To implement an intrusion detection security system by using Raspberry Pi together with various IoT sensors.

- To carry out a performance study for the connectivity between the sensor, Raspberry Pi and communication platform.

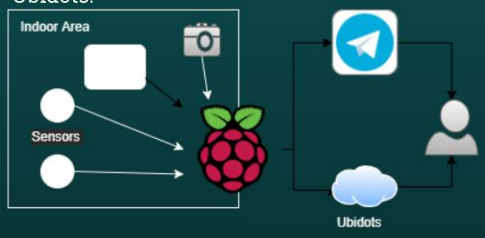
-To enhance the system robustness by reducing the false alarm issue on system that using IoT sensor.

METHODOLOGY

Agile Model. For every iteration, system will be tested. The system can break into small incremental build by building, deploy and testing each of the small build one by one.

SYSTEM ARCHITECTURE

Raspberry Pi will be collecting the sensor reading. When the door is opened, the video recording will start, and Object Detection Algorithm will check whether the object enter is a person. Then it will notify the user through Telegram. User can monitor the data through Ubidots.

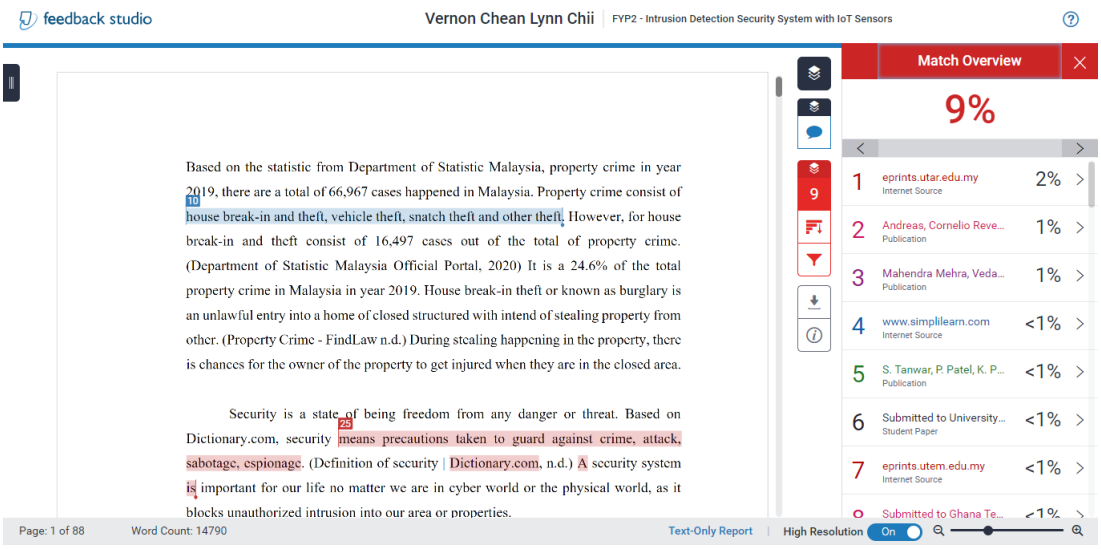


CONCLUSION

In this project, the IoT sensors are involved in the intrusion detection security system. Object detection algortihm is added to enhance the IoT sensor data to reduce false alarm.

Bachelor of Information Technology (Honours) Communications and Networking

PLAGIARISM CHECK RESULT



PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF _INFORMATION AND COMMUNICATION TECHNOLOGY_

Full Name(s) of Candidate(s)	VERNON CHEAN LYNN CHII
ID Number(s)	18ACB06319
Programme / Course	COMMUNICATIONS AND NETWORKING
Title of Final Year Project	INTRUSION DETECTION SECURITY SYSTEM WITH IOT SENSORS

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: _9_ **%** **Similarity by source** Internet Sources: _5_ % Publications: _4_ % Student Papers: _2_%	
Number of individual sources listed of more than 3% similarity: _0_	
Parameters of originality required and limits approved by UTAR are as follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr. Goh Hock Guan

Date: 20/8/2021

Signature of Co-Supervisor

Name: _____

Date: _____

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB06319
Student Name	VERNON CHEAN LYNN CHII
Supervisor Name	Ts Dr Goh Hock Guan

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. _____ (Signature of Student) Date: 17th August2021	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. _____ (Signature of Supervisor) Date: 20/8/2021