Faculty of Electrical and Electronic Engineering Technology

DESIGN AND DEVELOPMENT OF AN AUTOMATIC HAND SANITIZER WITH DOOR CONTROL SYSTEM USING NODEMCU

NURAIN BINTI AHMAD HARUN

Bachelor of Electronics Engineering Technology with Honours

2021

DESIGN AND DEVELOPMENT OF AN AUTOMATIC HAND SANITIZER WITH DOOR CONTROL SYSTEM USING NODEMCU

NURAIN BINTI AHMAD HARUN

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Electronics Engineering Technology with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I declare that this project report entitled "Design and Development of an Automatic Hand Sanitizer with Door Control System using NodeMcu" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : NURAIN BINTI AHMAD HARUN

Date : 10/1/2022

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering Technology with Honours.

Signature :

Supervisor Name : AHMAD FAIRUZ BIN MUHAMMAD AMIN

Date : 11th JANUARY 2022

Signature :

Co-Supervisor :

Name (if any)

Date :

DEDICATION

This project is dedicated to Allah, God Almighty, my creator, my pillar of strength, my source of inspiration, wisdom, knowledge, and understanding. I also dedicated this project to my beloved parents and friends, who have supported me throughout the process and ensured that I give it everything I have to finish what I have started. To my best supervisor (Ts Ahmad Fairuz), who has been impacted by this quest in every way possible. Thank you very much. My love for you all is unquantifiable. God's blessings on you.

DEDICATION

This project is dedicated to Allah, God Almighty, my creator, my pillar of strength, my source of inspiration, wisdom, knowledge, and understanding. I also dedicated this project to my beloved parents and friends, who have supported me throughout the process and ensured that I give it everything I have to finish what I have started. To my best supervisor (Ts Ahmad Fairuz), who has been impacted by this quest in every way possible. Thank you very much. My love for you all is unquantifiable. God's blessings on you.

ABSTRACT

In the current era of globalization, the "Internet of Things," synonymous with the term "IoT," is becoming increasingly popular as technology experts and people talk. This is because it is conceptualized where machines, sensors, equipment, and devices are connected to the Internet. This paper uses the Internet as an intermediary between users, systems, and devices. This allows the system to be controlled and accessed by administrators to detect if employees are experiencing symptoms of an increasingly contagious disease that is a coronavirus. Many consumers often take lightly about sanitation practices and body temperatures above normal limits because they feel such things are normal. According to the highest cluster statistics in Malaysia on this epidemic, the employment sector cluster is the highest cluster to get the subsequent epidemic infections. This system is specially designed for a premise with 20-30 employees. In this system, each employee has a unique code found on their RFID smart card to store their data. Whenever workers want to enter the premises, they check their body temperature and stick their smart cards on the sensor. Workers at a normal body temperature can access the entrance and sanitize their hands first and vice versa. The employee's date, time, and body temperature will always be recorded each time the employee sticks their smart card on the sensor. With this, the admin can detect employees who have symptomatic symptoms, but the admin can also see the movement of employees in and out at the main door of the premises.

ABSTRAK

Di era globalisasi kini, "Internet of Things" atau sinonim dengan sebutan "IoT" semakin popular menjadi bualan pakar teknologi dan rakyat. Hal ini kerana ia berkonsepkandi mana mesin, sensor, peralatan dan peranti dihubungkan dengan internet. Di dalam kertaskerja ini, Internet digunakan sebagai pengantara antara pengguna, sistem dan peranti, ini membolehkan sistem dikawal dan diakses oleh pentadbir untuk mengesan sekiranya terdapat pekerja yang mengalami gejala daripada penyakit yang kian menular iaitu coronavirus. Ramai pengguna sering mengambil ringan tentang amalan sanitasi dan suhu badan melebihi daripada had normal kerana mereka merasakan perkara tersebut adalah normal. Mengikut statistik kluster tertinggi di Malaysia mengenai wabak ini, kluster sektor pekerjaan merupakan kluster tertinggi untuk mendapat jangkitan wabak berikut. Sistem ini di reka khas untuk sebuah premis yang mempunyai 20-30 pekerja. Dalam sistem ini, setiap pekerja mempunyi kod unikyang terdapat pada kad pintar RFID mereka yang digunakan untuk menyimpan data mereka.Setiap kali pekerja ingin memasuki premis, pekerja harus periksa suhu badan mereka pada sensor dan menempelkan kad pintar mereka pada sensor. Pekerja yang berada pada suhu badan normal dapat mengakses pintu masuk dan akan mensanitasi tangan mereka terlebih dahulu dan sebaliknya. Tarikh , waktu dan suhu badan pekerja akan sentiasa direkod setiap kali pekerja menempelkan kad pintar mereka pada sensor. Dengan ini, admin bukan sahaja dapat mengesan pekerja yang mempunyai symptom bergejala, admin juga dapat mengesan pergerakan keluar masuk pekerja di pintu utama premis.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts Ahmad Fairuz for their precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for the financial support through financial claims which enables me to accomplish the project. Not forgetting my fellow colleague, Anis Nadhirah for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents, and family members for their love and prayer during the period of my study also for all the motivation and understanding.

Finally, I would like to thank all the fellow colleagues and classmates, the Faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

°	-	Degrees
°C	-	Degrees Celcius
cm	-	centimeter
mm	-	milimeter
V	-	Volt

LIST OF ABBREVIATIONS

AC	-	Alternating Current
DC	-	Direct Current
ESP	-	Espressif
EEPROM	-	Eletrically Erasable Programmable Read Only Memory
GND	-	Ground
GPIO	-	General Purpose Input/Output
HTTP	-	Hypertext Transfer Protocol
ID	-	Identity Document
IDE	-	Integrated Development Environment
LED	-	Light Emitting Diode
LCD	-	Liquid Crystal Display
MHz	-	Megahertz
OTP	-	One Time Password
PWM	-	Pulse Width Modulation
RFID	-	Radio Frequency Identification
SPI	-	Serial Peripheral Interface
UART	-	Universal Asynchronouse Receiver Transmitter
V	-	Voltage
VCC	-	Voltage Common Collector

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

According to WHO (World Health Organization) and Malaysia Ministry of Health (KKM), hand hygiene is the most critical prevention measure in the COVID-19 pandemic, a global epidemic, and entails often washing hands with water and soap and hand sanitizing using hand sanitizer. We must practice proper handwashing to remove the majority of germs on our hands. Alcohol-based hand sanitizers are preferred over soap and water in most healthcare settings because they are easier to absorb and are more efficient at removing bacteria.

The smart sanitizer will be placed at the main entrance along with a temperature sensor. Users need to scan RFID cards first, and then the temperature sensor will detect user body temperature. If the user's body temperature is average, the door will open automatically to allow users to enter the premises. If user temperature exceeds expected levels, a red LED will light up and warn users cannot enter the premises. Users who are allowed to enter the premises will place their hands on the sensor, and the water pump will automatically spray the gel sanitizer onto their hands. Finally, each time a user scans at the RFID scanner, the server will receive the user detail, and the admin can read the user's data entering the premises.

1.2 PROBLEM STATEMENT

Limb sanitation has become crucial since the covid-19 pandemic, radically changing people's practices and lifestyles worldwide, including in our country. Sanitation can help to prevent illness and bacterial infections from spreading. Even though the government has made sanitation necessary, the community often does not take sanitation seriously before entering any facilities or dwellings because they believe they are not disease agents. There are many automatic hand sanitizers on the market now. However, they merely disinfect our hands and do nothing else.

They are beneficial, but we require something more. Combining an automatic hand sanitizer and a door control system can help solve this problem, particularly in businesses and offices. The industrial sector is the highest cluster that is easily infected with the virus. It was designed to limit human mobility before approaching specific locations, such as the main entrance. As a result, we devised a strategy for tracking employee attendance by scanning RFID tags and automatically measuring body temperature. While they sanitize their hands, the system will save all data in a database, which the administrator will access and track each employee. This initiative will aid in the endeavor to fix the problem without requiring anyone to touch anything. Other automated sanitizers are not like this.

1.3 PROJECT OBJECTIVE

The objective of this project is:

a) To develop smart attendance using RFID smart card and contactless body temperature connected with server and sensor.

b) To develop smart hand sanitizing for industry based on microcontrollers.

c) To develop an automatic door controller system using multisensory.

1.4 SCOPE OF PROJECT

The scopes of work for the project include the following areas:

Suitable for offices with staff not exceeding 30 employees. It is much safer and recommended due to its contactless nature and avoids the risk of infectious disease outbreaks.

This device is simple to operate and may be used by anyone. For example, if you are an employee, first place your RFID tag near the RFID scanner, then the thermal temp sensor will check your body temperature. If it is normal, it will command the smart sanitizer and place your hand under the smart sanitizer, and the sanitizer outlet will drop some amount into your hands. Then the microcontroller activates the servo motor, which opens the main entrance.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, this literature review aims to examine and conduct research on previously completed projects involving automatic sanitizers with door control systems and smart attendance systems. The research is based on a study that does not exceed five years from the project's implementation date. It is suitable to be used as a reference. in addition, the suitability of the search method for the implemented project will be discussed, reviewed, and determined. The pros and cons of hardware and system are crucial to evaluate, and it will be examined so that the selection hardware used is more accurate during assembly in this project. Finally, previous research findings and approaches will be compared and analyzed to ensure that they are the best and most reliable reference source for this project.

2.1.1 Introduction of Node MCU ESP-8266

The ESP-8266 is a Wi-Fi module with an IP/TCP protocol stack containing a complete or self-contained chip (SOC). Via Wi-Fi link, the ESP-8266 gives access to any microcontroller and can host any program or offload all Wi-Fi networking functions. It is highly long-lasting and can withstand even the most demanding industrial settings. This microcontroller has powerful onboard processing and storage capabilities, allowing it to be used with sensors and other applications. This microcontroller also has a self-calibrated radio frequency (RF) that will enable it to operate in any operating situation without the need for radio frequency components.

Figure 2.1: ESP-8266 microcontroller [1]

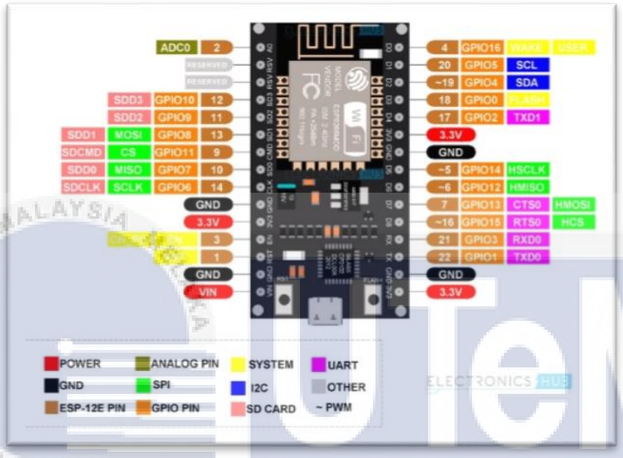


Figure 2.2: ESP-8266 Pinout Reference[2]

2.1.2 Introduction of Node MCU ESP-32

This microcontroller is an advanced version of the ESP-8266. ESP 32 also has a low-power co-processor and has a dual-core. It is designed for the lack of security in ESP-8266. This microcontroller offers a dual-core or clock frequency 160MHz to 240MHz. With the ESP 32, users may manage and monitor their devices over Wi-Fi or Bluetooth at a meager cost. The ESP-32 has a 32-bit processor with an extremely low energy co-processor and a couple of input/output connectors, incorporating digital-to-analog converters. For IoT, the ESP-32 is a reliable platform. This microcontroller features a temperature sensor and writes a 1024-bit OTP with PWM (soft) 16. The ESP-32 has 10 bits of sensors.

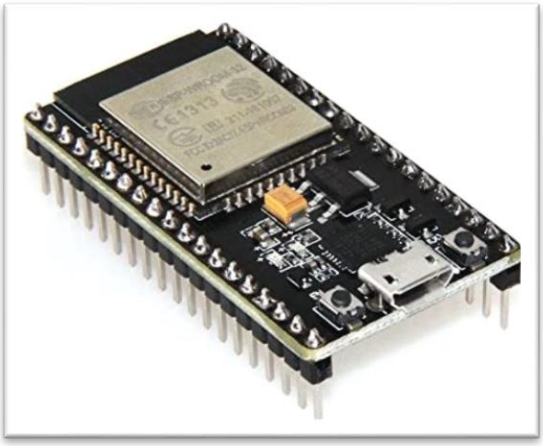


Figure 2.3: ESP-32 microcontroller[3]

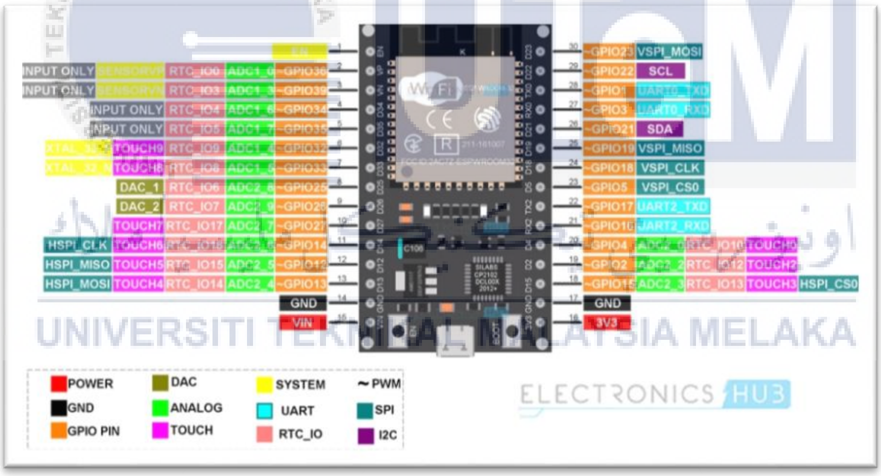


Figure 2.4: ESP-32 Pinout Reference [4]

2.1.3 Comparison of ESP-8266 and ESP-32

The ESP-32 and the ESP-8266 [5]SOCs are Wi-Fi-centric (Systems on Chop). There is a 32-bit CPU on both microcontrollers, with the ESP-8266 running at 160MHz single-core and the ESP-32 running at 80MHz to 240MHz twin core. Both the ESP-8266 and ESP-32 are unique toolkits. The ESP8266 uses less power than its competitor, yet one has more digital pins. ESP32 systems have more GPIO, allowing them to work with more complex and usable projects. It is more appropriate for any application that needs a microcontroller. Many ESP32 development boards have small cameras, and the board is competent. The following table provides a more detailed comparison:

Table 1 : Comparison of ESP8266 and ESP32

Feature	ESP-32	ESP-8266
Bluetooth	BLE & Bluetooth 4.2	No
SRAM	Yes	No
MCU	600 DMIPS Xtensa Dual-core 32-bit	L106 Xtensa Single-core 32 bit
Frequency of use	160MHz	80MHz
Software or Hardware PWM	16 channels/none	8 channels/non
Flash	Yes	No
ADC	Has 12 bit	Has 10 bit
GPIO	Has 34	Has 17
Sensor of temperature	Has (old version)	No
Sensor of touch	Yes	No
Wi-Fi 802.11 b/g/n	HT40	HT20
Sensor with a half effect	Yes	No

MAC Interface for Ethernet	Yes	No
CAN	Yes	No

2.1.4 Introduction of RFID RC522 Module

As we all know, RFID stands for Radio Frequency Identification, the non-contact wireless transmission of data using radiofrequency waves. The NXP MFRC522 controller is used in the RC522, a 13.56MHz RFID module. The module includes an RFID card and supports I2C, SPI, and UART. It's commonly used in time-keeping systems and other applications that require people or objects to be identified.

The RC522 is a radio frequency module with a key chain, an RFID reader, and an RFID card. The module runs in the 13.56MHz industrial (ISM) band. Hence no license is required. The module is widely used in 3.3V designs since it runs at that voltage. It's most commonly employed when a unique ID is needed to identify a particular person or thing.

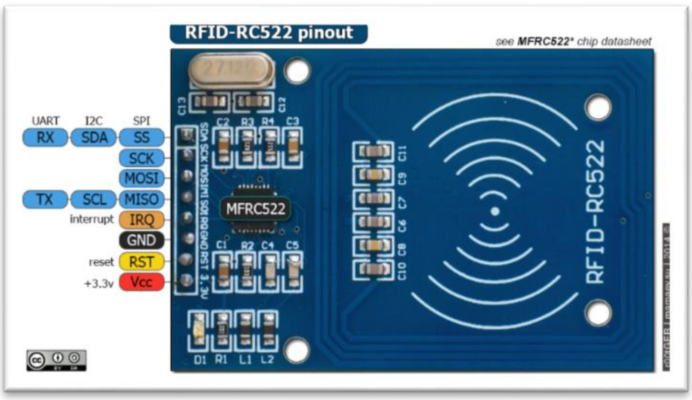


Figure 2.5: RFID RC522[6]

The keychain has 1 KB of memory, which can store one-of-a-kind information. Data can be read and write into these memory elements using the RC522 reader module. The reader can read only passive tags with a frequency of 13.56MHz. I use RFID RC522 to apply a smart attendance system and access control system in my project.

2.2 TECHNOLOGY DEVELOPMENT

2.2.1 ESP-32 with RFID: Access Control

Based on a project proposed by (Fernando Koyanagi,2018) [7], this project aims to create a program to read or write data to an RFID card. An RFID-RC522 module and a Wi-Fi NodeMCU-32S are used. A transceiver with a decoder, an antenna, and a transponder are the essential components of an RFID device in terms of operation. These cards have a reel integrated into them. As we approach them from the reader, they send out a radio signal via the antennae attached to the reader. The RFID reader receives the data from the tag, decodes it, and sends it to the server. This type of chip has 1k of memory inside it for the memory part. The EEPROM memory is divided into 16 sectors, each with four blocks containing 16 bytes.

The ESP-32 is then used as a microcontroller because it already has RF, Wi-Fi, and Bluetooth capabilities. Placing the keyring over the player reveals the options of 0 for data reading and 1 for data recording, all powered by USB and connected in the serial of the Arduino IDE; placing the keyring over the player reveals the options of 0 for data reading and 1 for data recording; placing the keyring over the player reveals the possibilities of 0 for data reading and 1 for data recording; placing the keyring over the player reveals the options of 0 for data reading and 1 for data recording; placing the keyring, We demonstrate that the reader recognized the number if the green led lights up after reading the chip or card. A red led indicates a problem, and that authentication was not completed have the chip keychain and the card. During the analysis of a card or chip, the green LED will light up, indicating that the identification has been finished and access has been authorized; however, when the red LED lights up, the data has not been validated.

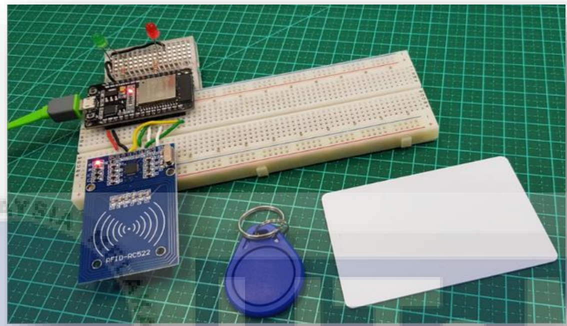


Figure 2.6: Hardware Component requirements for Access Control[7]

2.2.2 RFID based Contactless Body Temperature Screening Using Arduino and MLX90614 IR Temperature Sensor.

Based on a project proposed by Ashih Choudhary,2020)[8], this program aims to build an RFID-based Contactless Temperature Monitoring System and an Arduino. A contactless temperature sensor is used to monitor the temperature. When employees scan the RFID card, a non-contact infrared thermometer measures their body temperature and logs the employee's name and temperature directly to an excel sheet.

The Arduino Nano, MLX90614, EM18 RFID Reader, and Ultrasonic Sensor will construct this project. The distance between the thermometer and the person is calculated using the ultrasonic sensor. When the distance between a user and the thermometer is less than 25 cm, the thermometer will only measure the temperature.

It's similar to an RFID-based attendance system that keeps track of each person's body temperature.

The MLX90614 is a non-contact infrared thermometer that measures temperatures from -70 to +380°C. The MLX90614 is one such sensor that detects an object's temperature using infrared energy. To calibrate the object temperature value, the sensor measures both the object temperature and the ambient temperature. Connect the four leads to the board. The user has an accurate thermometer with a resolution of 0.01 and a precision of 0.5 degrees, or the user can use any microcontroller with an I2C interface to communicate with it**.**

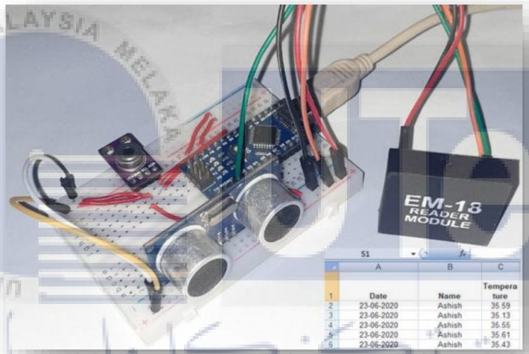


Figure 2.7: RFID Based Contactless Temperature Monitoring System using Arduino[8]

2.2.3 Automatic Hand Sanitizer Dispenser with COVID19 Live Updates

Based on a project proposed by (Ashih Choudhary,2020)[9], this project aims to create an automatic hand sanitizer dispenser with an LCD that shows the number of Coronavirus cases alive. The ESP32, Ultrasonic Sensor, 16x2 LCD Module, Water Pump, and Hand Sanitizer will be used in this project. They're getting live data from people who have been infected with Covid19 using Esri's API Explorer. An ultrasonic sensor detects the presence of hands beneath the sanitizer machine's outlet. It will continuously calculate the distance between the sanitizer outlet and itself, instructing the ESP to activate the pump if space is less than 15cm, allowing the sanitizer to be pushed out.

An ESP32, a Wi-Fi module that can readily connect to the Internet, serves as the central controller. The user must obtain data from the Internet and feed it to an ESP32 shown on a 16x2 LCD. An HTTP get request is used to retrieve the JSON file from the Internet. The water pump is connected to the ESP32 through a relay module. The Vcc and GND pins of the relay are linked to the ESP32's Vin and GND pins, respectively, and the relay's input pin to the ESP32's D19 pin. The Trig and Echo pins of the Ultrasonic sensor are connected to the Arduino's D5 and D18 pins.

To begin, include all of the necessary library files in the code. The HTTPClient library is used to get data from the HTTP server. The ArduinoJson library is used to phrase the data arrays. We'll use an ultrasonic sensor to calculate the distance inside the void ultra(continuously) method in the code program. If it's less than or equal to 15 cm, it'll turn on the pump for 2 seconds to force the sanitizer outside through the pipe. When a person places his hands below the outlet pipe, the distance between them narrows, causing the pump to activate.

Figure 2.8: COVID19 Live Tracker and Automatic Hand Sanitizer Dispenser[9]

2.2.4 Smart Sanitizer with Auto Lock Control System

Based on the project proposed by (Smit Babariya 2020)[10], This project aims to develop a low-cost smart hand sanitizer dispenser with a door controller. It is based on the ESP-32 (Microcontroller), Thermal temperature sensor, Ultrasonic rangefinder sensor (an ultrasonic sensor is used to check the presence of hands below the outlet of the sanitizer machine), and RFID (attendance if you are an employee) and can help security guards solve challenges at various stations such as main entrance doors, office gates, and so on.

The sanitizer outlet will drop some amount into the user's hands when they put their RFID tag near the RFID scanner and then place their hand under the smart sanitizer. The thermal temp sensor will check body temperature at that time. If it is expected, the microcontroller will instruct the servo motor to open the entrance door and enable a person to enter while illuminating a green LED.

Otherwise, the door will not unlock or open. Still, both LEDs will continue to blink, conveying the user's entry time, date, and body temperature data to a sheet storing all personal data. As a result, the administrator will use a spreadsheet to keep track of all personnel information. In this project, the smart hand sanitizer can also be integrated with an attendance system.

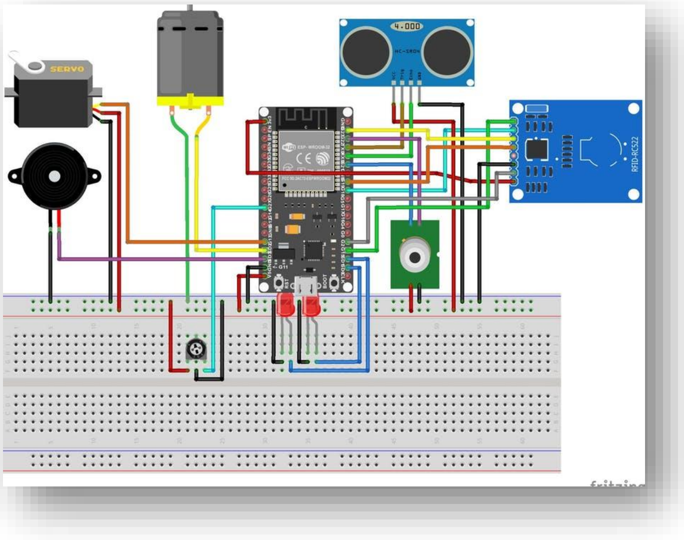


Figure 2.9: Smart Hand Sanitizer Hardware connection simulation[10]

2.3 SUMMARIZATIONS AND COMPARISON FROM THE PREVIOUS PROJECT

Based on several projects that have been implemented in the literature review that has been studied, the smart sanitizer project is a project that is primarily new in the field of IoT, especially when there is a pandemic coronavirus 19 that has hit the world. The previously built and designed project makes the public aware of sanitation practices to prevent exposure to the virus. For example, a project implemented by Ashih Choudhary and Smit Babariya emphasizes consumers for sanitation first. They use the ESP-32 microcontroller to control all of their project functions. This project has been inspired to modify the following tasks from their implemented projects to many uses. This project added a section to detect a person's physical body temperature using the MLX90614 IR temperature sensor like the project implemented by Ashih Choudhary.

Next, this project got the inspiration to improve by adding an attendance system using RFID Smart Card like the RFID -related project implemented by Alsan Parajuli and Fernando Koyanagi. They made an arrival system using RFID Smart Card. The projects implemented have inspired me to build a smart sanitizer project with a door control system using smart attendance. It is to make sure the employees are in normal temperature conditions (< 37°C) and sanitize their hands before entering the premises. If their body temperature exceeds the normal temperature, they cannot enter the premises. In addition, the admin always gets updates on the details of workers entering and leaving the premises.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter will go through the project implementation approach, which contains a project block diagram and a project flow chart. To provide a complete understanding of the project's flow, each function will be shown as a flow chart. In addition, the methods for joining the networks will be included, and circuit diagrams, which will be discussed. The use of tools like the Arduino IDE, and Thingspeak will be discussed after that. To ensure that the smart sanitizer system works as planned, this methodology is essential.

3.2 PROJECT WORKFLOW

To identify and ensure that a project's effectiveness runs smoothly, creating an organized and efficient workflow chart is crucial. Making a well-organized and planned plan is essential to a successful endeavor. The next stage is to conduct research when you have completed your planning. To make things easier, a project must be implemented. Because of thorough research, every potential issue can be identified and avoided during project implementation. As a result, the project's design is created, followed by the project's implementation. The project was then analyzed to determine its effectiveness after it was completed.

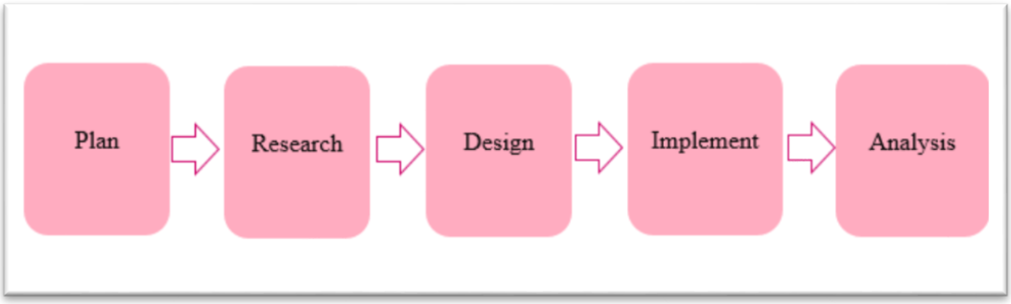


Figure 3.1: Project workflow

3.2.1 Planning

The first step in the planning process is to define project objectives so that the project's implementation goals are clear. This project aims to create a smart attendance with automatic sanitizer that includes door control system factors. To meet the objective criteria that have been set, some aspects are considered and reviewed. The Gantt chart was created to help the project be completed promptly. The project's scope is also designed so that the project's limits can be identified.

3.2.1.1 Gant Chart PSM 2

The Gantt Chart will assist in completing this PSM1 from day one, when I am briefed on the PSM1, until the day of the presentation, when everything will be arranged and represented in the Gantt Chart.

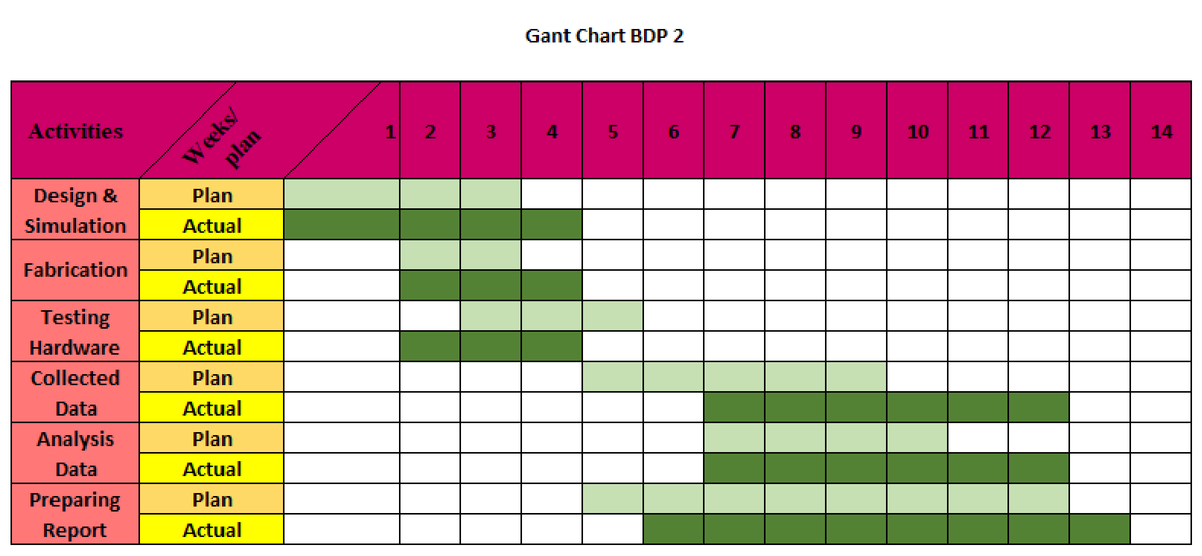


Figure 3.2: Gant chart of project

3.2.1.2 Research and Data Gathering

The most important aspect of this research is identifying the sources and issues that contribute to public apathy toward the epidemic of coronavirus infectious diseases spread throughout the world, including our own country. This is because they frequently fail to use disinfection gel and check their body temperature before entering a facility. Various concerns are being investigated to find a solution to the difficulties listed below. Several past projects have been researched and compared to aid in completing the key and improving the system to be constructed. Reading prior studies and journal articles is one method of acquiring information and data.

Other options were considered, such as studying websites and reading community comments, to aid in the hunt for other ideas. As a result, a project plan is created, and the hardware is chosen based on the information and data gathered. A mobile system-based solution was selected to allow the system to operate with security systems, maintain user privacy, be user-friendly, and accessible from anywhere. The ESP-32 was chosen as the central control unit for this project because it met all its requirements.

3.2.1.3 Design the Automatic Sanitizer with Door Control System and Smart Attendance.

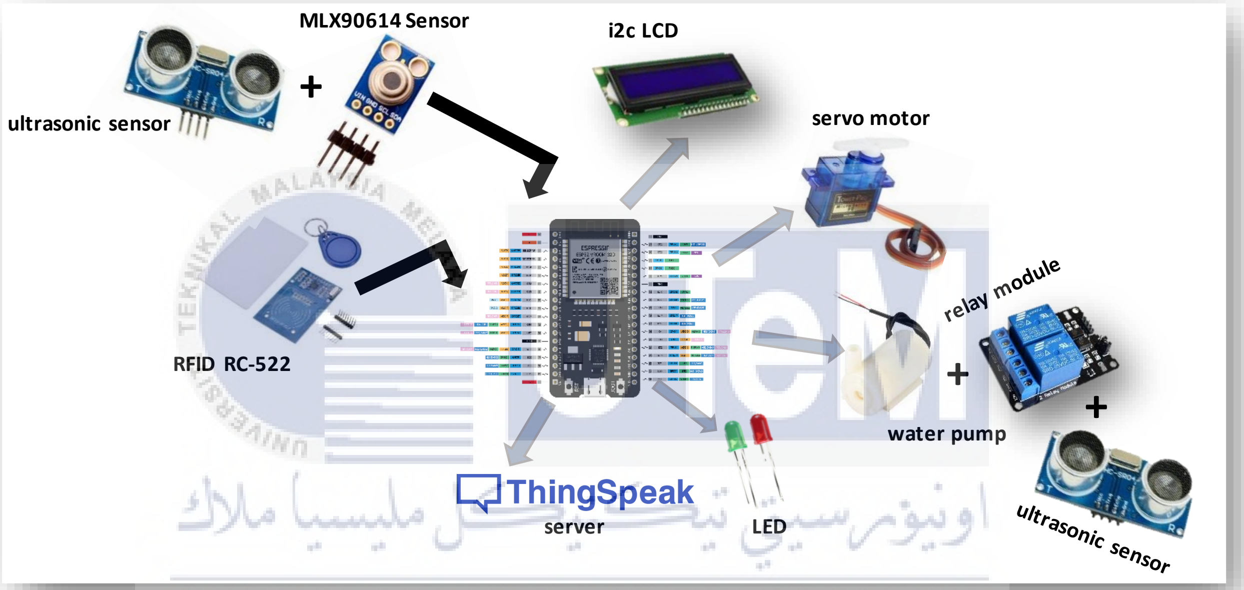


Figure 3.3: Smart Sanitizer with Door Control System Design

The diagram describes how Smart Sanitizer with Door Control System works in general. Users can enter the door when they scan attendance using an RFID tag scan. Users who have registered, their ID and details will be displayed on the web admin. After users scan attendance utilizing an RFID tag scan, users will go through the MLX90614 sensor to check their body temperature. This sensor was chosen because it can measure the physical temperature of a person's body. After measuring the body temperature, there is a decision on whether the user can access enter the door or not. The sensor will send a signal to the ESP-32 microcontroller. If the user's physical temperature exceeds 37 degrees Celsius, the buzzer will emit a signal sound. The sound of the signal means the user is not allowed to access the door.

However, if the user's physical temperature is less than 37 °C, the ESP-32 microcontroller will send the signal received from the MLX90614 sensor to the servo motor. The servo motor is used to control the door. The servo motor will move 90 degrees, and the user can enter the door, then the user will place their hand in front of the ultrasonic sensor. The ultrasonic sensor will detect the user's hand according to a predetermined distance, and the water pump will be turned on to discharge the sanitizer gel through the pipe. Users will wash their hands after entering the door.

3.2.1.4 System Access Using Smart Sanitizer with Door Control System Flowchart.

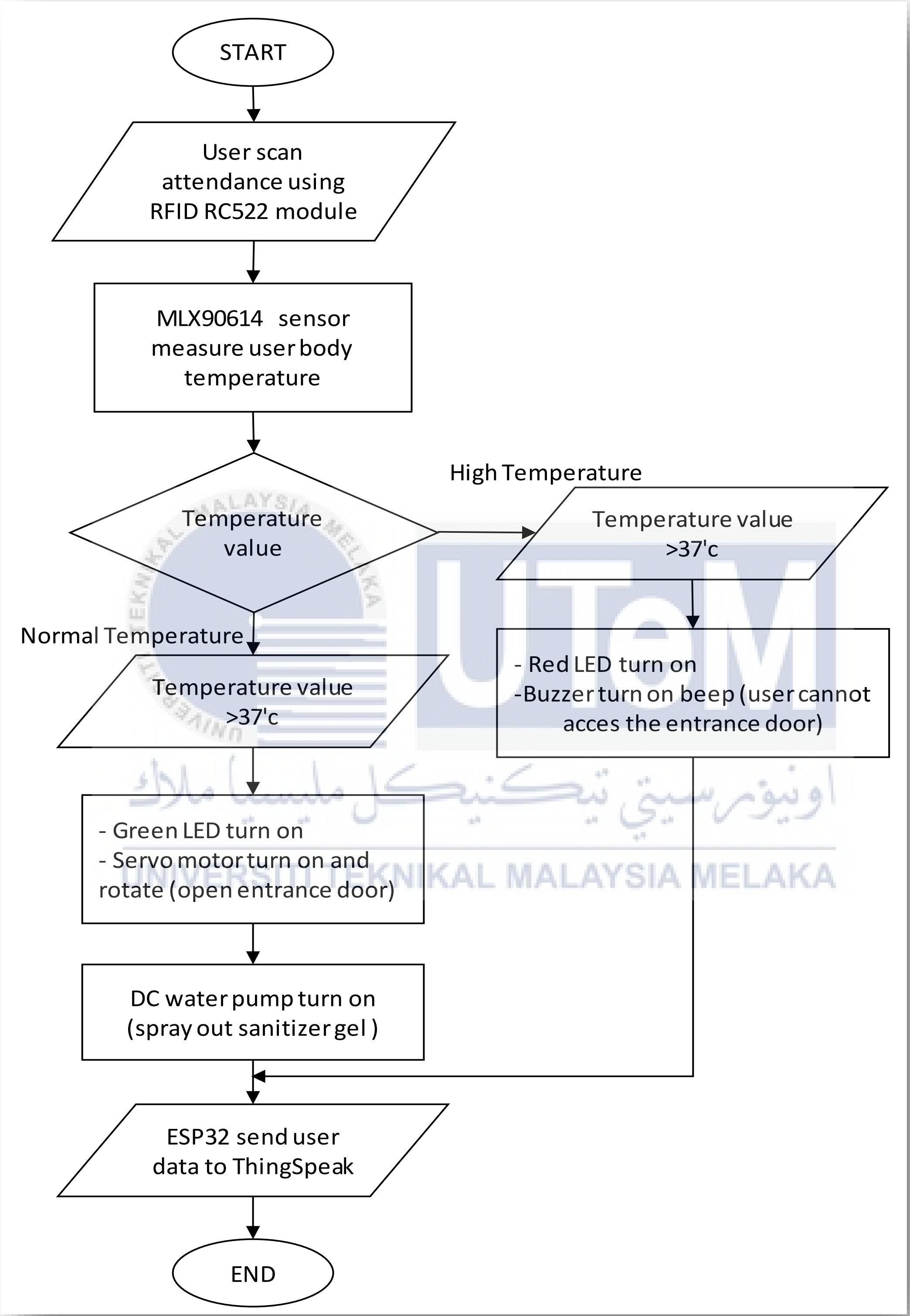


Figure 3.4: Smart Sanitizer with Door Control System Flowchart

3.3 HARDWARE SPECIFICATIONS

This research focuses on the materials and hardware used in the project, such as NodeMcu ESP-32, RFID RC522 Module, MLX90614 sensor, Ultrasonic sensor, and servo motor used to control this smart sanitizer with a door control system.

3.3.1 NodeMcu ESP-32

The ESP-32 microcontroller series is a low-cost, low-power microcontroller series featuring built-in Wi-Fi and dual-mode Bluetooth. The ESP32 series contains built-in antenna switches, power amplifiers, low-noise receive amplifiers, filters, power-management modules, and a Tensilica Xtensa LX6 CPU in dual-core single-core versions. It is a microcontroller that has been upgraded from the esp8266 microcontroller in this project. ESP-32 is used as a microcontroller that will send and receive instructions from servers and sensors. For example, when the user attaches the card to the sensor, ESP-32 will send the ID number found in the RFID card to the server.

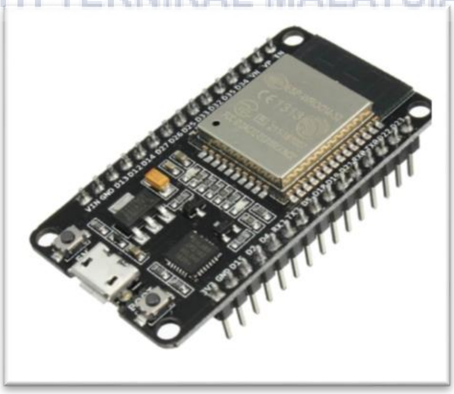


Figure 3.5: NodeMcu ESP-32 Board[3]

3.3.2 RFID RC-522 Module

The MFRC522 controller is used in the RC522, a 13.56MHz RFID module. The module includes an RFID card and supports I2C, SPI, and UART. It's often used in attendance systems and other applications that need to identify people or objects. The RFID RC522 is a radio frequency module with an RFID reader, an RFID card, and a keychain. The module runs on the industrial (ISM) band at 13.56MHz and does not require a license. Because the module typically runs at 3.3V, it is widely employed in 3.3V designs. It's commonly utilized in applications where a particular person or thing needs to be identified by a unique ID. In this project, we use RFID to identify the ID and details of staff who have registered. When the staff scans their RFID card, we will record the staff details, staff's entry, and exit times.

Figure 3.6: RFID RC522 Module[11]

3.3.3 MLX90614 Sensor

The MLX90614 is a Contactless Infrared (IR) Digital Temperature Sensor that monitors a specific object's temperature between -70°C and 382.2°C. The sensor measures the object's temperature using infrared rays without any physical contact and communicates with the microcontroller via the I2C protocol. This sensor is used in this project to detect the user's body temperature. If the user's body temperature is at a normal temperature (<37 degrees Celsius), the user can access the entrance. If the user's body temperature exceeds the normal temperature, the user will not access the door.

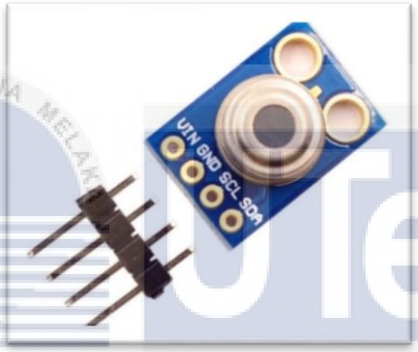


Figure 3.7: MLX90614 Non-Contact IR Temperature Sensor[12]

3.3.4 Ultrasonic Sensor HC-SR04

Two ultrasonic transducers are at the core of the HC-SR04 Ultrasonic Distance Sensor The one serves as a transmitter, converting electrical signals into ultrasonic sound pulses at a frequency of 40 kHz. The receiver listens for the pulses that have been broadcast. Two ultrasonic transducers form the heart of the HC-SR04 Ultrasonic Distance Sensor. The one serves as a transmitter, converting electrical impulses into ultrasonic sound pulses at a frequency of 40 kHz. The receiver listens for the pulses that have been broadcast. If it receives them, it generates an output pulse, the width of which can be used to calculate the pulse's travel distance.

The sensor is compact and straightforward to include in any robotics project. It delivers outstanding non-contact range detection with a 3mm accuracy range of 2 cm to 400 cm (roughly an inch and 13 feet). Because it works on 5V, it may be immediately linked to an Arduino or any other 5V logic microcontroller. Ultrasonic sensors are used in this project to detect the user at a predetermined distance physically. For example, when staff places their hands with a set distance on the sensor, the sensor will instruct the water pump to spray the sanitizer gel liquid into staff hands.

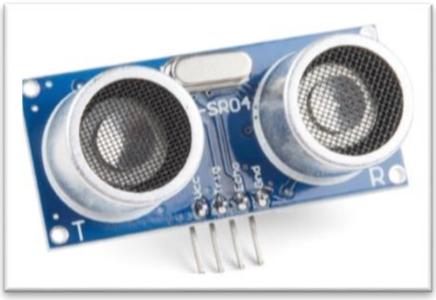


Figure 3.8: Ultrasonic sensor HC-SR04[13]

3.3.5 Servo Motor

A servo motor is an electrical device that can push or spin an item with pinpoint accuracy. We may rotate an object at precise angles or distances using a servo motor. It is simply constructed consisting of a simple motor that is controlled by a servo mechanism. If the motor is DC powered, it is referred to as a DC servo motor, and if the motor is AC powered, it is referred to as an AC servo motor. We can acquire a servo motor with a lot of torque in a tiny and light package.

These characteristics are utilized in various applications such as toy cars, RC helicopters and planes, robotics, and machines. Servo motors are used in this project to open the doors of the premises if the body temperature is at normal temperature (<37 °C). The servo motor will rotate 90° degrees when receiving instructions from the microcontroller, and the servo motor will not spin if the temperature exceeds the normal level. It indicates that staff is prohibited from entering the premises.

Figure 3.9: Servo Motor[14]

3.3.6 Relay Module

This device is known as a relay, and it is essentially an electromagnetic switch that turns on and off more considerable power using a small amount of power. The invention of the relay and the need for a relay on a circuit arose from the need to control large amounts of power and switch the circuit. Relays act as a bridge to allow a small current to flow into a larger one. To put it another way, relays can act as switches, turning on or off circuits, or as amplifiers, converting small currents into large currents.

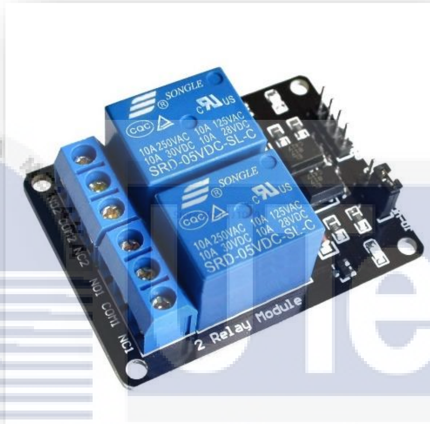


Figure 3.10: Relay Module[15]

3.3.7 I2C LCD

I2C LCD is a simple display module that can make displaying more convenient. It can help makers focus on the core of their work by reducing making it. The name 'character LCD' comes from the fact that this LCD type is ideal for displaying text and numbers. The I2C LCD module used in this tutorial has a small add-on circuit mounted on the back. A PCF8574 chip (for I2C communication) and a potentiometer for adjusting the LED backlight are included in this module. The advantage of an I2C LCD is the ease with which it can be wired. To control the LCD, we only need two data pins.

Figure 3.11: I2C LCD [16]

3.4 CIRCUIT DESIGN

3.4.1 Connection of the ESP32 to hardware component.

On this topic, NodeMcu(ESP32) and other hardware components are needed to make this connection. Firstly, the I2C LCD is connected to ESP32 to ensure that the LCD will display light up if the link is on. The LCD pinout connected with I2C IC (PCF8574) then connected to ESP32. Then, to identify the ID and details of registered staff, we need to join the RFID RC522 pinout to ESP32. Each card has its unique code, and it is specially registered for one user only. Each user has a unique code to track their attendance each time they scan their RFID card on the module reader.

Then the MLX90614 sensor is connected to the nodemcu to read the user's body temperature, and the MLX90614 pinout will be connected to the HC-SR04 ultrasonic sensor. Ultrasonic sensors are used to measure the object's distance while detecting the temperature so that the temperature reading is read accurately. Next, 2 LEDs (red and green) are connected to the NodeMcu, as well as the buzzer. The red LED will light up when the MLX90614 detects the temperature above the normal level, and the buzzer will emit a warning sound. If the user's temperature is normal, the green LED will light up.

After, a servo motor was used in this project as an automatic door actuator. Servo motor is connected to NodeMcu to function. When the user's temperature is normal, the servo motor rotates 180 degrees to open the door and vice versa. The module relay is connected to the ESP32 pinout to turn the water pump function as an auto sanitizer. Ultrasonic sensors are also connected to the relay and water pump. The ultrasonic sensor will detect at a set distance, and then the water pump will discharge the sanitizer liquid.

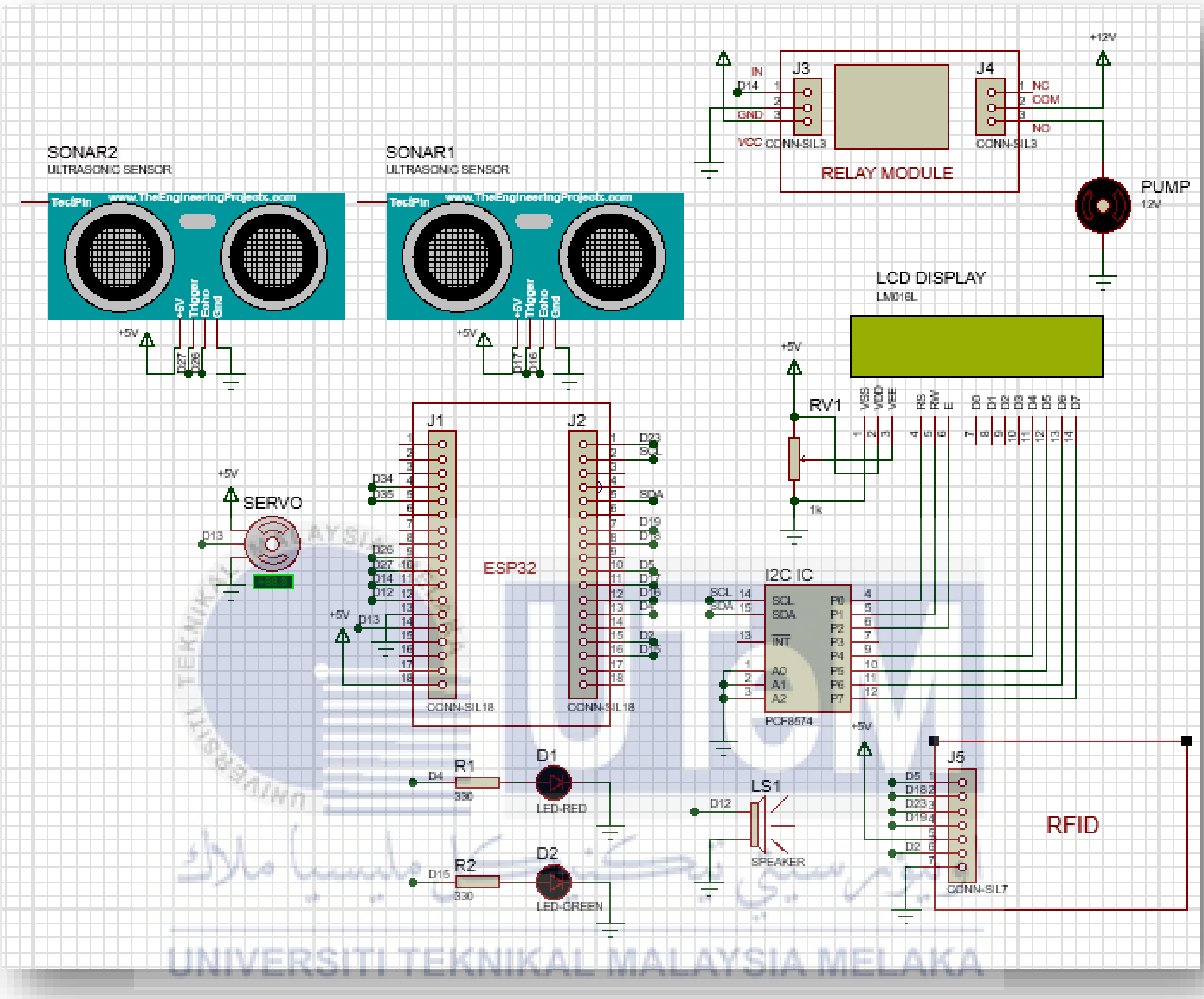


Figure 3.12: Connection of ESP32 to hardware component

3.5 SOFTWARE IMPLEMENTATION

3.5.1 Software System

This research uses a methods approach to develop a control system such as ThingSpeak and Arduino IDE. This software improves the system's programming customization and simulation.

3.5.1.1 ThingSpeak server

ThingSpeak is a cloud-based IoT analytics software that lets users aggregate, visualize, and analyze live data streams. Users can submit data from devices to ThingSpeak, generate real-time visualizations of live data, and issue alarms. Thingspeak server is used in this project as a database to record userID, user name, temperature, and time each time a user scans their RFID card.

Figure 3.13: ThingSpeak server [17]

3.5.1.2 Arduino IDE

The Arduino IDE stands for Integrated Development Environment. The Arduino IDE is a free and open-source tool that allows you to write and compile code for the Arduino Module. Its Arduinos official software simplifies code compilation. It runs on the Java Platform and is compatible with operating systems such as MAC, Windows, and Linux. It includes built-in functions and commands helpful in debugging, editing, and compiling code in the environment. On the board of each of them is a microcontroller that has been programmed and accepts data in code. So, in this project, we use Arduino IDE to create code that can instruct the microcontroller and another device.

Figure 3.14: Arduino IDE software[18]

CHAPTER 4

RESULT AND DISCUSSION

4.1 INTRODUCTION

The flow of the project's operations and the content of the final product will be shown in this chapter. This project demonstration will show how the hardware and software work together. This chapter will discuss the relationship between components and coding and how they work together. In addition, this chapter will go over the project's process, flowchart, table, graph, and related figure. This project took almost a year to complete, beginning with research, literature review, and other processes.

4.2 PROJECT HARDWARE LAYOUT

Figure 4.1 shows the Smart Sanitizer with Door Control System using ESP32 prototype, which presents the system that will duplicate according to the actual situation if the system is implemented in the office. As stated in the project's scope, the project is specially created suitable located in a small office, office entrance. This prototype represents a medium-sized building (approximately an office) with an RFID reader to detect RFID smart cards. Staff (users) need to scan their smart card on an RFID reader, then ultrasonic sensor and MLX90614 (temperature sensor) are connected to ESP32. Staff needs to scan their body temperature by placing their forehead on the temperature sensor at a predetermined distance. Two LEDs (red and green) are connected to ESP32. Then, the servo motor is connected to the ESP32 as an auto door lock control, where the door will open automatically. The smart sanitizer, relay module, ultrasonic, and water pump connected to the ESP32.

Figure 4.1: Final Prototype of Smart Sanitizer with Door Control System using ESP32

4.3 PROJECT DEMONSTRATION

4.3.1 Smart Sanitizer with Door Control system using ESP32

Smart sanitizer with the door control system is a combined hardware and software project. The project can reduce the risk of infection of diseases such as the coronavirus by detecting a person's body temperature. The LCD will display instructions to staff users (users) to scan their smart cards on the RFID reader in the first step. Each RFID smart card has a unique code, and the unique code is registered and valid for only one user. The card also has registered user details such as name, staff ID, department, and telephone number.

Figure 4.2: LCD display instruction to scan ID card

After the user scans their ID card on the RFID reader, the LCD that has been programmed will display instructions for users to scan their body temperature. Accurate body temperature can only be measured on the forehead at a predetermined distance. Ultrasonic sensors have been programmed to measure distances not exceeding 10cm to obtain the best readings for human body temperature. The pre-programmed MLX90614 temperature sensor will detect body temperature in two situations. Normal body temperature does not exceed 36 degrees Celsius. If the temperature exceeds 37.5 degrees, Celsius is considered a fever. The pre-programmed LCD will display "YOU ARE NOT FEVER" when the temperature is normal. If the temperature is normal, the servo motor will receive instructions from the pre-programmed ESP32. The servo motor will rotate at 180 degrees to open the door (automatic door). The green LED gets a command to light up as a sign the user is in normal temperature and allowed to enter the main entrance automatically, and vice versa when the body temperature exceeds 37 degrees Celsius.

Table 2: The LCD Display Image and Instructions

LCD Display	Instruction
	MLX90614 detects a temperature below 36 degrees Celsius, and the LCD receives a command from the pre-programmed ESP32, the LCD, "YOU ARE NOT FEVER." The green LED lights up.
	User data will be sent to the server (Thingspeak).

	LCD "PLEASE TAKE SANITIZER" instructs users who are at a normal temperature to sanitize their hands on the automatic sanitizer.
	MLX90614 detects a temperature above 37 degrees Celsius, and the LCD receives a command from the pre-programmed ESP32, the LCD "YOU ARE FEVER." The red LED lights up.

Users who are at a normal temperature will be allowed to enter. The servo motor gets instructions from the ESP32 to rotate 180°, and the servo is mounted on the main door as an automatic door. The door opens automatically when the MLX90614 sensor detects the temperature at a normal level. Then, users have to sanitize their hands after entering the entrance. Relay modules and water pumps are mounted on the ESP32 to receive instructions to generate automatic sanitation. With higher horsepower submersible pumps, a pump relay switch tells the pump when to turn on and off. Pump relay switches are buffers between a heavy-duty water pump and the pump pressure control switch. The ultrasonic sensor is installed once with the water pump, and when it detects an object less than 10cm, it will give instructions on the relay and the water pump to spray the sanitary gel.

4.3.2 ThingSpeak Data Demonstration

ThingSpeak is a platform to receive a database from any user who accesses the RFID reader on the Smart Sanitizer with Door Control System using ESP32. Data such as date and time entry, User ID, username, phone number, position, and the department will be sent to the ThingSpeak server. The ESP32 has been coded through the IDE to send the data to ThingSpeak via an internet connection. At the start of the project, when pressing the push-on button, ESP32 is programmed to delay for 10 seconds because ThingSpeak needs about 10 seconds to be active before receiving data. In the date and time entry section, we can select the time zone in ThingSpeak. On this project, I used the date and time zone from (GMT +08:00) Kuala Lumpur to get a more accurate time and date here.

Figure 4.3: The Sign In Interface for ThingSpeak[19]

In thingspeak, table data will be generated automatically. Figure 4.4 below shows a graph plot of the temperature reading and the date when the sensor read the smart card and the sensor reading at temperature.

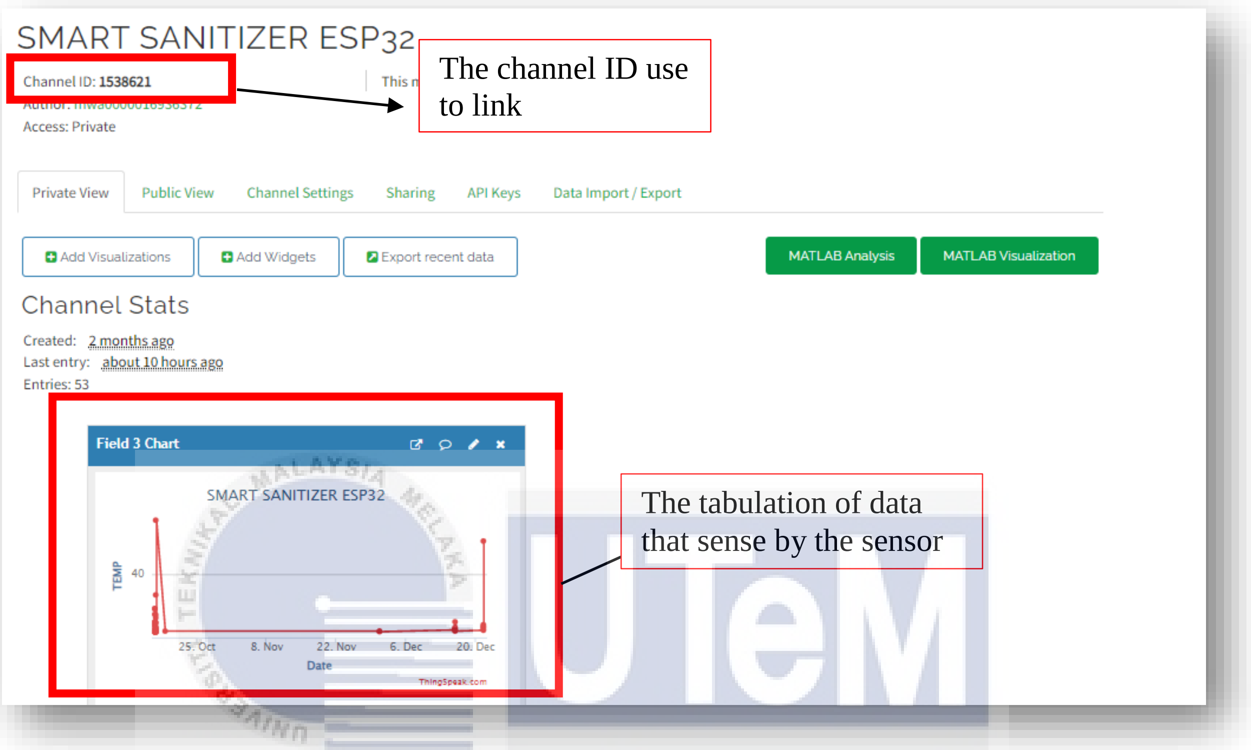


Figure 4.4: The Web of ThingSpeak[17]

4.4 DATA AND ANALYSIS OF THE PROJECT

4.4.1 The distance of the sensor reading on the object

This section will examine the reading distance of the sensor measurement on the object. We will assess the exact distance for the sensor to take readings in this project using ultrasonic sensors to detect the distance of an object. Two ultrasonic sensors were used in this project. The first ultrasonic sensor was installed close to the MLX90614 (temperature sensor). The accuracy of the appropriate distance to obtain the exact temperature on the object (human body) is less than 10cm. The ultrasonic and temperature sensors remain running while the project is running. However, accurate temperature readings will only be taken. Data is sent to the server when the user has scanned their smart card and scans the temperature on the temperature sensor at a distance not exceeding 10cm, in this project using the Arduino IDE on the serial port of the monitor to obtain the ultrasonic reading data of the sensor on the object, as well as the temperature sensor on the object.

Table 4 below shows the readings that have been recorded five times for the object distance while the temperature sensor takes the temperature reading on the object.

Table 3 : Distance reading (cm) vs Temperature sensor reading (°C)

Distance Reading (cm)	Temperature sensor reading(°C)
54	32.51
7	37.61
47	33.81
9	35.25
19	33.95

Object distance detect (<10cm) for temperature sensor reading then send

Figure 4.5 shows a graph plot for the distance reading instead of the sensor temperature reading. At a distance below 10cm, only the temperature sensor will take a temperature reading on the object. Based on the data analysis, at a distance of 7cm, the temperature sensor has taken temperature readings at 37.61°C (high temperature) and 9cm at 35.35°C (normal temperature).

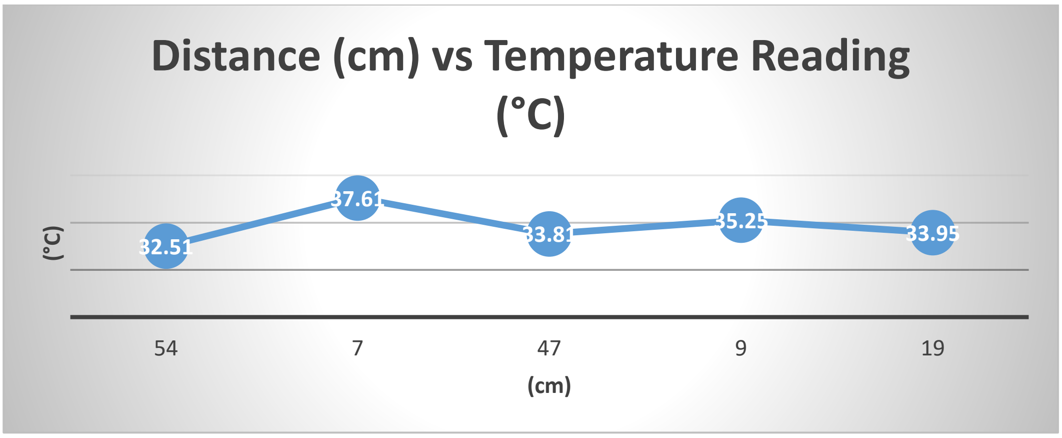


Figure 4.5: Distance reading (cm) vs Temperature reading (°C)

A second ultrasonic sensor is mounted on the auto sanitizer section. In this section, the relay module and water pump components are installed to pump and remove the sanitary gel. Ultrasonic sensors have been programmed to measure objects at distances not exceeding 10cm. When the user places his hand on the ultrasonic sensor with a distance not exceeding 10cm, the relay module and water pump receive instructions and pump the sanitizer automatically. Users do not need to touch the sanitary ware like other premises that we usually see. In this project, it is touchless.

Table 5 below shows the readings that have been recorded five times for the sensor to detect the distance of the object and how long it takes for the water pump to pump water automatically.

Table 4 : Distance reading (cm) vs Time Taken for waterpump (s)

Distance Reading (cm)	Time Taken(s)
0	0
7	1.98
5.5	1.45
9	1.32
8.8	1.56

Figure 4.6 shows a graph plot for the distance reading with the time reading for the water pump to pump the liquid out. When the sensor detects an object below 10cm near the sensor, the water pump automatically pumps the sanitary liquid.

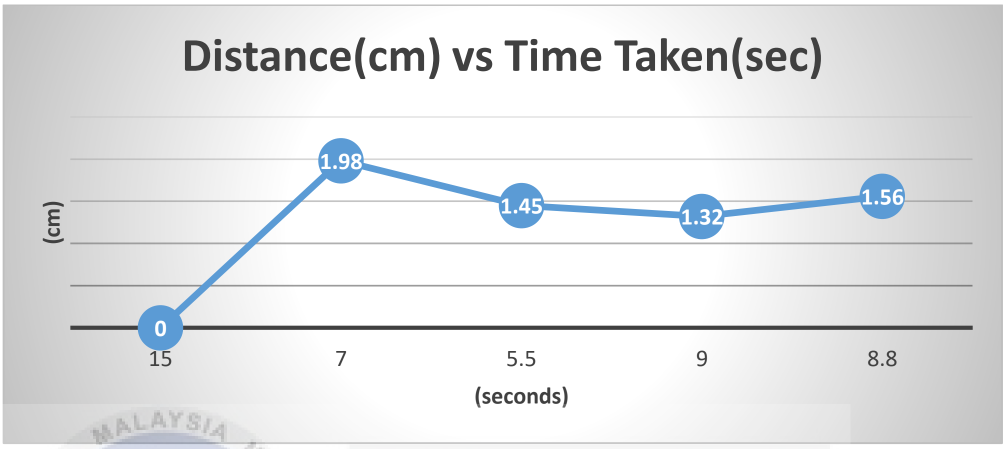


Figure 4.6: Distance reading (cm) vs Time Taken for waterpump(s)

4.4.2 The time taken by the servo motor to rotate for automatic door control

In this section, we will describe and analyze the movement of a servo motor. The servo motor can precisely control the rotation of the motor shaft. We can use a code or inputs like a joystick, push-button, or potentiometer to set the exact rotation angle. Servo motors were used to open the main entrance as an automatic door. Only users who are in a state of normal body temperature are allowed to enter. To control the movement and time taken for the servo motor to rotate, we will create a program code on the IDE for the servo motor. The program will be uploaded into the ESP32, and the ESP32 will give instructions to the servo motor. The program code will set up the servo motor to rotate 180 degrees, program code myservo. write(180). Next, set up the code section for the time taken by the servo motor to rotate 180 ° as the entrance door opens automatically for 9 seconds, and the servo motor will rotate at 0 ° for 9 seconds to close the main door.

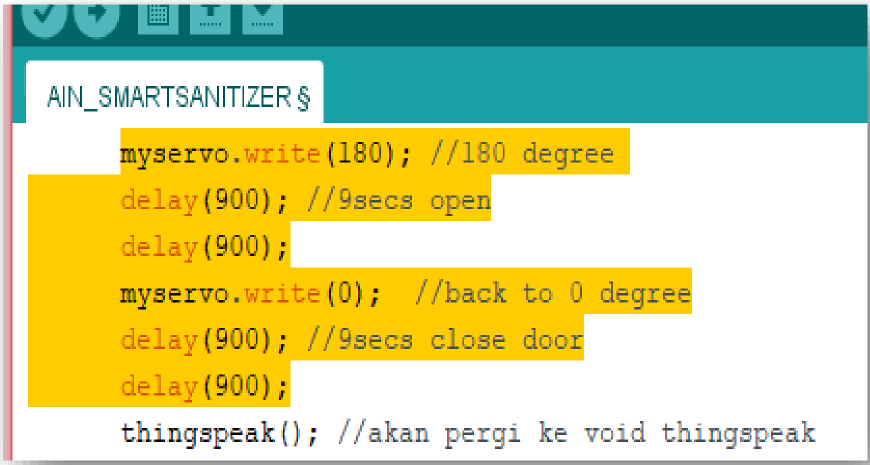
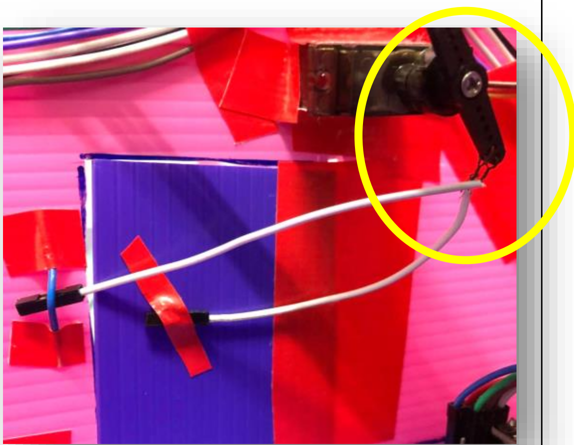
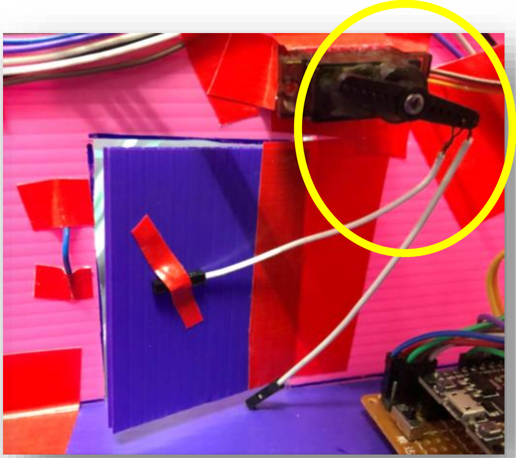


Figure 4.7: Program code for servo motors on the Arduino IDE

Table 5 shows a picture of a servo motor functioning as an automatic door. The door will open automatically when the servo motor rotates. Then, the door closes again when the servo motor rotates back to its original position.

Table 5 : The servo motor works to open the door automatically

Front	Back (Servo Motor)
The door is remain locked. The servo motor does not rotate	
	

<table>
<tr><td colspan="2">The door opens automatically. The servo motor rotates to open the door</td></tr>
<tr><td>
</td><td></td></tr>
</table>

4.4.3 Database records received on the server

In this project, database records are vital. Because from there will be able to track data every time users scan their smart card on the RFID reader to access the entrance. The server used to submit data to the server is ThingSpeak. As described in the passage above, ThingSpeak is an IoT analytics platform that allows users to aggregate, visualize and analyze live data streams in the cloud. We use this platform to receive a database. Important data such as user ID, date and time, temperature reading, name, phone number, and others can be seen in the record. Record data will remain and will not be lost. It dramatically simplifies the administration of an organization to check data records. ESP32 will receive input from the components while the component is working. To send input data to the server, the program code will be set on the Arduino IDE and uploaded on ESP32.

```
IN_SMARTSANITIZER §
}
void thingspeak(){
        if (millis() - thingSpeakTimer > 16000) {
        Blynk.virtualWrite(V0,"https://api.thingspeak.com/update?api_key=ZK31H9K6721NO9BI&field1=" + String(ID) + "&field2=" +
        String(NAMA) + "&field3=" + String(TEMP)+ "&field4=" + String(POSITION)+ "&field5=" + String(DEPARTMENT) + "&field6=" + String(PHONE));//kt blynk add v0, set pin v0
        Serial.println("DATA SENDED TO THINGSPEAK");
        Serial.println("ID" + String(ID) + "NAME" + String(NAMA) + "TEMP:" + float(TEMP) + "C  "+ "POSITION" +
        String(POSITION) + "DEPARTMENT" + String(DEPARTMENT)+ "PHONE" + String(PHONE) );
        lcd.setCursor(0,0);
        lcd.print("  DATA SUBMITTED   ");
        lcd.setCursor(0,1);
        lcd.print("  TO SERVER   ");
        delay(2000);
        thingSpeakTimer = millis();
        ID = "";
        NAMA = "";
        TEMP = 0;
        POSITION = "";
        DEPARTMENT = "";
        PHONE = "";
    }
```

Figure 4.8: Program code for submit data to ThingSpeak on the Arduino IDE

Each time a user scans an ID card and scans a body temperature, their data will be sent directly to the ThingSpeak server. Whether they can access the entrance or not, permanent data will be recorded and stored. To get the data transmitted and recorded, you need to enter on the platform ThingSpeak. In the ThingSpeak forum, there is a data import and export section. Just click on the data import and export download section, and the data in the Excel file will be downloaded. After pressing the download button, we will get an excel file that contains all the data that has been recorded. Figure 4.9 shows how to import and export data that has been recorded on thingspeak. Figure 4.10 shows an excel file that contains record data obtained from the current thingspeak server.

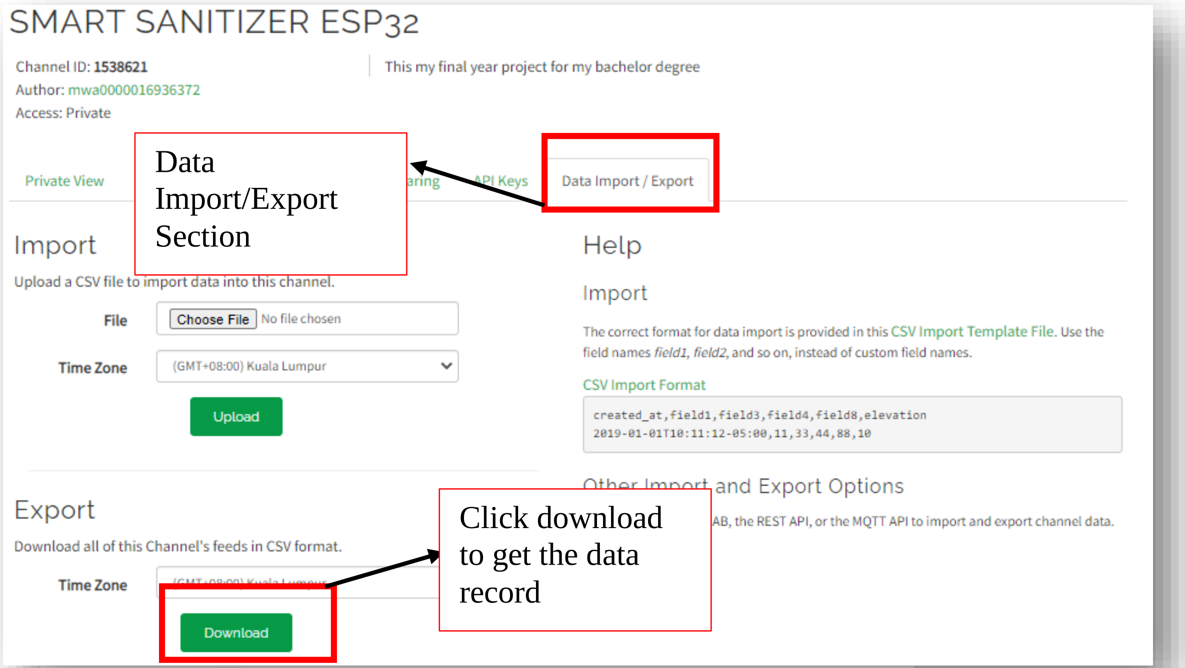
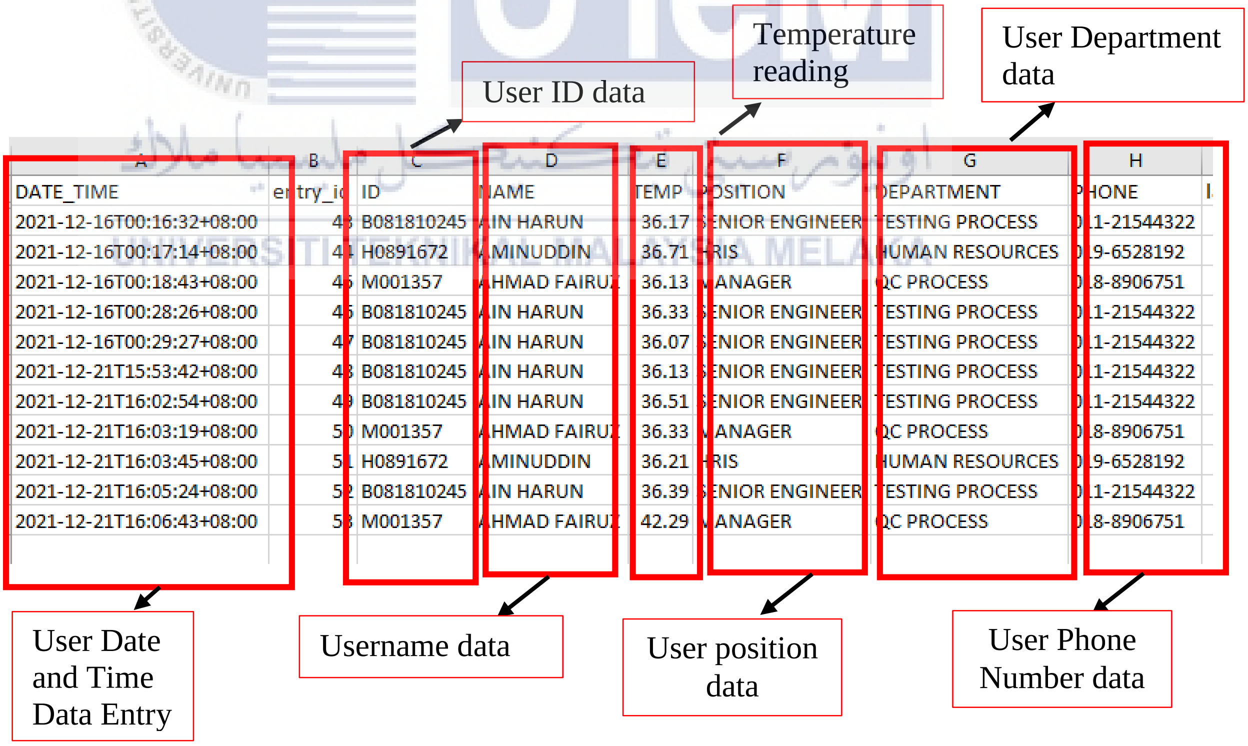


Figure 4.9: ThingSpeak Data Import/Export



A	B	C	D	E	F	G	H
DATE_TIME	entry_id	ID	NAME	TEMP	POSITION	DEPARTMENT	PHONE
2021-12-16T00:16:32+08:00	43	B081810245	AIN HARUN	36.17	SENIOR ENGINEER	TESTING PROCESS	01-21544322
2021-12-16T00:17:14+08:00	44	H0891672	AMINUDDIN	36.71	HRIS	HUMAN RESOURCES	09-6528192
2021-12-16T00:18:43+08:00	45	M001357	AHMAD FAIRUZ	36.13	MANAGER	QC PROCESS	08-8906751
2021-12-16T00:28:26+08:00	46	B081810245	AIN HARUN	36.33	SENIOR ENGINEER	TESTING PROCESS	01-21544322
2021-12-16T00:29:27+08:00	47	B081810245	AIN HARUN	36.07	SENIOR ENGINEER	TESTING PROCESS	01-21544322
2021-12-21T15:53:42+08:00	48	B081810245	AIN HARUN	36.13	SENIOR ENGINEER	TESTING PROCESS	01-21544322
2021-12-21T16:02:54+08:00	49	B081810245	AIN HARUN	36.51	SENIOR ENGINEER	TESTING PROCESS	01-21544322
2021-12-21T16:03:19+08:00	50	M001357	AHMAD FAIRUZ	36.33	MANAGER	QC PROCESS	08-8906751
2021-12-21T16:03:45+08:00	51	H0891672	AMINUDDIN	36.21	HRIS	HUMAN RESOURCES	09-6528192
2021-12-21T16:05:24+08:00	52	B081810245	AIN HARUN	36.39	SENIOR ENGINEER	TESTING PROCESS	01-21544322
2021-12-21T16:06:43+08:00	53	M001357	AHMAD FAIRUZ	42.29	MANAGER	QC PROCESS	08-8906751

Figure 4.10: Excel file for the database that has been recorded

4.5 COMPARISON BETWEEN SMART SANITIZER WITH DOOR CONTROL SYSTEM WITH REGULAR SANITATION.

The smart sanitizer with door control system project is an IoT project that can help people out there. It is different compared to the sanitizers out there. This project is touchless, requires no touch. All processes are touchless from the beginning of the movement to the end. It can prevent infectious diseases because infectious diseases are easily spread through contact. Table 6 shows the comparison and explanation for smart sanitizer with a door control system with other automatic sanitizers.

Table 6: Comparison between smart sanitizer with regular sanitation

Method	**Smart Sanitizer with Door Control System**	**Regular Sanitation**
User Data	User data has been registered on an RFID smart card. Users only need to scan their smart card on the RFID reader, and user data will be sent to the server. User information and data are authentic and accurate. Data recorded and sent on a server is secure and not extinct.	Users need to write their data on a record book. It took a long time. The time, date, and data that users record are also mostly incomplete and inaccurate as they can falsify their information to enter the premises.
Temperature reading	Temperature readings are more accurate. Because the sensor only detects readings not exceeding 10cm. Users need to place their forehead on the temperature sensor without touching it.	Some temperature reader devices require the user to hold the device and measure its temperature. The readings taken are less accurate.
User enters the door	The automatic door only opens when the user is at a normal temperature. The user will enter the door without touching it. When the user's body temperature exceeds the normal limit, the user cannot access the entrance because the door is not open.	Users can enter the door even if their body temperature is above the normal temperature. Users also have to open the entrance themselves, and it takes a touch to open the door.
Sanitation methods	It's touchless! Users place their hands a distance not exceeding 10cm on the sensor, and then the water pump will release sanitary liquid on the user's hands. It's safer.	Users need to squeeze the bottle to sanitize their hands. It takes a longer time than smart sanitizer.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Many things will be analyzed and improvised at the end of this smart sanitizer with a door control system using the ESP32 final version. This is a challenging project. It requires a lot of resources and helps from many educated and skilled people. As recommended on this project's scope, this project is suitable for use in an organization that has approximately 20 to 30 employees. The first objective of this project is to develop smart attendance using RFID smart cards and contactless body temperature connected with servers and sensors. The first objective was successfully achieved. Each employee had a different RFID card code in it. Each time they scan their RFID card on the reader, the employee's data will continue to be sent to the server automatically. By using ThingSpeak, every piece of data can be recorded. As a result, we devised a strategy for tracking employee attendance by scanning RFID tags, automatically measuring body temperature, and submitting the data.

The next objective in this project is to develop smart hand sanitizing for industry based on microcontrollers has been achieved. Users who are allowed to enter the premises will place their hands on the sensor, and the water pump will automatically spray the gel sanitizer onto their hands. It is touchless. So, users do not have to press the pump to sanitize their hands, they place their hands at a set distance on the sensor, and they will sanitize their hands automatically. Sanitizing hands can prevent bacterial infection. The final objective of this project that has been achieved is to develop an automatic door controller system using multisensory. Workers at a normal body temperature can access the entrance door, and workers who are at temperatures above normal limits are not allowed to access the entrance to the premises. It helps prevent being easily infected with those who have symptoms. The main thing to know the asymptomatic individual is that their body temperature is above normal temperature.

5.2 Future Works

This project might be improved in specific places if this project has more time and resources. This project is currently aimed at offices, factories, and company employees. For future work, the most important thing to improve is to add functionality for visitors to shops, hotels, banks, and hospitals, among other places, so that this machine can serve both employees and visitors. Maybe, on further improvement, the development of the server needs to use a different server like phpMyadmin and XAMPP server. For a more user-friendly experience, the apps' fancy and professional interfaces for registering user ID and recording data should be improved. So that both the employee and the administrative staff can keep track of their information.

REFERENCES

[1] "NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet." https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet (accessed Jun. 18, 2021).

[2] "NodeMCU-Pinout-Image.jpg (1200×859)." https://www.electronicshub.org/wp-content/uploads/2021/02/NodeMCU-Pinout-Image.jpg (accessed Jun. 18, 2021).

[3] "HiLetgo ESP-WROOM-32 ESP32 ESP-32S Development Board 2.4GHz Dual-Mode WiFi + Bluetooth Dual Cores Microcontroller Processor Integrated with Antenna RF AMP Filter AP STA for Arduino IDE: Buy Online at Best Price in UAE - Amazon.ae." https://www.amazon.ae/HiLetgo-ESP-WROOM-32-Development-Microcontroller-Integrated/dp/B0718T232Z (accessed Jun. 18, 2021).

[4] "ESP32-Pinout-1.jpg (1000×518)." https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-Pinout-1.jpg (accessed Jun. 18, 2021).

[5] E. Hub, "ESP32 vs ESP8266 | Circuits4you.com." 2021, [Online]. Available: https://circuits4you.com/2019/03/02/esp32-vs-esp8266/.

[6] "RC522-RFID-CARD-READERS-Pinout.png (980×540)." https://microcontrollerslab.com/wp-content/uploads/2020/01/RC522-RFID-CARD-READERS-Pinout.png (accessed Jun. 18, 2021).

[7] F. Koyanagi, "ESP32 With RFID: Access Control : 13 Steps," *Instructables*. 2018, [Online]. Available: https://www.instructables.com/ESP32-With-RFID-Access-Control/.

[8] "RFID based Contactless Body Temperature Screening using Arduino and MLX90614 IR Temperature Sensor." https://circuitdigest.com/microcontroller-projects/rfid-based-contactless-body-screening-using-mlx90614-infrared-termperature-sensor-and-arduino (accessed Jun. 18, 2021).

[9] A. Choudhary, "Automatic Hand Sanitizer Dispenser with COVID19 (Corona Virus) Live Updates." 2020, [Online]. Available: https://circuitdigest.com/microcontroller-projects/automatic-hand-sanitizer-dispenser-with-covid19-live-updates.

[10] "Smart Hand Sanitizer - Arduino Project Hub." https://create.arduino.cc/projecthub/smd-innovation/smart-hand-sanitizer-903adc (accessed Jun. 18, 2021).

[11] "Mifare RC522 RFID Kit." https://sg.cytron.io/p-mifare-rc522-rfid-kit (accessed Jun. 18, 2021).

[12] "MLX90614 Non-Contact IR Temperature Sensor Pinout, Datasheet, Equivalents & Specs." https://components101.com/sensors/melexis-mlx90614-contact-less-ir-temperature-sensor (accessed Jun. 18, 2021).

[13] "Ultrasonic Distance Sensor - HC-SR04 - SEN-15569 - SparkFun Electronics." https://www.sparkfun.com/products/15569 (accessed Jun. 18, 2021).

[14] "Servo Motor Basics, Working Principle & Theory." https://circuitdigest.com/article/servo-motor-working-and-basics (accessed Jun. 18, 2021).

[15] "2 Channels 5V Relay Module | Shopee Malaysia." https://shopee.com.my/2-Channels-5V-Relay-Module-i.40459773.780479588 (accessed Dec. 24, 2021).

[16] "1602 (16x2) LCD Display with I2C/IIC interface - Blue Backlight buy online at Low Price in India - ElectronicsComp.com." https://www.electronicscomp.com/1602-blue-lcd-display-with-i2c-interface (accessed Dec. 24, 2021).

[17] "IoT Analytics - ThingSpeak Internet of Things." https://thingspeak.com/ (accessed Dec. 24, 2021).

[18] "Installing Arduino IDE - learn.sparkfun.com." https://learn.sparkfun.com/tutorials/installing-arduino-ide/all (accessed Dec. 24, 2021).

[19] "Sign In - ThingSpeak IoT." https://thingspeak.com/login?skipSSOCheck=true (accessed Dec. 24, 2021).

APPENDICES

CODING

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <SPI.h>
#include <MFRC522.h> //rfid lib
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
#include <NewPing.h> //ir sensor
#include <Adafruit_MLX90614.h>
Servo myservo;
#define PUMP  14
#define LEDRED  4
#define LEDGREEN  15
#define BUZZER  12
#define TRIGGER_PIN1  17
#define ECHO_PIN1     16
#define TRIGGER_PIN2  27
#define ECHO_PIN2     26
#define MAX_DISTANCE 200
NewPing sonar1(TRIGGER_PIN1, ECHO_PIN1, MAX_DISTANCE);
NewPing sonar2(TRIGGER_PIN2, ECHO_PIN2, MAX_DISTANCE);
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
#define RST_PIN 2
#define SS_PIN 5
int servoPin = 13;
char auth[] = " TOKEN";
char ssid[] = "wifi name ";
char pass[] = " wifi password ";
LiquidCrystal_I2C lcd(0x27, 16, 2);
MFRC522 mfrc522(SS_PIN, RST_PIN);
int flagDataIn = 0;
String data1 = "";
String content = "";
float t;
unsigned long thingSpeakTimer = 0;
String database[] = {"rfid card unique code"}; String Name[] = {"name"};
String idstaff[] = {"id"};
String posisi[] = {"title"};
String dept[] = {"department"};
String tel[] = {"no hp"};
int flag = 0;
String ID = "";
String NAMA = "";
String POSITION = "";
```

```
String DEPARTMENT = "";
String PHONE = "";
float TEMP = 0;
void setup() { //buat setup
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
pinMode(LEDRED,OUTPUT);
pinMode(LEDGREEN,OUTPUT);
pinMode(BUZZER,OUTPUT);
pinMode(PUMP,OUTPUT);
digitalWrite(PUMP,HIGH);
SPI.begin();
mfrc522.PCD_Init();
thingSpeakTimer = millis();
mlx.begin();
lcd.begin();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("  AIN HARUN  ");
lcd.setCursor(0,1);
lcd.print("  IOT PROJECT  ");
delay(10000);
lcd.setCursor(0,0);
lcd.print("  PLEASE SCAN   ");
lcd.setCursor(0,1);
lcd.print("   YOUR ID CARD    ");
myservo.setPeriodHertz(50);
myservo.attach(servoPin);
myservo.write(0);
delay(2000); }
void loop() {
//Blynk.run();
readRFID();
ultrasonic();
t = (mlx.readObjectTempC() + 2 );
Serial.print("Object temp = ");
Serial.print(t);
Serial.print(" C   ");
if( t >= 36 && t <= 37.5 && sonar1.ping_cm() < 10 && flag == 1){           TEMP = t;
flag = 2;
lcd.setCursor(0,0);
lcd.print("  YOU ARE NOT   ");
lcd.setCursor(0,1);
lcd.print("     FEVER      ");
digitalWrite(LEDGREEN,HIGH);
digitalWrite(LEDRED,LOW);
digitalWrite(BUZZER,HIGH);
myservo.write(180);
delay(900);
myservo.write(0);
```

```
delay(900);
thingspeak();
digitalWrite(BUZZER,LOW);
digitalWrite(LEDGREEN,LOW);
digitalWrite(LEDRED,LOW);
lcd.setCursor(0,0);
lcd.print("  PLEASE TAKE   ");
lcd.setCursor(0,1);
lcd.print("   SANITIZER    ");
delay(1000);}
else if( t > 37.5 && sonar1.ping_cm() < 10 && flag == 1){
TEMP = t;
flag = 2;
lcd.setCursor(0,0);
lcd.print(" YOU ARE FEVER!!!");
lcd.setCursor(0,1);
lcd.print("   STAY AT HOME  ");
digitalWrite(LEDGREEN,LOW);
digitalWrite(LEDRED,HIGH);
digitalWrite(BUZZER,HIGH);
delay(3000);
thingspeak();
digitalWrite(BUZZER,LOW);
digitalWrite(LEDGREEN,LOW);
digitalWrite(LEDRED,LOW);
lcd.setCursor(0,0);
lcd.print("  PLEASE SCAN   ");
lcd.setCursor(0,1);
lcd.print("   YOUR ID CARD    "); }
if(sonar2.ping_cm() < 10 && sonar2.ping_cm() > 0 && flag == 2){    flag = 0;
digitalWrite(PUMP,LOW);
delay(500);
digitalWrite(PUMP,HIGH);
lcd.setCursor(0,0);
lcd.print("  PLEASE SCAN   ");
lcd.setCursor(0,1);
lcd.print("   YOUR ID CARD    ");}
if (flagDataIn == 1) {
flagDataIn = 0;
flag = 1;
for (int i = 0; i < sizeof(database); i++) {
if (data1 == database[i]) {          Serial.println(data1 + ": " + Name[i]);
ID = idstaff[i];
NAMA = Name[i];
POSITION = posisi[i];
DEPARTMENT = dept[i];
PHONE = tel[i];
Serial.println("ID:" + String(ID) + "NAMA:" + String(NAMA) + "");
lcd.setCursor(0,0);
lcd.print("NAME:" + String(NAMA) + "         ");
```

```
lcd.setCursor(0,1);
lcd.print("ID:" + String(ID) + "         ");
delay(1500);
lcd.setCursor(0,0);
lcd.print("  PLEASE SCAN   ");
lcd.setCursor(0,1);
lcd.print("YOUR TEMPERATURE");}}}}
void readRFID() {
if (flagDataIn == 0){
if ( !mfrc522.PICC_IsNewCardPresent() ) {
return;  }
if ( !mfrc522.PICC_ReadCardSerial() ) {
return;    }
byte letter;
for ( byte i = 0; i < mfrc522.uid.size; i++ ) {
content.concat(String(mfrc522.uid.uidByte[i], HEX));
if ( i < mfrc522.uid.size - 1 ) content += "-";}
content.toUpperCase();
Serial.println();
Serial.println("UID tag :'" + content + "'");
data1 = content;
content = "";
flagDataIn = 1;
delay(1000);}}
void ultrasonic(){
delay(50);
Serial.println("P1:" + String(sonar1.ping_cm()) + "cm  P2:" + String(sonar2.ping_cm()) +
"cm  "); }
void thingspeak(){
if (millis() - thingSpeakTimer > 16000) {
Blynk.virtualWrite(V0,"https://api.thingspeak.com/update?api_key=TknThingSpeak&field
1=" + String(ID) + "&field2=" +
String(NAMA) + "&field3=" + String(TEMP)+ "&field4=" + String(POSITION)+
"&field5=" + String(DEPARTMENT) + "&field6=" + String(PHONE));//kt blynk add v0,
set pin v0
Serial.println("DATA SENDED TO THINGSPEAK");
Serial.println("ID" + String(ID) + "NAME" + String(NAMA) + "TEMP:" + float(TEMP) +
"C  "+ "POSITION" +
String(POSITION) + "DEPARTMENT" + String(DEPARTMENT)+ "PHONE" +
String(PHONE) );
lcd.setCursor(0,0);
lcd.print("  DATA SUBMITTED   ");
lcd.setCursor(0,1);
lcd.print("  TO SERVER   ");
delay(2000);
thingSpeakTimer = millis();
ID = "";
NAMA = "";
TEMP = 0;
POSITION = "";
```

```
DEPARTMENT = "";
PHONE = "";}else {
Serial.println("Please wait for 15 second delay to finish");
lcd.setCursor(0,0);
lcd.print(" PLEASE RESCAN ");
lcd.setCursor(0,1);
lcd.print("     AGAIN     ");
delay(1500);
lcd.setCursor(0,0);
lcd.print("  PLEASE SCAN   ");
lcd.setCursor(0,1);
lcd.print("   YOUR CARD    ");}}
```