CHAPTER 1 – INTRODUCTION

1.1 GENERAL

The sanitizer detection block of this project involves using an MQ3 sensor to detect the presence of sanitizer in a person's hands. The MQ3 sensor detects the alcohol content in the sanitizer and sends a signal to the microcontroller. If the alcohol content is above a certain threshold, the microcontroller sends a signal to the Servo motor, which opens the door for the person.

The RFID-based attendance-making system is another important block of this project. The system uses an RFID reader to detect the presence of an RFID tag carried by a person. The MQ3 sensor detects the alcohol content in the person's hand when he sanitized his hands and indicating whether they have used sanitizer or not. The system then logs the person's attendance in a database.

By combining these two blocks, this project provides an effective way to ensure that people entering a building have sanitized their hands, while also keeping track of their attendance. The system can be used in various settings, such as offices, schools, hospitals, and other public places.

Overall, this project demonstrates the use of advanced technology to improve public health and safety. By detecting the presence of sanitizer and keeping track of attendance, this system can help prevent the spread of infectious diseases and ensure a safe environment for everyone.

1.2 VIRAL PANDEMIC

An epidemic is a sudden outbreak of a disease that affects a large number of people within a specific community or region. In contrast, a pandemic is a global outbreak of a disease that can affect people in all parts of the world. Pandemics are often caused by viruses that are highly contagious and can spread quickly from person to person.

One of the most well-known pandemics in history is the Spanish flu pandemic of 1918-1919, which infected an estimated 500 million people worldwide and caused the deaths of millions. More recently, the COVID-19 pandemic, caused by the SARS-CoV-2 virus, has affected millions of people worldwide and led to significant disruptions in daily life.

Pandemics can have severe social, economic, and health consequences. They can overwhelm healthcare systems, cause shortages of essential supplies, and disrupt global trade and travel. Governments and public health organizations around the world work together to monitor the spread of infectious diseases and implement measures to contain their spread and mitigate their impact.

1.3 PREVENTING METHODS

Following the WHO's instructions can help keep you and others safe during the pandemic. The instructions include:

1. Get vaccinated: Vaccination is an important tool in preventing the spread of COVID-19. When it's your turn, get vaccinated and follow local guidance on vaccination.

2. Keep physical distance: Keep at least 1 meter (3 feet) distance from others, even if they don't appear to be sick. Avoid crowds and close contact to reduce the risk of transmission.

3. Wear a properly fitted mask: Wear a mask when physical distancing is not possible, in poorly ventilated settings, or when required by local guidelines. Make sure the mask covers your nose, mouth, and chin.

4. Clean your hands frequently: Use alcohol-based hand rub or soap and water to clean your hands frequently. This can help prevent the spread of the virus.

5. Cover your mouth and nose when you cough or sneeze: Use a bent elbow or tissue to cover your mouth and nose when coughing or sneezing. Dispose of used tissue immediately and clean your hands regularly.

6. Self-isolate if you develop symptoms: If you develop symptoms or test positive for COVID-19, self-isolate until you recover. This can help prevent the spread of the virus to others.

1.4 IMPORTANCE OF HAND SANITIZER

Wearing gloves can be helpful in certain situations, such as when handling contaminated materials or when performing medical procedures. However, it is important to remember that gloves are not a substitute for proper hand hygiene. Even when wearing gloves, it is still important to wash your hands frequently with soap and water or use hand sanitizer.

One of the risks of wearing gloves is that they can transfer germs from one surface to another, especially if they are not changed or removed frequently. This can lead to cross-contamination and the spread of germs. When removing gloves, it is important to do so carefully to avoid touching the outside of the glove and contaminating your hands.

It is also important to note that the virus can survive on gloves for several hours, which means that wearing gloves is not a foolproof method of protection against infection. In contrast, hand sanitizer can kill germs in as little as 15 seconds and is a more effective method of disinfection.

Health workers wear gloves only for specific tasks where they are necessary, such as during medical procedures or when handling contaminated materials. They are trained to use gloves properly and to follow strict protocols for their use and disposal. The general public should follow similar guidelines and use gloves only when necessary, while still maintaining proper hand hygiene.

1.5 CARELESSNESS OF PEOPLE

Yes, forgetting to sanitize your hands or avoiding hand sanitizer altogether can increase the risk of transmitting viruses from one person to another. Hand sanitizer is an important tool in preventing the spread of infectious diseases, especially when it is not possible to wash your hands with soap and water.

When people come into contact with surfaces or objects that may be contaminated with viruses, they can inadvertently transfer those germs to their hands. If they do not sanitize their hands or wash them with soap and water, they can potentially spread those germs to others through direct contact or by touching surfaces that others may come into contact with.

Hand sanitizer is a convenient and effective way to kill germs on your hands when you do not have access to soap and water. It is important to use hand sanitizer properly, by applying it to all surfaces of your hands and rubbing them together until they are dry.

By making a habit of using hand sanitizer regularly, you can help reduce the risk of transmitting viruses to others and protect yourself from infection. It is also important to remember to sanitize your hands before and after touching your face, before eating or preparing food, and after using the restroom.

CHAPTER 2 – I / O DEVICES

2.1 MQ3 SENSOR

It is a low-cost semiconductor sensor that can detect the presence of alcohol gases at concentrations from 0.05 mg/L to 10 mg/L. The sensitive material used for this sensor is SnO_2, whose conductivity is lower in clean air. Its conductivity increases as the concentration of alcohol gases increases.

The MQ3 sensor is capable to detect ethanol that in gas form. MQ3 alcohol sensor module can be easily interfaced with Microcontrollers, Arduino Boards, Raspberry Pi, etc. This module provides both digital and analog outputs. The MQ3 alcohol sensor operates on 5V DC and consumes approximately 800mW. It can detect alcohol concentrations ranging from 25 to 500 ppm.

When a SnO_2 semiconductor layer is heated to a high temperature, oxygen is adsorbed on the surface. When the air is clean, electrons from the conduction band of tin dioxide are attracted to oxygen molecules. This creates an electron depletion layer just beneath the surface of the SnO_2 particles, forming a potential barrier. As a result, the SnO_2 film becomes highly resistive and prevents electric current flow. In the presence of alcohol, however, the surface density of adsorbed oxygen decreases as it reacts with the alcohol, lowering the potential barrier. As a result, electrons are released into the tin dioxide, allowing current to freely flow through the sensor.

FIG 2.1.1 MQ3 SENSOR

2.2 RFID

An **RFID** or radio frequency identification system consists of two main components, a tag attached to the object to be identified, and a reader that reads the tag. A reader consists of a radio frequency module and an antenna that generates a high frequency electromagnetic field. Whereas the tag is usually a passive device (it does not have a battery). It consists of a microchip that stores and processes information, and an antenna for receiving and transmitting a signal.

When the tag is brought close to the reader, the reader generates an electromagnetic field. This causes electrons to move through the tag's antenna and subsequently powers the chip. The chip then responds by sending its stored information back to the reader in the form of another radio signal. This is called a backscatter. The reader detects and interprets this backscatter and sends the data to a computer or microcontroller.

FIG 2.2.1 RFID Reader

The RC522 RFID module based on the **MFRC522 IC from NXP** is one of the cheapest RFID options you can get online for less than four dollars. It usually comes with an RFID card tag and a key fob tag with 1KB of memory. And the best part is that it can write a tag that means you can store any message in it. The RC522 RFID reader module is designed to create a 13.56MHz electromagnetic field and communicate with RFID tags (ISO 14443A standard tags). The reader can communicate with a microcontroller over a 4-pin SPI with a maximum data rate of 10 Mbps. It also supports communication over I2C and UART protocols.

The RC522 RFID module can be programmed to generate an interrupt, allowing the module to alert us when a tag approaches it, instead of constantly asking the module "Is there a card nearby?". The module's operating voltage ranges from 2.5 to 3.3V, but the good news is that the logic pins are 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using a logic level converter.

2.3 SERVO MOTOR

There are many motors to pick from, but it's important to pick the right one for the job. If your project requires precise positioning, a servo motor is usually the best option. Instruct them where to point, and they'll do it for you. Servos are motors that allow you to precisely control physical movement because they generally move to a position rather than continuously rotating. They are simple to connect and control because the motor driver is built right into them.

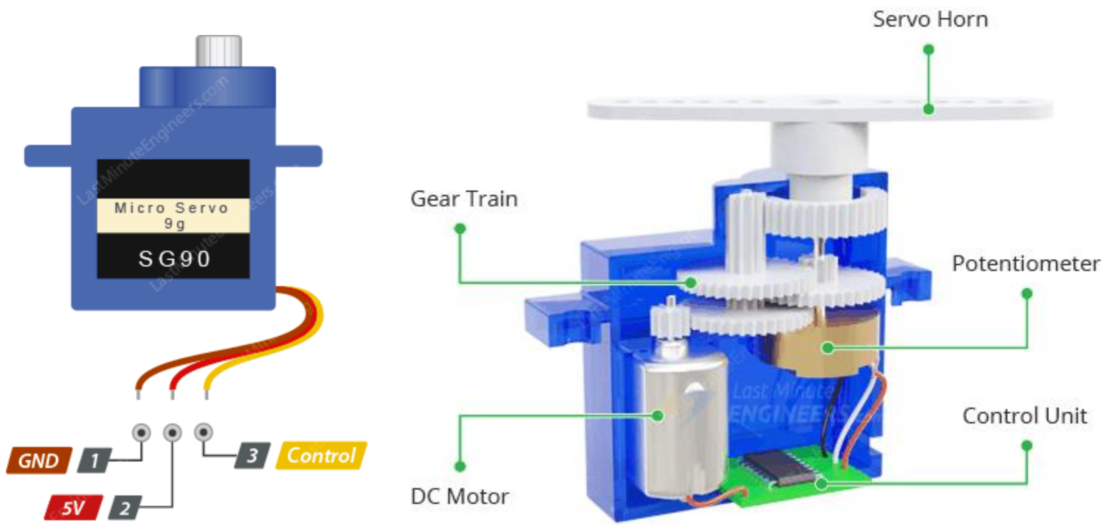

FIG 2.3.1 SERVO MOTOR

Servos contain a small DC motor connected to the output shaft through gears. The output shaft drives a servo horn and is also linked to a potentiometer (pot). The potentiometer provides position feedback to the error amplifier in the control unit, which compares the current position of the motor to the target position. In response to the error, the control unit adjusts the motor’s current position so that it matches the desired position. In control engineering, this mechanism is known as a servomechanism, or servo for short. It is a closed-loop control system that uses negative feedback to adjust the motor’s speed and direction to achieve the desired result. You can control the servo motor by sending a series of pulses to it. A typical servo motor expects a pulse every 20 milliseconds (i.e., the signal should be 50Hz). The length of the pulse determines the position of the servo motor.

CHAPTER 3 – DEVELOPMENT BOARDS

3.1 ARDUINO UNO

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong.

Arduino Uno Specification:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Inout Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Length: 68.6 mm
- Width: 58.4 mm
- Weight: 25 g

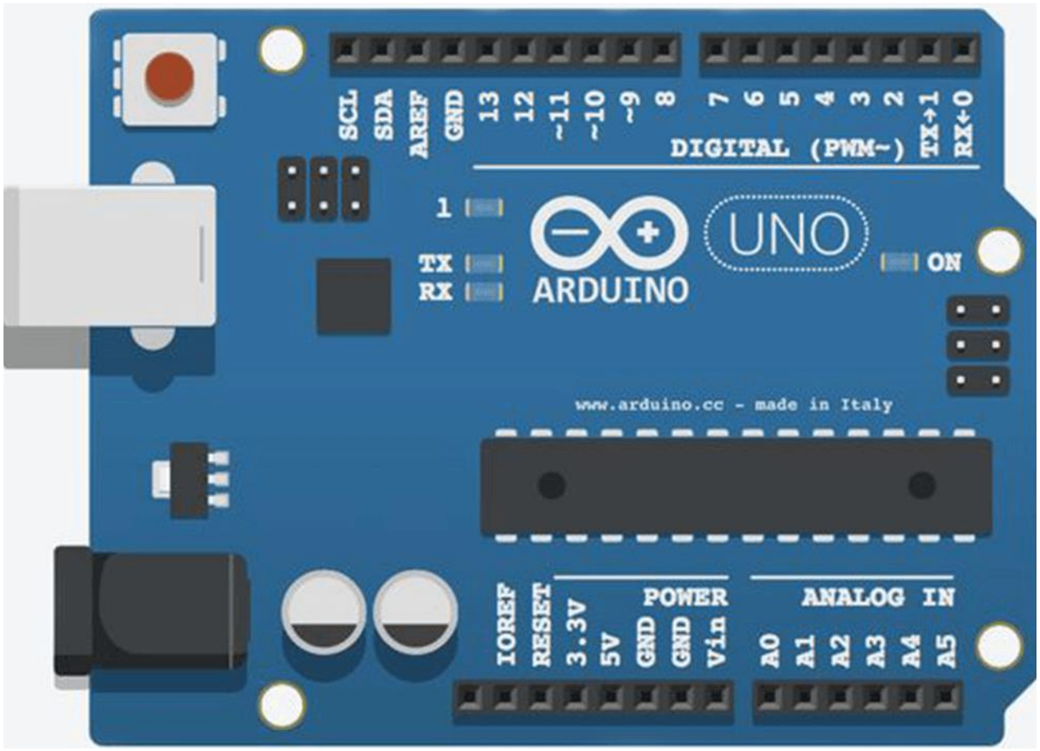

FIG 3.1.1 ARDUINO UNO (ATmega328P)

The Arduino Uno board can be powered via a USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

Vin - The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power sources). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V - This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3 - A 3.3 volt supply generated by the on-board regulator. The maximum current draw is 50 mA.

GND - Ground pins.

IOREF - This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

3.2 NODE MCU (ESP 8266)

The ESP8266 is a low-cost WiFi-enabled microcontroller, created and developed by Espressif Systems, a Shanghai-based Chinese company. The ESP8266 is equipped with a Tensilica Xtensa® 32-bit LX106 RISC microprocessor that operates at an adjustable clock frequency of 80 to 160 MHz and supports RTOS. The ESP8266 incorporates an 802.11b/g/n HT40 Wi-Fi transceiver, allowing it to connect to a Wi-Fi network to access the internet (Station mode) or to create its own Wi-Fi wireless network (Soft access point mode) to which other devices can connect.

3.2.1 ESP8266 NodeMCU

This is one of the most popular ESP8266 development boards. It has 4MB of flash memory, 11 GPIO pins, and one ADC pin with 10-bit resolution. This is what we use more often in our Wi-Fi and IoT projects. It is extremely versatile and perfect for beginners.

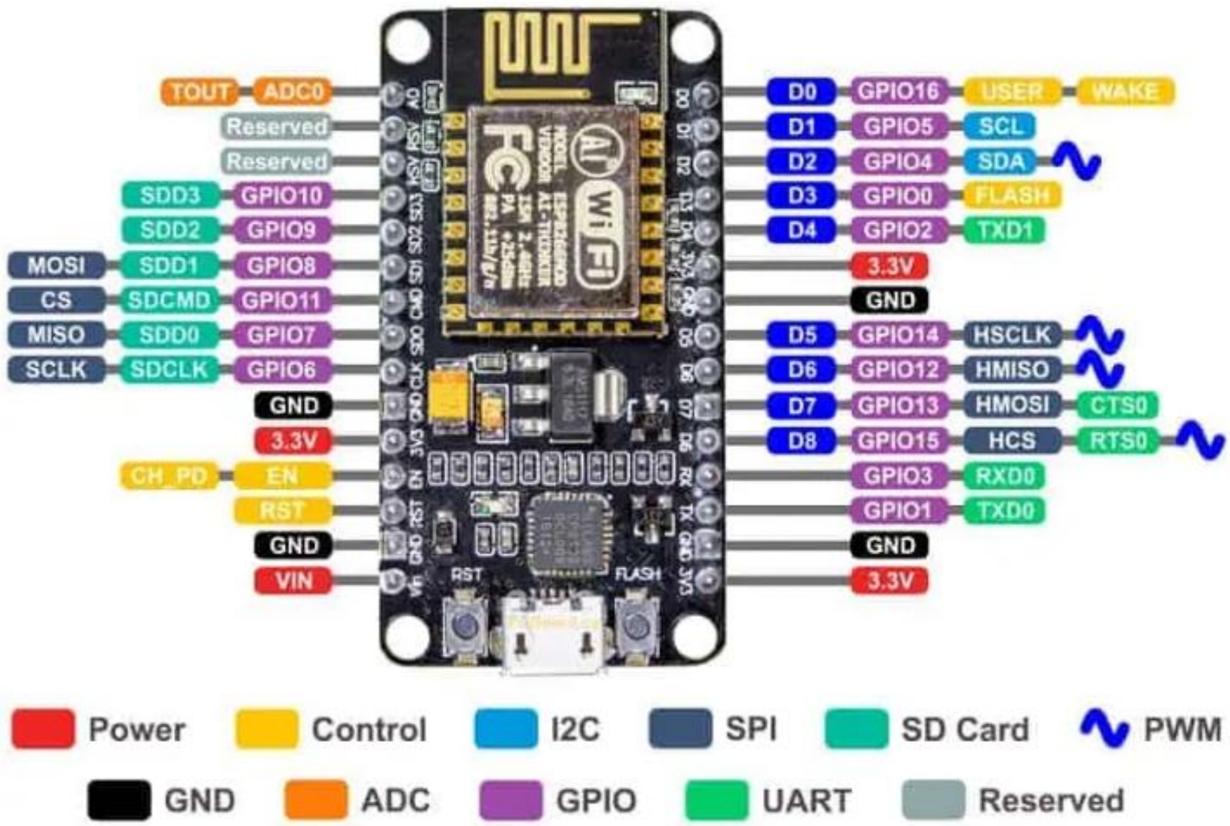

FIG 3.2.1 NODE MCU

3.2.2 PIN SPECIFICATION

POWER PIN - There are four power pins. VIN pin and three 3.3V pins.VIN can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the NodeMCU module – you can also supply 5V regulated to the VIN pin. 3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components. GND are the ground pins of NodeMCU/ESP8266.

I2C PIN are used to connect I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO PINS - NodeMCU/ESP8266 has 17 GPIO pins which can be assigned to functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC CHANNEL - The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART PINS - NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

PWM PINS -The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 µs to 10000 µs (100 Hz and 1 kHz).

SDIO PINS - NodeMCU/ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

CHAPTER 4 – PROPOSED SYSTEM

4.1 PANDEMIC PREVENTION SYSTEM BASED ON MQ3 SENSOR

The pandemic prevention system based on the MQ3 sensor with RFID and servo motor offers an effective way to reduce the spreading of viruses in public places. The system can detect the presence of hand sanitizer, which is essential in preventing the spread of infectious diseases. The MQ3 sensor can detect the ethanol content in the hand sanitizer, and this information is used to trigger the servo motor to open or close the door.

Moreover, the RFID-based attendance-making system ensures that everyone entering the premises has used hand sanitizer. The MQ3 sensor detects the alcohol content in the person's hand when he sanitizes his hands and, indicating whether they have used sanitizer or not. The system then logs the person's attendance in a database, providing a way to track people's movements and prevent the spread of viruses.

The MQ3 sensor is a crucial component of this project as it can detect ethanol gas accurately. It provides both digital and analog outputs, making it easy to interface with microcontrollers such as Arduino and Raspberry Pi. With its high sensitivity, low power consumption, and ease of use, the MQ3 sensor is an ideal choice for detecting hand sanitizer.

Overall, this project offers an effective and efficient way to prevent the spread of infectious diseases. By detecting hand sanitizer and keeping track of sanitation of hand, the system can help create a safe and healthy environment for everyone. This project demonstrates the power of electronics in solving real-world problems and improving public health and safety.

4.1.1 AUTOMATIC DOOR LOCK SYSTEM INTEFACED WITH MQ3 SENSOR

The MQ3 sensor used in this system is a gas sensor that can detect the presence of alcohol vapor in the air. This makes it suitable for detecting whether hands have been sanitized with an alcohol-based solution, which is commonly used in hospitals, clinics, and other public places. The MQ3 sensor produces an analog voltage output that is proportional to the concentration of alcohol vapor in the air.

The Arduino Uno is used as the microcontroller in this system. It receives the inverter circuit signal. If the inverter first pin connected to ground, the Arduino sends a signal to the servo motor to open the door. The servo motor is used to control the movement of the door, which can be a sliding door, swing door, or any other type of door.

BLOCK DIAGRAM

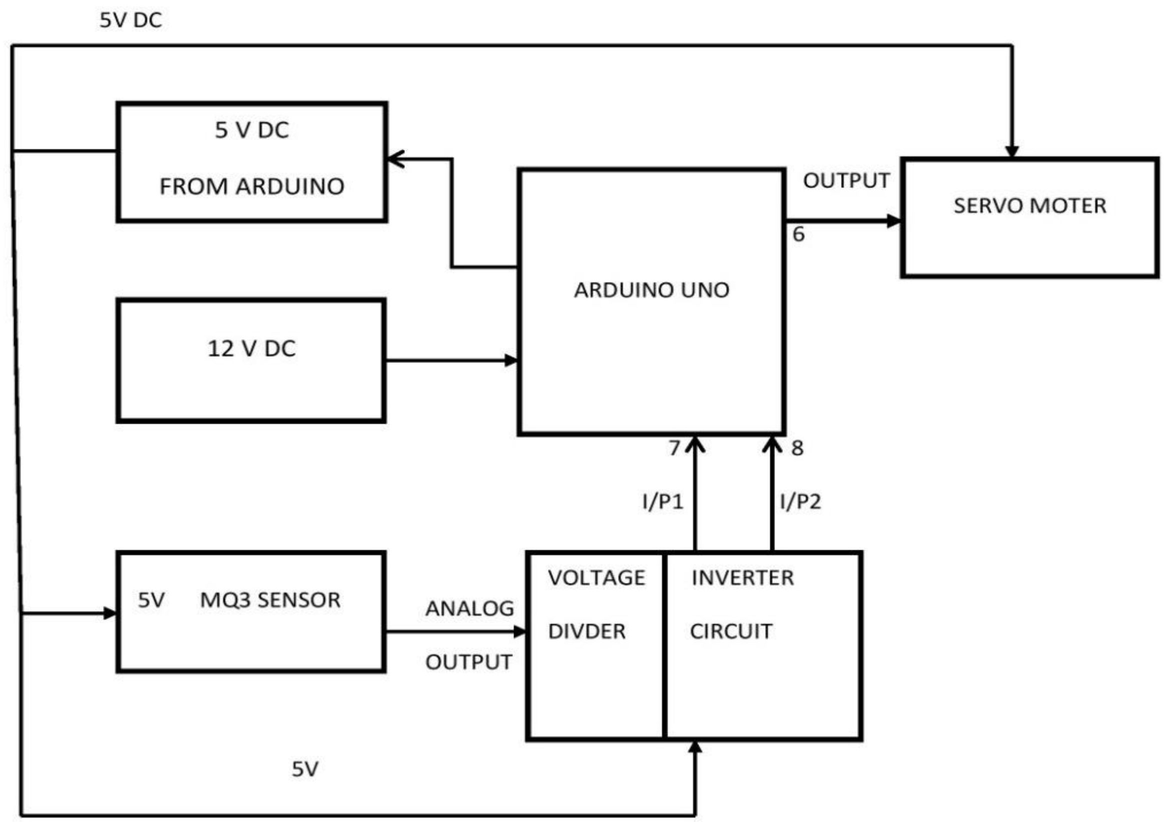

FIG 4.1.1 AUTOMATIC DOOR LOCK SYSTEM WITH MQ 3 SENSOR

To control the direction of the servo motor, an inverter circuit is used. This circuit consists of a transistor and resistors, which are connected to the Arduino UNO. When the inverter circuit's first pin connected to ground then Arduino allows the current to flow through the servo motor in one direction. When the inverter circuit's second pin connected to ground then Arduino allows the current to flow through the servo motor in the opposite direction. This allows the servo motor to rotate in both clockwise and anticlockwise directions, which is necessary to open and close the door.

The door is kept open for a few seconds after the analog output from the MQ3 sensor reaches the threshold level. This is done to ensure that people have enough time to pass through the door. The duration for which the door remains open can be controlled by using the right amount of resistor at the output of the MQ3 sensor. A larger resistor value will result in a longer delay before the door is closed.

The LED indicators used in this system are used to indicate whether hands have been sanitized or not. If the MQ3 sensor output is LOW, indicating that there is no alcohol vapor in the air, the RED LED is in ON condition and the GREEN LED is in OFF condition. This indicates that the hands have not been sanitized and the person should sanitize their hands before entering. If the MQ3 sensor output is HIGH, indicating that there is alcohol vapor in the air, the RED LED is in OFF condition and the GREEN LED is in ON condition. This indicates that the hands have been sanitized and the person can enter.

In conclusion, the automatic door opening system using a servo motor and MQ3 sensor is a useful system that can be used to ensure that people sanitize their hands before entering a public place. The system is based on a simple principle of detecting the presence of alcohol vapor in the air, which is an indication of whether hands have been sanitized or not. The system is easy to implement and can be used in a variety of applications, such as hospitals, clinics, airports, and other public places.

4.1.2 RFID ATTENDANCE SYSTEM INTERFACED WITH MQ3 SENSOR

Radio Frequency Identification (RFID) technology has become increasingly popular in recent years due to its ability to identify and track objects wirelessly. The system comprises two primary components: RFID tags and readers. The reader is a device that has one or more antennas that emit radio waves and receive signals back from the RFID tag. The tag contains a unique identification number (UID) that can be used to retrieve data about the object to which it is attached, such as its name, email address, or mobile number.

In this attendance-making system, the RFID reader is interfaced with a NODE MCU (ESP 8266) to enable the attendance to be saved online. By connecting the NODE MCU to the internet using hotspot technology, attendance data can be transmitted and stored in the cloud. The NODE MCU is also interfaced with an MQ3 sensor and a transistor.

BLOCK DIAGRAM

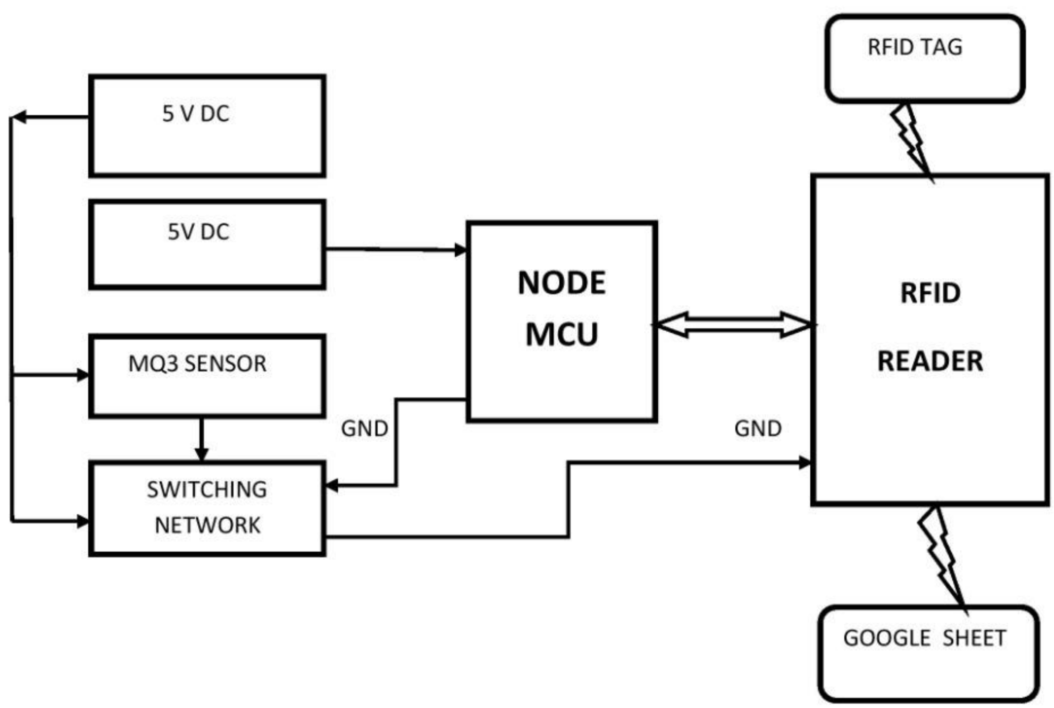

FIG 4.1.2 ATTENDANCE SYSTEM WITH MQ3 SENSOR

The MQ3 sensor measures the alcohol content in the air, and its output is used to control the ON/OFF state of the transistor. If the transistor is ON, the attendance can be registered, but if it is OFF, the system cannot register attendance. This feature ensures that employees or students who does not sanitize their hands unable to register attendance.

After the RFID reader reads the UID of a card, the NODE MCU records the attendance data and updates it in a Google sheet. The use of RFID technology in this system reduces the need for physical contact and, therefore, minimizes the risk of spreading viruses and bacteria through hand contact. Additionally, the use of an alcohol sensor ensures that the attendance system is used responsibly and does not encourage risky behavior.

In conclusion, the attendance-making system based on RFID technology and NODE MCU has numerous benefits, including the ability to save attendance data online, reduce physical contact, promote responsible behaviour and mandatory the hand sanitation. The system is easy to use and can be implemented in a wide range of settings, including offices, schools, and hospitals.

4.1.3 INTERFACING DOOR LOCK SYSTEM AND RFID ATTENDANCE SYSTEM THROUGH MQ3 SENSOR

The interfacing of a door lock system and an RFID attendance system through an MQ3 sensor is a great innovation in pandemic prevention technology. This system combines two different systems into one device and ensures that people entering public places have sanitized their hands. In this system, the Arduino UNO and servo motor are used for the door lock system, while the Node MCU and RFID reader are used for the attendance system.

The MQ3 sensor plays a crucial role in this system by detecting the presence of ethanol in a person's hand after they have sanitized. When a person sanitizes their hands, the ethanol gas is released and detected by the MQ3 sensor. The data is then sent to the inverter circuit, which has two outputs connected to the Arduino UNO digital pins. The first output is connected to ground and the Arduino UNO commands the servo motor to open the door. The second output is also connected to ground, and the Arduino UNO commands the servo motor to close the door.

In addition to controlling the door lock system, the inverter circuit also has a separate single transistor circuit that controls the attendance system. This transistor is controlled by the output of the MQ3 sensor. If the transistor is in the ON state, the system allows the person to register their attendance. If the transistor is in the OFF state, the system does not allow the person to register their attendance.

The MQ3 sensor is the core of this project because it detects the sanitizer and ensures that people entering public places have sanitized their hands. If a person does not sanitize their hands, they cannot register their attendance or enter public places like hospitals, schools, colleges, shopping malls, small shops, airports, and so on. By implementing this system, we can make hand sanitation mandatory and reduce the transmission of viruses through hands.

Overall, this device is very useful during pandemic situations because it ensures that people entering public places have sanitized their hands. It is a simple and effective way to prevent the spread of viruses and ensure the safety of the public. With this system in place, we can make public places safer and reduce the risk of infection.

CIRCUIT DIAGRAM:

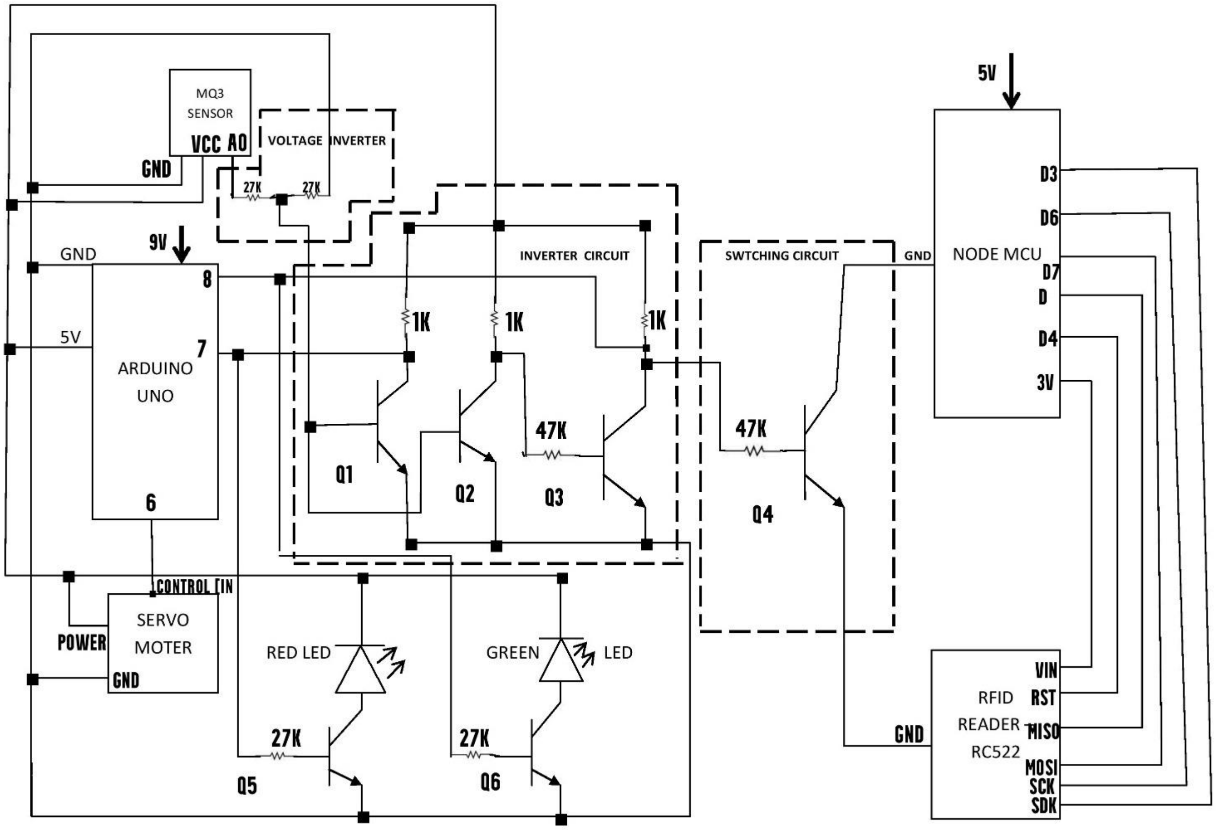

FIG 4.1.3 PANDEMIC PREVENTION SYSTEM USING MQ 3 SENSOR

The implementation of this device is also very cost-effective and easy to install. It can be installed in various public places like hospitals, schools, colleges, shopping malls, airports, and other public places to prevent the spread of viruses through hands. The device can be easily integrated with existing door lock and attendance systems, making it easy to adopt.

Furthermore, this device also helps in promoting hygiene and cleanliness in public places. It encourages people to sanitize their hands and ensures that they follow the hygiene protocols. This can be especially useful in places where there are high chances of virus transmission.

In addition to the benefits during the pandemic, this device can also be used in various other applications. For instance, it can be used in industries that deal with hazardous chemicals. It can detect the presence of chemicals on a person's hand and ensure that they are not allowed to enter a certain area until they have washed their hands properly.

Moreover, this device can also be used in homes to monitor the hygiene of family members. It can be connected to the door lock system and attendance system of a house and ensure that only people who have sanitized their hands are allowed to enter the house. This can be especially useful during flu seasons or other times when there is a high risk of infection.

In conclusion, the interfacing of a door lock system and an RFID attendance system through an MQ3 sensor is a great innovation in pandemic prevention technology. It is a simple and effective way to prevent the spread of viruses through hands and ensure the safety of the public. This device is cost-effective, easy to install, and can be used in various public places and industries. By implementing this system, we can make public places safer and promote hygiene and cleanliness.

4.2 APPLICATIONS & ADVANTAGES

4.2.1 APPLICATIONS

Automatic Door Opening System with MQ3 Sensor and Servo Motor:

1. Hospitals and clinics: to ensure that all visitors sanitize their hands before entering the facility, thus reducing the risk of spreading infections.

2. Schools and universities: to ensure that students and teachers sanitize their hands before entering the classroom or laboratory, thus creating a safer learning environment.

3. Airports and public transportation: to ensure that passengers sanitize their hands before boarding, thus reducing the risk of spreading infections.

4. Retail Stores: Retail stores can use the automatic door opening system to ensure that all customers entering the store sanitize their hands before handling merchandise.

RFID Attendance System Interfaced with MQ3 Sensor:

1. Offices and workplaces: to ensure that employees sanitize their hands before registering their attendance, thus promoting a healthier and safer work environment.

2. Schools and universities: to ensure that students sanitize their hands before registering their attendance, thus promoting better hygiene practices.

3. Hospitals and clinics: to ensure that staff and visitors sanitize their hands before entering the facility, thus reducing the risk of spreading infections.

Both of these devices can be used in a wide range of settings where hand sanitation is important for maintaining a safe and healthy environment. They can also be used in combination with other devices, such as temperature sensors or facial recognition systems, to create a more comprehensive safety system.

For example, in a hospital setting, the automatic door opening system can be combined with temperature sensors to ensure that visitors with fever or other symptoms of illness are not allowed to enter the facility. Similarly, the RFID attendance system can be combined with facial recognition systems to ensure that only authorized personnel are allowed to enter restricted areas.

Overall, these devices have the potential to improve hygiene practices, reduce the spread of infections, and create a safer and healthier environment in a variety of settings.

4.2.2 ADVANTAGES

AUTOMATIC DOOR LOCK SYSTEM INTERFACED WITH MQ3 SENSOR

The automatic door opening system using a servo motor and MQ3 sensor is a useful system that can be used to ensure that people sanitize their hands before entering a public place. The system is based on a simple principle of detecting the presence of alcohol vapor in the air, which is an indication of whether hands have been sanitized or not. The system is easy to implement and can be used in a variety of applications, such as hospitals, clinics, airports, and other public places.

The advantages of the system include:

1. Improved hygiene:

The system promotes good hygiene practices by ensuring that people sanitize their hands before entering a public place. This helps to prevent the spread of germs and bacteria, which can cause diseases.

2. Automatic operation:

The system operates automatically, which means that there is no need for manual intervention. This makes it a convenient and user-friendly solution for public places where large numbers of people pass through.

3. Accurate detection:

The MQ3 sensor used in the system is highly accurate and can detect even small amounts of alcohol vapor in the air. This ensures that people who have sanitized their hands are identified accurately.

4. Easy to use:

The system is easy to use and requires no special training or expertise. Anyone can use it without difficulty, which makes it a convenient solution for public places.

5. Cost-effective:

The system is cost-effective and does not require a large investment. This makes it a practical solution for small businesses and public places with limited budgets.

RFID BASED ATTENDANCE SYSTEM INTERFACED WITH MQ3 SENSOR

The RFID attendance system interfaced with an MQ3 sensor also has numerous benefits, including the ability to save attendance data online, reduce physical contact, promote responsible behavior, and mandatory hand sanitation.

The advantages of this system include:

1. Online data storage:

Attendance data is saved online, making it easy to access and retrieve. This eliminates the need for physical storage devices such as memory cards, which can be lost or damaged.

2. Portability and safety:

The online storage of attendance data makes the system portable and safe. It can be accessed from anywhere with an internet connection, which makes it a convenient solution for businesses and organizations with multiple locations.

3. Low risk of virus transmission:

The use of RFID technology reduces the need for physical contact, which minimizes the risk of spreading viruses and bacteria through hand contact.

4. Accurate attendance tracking:

The system is highly accurate and can track attendance data in real-time. This eliminates the need for manual attendance tracking, which can be time-consuming and prone to errors.

5. Encourages responsible behavior:

The use of an alcohol sensor ensures that the attendance system is used responsibly and does not encourage risky behavior. Employees or students who do not sanitize their hands are unable to register attendance, which promotes good hygiene practices.

In conclusion, both the automatic door opening system and the RFID attendance system interfaced with an MQ3 sensor are innovative solutions that promote good hygiene practices and help prevent the spread of germs and bacteria. These systems are easy to use, accurate, and cost-effective, making them practical solutions for public places, offices, schools, hospitals, and other settings.

CHAPTER 5 – COST ESTIMATION

5.1 COST OF ATTENDANCE SYSTEM INTERFACED WITH MQ3 SENSOR

TABLE 5.1

SI NO	COMPONENT	SPECIFICATION	QUANTITY	COST
1	NODE MCU	ESP 8266	1	350
2	RFID READER	RC 522	1	150
3	RFID CARD		2	40
4	DOT PCB	7x7 (cm)	1	20
5	MQ 3 SENSOR	MQ 3	1	150
6	TRANSISTOR	2N2222A	6	30
7	RESISTOR	1K 27K 47K	3 4 2	3 4 2

TOTAL COST IS – ₹749

5.2 COST OF AUTOMATIC DOOR LOCK SYSTEM INTERFACED WITH MQ3 SENSOR

TABLE 5.2

SI NO	COMPONENT	SPECIFICATION	QUANTITY	COST
1	ARDUINO UNO	ATmega328P	1	780
2	SERVO MOTOR	9G	1	130
3	TRANSISTOR	2N2222A	6	30
4	RESISTOR	1K 27K 47K	3 4 2	3 4 2
5	DOT PCB	7x5 (cm)	1	20
6	MQ 3 SENSOR	MQ 3	1	150

TOTAL COST IS – ₹ 1,119

5.3 COST OF INTERFACING DOOR LOCK SYSTEM AND RFID ATTENDANCE SYSTEM THROUGH MQ3 SENSOR

TABLE 5.3

SI NO	COMPONENT	SPECIFICATION	QUANTITY	COST
1	ARDUINO UNO	ATmega328P	1	780
2	NODE MCU	ESP 8266	1	350
3	MQ 3 SENSOR	MQ 3	1	150
4	SERVO MOTOR	9G	1	130
5	RFID READER	RC 522	1	150
6	RFID CARD		2	40
7	TRANSISTOR	2N2222A	6	30
8	RESISTOR	1K 27K 47K	3 4 2	3 4 2
9	DOT PCB	7x5 (cm) 7x7 (cm)	1 1	20 20
10	JUMPER WIRES	M-M M-F	10	20
11	POWER ADAPTER	9 V	1	110

TOTAL COST IS – ₹ 1809

CHAPTER 6 – CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

We Conclude by Using these two devices can reduce the transmission of viruses through hands. When a product can reach all category of people then the product is known as a successful product in the market and make a profit. The MQ3 sensor, NODE MCU and Arduino uno are very low-cost devices, So small shops also can easily afford these products. Definitely, this product can contribute to reducing the transmission of viruses in pandemic situations and also make a profit to the manufacturer, and very helpful to people to avoid virus transmission.

6.2 FUTURE SCOPE

The future scope for a device interfaced with mask detection and temperature detection in clinics and hospitals is quite promising. Here are some potential benefits and opportunities:

1. Improved safety: By detecting and alerting individuals who are not wearing masks or have a high temperature, the device can help prevent the spread of contagious diseases in healthcare settings. This can help keep both patients and healthcare workers safer.

2. Enhanced efficiency: The device can automate the process of checking whether patients or visitors are wearing masks and have a normal temperature. This can save time for healthcare workers and streamline the check-in process, making it more efficient.

3. Better data collection: The device can collect and store data on the number of patients and visitors who enter the clinic or hospital, as well as their mask usage and temperature readings. This data can be used to identify trends and patterns over time, which can help with decision-making and resource allocation.

4. Potential for integration with other systems: The device could be integrated with other healthcare systems, such as electronic medical records or appointment scheduling software, to provide a more comprehensive view of patient information.

5. Potential for expansion into other industries: The technology used in the device could be applied to other industries, such as retail or transportation, to help monitor and control the spread of contagious diseases in public settings.

Overall, the future scope for a device interfaced with mask detection and temperature detection in clinics and hospitals is promising, with potential benefits for safety, efficiency, data collection, and integration with other systems.

CHAPTER 7 – APPENDIX

ARDUINO PROGRAM

```
#include <Servo.h>
#define LEFT 7
#define RIGHT 8
Servo myservo;
int angle =90;    // initial angle  for servo
int angleStep =10;
void setup() {
  Serial.begin(9600);          //  setup serial
  myservo.attach(6);  // attaches the servo on pin 9 to the servo object
  pinMode(LEFT,INPUT_PULLUP); // assign pin 12 ass input for Left button
  pinMode(RIGHT,INPUT_PULLUP);
  myservo.write(angle);// send servo to the middle at 90 degrees
}
void loop() {
  while(digitalRead(RIGHT) == LOW){
    if (angle > 0 && angle <= 180) {
      angle = angle - angleStep;
       if(angle < 0){
        angle = 0;
       }else{
      myservo.write(angle); // move the servo to desired angle
      Serial.print("Moved to: ");
```

```
    Serial.print(angle);   // print the angle
    Serial.println(" degree");
     }
   }
  delay(100); // waits for the servo to get there
  }
  while(digitalRead(LEFT) == LOW){
   if (angle >= 0 && angle <= 180) {
    angle = angle + angleStep;
    if(angle >180){
      angle =180;
     }else{
    myservo.write(angle); // move the servo to desired angle
    Serial.print("Moved to: ");
    Serial.print(angle);   // print the angle
    Serial.println(" degree");


    }
   }
 delay(100); // waits for the servo to get there
  }
}
```

NODE MCU PROGRAM TO SAVE DATA IN RFID

```
#include <SPI.h>

#include <MFRC522.h>

//------------------------------------------------

//GPIO 0 --> D3

//GPIO 2 --> D4

const uint8_t RST_PIN = D3;

const uint8_t SS_PIN = D4;

//------------------------------------------------

MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

//------------------------------------------------

/* Be aware of Sector Trailer Blocks */

int blockNum = 4;

byte block_data[16];

/* Create array to read data from Block */

/* Length of buffer should be 4 Bytes

more than the size of Block (16 Bytes) */

byte bufferLen = 18;

byte readBlockData[18];

//------------------------------------------------

MFRC522::StatusCode status;

//------------------------------------------------
```

```
void setup()
{
  //------------------------------------------------------
  //Initialize serial communications with PC
  Serial.begin(9600);
  //------------------------------------------------------
  //Initialize SPI bus
  SPI.begin();
  //------------------------------------------------------
  //Initialize MFRC522 Module
  mfrc522.PCD_Init();
  Serial.println("Scan a MIFARE 1K Tag to write data...");
  //------------------------------------------------------
}
* loop() function
void loop()
{
  //------------------------------------------------------------------------------
  /* Prepare the ksy for authentication */
  /* All keys are set to FFFFFFFFFFFFh at chip delivery from the factory */
  for (byte i = 0; i < 6; i++){
    key.keyByte[i] = 0xFF;
  }
```

```
//-----------------------------------------------------------------------------------
/* Look for new cards */
/* Reset the loop if no new card is present on RC522 Reader */
if ( ! mfrc522.PICC_IsNewCardPresent()){return;}
//-----------------------------------------------------------------------------------
/* Select one of the cards */
if ( ! mfrc522.PICC_ReadCardSerial()) {return;}
//-----------------------------------------------------------------------------------
Serial.print("\n");
Serial.println("**Card Detected**");
/* Print UID of the Card */
Serial.print(F("Card UID:"));
for (byte i = 0; i < mfrc522.uid.size; i++){
  Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
  Serial.print(mfrc522.uid.uidByte[i], HEX);
}
Serial.print("\n");
/* Print type of card (for example, MIFARE 1K) */
Serial.print(F("PICC type: "));
MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
Serial.println(mfrc522.PICC_GetTypeName(piccType));
//-----------------------------------------------------------------------------------
blockNum = 4;
```

```
toBlockDataArray("data 1"); //Student ID
WriteDataToBlock(blockNum, block_data);
ReadDataFromBlock(blockNum, readBlockData);
dumpSerial(blockNum, readBlockData);
blockNum = 5;
toBlockDataArray("data 2"); //First Name
WriteDataToBlock(blockNum, block_data);
ReadDataFromBlock(blockNum, readBlockData);
dumpSerial(blockNum, readBlockData);
blockNum = 6;
toBlockDataArray("data 3"); //Last Name
WriteDataToBlock(blockNum, block_data);
ReadDataFromBlock(blockNum, readBlockData);
dumpSerial(blockNum, readBlockData);
blockNum = 8;
toBlockDataArray("data 4"); //Father's Name
WriteDataToBlock(blockNum, block_data);
ReadDataFromBlock(blockNum, readBlockData);
dumpSerial(blockNum, readBlockData);
blockNum = 9;
toBlockDataArray("data 4"); //Date of Birth
WriteDataToBlock(blockNum, block_data);
ReadDataFromBlock(blockNum, readBlockData);
```

```
  dumpSerial(blockNum, readBlockData);

  blockNum = 10;

  toBlockDataArray("+917603906795"); //Phone Number

  WriteDataToBlock(blockNum, block_data);

  ReadDataFromBlock(blockNum, readBlockData);

  dumpSerial(blockNum, readBlockData);

}

* Writ() function

void WriteDataToBlock(int blockNum, byte blockData[])

{

   Serial.print("Writing data on block ");

   Serial.print(blockNum);

  //------------------------------------------------------------------------------

  /* Authenticating the desired data block for write access using Key A */

  status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
blockNum, &key, &(mfrc522.uid));

  if (status != MFRC522::STATUS_OK){

    Serial.print("Authentication failed for Write: ");

    Serial.println(mfrc522.GetStatusCodeName(status));

    return;

  }

  //------------------------------------------------------------------------------
```

```
  else {
    Serial.println("Authentication success");
  }
  //----------------------------------------------------------------------------
  /* Write data to the block */
  status = mfrc522.MIFARE_Write(blockNum, blockData, 16);
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Writing to Block failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  else
  {Serial.println("Data was written into Block successfully");}
  //----------------------------------------------------------------------------
}
/****************************************************************************************************
 * ReadDataFromBlock() function

****************************************************************************************************/
void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
   Serial.print("Reading data from block ");
```

```
 Serial.println(blockNum);

//----------------------------------------------------------------------------

/* Authenticating the desired data block for Read access using Key A */

status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
blockNum, &key, &(mfrc522.uid));

//----------------------------------------------------------------------------

if (status != MFRC522::STATUS_OK){

 Serial.print("Authentication failed for Read: ");

 Serial.println(mfrc522.GetStatusCodeName(status));

 return;

}

else {

  Serial.println("Authentication success");

}

//----------------------------------------------------------------------------

/* Reading data from the Block */

status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);

if (status != MFRC522::STATUS_OK){

  Serial.print("Reading failed: ");

  Serial.println(mfrc522.GetStatusCodeName(status));

  return;

}
```

```
  else {
    Serial.println("Block was read successfully");
  }
  //----------------------------------------------------------------------------
}
/****************************************************************
**********************************
 * dumpSerial() function

****************************************************************
**********************************/
void dumpSerial(int blockNum, byte blockData[])
{
   Serial.print("\n");
   Serial.print("Data in Block:");
   Serial.print(blockNum);
   Serial.print(" --> ");
   for (int j=1 ; j<16 ; j++){
     Serial.write(readBlockData[j]);
   }
   Serial.print("\n");Serial.print("\n");
}
/****************************************************************
**********************************
```

```
 * dumpSerial() function

*****************************************************************
*********************************/
void toBlockDataArray(String str)
{
  byte len = str.length();
  if(len > 16) len = 16;
  for (byte i = 0; i < len; i++) block_data[i] = str[i];
  for (byte i = len; i < 16; i++) block_data[i] = ' ';
}
```

PROGRAM FOR ATTENDANCE SYSTEM

```
#include <SPI.h>

#include <MFRC522.h>

#include <Arduino.h>

#include <ESP8266WiFi.h>

#include <ESP8266HTTPClient.h>

#include <WiFiClient.h>

#include <WiFiClientSecureBearSSL.h>

//-----------------------------------------------------------

const String web_app_url = "https://script.google.com/macros/s/AKfycbzI-
Ya5JNDScYDtnytAsUA750hYbPHeqoyZWA1XADxMGG0CMIzu9FOtwrWVW
3v--LsdRA/exec";

//-----------------------------------------------------------

#define WIFI_SSID "AK"

#define WIFI_PASSWORD "AK12345AK"

//-----------------------------------------------------------

int blocks[] = {4,5,6,8,9,10};

#define blocks_len  (sizeof(blocks) / sizeof(blocks[0]))

//-----------------------------------------------------------

//GPIO 0 --> D3

//GPIO 2 --> D4

//GPIO 4 --> D2

#define RST_PIN  0
```

```
#define SS_PIN   2

#define BUZZER   4

//-----------------------------------------

MFRC522 mfrc522(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

MFRC522::StatusCode status;

//-----------------------------------------

/* Be aware of Sector Trailer Blocks */

int blockNum = 2;

/* Create another array to read data from Block */

/* Legthn of buffer should be 2 Bytes more than the size of Block (16 Bytes) */

byte bufferLen = 18;

byte readBlockData[18];

//-----------------------------------------
```

// Fingerprint for demo URL, expires on **Monday, May** 2, **2022 7:20:58 AM, needs to be updated well before this date**

```
//const uint8_t fingerprint[20] = {0x9a, 0x87, 0x9b, 0x82, 0xe9, 0x19, 0x7e, 0x63, 0x8a, 0xdb, 0x67, 0xed, 0xa7, 0x09, 0xd9, 0x2f, 0x30, 0xde, 0xe7, 0x3c};

//const uint8_t fingerprint[20] = {0x9a, 0x71, 0xde, 0xe7, 0x1a, 0xb2, 0x25, 0xca, 0xb4, 0xf2, 0x36, 0x49, 0xab, 0xce, 0xf6, 0x25, 0x62, 0x04, 0xe4, 0x3c};

//9a 87 9b 82 e9 19 7e 63 8a db 67 ed a7 09 d9 2f 30 de e7 3c

//9a 71 de e7 1a b2 25 ca b4 f2 36 49 ab ce f6 25 62 04 e4 3c

//-----------------------------------------
```

```
/*****************************************************************
*********************************

 * setup() function

*****************************************************************
*********************************/

void setup()

{

  //---------------------------------------------

  /* Initialize serial communications with the PC */

  Serial.begin(9600);

  //Serial.setDebugOutput(true);

  //---------------------------------------------

  //WiFi Connectivity

  Serial.println();

  Serial.print("Connecting to AP");

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  while (WiFi.status() != WL_CONNECTED){

    Serial.print(".");

    delay(200);

  }

  Serial.println("");

  Serial.println("WiFi connected.");

  Serial.println("IP address: ");
```

```
  Serial.println(WiFi.localIP());
  Serial.println();
  //-----------------------------------------------------
  /* Set BUZZER as OUTPUT */
  pinMode(BUZZER, OUTPUT);
  //-----------------------------------------------------
  /* Initialize SPI bus */
  SPI.begin();
  //-----------------------------------------------------
}
/*******************************************************************
**********************************
 * loop() function
*******************************************************************
**********************************/
 void loop()
{
  //---------------------------------------------------------------------------
  /* Initialize MFRC522 Module */
  mfrc522.PCD_Init();
  /* Look for new cards */
  /* Reset the loop if no new card is present on RC522 Reader */
  if ( ! mfrc522.PICC_IsNewCardPresent()) {return;}
```

```
/* Select one of the cards */
if ( ! mfrc522.PICC_ReadCardSerial()) {return;}
/* Read data from the same block */
Serial.println();
Serial.println(F("Reading last data from RFID..."));
//--------------------------------------------------------------------
String fullURL = "", temp;
for (byte i = 0; i < blocks_len; i++) {
  ReadDataFromBlock(blocks[i], readBlockData);
  if(i == 0){
    temp = String((char*)readBlockData);
    temp.trim();
    fullURL = "data" + String(i) + "=" + temp;
  }
  else{
    temp = String((char*)readBlockData);
    temp.trim();
    fullURL += "&data" + String(i) + "=" + temp;
  }
}
//Serial.println(fullURL);
fullURL.trim();
fullURL = web_app_url + "?" + fullURL;
```

```
  fullURL.trim();
  Serial.println(fullURL);
  //-----------------------------------------------------------------------
  digitalWrite(BUZZER, HIGH);
  delay(200);
  digitalWrite(BUZZER, LOW);
  delay(200);
  digitalWrite(BUZZER, HIGH);
  delay(200);
  digitalWrite(BUZZER, LOW);
  delay(3000);
  //-----------------------------------------------------------------------
  if (WiFi.status() == WL_CONNECTED) {
    //-------------------------------------------------------------------------
    std::unique_ptr<BearSSL::WiFiClientSecure>client(new
BearSSL::WiFiClientSecure);
    //-------------------------------------------------------------------------
    //uncomment following line, if you want to use the SSL certificate
    //client->setFingerprint(fingerprint);
    //or uncomment following line, if you want to ignore the SSL certificate
    client->setInsecure();
    //-------------------------------------------------------------------------
    HTTPClient https;
```

```
  Serial.print(F("[HTTPS] begin...\n"));
  //---------------------------------------------------------------------------
  if (https.begin(*client, (String)fullURL)){
    //-------------------------------------------------------------
    // HTTP
    Serial.print(F("[HTTPS] GET...\n"));
    // start connection and send HTTP header
    int httpCode = https.GET();
    //-------------------------------------------------------------
    // httpCode will be negative on error
    if (httpCode > 0) {
      // HTTP header has been send and Server response header has been handled
      Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
      // file found at server
    }
    //-------------------------------------------------------------
    else
    {Serial.printf("[HTTPS] GET... failed, error: %s\n",
https.errorToString(httpCode).c_str());}
    //-------------------------------------------------------------
    https.end();
    delay(1000);
  }
```

```
      else {
     Serial.printf("[HTTPS} Unable to connect\n");
   }
  }
}
/*****************************************************************
*********************************
 * ReadDataFromBlock() function

*****************************************************************
*********************************/
void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
  //----------------------------------------------------------------------------
  /* Prepare the ksy for authentication */
  /* All keys are set to FFFFFFFFFFFFh at chip delivery from the factory */
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  //----------------------------------------------------------------------------
  /* Authenticating the desired data block for Read access using Key A */
  status =
mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A,
blockNum, &key, &(mfrc522.uid));
```

```
//------------------------------------------------------------------------s
if (status != MFRC522::STATUS_OK){
   Serial.print("Authentication failed for Read: ");
   Serial.println(mfrc522.GetStatusCodeName(status));
   return;
}
//------------------------------------------------------------------------
else {
  Serial.println("Authentication success");
}
//------------------------------------------------------------------------
/* Reading data from the Block */
status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
if (status != MFRC522::STATUS_OK) {
  Serial.print("Reading failed: ");
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
//------------------------------------------------------------------------
else {
  readBlockData[16] = ' ';
  readBlockData[17] = ' ';
  Serial.println("Block was read successfully");
```

```
  }
  //-----------------------------------------------------------------------------
}
```

PROGRAM FOR GOOGLE SHEET SCRIPT

```
var ss = SpreadsheetApp.openById('1OxRmZFp5vvE00WrcwsoGP-
zVVKBAnR9iMtm-l7vL8-0');

var sheet = ss.getSheetByName('Sheet1');

var timezone = "asia/calcutta"

var hours = 5.30

function doGet(e){

  Logger.log( JSON.stringify(e) );

  //----------------------------------------------------------------------------------

  //write_google_sheet() function in esp32 sketch, is send data to this code block

  //----------------------------------------------------------------------------------

  //get gps data from ESP32

  if (e.parameter == 'undefined') {

    return ContentService.createTextOutput("Received data is undefined");

  }

  //----------------------------------------------------------------------------------

  var Curr_Date =new Date(new Date().setHours(new Date().getHours() + hours));

  var Curr_Time     = Utilities.formatDate(Curr_Date, timezone, 'HH:mm:ss');

  var student_id    = stripQuotes(e.parameters.data0);

  var first_name    = stripQuotes(e.parameters.data1);

  var last_name     = stripQuotes(e.parameters.data2);

  var father_name   = stripQuotes(e.parameters.data3);

  var date_of_birth = stripQuotes(e.parameters.data4);
```

```
  var phone_number  = stripQuotes(e.parameters.data5);
  //Logger.log('name=' + student_id);
  //----------------------------------------------------------------------------------
  var nextRow = sheet.getLastRow() + 1;
  sheet.getRange("A" + nextRow).setValue(Curr_Date);
  sheet.getRange("B" + nextRow).setValue(Curr_Time);
  sheet.getRange("C" + nextRow).setValue(student_id);
  sheet.getRange("D" + nextRow).setValue(first_name);
  sheet.getRange("E" + nextRow).setValue(last_name);
  sheet.getRange("F" + nextRow).setValue(father_name);
  sheet.getRange("G" + nextRow).setValue(date_of_birth);
  sheet.getRange("H" + nextRow).setValue(phone_number);
  //----------------------------------------------------------------------------------
  //returns response back to ESP32
  return ContentService.createTextOutput("Data is stored in Google Sheet.");
  //----------------------------------------------------------------------------------
}
function stripQuotes( value ) {
  return value.toString().replace(/^["']|['"]$/g, "");
}
//Extra Function. Not used in this project.
//planning to use in future projects.
//this function is used to handle POST request
```

```
function doPost(e) {
  var val = e.parameter.value;
  if (e.parameter.value !== undefined){
    var range = sheet.getRange('A2');
    range.setValue(val);
  }
}
```

CHAPTER 8 – REFERENCES

[1]. De Sousa A. Disulfiram ethanol reaction in a patient abstinent from alcohol caused by hand sanitizing. *Alcohol Alcohol.* 2020;55:349. [PubMed] [Google Scholar]

[2]. V. Verhoeven, G. Tsakitzidis, H. Philips and P. Van Royen, "Impact of the COVID-19 pandemic on the core functions of primary care: Will the cure be worse than the disease? A qualitative interview study in Flemish GPs", *BMJ Open*, vol. 10, no. 6, Jun. 2020.

[3]. T. P. B. Thu, P. N. H. Ngoc, N. M. Hai and L. A. Tuan, "Effect of the social distancing measures on the spread of COVID-19 in 10 highly infected countries", *Sci. Total Environ.*, vol. 742, Nov. 2020.

[4]. J.-P. Bonardi, Q. Gallea, D. Kalanoski and R. Lalive, "Fast and local: How did lockdown policies affect the spread and severity of the COVID-19", *Covid Econ.*, vol. 23, pp. 325-351, Jun. 2020.

[5]. N. Askitas, K. Tatsiramos and B. Verheyden, "Lockdown strategies mobility patterns and COVID-19", *arXiv:2006.00531*, 2020.

[6]. Brewer C, Streel E. Is alcohol in hand sanitizers absorbed through the skin or lungs? Implications for disulfiram treatment. *Alcohol Alcohol.* 2020;55:354–356. [PMC free article] [PubMed] [Google Scholar]

[7]. R. A. Fahey and A. Hino, "COVID-19 digital privacy and the social limits on data-focused public health responses", *Int. J. Inf. Manage.*, vol. 55, Dec. 2020.