CHAPTER-1
INTRODUCTION

1.1 OVERVIEW

The corona virus disease is a major problem in the future world. Presently there is no medicine or vaccine found in the present world. As there is a severe attack in this world, the people are suffering from the corona disease. The corona disease is not a simple virus attack, it makes severe to the human by infecting the respiratory system. The virus disease is heavily spreading in the world, as the nations are trying to monitor and maintain the spread of corona in the nation and other nations. The world is suffering a lot due to this corona virus.

There is a strict evaluation everywhere to control the corona disease and spread to the nation. The hospital and the nurse people are suffering to cure the affected people and stop spreading the virus to the neighboring people. The mouth spread is controlled with the mask cloth and the human hand is controlled by the hand wash sanitizer. The hand touch while pressing the dispenser usage also spreads from human to human. There should be a system for the places to provide access to specific people and an automatic hand sanitizer dispenser, to control and maintain the spread from human to human.

1.2 OBJECTIVE OF THE PROJECT

The main objective of this project is to provide safe entrances to people into places and make sure no one is affected by this virus. To make sure that everyone is sanitized and a proper precautionary method is implemented for everyone to be safe in this pandemic period since we have seen the rapid increase of cases due to the fast spread of the virus.

- Decreases person to person contact as much as possible.
- Helps sanitization of hands to keep ourselves safe and clean.
- Access to only limited and safe people to enter the space.
- Decreases labor by making everything automated.

1.3 PROPOSED SYSTEM

The proposed system is designed to control and monitor the access system and the sanitization process automatically by the predefined specifications given during coding in Arduino. Here in this project we can automatically control by setting the conditions of the sensors and the condition controlling will be done by Arduino by using the specified conditions in the code.

The Arduino works as a micro controller which is interfaced with various sensors and components and can be used to configures as required by the user, the model used in the current project is the Arduino UNO which is the most compatible board for our requirement

Here we are using an RFID tag and an RFID reader for providing the access into the space which is controlled by the Arduino based on the conditions. When the person tries to enter into the space the RFID scans the card and then the access is given to the person whose tag reads and negative, if the card reads it as positive the buzzer buzzes.

Two Ultrasonic sensors which are interfaced with the Servo motors are used to dispense the hand sanitizer and the tissue papers to the user of this prototype.

The current model uses an MLX90614 contactless temperature sensor which checks the temperature through IR waves and if the temperature is beyond 101-degree Fahrenheit then the buzzer buzzes.

It also consists of a Smart Bin which helps in dispensing the waste from the tissue paper or any waste that the person or the user wants to throw. This helps in keeping the place neat and clean.

In addition, we used a robotic arm which is a combination of 10K potentiometers and servo motors which help in delivering the products without touching the products.

1.4 BLOCK DIAGRAM OF MINI PROJECT

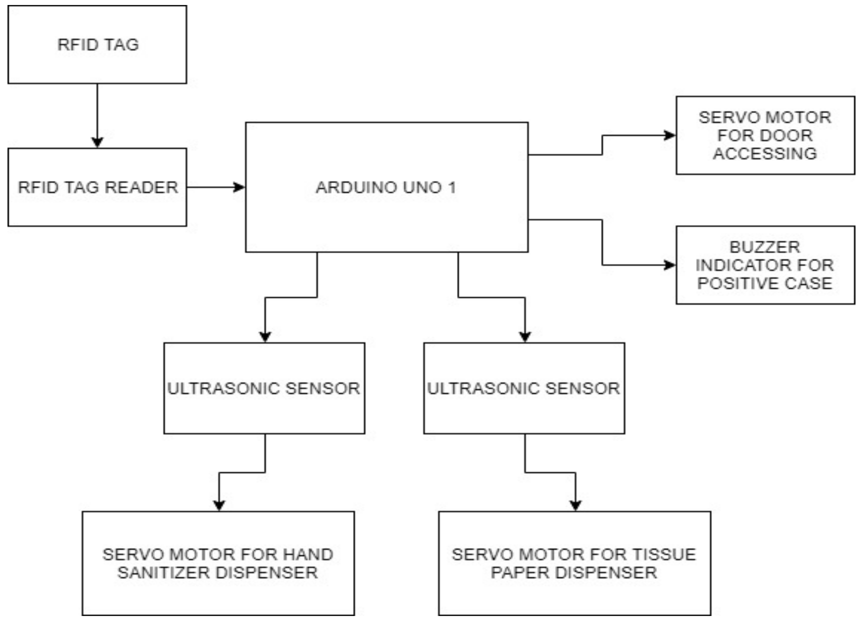

Figure: 1.1 Block Diagram of Mini Project

1.5 BLOCK DIAGRAM OF MAJOR PROJECT

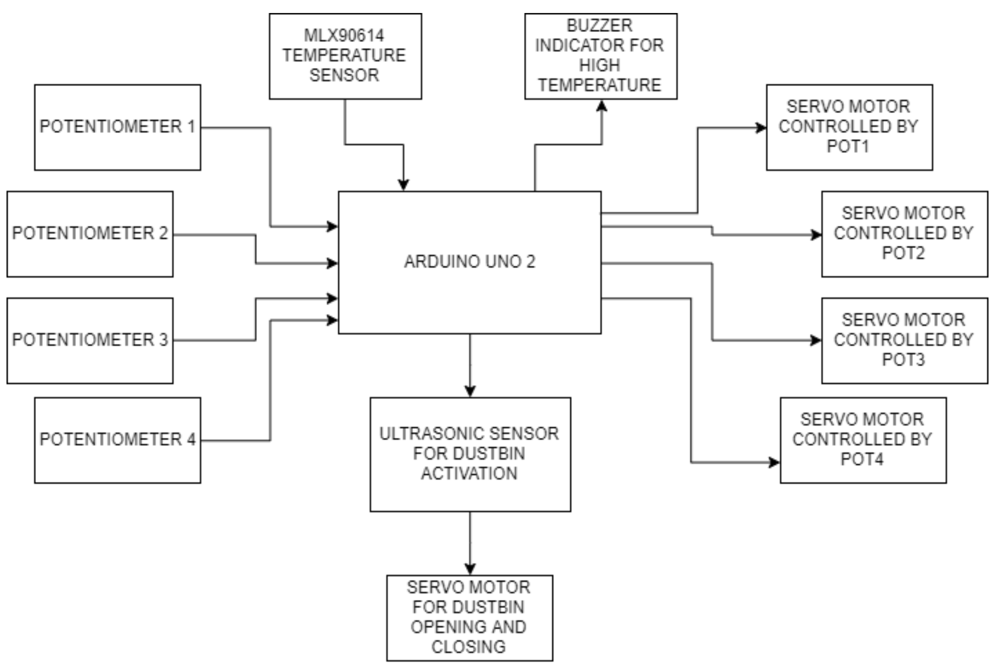

Figure: 1.2 Block Diagram of Major Project

1.6 WORKING

The working is as follows when the person scans the RFID tag the reader reads the tag and the access is given accordingly and a buzzer buzzes when the access is denied, a non-contact temperature sensor is used, when the temperature is above 101 degree Fahrenheit then the buzzer buzzes and then the ultrasonic sensors are used to sense the distance between the hands of the user so as to dispense the hand sanitizer and tissue paper and dustbin as per requirement, a robotic arm is a combination which helps deliver the products.

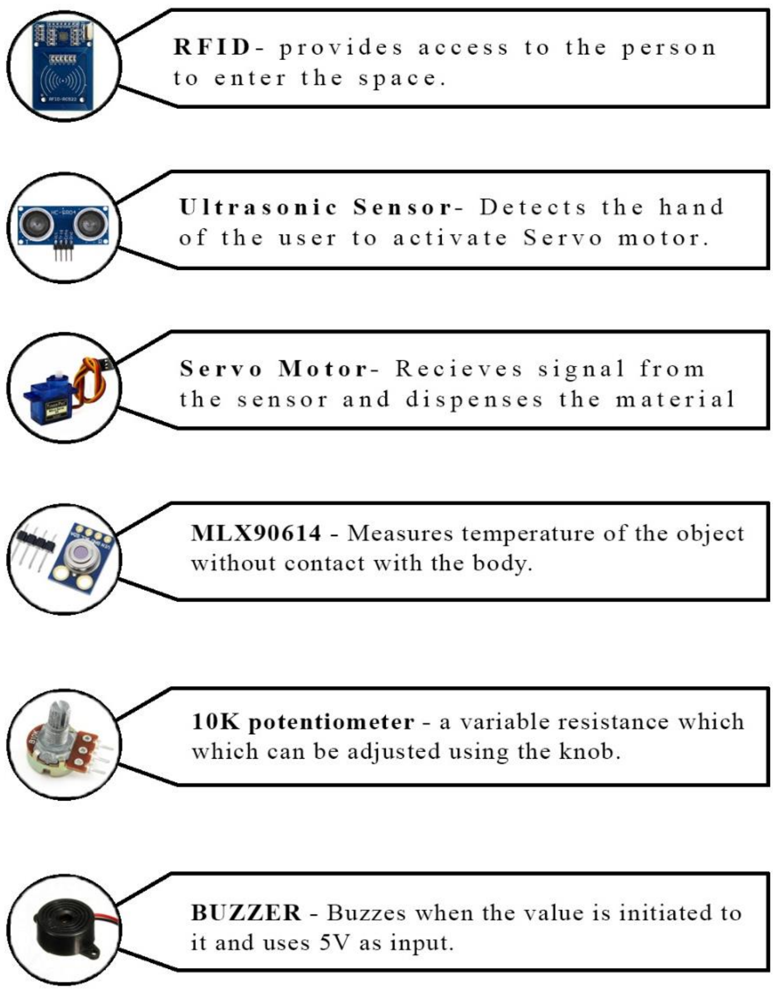

Figure 1.3 Work Flow

1.7 COMPONENTS AND TOOLS USED

1.7.1 HARD WARE TOOLS

- **Arduino Uno (ATmega328) –** Arduino Uno is electronic circuit that can be programmed to carry out a vast of task. They can be programmed to be timers or to control a production line and much more. The Arduino Uno is the most used microcontroller because of its vast compatibility with all the sensors and is known as one of the best microcontrollers out there currently.
- **RC522 RFID Module - Radio-frequency identification** (**RFID**) uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter. It reads the tag using a electromagnetic field and gives access accordingly.
- **HC-SRO4 Ultrasonic Sensor -** An **Ultrasonic sensor** is an electronic device that measures the distance of a target object by emitting Ultrasonic sound waves, and converts the reflected sound into electrical signal. Ultrasonic waves travel faster than speed of audible sound.
- **SG-90 Servo motor -** A servo motor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servo motors.
- **MLX90614 Temperature Sensor -** The sensor is a non-contact temperature sensor which senses the temperature of an object while absorbing the IR waves that are dispensed from the body itself, once the sensor captures the IR waves, it calculates the temperature accordingly.
- **10K Potentiometers –** The 10K potentiometers are the variable resistors which can be increased or decreased with the help of a knob attached to the body of the potentiometer, the maximum value of the potentiometer is 10k and the analog value ranges from 0-1023.
- **Buzzer -** The buzzer is the component which is used at two places in the project, one is the RFID tag reader and the other is the temperature sensor, both of these buzzers are interfaced in such a way that they buzz when the access is denied or the temperature is above the mentioned temperature in the code.

1.8 SOFTWARE TOOLS

Arduino IDE 1.0 is used. ”UNO” means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0/ the Uno board and version 1.0 of Arduino software (IDE) were the reference versions of Arduino , now evolved to newer releases . The Uno board is the first in series of USB Arduino boards, and the reference model for the Arduino platform, for an extensive list of current, past or outdated boards.

- Proteus VSM is a digital simulation and circuit design tool. It provides the plat form for circuit layout, circuit design and real time simulation.
- Arduino software is the tool to burn the .hex file on the chip.

CHAPTER-2

EMBEDDED SYSTEMS

2.1 INTRODUCTION TO EMBEDDED SYSTEMS

Our day-to-day life is becoming more and more dependent on "Embedded Systems and Digital Technologies". Embedded technologies are bonding into our daily activities even without our knowledge. We know the fact that the refrigerator, washing machine, microwave oven, air conditioner, television, DVD players, and music systems that we use in our home are built around an embedded system. It is nothing but an intelligent embedded system. In your vehicle itself the presence of specialized embedded systems vary from intelligent head lamp controllers, engine controllers and ignition control Systems to complex air bag control systems to protect you from a severe accident. People experience the power of embedded systems and enjoy the features and comfort provided by them. Most of us are totally unaware or ignorant of the intelligent embedded systems giving us so much comfort and security. Embedded systems are like reliable servants-they don't like to reveal their identity and neither they complain about their workloads to their owners or bosses. They are always working behind the scenes and are dedicated to their assigned task till their last breath. This book gives you an overview of embedded systems, the various steps involved in their design and development and the major domains where they are deployed.

2.2 FEATURES OF EMBEDDED SYSTEMS

The versatility of the embedded computer system lends itself to utility in all kinds of enterprises, from the simplification of deliverable products to a reduction in costs in their development and manufacture. Usually a low power consumption CPU with a limited amount of memory is used in embedded systems. Complex systems with rich functionality employ special operating systems that take into account major characteristics of embedded systems.

Embedded operating systems have minimized footprint and may follow real-time operating system specifics. The special computers system is usually less powerful than general-purpose systems, although some expectations do exist where embedded systems are very powerful and complicated. Usually a low power consumption CPU with a limited amount of memory is used in embedded systems. Some embedded systems have to operate in extreme environment conditions such as very high temperature & humidity Many embedded systems use very small operating systems; most of these provide very limited operating system capabilities. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance some embedded systems are mass-produced, bene fitting from economies of scale. Some embedded systems have to operate in extreme environment conditions such as very high temperature & humidity. For high volume systems such as portable music players or mobile phones, minimizing cost is usually the primary design consideration. Engineers typically select hardware that is just "good enough" to implement the necessary functions. For low volume or prototype embedded systems, general purpose computers may be adapted by limiting the programs or by replacing the operating system with a real-time operating system.

2.3 CHARACTERISTICS OF EMBEDDED SYSTEMS

Embedded computing systems generally exhibit rich functionality complex functionality is usually the reason for introducing cups into the design. However, they also exhibit many non-functional requirements that make the task especially challenging:

- Real-time deadlines that will cause system failure if not met.
- Multi-rate operation.
- In many cases, low power consumption.
- Low manufacturing cost, which often means limited code size.

Workstation programmers often concentrate on functionality. They may consider the performance characteristics of a few computational kernels of their software, but rarely analyze the total application. They almost never consider power consumption and manufacturing cost. The need to juggle all these requirements makes embedded system programming very challenging and is the reason why embedded system designers need to understand computer architecture.

2.4 EMBEDDED SYSTEM ARCHITECTURE

Every Embedded system consists of a custom-built hardware built around a central processing unit. This hardware also contains memory chips onto which the software is loaded.

The operating system runs above the hardware and the application software runs above the operating system. The same architecture is applicable to any computer including desktop computer. However these are significant differences. It is not compulsory to have an operating system in every embedded system. For small applications such as remote control units, air conditioners, toys etc.

The operating system runs above the hardware and the application software runs above the operating system. The same architecture is applicable to any computer including desktop computer. However these are significant differences. It is not compulsory to have an operating system in every embedded system. For small applications such as remote control units, air conditioners, toys etc.

2.5 TYPES OF EMBEDDED SYSTEM

Based on functionality and performance embedded systems categorized as 4 types

1. Stand-alone embedded systems
2. Real time embedded systems
3. Networked information appliances
4. Mobile devices

1. Standalone embedded systems:

As the name implies, stand-alone systems work in standalone mode. They take i/p. Process them and produce the desire o/p. The i/p can be an electrical signal from transducer or temperature signal or commands from human being. The o/p can be electrical signal to drive another system an led or LCD display

Ex digital camera, microwave oven, CD player, air conditioner etc.

2. Real time embedded systems:

In this type of an embedded system a specific work has to be complete in a particular period of time

Hard real time systems:- embedded real time used in missiles.

Soft real time systems:- DVD players

3. Networked information appliances:

Embedded systems that are provided with n/w interfaces and accessed by n/w such as local area n/w or internet are called network information appliances.

EX. a web camera is connected to the internet. Camera can send pictures in real time to any computers connected to the internet.

4. Mobile devices:

Actually it is a combination of both VLSI and Embedded system Mobile devices such as mobile phone, personal digital assistants, smart phones etc are special category of embedded system.

Embedded systems are based on the concept of the microcontroller, a single integrated circuit that contains all the technology required to run an application.

- Central processing Unit.
- Input/output interfaces (such as serial ports)
- Peripherals (such as timers)
- ROM ,EEPROM (or) Flash memory for program storage
- RAM for data storage Clock generator

By integrating all of these features into a single chip it is possible to greatly reduce the number of chips and wiring necessary to control an electronic device, dramatically reducing its complexity, size and cost.

- **Size & Weight**: Microcontrollers are designed to deliver maximum performance for Minimum size and weight. A centralized on-board computer system would greatly Outweigh a collection of microcontrollers.
- **Efficiency**: Microcontrollers are designed to perform repeated functions for long periods of time without failing or requiring service.
- **Micro Controller**: It is a chip through which we can connect many other devices and also those are controlled by the program the program which burn into it. Here we are using Arduino as the main device to control.

CHAPTER-3

HARDWARE MODULES

3.1 ARDUINO UNO (ATMEGA 328)

3.1.1 Feature:

- High performance, low power AVR 8bit micro controller
- Advanced RISC architecture
  - 331 power full instructions –most single clock cycle execution
  - 32×8 general purpose working registers
  - Fully static operation
  - Up to 20 MIPS throughput at 20MHz
  - On-chip 2-cycle multiplier
- Special micro controller features
  - Power- on reset and programmable brown-out detection
  - Internal calibrated oscillator
  - External and internal interrupt sources
  - 6 sleep modes: Idle ,ADC noise reduction, power save , power down ,
- I/O and packages
  - 23 programmable I/O Lines
  - 28- pin PDIP, 32-lead TQFP , 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating voltages
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 105°C
- Speed Grade:
  - 0 - 4MHz @ 1.8 - 5.5V
  - 0 - 10MHz @ 2.7 - 5.5V
  - 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
  - Active Mode: 0.2mA

- Power-down Mode: 0.1µA
- Power-save Mode: 0.75µA (Including 32 kHz RTC)

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, and Max MSP.) The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free. The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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 a AC-to-DC adapter or battery to get started.

3.2 RC522 RFID Module

- **Introduction**

This RFID module is categorized into two parts, an RFID Tag and an RFID Reader these both together work to form a full RFID system, we will be discussing more about the Tags and the Reader further into the report.

- **RFID Tag**

RFID tags are made out of three pieces: a microchip (an integrated circuit which stores and processes information and modulates and demodulates radio-frequency (RF) signals), an antenna for receiving and transmitting the signal and a substrate.

RFID tags can be either passive, active or battery-assisted passive. An active tag has an on-board battery and periodically transmits its ID signal. A battery-assisted passive tag has a small battery on board and is activated when in the presence of an RFID reader. A passive tag is cheaper and smaller because it has no battery; instead, the tag uses the radio energy transmitted by the reader.

The RFID tag receives the message and then responds with its identification and other information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information. Since tags have individual serial numbers, the RFID system design can discriminate among several tags that might be within the range of the RFID reader and read them simultaneously.

Figure 3.1 RFID Tag Internal Structure

- **RFID Reader**

  An RFID reader is the brain of the RFID system and is necessary for any system to function. Readers**,** also called interrogators, are devices that transmit and receive radio waves in order to communicate with RFID tags. RFID readers are typically divided into two distinct types – Fixed RFID Readers and Mobile RFID Readers. Fixed readers stay in one location and are typically mounted on walls, on desks, into portals, or other stationary locations.

A common subset of fixed readers is integrated readers. An integrated RFID reader is a reader with a built-in antenna that typically includes one additional antenna port for the connection of an optional external antenna as well. Integrated readers are usually aesthetically pleasing and designed to be used for indoor applications without a high traffic of tagged items.

Mobile readers are handheld devices that allow for flexibility when reading RFID tags while still being able to communicate with a host computer or smart device. There are two primary categories of Mobile RFID readers – readers with an onboard computer, called Mobile Computing Devices, and readers that use a Bluetooth or Auxiliary connection to a smart device or tablet, called Sleds.

Fixed RFID Readers typically have external antenna ports that can connect anywhere from one additional antenna to up to eight different antennas. With the addition of a multiplexer, some readers can connect to up to 32 RFID antennas. The number of antennas connected to one reader depends on the area of coverage required for the RFID application. Some desktop applications, like checking files in and out, only need a small area of coverage, so one antenna works well. Other applications with a larger area of coverage, such as a finish line in a race timing application typically require multiple antennas to create the necessary coverage zone.

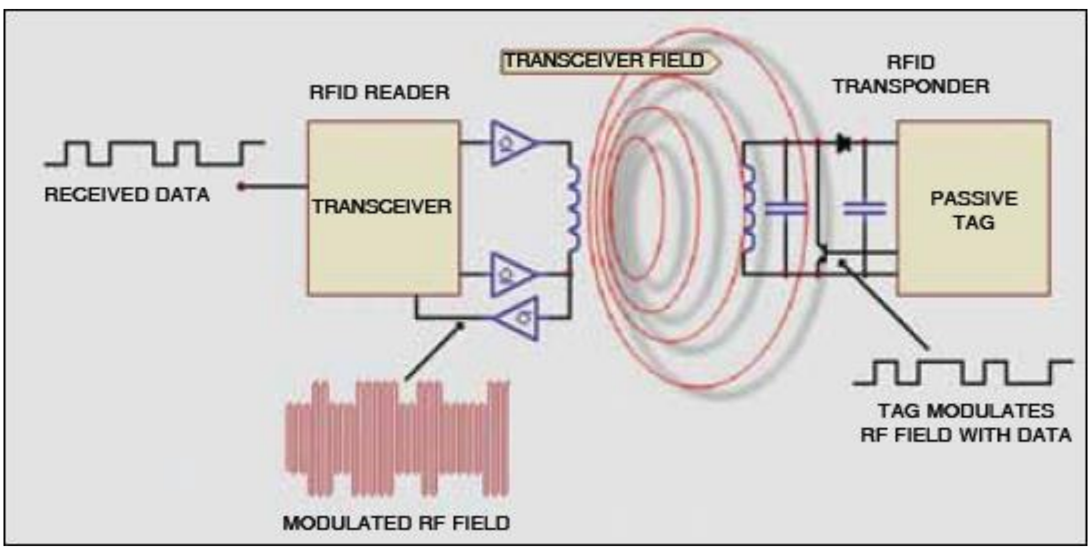

Figure 3.2 RFID Reader Internal Structure

- **Specifications**

Table no: 3.1 RC522 RFID specifications

5 Centimeters	Read Range
10Mbps	Max Data Rate
13.56MHz	Operating Frequency
40mm* 60mm	Body Size
2.5 – 3.3 V	Operating Voltage
13-26mA	Max Current During Measuring

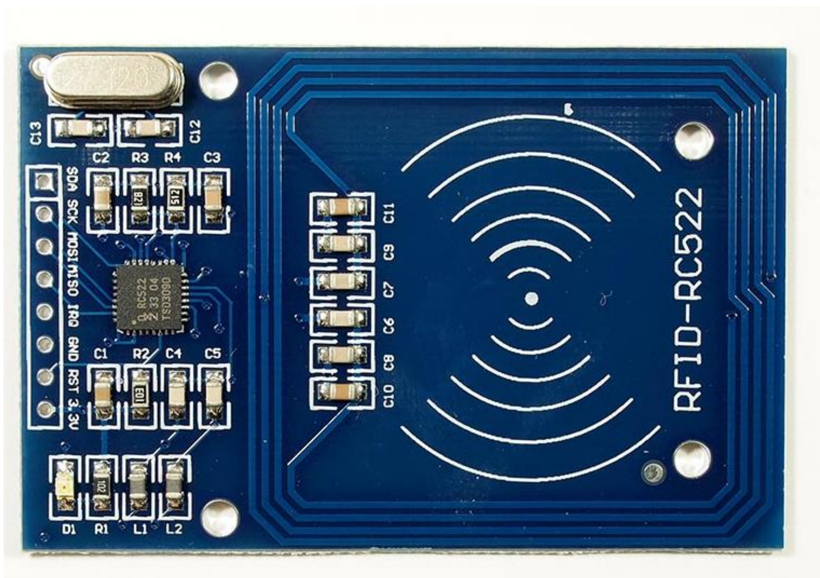

Figure: 3.3 RC522 RFID Module

- **Working Principle**

The RC522 RFID Reader module is designed to create a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags (ISO 14443A standard tags). The reader can communicate with a microcontroller over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. It also supports communication over I2C and UART protocols.

The module comes with an interrupt pin. It is handy because instead of constantly asking the RFID module, the module will alert us when a tag comes into its vicinity.

The operating voltage of the module is 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 any logic level converter.

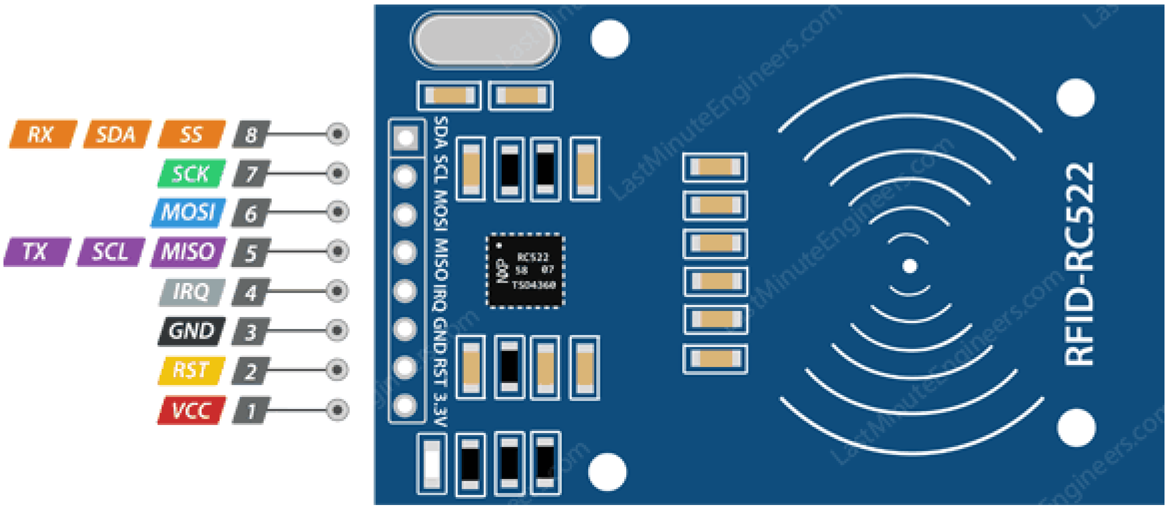

Figure: 3.4 RFID Pin Description

VCC supplies power for the module. This can be anywhere from 2.5 to 3.3 volts. You can connect it to 3.3V output from your Arduino. Remember connecting it to 5V pin will likely destroy your module.

RST is an input for Reset and power-down. When this pin goes low, hard power-down is enabled. This turns off all internal current sinks including the oscillator and the input pins are disconnected from the outside world. On the rising edge, the module is reset.

GND is the Ground Pin and needs to be connected to GND pin on the Arduino.

IRQ is an interrupt pin that can alert the microcontroller when RFID tag comes into its vicinity.

MISO/ SCL/ Tx pin acts as Master-In-Slave-Out when SPI interface is enabled, acts as serial clock when I2C interface is enabled and acts as serial data output when UART interface is enabled.

MISO (Master Out Slave In) is SPI input to the RC522 module.

SCK (Serial Clock) accepts clock pulses provided by the SPI bus Master i.e. Arduino.

SS/ SDA/ Rx pin acts as Signal input when SPI interface is enabled, acts as serial data when I2C interface is enabled and acts as serial data input when UART interface is enabled.

3.3 HC-SRO4 Ultrasonic Sensor

- **Introduction**

Ultrasonic transducers and ultrasonic sensors are devices that generate or sense ultrasound energy. They can be divided into three broad categories: transmitters, receivers and transceivers. Transmitters convert electrical signals into ultrasound, receivers convert ultrasound into electrical signals, and transceivers can both transmit and receive ultrasound.

In a similar way to radar and sonar, ultrasonic transducers are used in systems which evaluate targets by interpreting the reflected signals. For example, by measuring the time between sending a signal and receiving an echo the distance of an object can be calculated.

The design of transducer can vary greatly depending on its use: those used for medical diagnostic purposes, for example the range-finding applications listed above, are generally lower power than those used for the purpose of changing the properties of the liquid medium, or targets immersed in the liquid medium, through chemical, biological or physical (e.g. erosive) effects. The latter class include ultrasonic probes and ultrasonic baths, which apply ultrasonic energy to agitate particles, clean, erode, or disrupt biological cells, in a wider range of materials. Typically, a microcontroller is used for communication with an ultrasonic sensor.

Figure 3.5 HC-SR04 Ultrasonic Sensor

When the system is initiated, the signal is sent to the Ultrasonic sensor and the response is as follows. The duty cycle of this trigger signal is 10µS for the HC-SR04 ultrasonic sensor. When triggered, the ultrasonic sensor generates eight acoustic (ultrasonic) wave bursts and initiates a time counter. As soon as the reflected (echo) signal is received, the timer stops. The output of the ultrasonic sensor is a high pulse with the same duration as the time difference between transmitted ultrasonic bursts and the received echo signal.

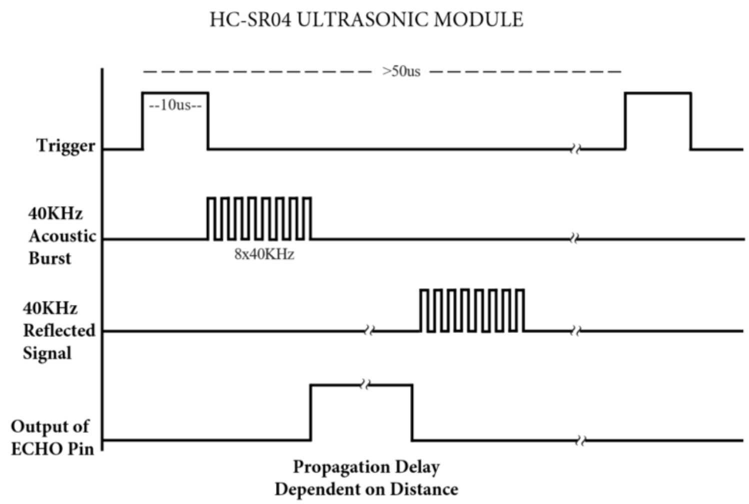

Figure 3.6 Representation of HC-SR04 Response

The microcontroller interprets the time signal into distance using the following functions:

$$Distance\ (cm) = \frac{echo\ pulse\ width\ (uS)}{58}$$

$$Distance\ (inch) = \frac{echo\ pulse\ width\ (uS)}{148}$$

Theoretically, the distance can be calculated using the TRD (time/rate/distance) measurement formula. Since the calculated distance is the distance traveled from the ultrasonic transducer to the object and back to the transducer it is a two-way trip. By dividing this distance by 2, you can determine the actual distance from the transducer to the object. Ultrasonic waves travel at the speed of sound (343 m/s at 20°C). The distance between the object and the sensor is half of the distance traveled by the sound wave. The following equation calculates the distance to an object placed in front of an ultrasonic sensor:

$$distance = \frac{time\ taken\ x\ speed\ of\ sound}{2}$$

- **Specifications**

Table no: 3.2 HC-SR04 Ultrasonic Sensor Specifications

+5V	Operating Voltage
2cm to 450cm	Theoretical Measuring Distance
2cm to 80cm	Practical Measuring Distance
3mm	Accuracy
<15mA	Operating Current
40Hz	Operating frequency

- **Working principle**

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below.

Figure 3.7 Working of HC-SR04 Ultrasonic Sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return.

Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading.

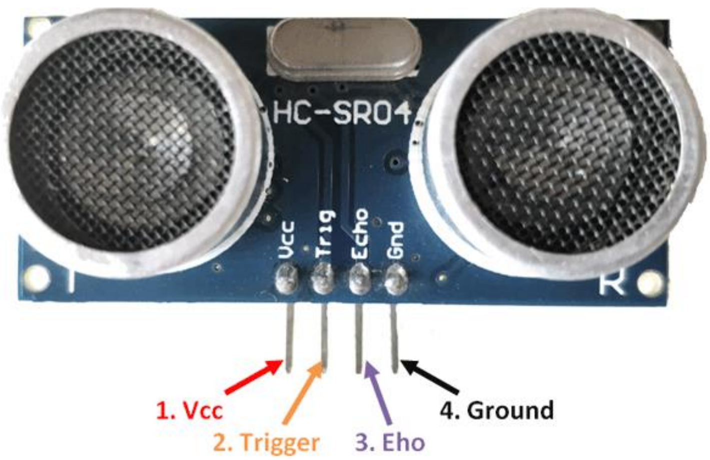

Figure 3.8 Pin diagram of HC-SR04 Ultrasonic Sensor

VCC the Vcc pin powers the sensor, typically with +5V

Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.

Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.

Ground this pin is connected to the Ground of the system.

The major use of the Ultrasonic sensor is in automobile self-parking technology and anti-collision safety systems.

3.4 SG-90 Servo Motor

A **servomotor** is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

A servomotor is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The motor is paired with some type of position encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller. If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped). This type of servomotor is not widely used in industrial motion control, but it forms the basis of the simple and cheap servos used for radio-controlled models.

More sophisticated servomotors use optical rotary encoders to measure the speed of the output shaft and a variable-speed drive to control the motor speed. Both of these enhancements, usually in combination with a PID control algorithm, allow the servomotor to be brought to its commanded position more quickly and more precisely, with less overshooting.

There are lots of servo motors available in the market and each one has its own specialty and applications. The following two paragraphs will help you identify the right type of servo motor for your project/system.

Most of the hobby Servo motors operates from 4.8V to 6.5V, the higher the voltage higher the torque we can achieve, but most commonly they are operated at +5V. Almost all hobby servo motors can rotate only from 0° to 180° due to their gear arrangement.

Next comes the most important parameter, which is the **torque** at which the motor operates. Again there are many choices here but the commonly available one is the 2.5kg/cm torque which comes with the Towerpro SG90 Motor. This 2.5kg/cm torque means that the motor can pull a weight of 2.5kg when it is suspended at a distance of 1cm. So if you suspend the load at 0.5cm then the motor can pull a load of 5kg similarly if you suspend the load at 2cm then can pull only 1.25. Based on the load which you use in the project you can select the motor with proper torque. The below picture will illustrate the same.

Figure 3.9 SG-90 Servo Motor

As we know there are three wires coming out of this motor. The description of the same is given on top of this page. To make this motor rotate, we have to power the motor with +5V using the Red and Brown wire and send PWM signals to the Orange colour wire. Hence we need something that could generate PWM signals to make this motor work, this something could be anything like a 555 Timer or other Microcontroller platforms like Arduino, PIC, ARM or even a microprocessor like Raspberry Pie.

They are small in size but pack a big punch and are very energy-efficient. These features allow them to be used to operate remote-controlled or radio-controlled toy cars, robots and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing, pharmaceutics and food services.

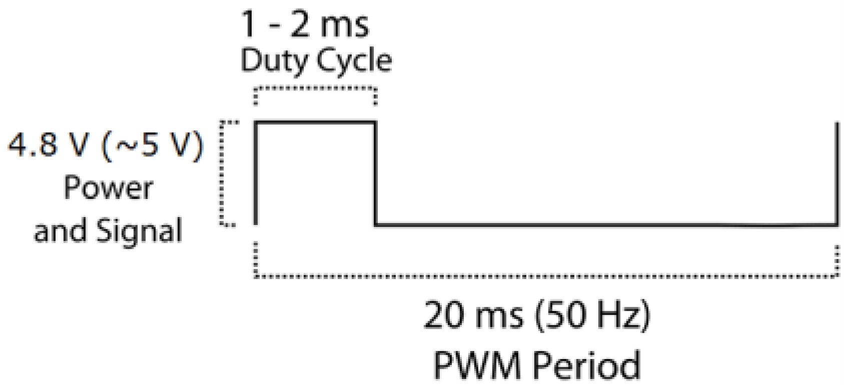

Figure 3.10 Response of SG-90 Servo Motor

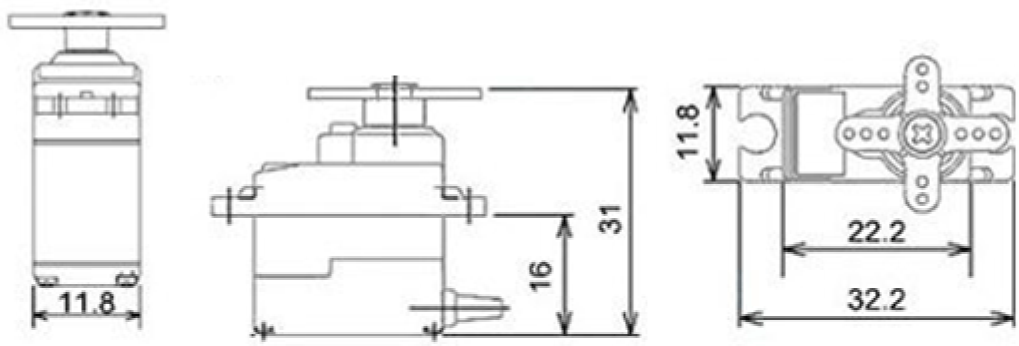

Figure 3.11 Dimensions of SG-90 Servo Motor

- **Specifications**

Table no: 3.3 SG-90 Servo Motor specifications

+5V	Operating Voltage
2.5kg/cm	Torque
0.1s/60°	Operating Speed
Plastic	Gear Type
0°-180°	Rotation
9gm	Weight

3.5 MLX90614 Temperature sensor

The MLX90614 is a **Contactless Infrared (IR) Digital Temperature Sensor** that can be used to measure the temperature of a particular object ranging from -70° C to 382.2°C. The sensor uses IR rays to measure the temperature of the object without any physical contact and communicates to the microcontroller using the I2C protocol.

The key feature of MLX90614 is that it is a contactless IR temperature sensor with high accuracy. So, it can be used in industries to **measure the temperature of moving objects** like a rotating motor shaft. Due to its high accuracy and precision, it is also used in a wide range of **commercial, health care, and household applications** like room temperature monitoring, body temperature measurement, etc.

Figure 3.12 MLX90614 Temperature Sensor

As mentioned earlier, the MLX90614 sensor can measure the temperature of an object without any physical contact with it. This is made possible with a law called **Stefan-Boltzmann Law**, which states that all objects and living beings emit IR Energy and the intensity of this emitted IR energy will be directly proportional to the temperature of that object or living being. So the MLX90614 sensor calculates the temperature of an object by measuring the amount of IR energy emitted from it.

The MLX90614 Temperature sensor is manufactured by a company called **Melexis**. The sensor is factory calibrated and hence it acts like a **plug and play sensor module** for speeding up development processes. The MLX90614 consists of two devices embedded as a single sensor, one device acts as a sensing unit and the other device acts as a processing unit.

The sensing unit an **Infrared Thermopile Detector** called **MLX81101** which senses the temperature and the processing unit is a **Single Conditioning ASSP** called MLX90302 which converts the signal from the sensor to digital value and communicates using I2C protocol. The MLX90302 has a low noise amplifier, 17-bit ADC and a powerful DSP which helps the sensor to have high accuracy and resolution.

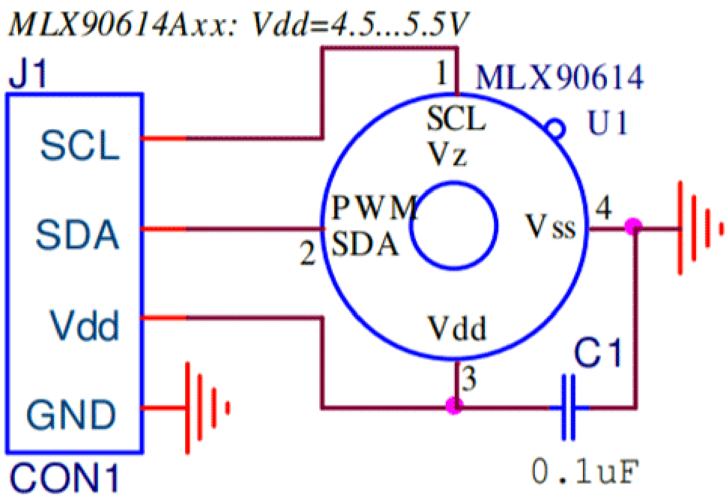

Figure 3.13 Internal circuit of MLX90614 Sensor

The sensor requires no external components and can be directly interfaced with a microcontroller like Arduino. As you can see above the power pins (Vdd and Gnd) can be directly used to power the sensor, typically 5V can be used, but there are other versions of this sensor which can operate on 3.3V and 7V as well. The capacitor C1 is optional and is used to filter noise and provide optimum EMC. The signal pins (SCL and SDA) for used for I2C communication and can be connected directly to microcontroller operating on 5V logic.

Every MLX90614 has a default **I^{2}C address** of **0x5A**, but that address can be re-written one of the major features supported by the device. By reconfiguring the address of an MLX90614, you can add multiple devices (up to 127!) to the same bus to get a larger temperature map. The MLX90614's data can also be read via a PWM interface. In this use case just one wire is required to read from the sensor: SDA. To use the PWM interface, the

MLX90614 has to be configured over the SMBus to produce it. By configuring the sensor's range -- setting minimum and/or maximum temperature values -- the PWM output can be turned into a "thermal relay" signal.

The MLX90614 is available in the standard TO-39 package making it easy to mount on a breadboard. Do note that this is a 4-pin TO39 package and hence the footprint will differ. the dimensions of MLX90614 is shown below.

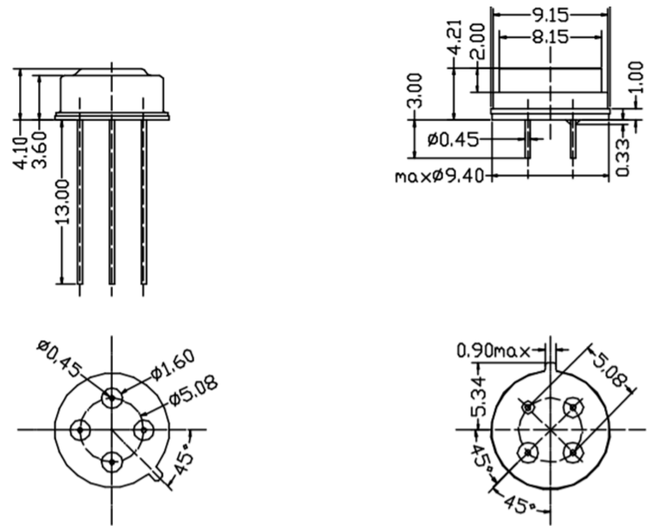

Figure 3.14 Dimensions of MLX90614 Sensor

- **Specifications**

Table no: 3.4 MLX90614 temperature sensor specifications

+5V	Operating Voltage
-70°C to 382°C	Object Temperature
-40°C to 125°C	Ambient Temperature
0.02°C	Accuracy
80°	Field of View
2cm-5cm	Distance Between Sensor and Object

3.6 10K Potentiometer

A **potentiometer** is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a **variable resistor** or **rheostat**.

The measuring instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage) the component is an implementation of the same principle, hence its name.

Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

Figure 3.15 10K Potentiometer

Potentiometers consist of a resistive element, a sliding contact (wiper) that moves along the element, making good electrical contact with one part of it, electrical terminals at each end of the element, a mechanism that moves the wiper from one end to the other, and a housing containing the element and wiper.

Many inexpensive potentiometers are constructed with a resistive element formed into an arc of a circle usually a little less than a full turn and a wiper sliding on this element when rotated, making electrical contact. The resistive element can be flat or angled. Each end of the resistive element is connected to a terminal on the case. The wiper is connected to a third terminal, usually between the other two. On panel potentiometers, the wiper is usually the center terminal of three. For single-turn potentiometers, this wiper typically travels just under one revolution around the contact. The only point of ingress for contamination is the narrow space between the shaft and the housing it rotates in.

The resistive element of inexpensive potentiometers is often made of graphite. Other materials used include resistance wire, carbon particles in plastic, and a ceramic/metal mixture called cermet. Conductive track potentiometers use conductive polymer resistor pastes that contain hard-wearing resins and polymers, solvents, and lubricant, in addition to the carbon that provides the conductive properties.

Potentiometers are rarely used to directly control significant amounts of power (more than a watt or so). Instead, they are used to adjust the level of analog signals (for example volume controls audio equipment), and as control inputs for electronic circuits. For example, a light dimmer uses a potentiometer to control the switching of a TRIAC and so indirectly to control the brightness of lamps.

Potentiometers also known as POT, are nothing but variable resistors. They can provide a variable resistance by simply varying the knob on top of its head. It can be classified based on two main parameters. One is their **Resistance (R-ohms)** itself and the other is its **Power (P-Watts)** rating.

The value or resistance decides how much opposition it provides to the flow of current. The greater the resistor value the smaller the current will flow. Some standard values for a potentiometer are 500Ω, 1K, 2K, 5K, 10K, 22K, 47K, 50K, 100K, 220K, 470K, 500K, 1 M. Resistors are also classified based on how much current it can allow; this is called Power (wattage) rating. The higher the power rating the bigger the resistor gets and it can also more current. For potentiometers the power rating is 0.3W and hence can be used only for low current circuits.

Such resistors need to be selected based on the requirement of the user and the application that needs its implementation.

- **Working principle**

As far as we know resistors should always have two terminals but, why a **potentiometer** has three terminals and how to we use these terminals. It is very easy to understand the purpose of these terminals by looking at the diagram below.

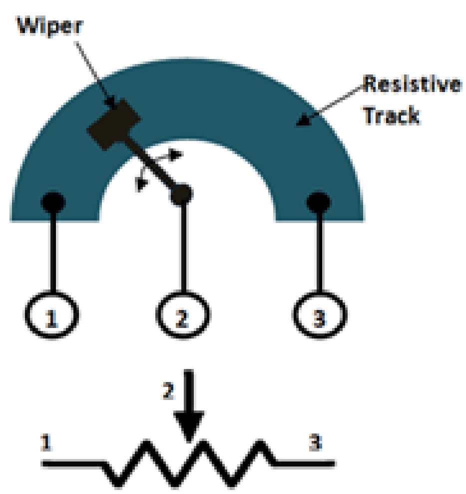

Figure 3.16 10K Potentiometer Internal Working

The diagram shows the parts present inside a potentiometer. We have a resistive track whose complete resistance will be equal to the rated resistance value of the POT.

As the symbol suggests a potentiometer is nothing but a resistor with one variable end. Let us assume a 10k potentiometer, here if we measure the resistance between terminal 1 and terminal 3 we will get a value of 10k because both the terminals are fixed ends of the potentiometer. Now, let us place the wiper exactly at 25% from terminal 1 as shown above and if we measure the resistance between 1 and 2 we will get 25% of 10k which is 2.5K and measuring across terminal 2 and 3 will give a resistance of 7.5K.

So, the terminals 1 and 2 or terminals 2 and 3 can be used to obtain the variable resistance and the knob can be used to vary the resistance and set the required value.

User-actuated potentiometers are widely used as user controls, and may control a very wide variety of equipment functions. The widespread use of potentiometers in consumer electronics declined in the 1990s, with rotary incremental encoders, up/down push-buttons, and other digital controls now more common. However, they remain in many applications, such as volume controls and as position sensors. Most of them are made in various sizes and shapes.

It can be used for the applications of voltage and current control circuits, used as the volume control knobs in radios, tuning or controlling circuits and the analog input control knobs and the potentiometers can also be used to control other sensors such as servo motors, there are various usages and applications for the potentiometers, and the dimensions of the resistor are shown in the figure below.

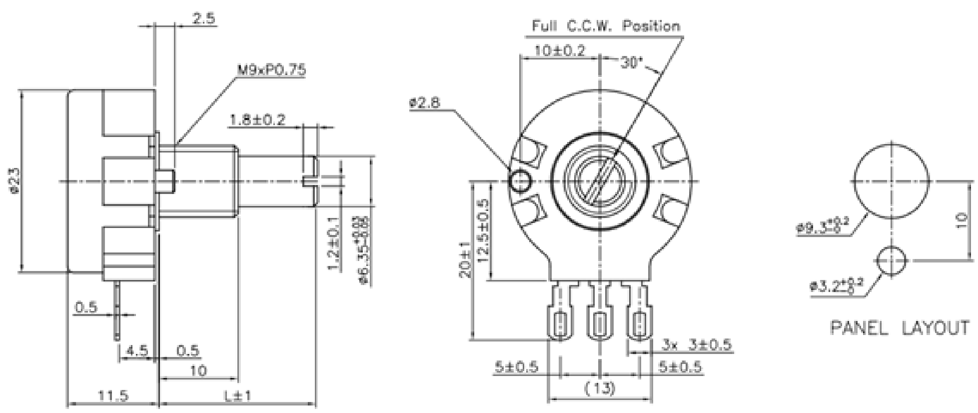

Figure 3.17 10K Potentiometer Dimensions

- **Specifications**

Table no: 3.5 10K Potentiometer specifications

Rotary	Type
500Ω to 1M	Available in Values
0.3W	Power Rating
200Vdc	Maximum Input Voltage
2000K cycles	Rotational Life
Linear	Taper

3.7 Buzzer

Piezoelectric buzzers, or piezo buzzers, as they are sometimes called, were invented by Japanese manufacturers and fitted into a wide array of products during the 1970s to 1980s. This advancement mainly came about because of cooperative efforts by Japanese manufacturing companies. In 1951, they established the Barium Titanate Application Research Committee, which allowed the companies to be "competitively cooperative" and bring about several piezoelectric innovations and inventions. A **buzzer** or **beeper** is an audio signaling device which maybe mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

Figure 3.18 Active Passive Buzzer

A **buzzer** is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard, Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beep sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep.

Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customized with help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at required time and require interval.

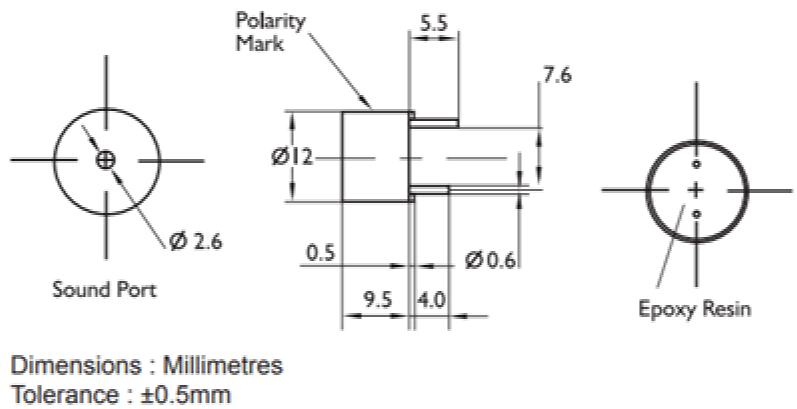

Figure 3.19 Buzzer Dimensions

- **Specifications**

Table no: 3.6 Active Passive Buzzer specifications

4-8V	Voltage
<30mA	Current
Continuous Beep	Sound Type
~2300Hz	Resonant Frequency
Positive and Negative	Terminals
Plastic Casing	Body Type

CHAPTER-4

SCHEMATIC DIAGRAM AND ITS WORKING

4.1 CIRCUIT DIAGRAM OF MINI PROJECT

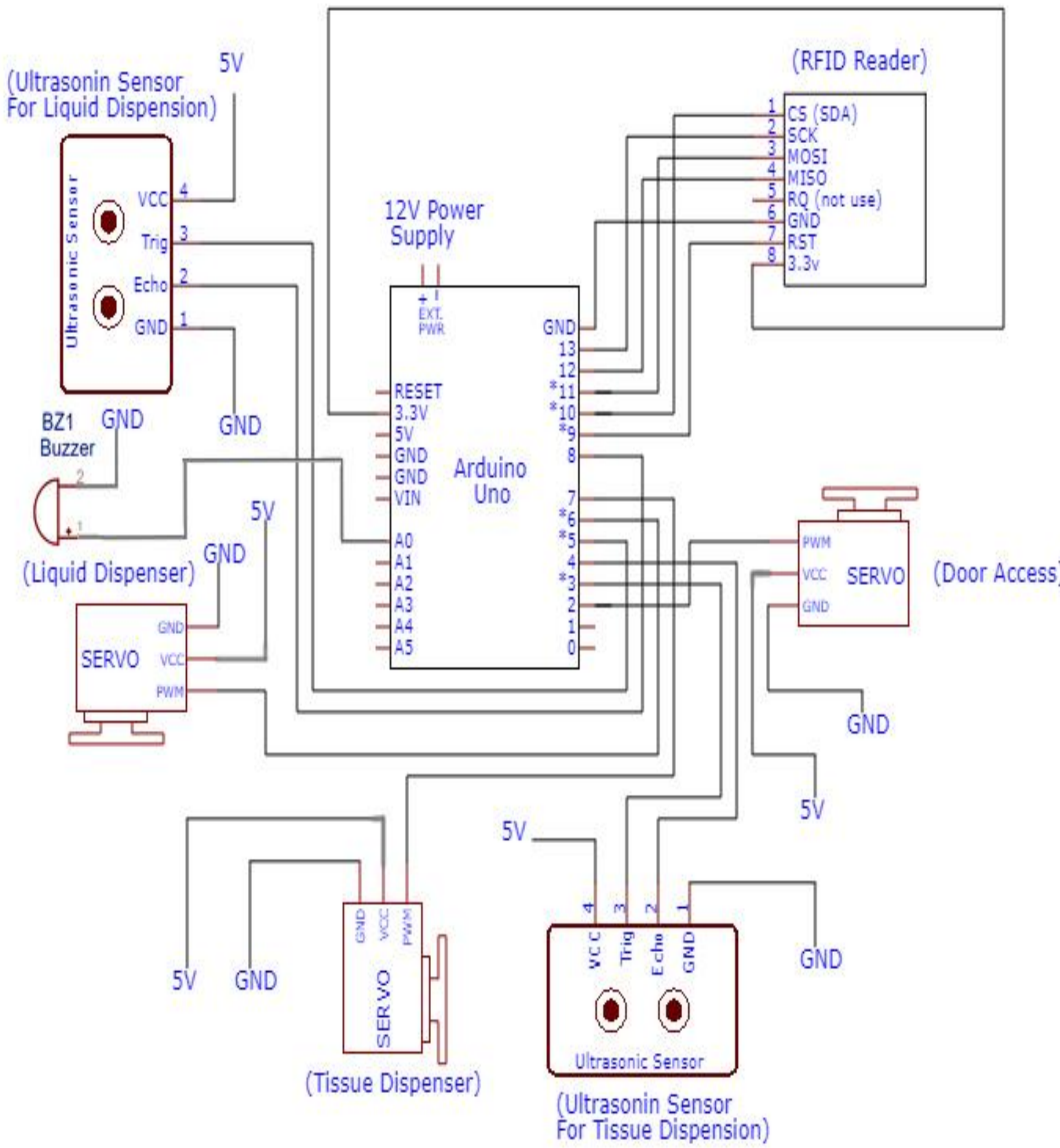

Figure: 4.1 Circuit Diagram of Mini Project

4.2 CIRCUIT DIAGRAM OF MAJOR PROJECT

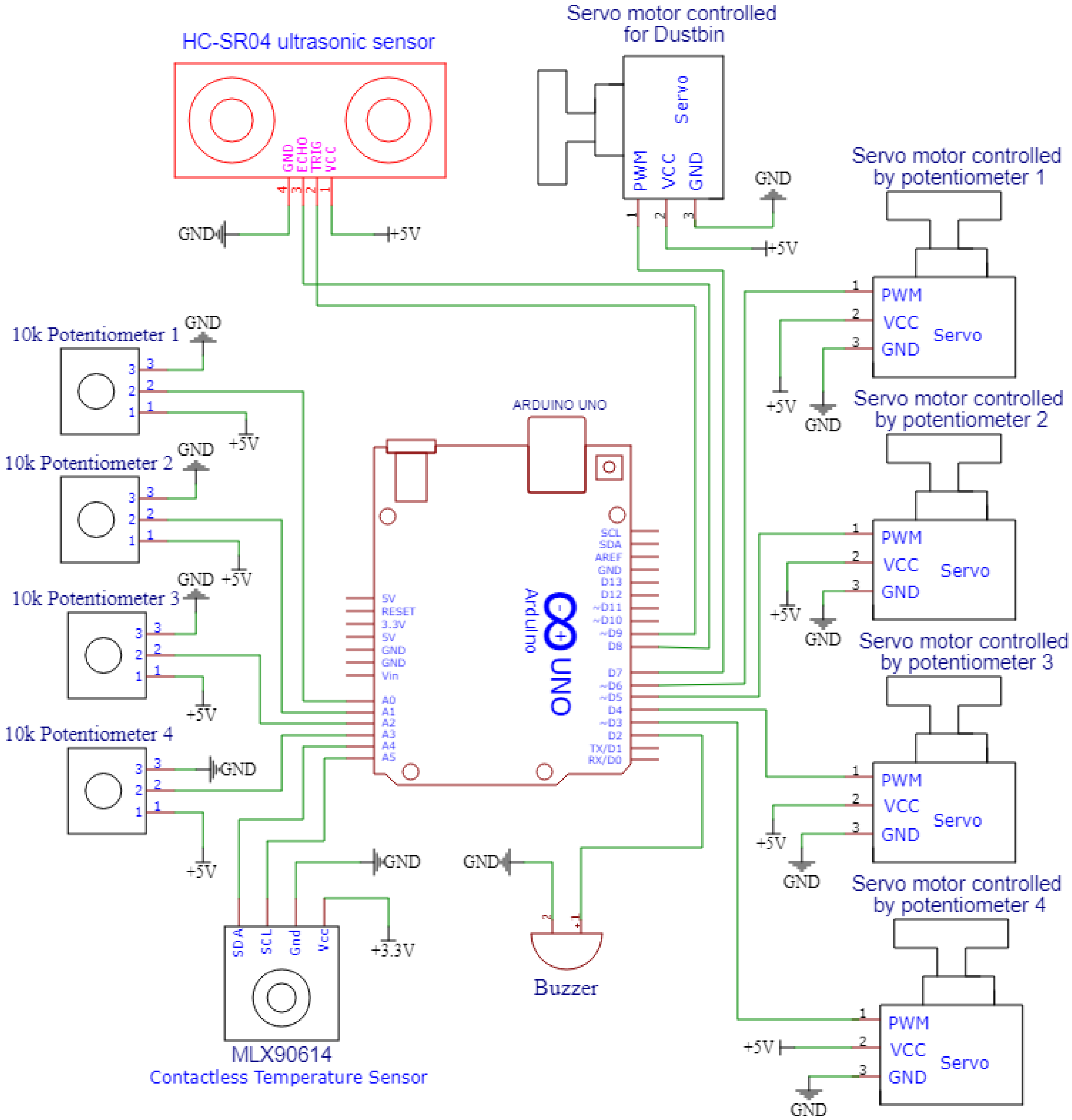

Figure: 4.2 Circuit Diagram of Major Project

4.3 FLOW CHART

The below figure shows the flow chart of the operation of the prototype of how the system performs, the range of RFID and MLX90614 is 5cm and the range of the Ultrasonic is set to 10cm, these ranges allow us to make it contactless prototype for safer precautions.

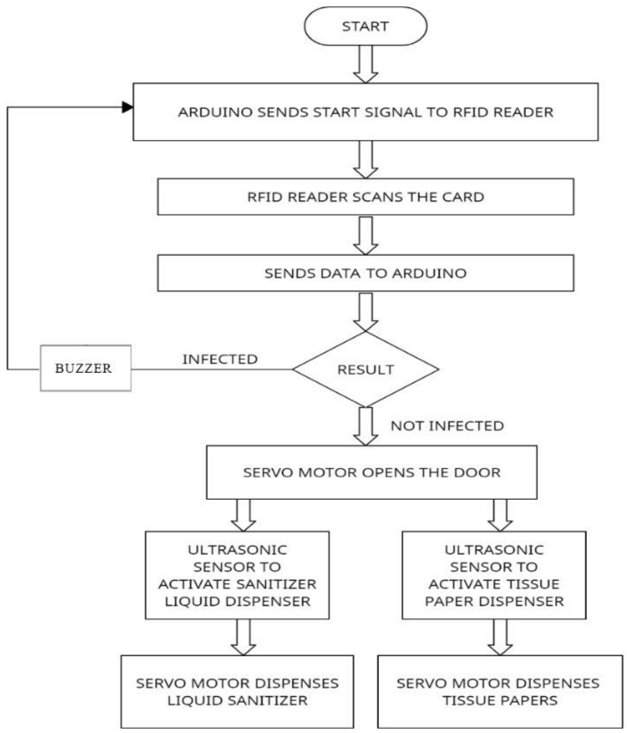

Figure: 4.3 Flow Chart of Mini Project

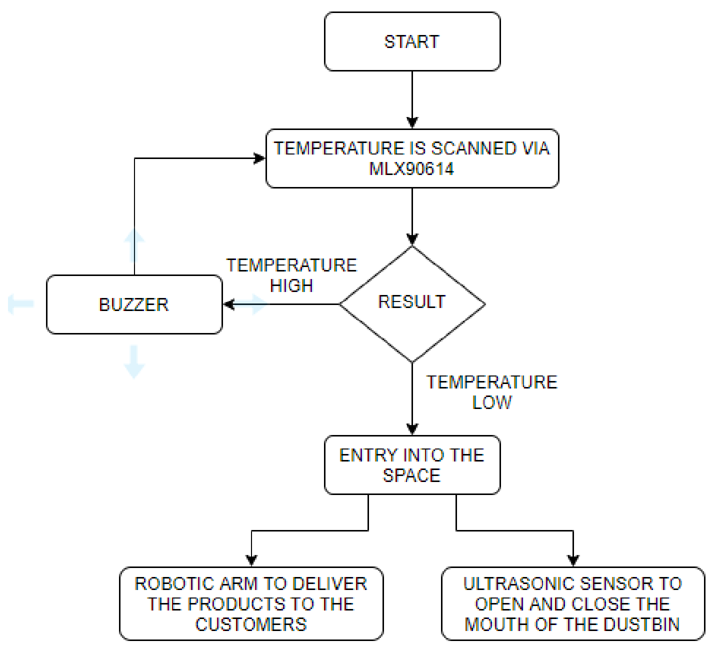

Figure: 4.4 Flow Chart of Major Project

CHAPTER-5

SOFTWARE IMPLEMENTATION

5.1 AURDINO SOFTWARE IDE USER'S GUIDE

Arduino is an open-source project, enabling hobbyists to easily take advantage of the powerful ATMega chips. The Arduino IDE is the software where you can write code and upload it to the ATMega chip. The code is then executed on the chip. Most 3D-printer electronics are Arduino-compatible, they use the ATMega chip and enable the user to upload their code using Arduino. This includes Megatronics, Minitronics and RAMPS. Before you can start using the electronics you need software 'firmware' that translates machine instructions into actual movements. There are a few options here, including Marlin and Sprinter and Repeater. The actual firmware is not discussed in this document. You can use Arduino to upload this firmware onto your electronics. This document will guide you in the steps you need to take.

To upload a firmware, you must first open the files using File → Open. Select the .ino file from the directory containing the firmware. Arduino will open several tabs with files. Next step is to select the correct electronics board. From the Tools menu, locate the Board item. This item should include a few sub items, including Megatronics, Minitronics, Arduino mega 2560 (RAMPS with mega 2560) and Arduino Mega 1280 (RAMPS with mega1280). Select the board that fits your electronics. Also we need to select the serial port the electronics is connected to. In the Tools menu, locate the Serial port item. This should include at least one item if the board is connected and the drivers are installed properly. If there are multiple items here, you need to find out which is the correct one by unplugging the board and checking which port was removed. Once you have set the board and serial port, you can upload the firmware by pressing File → Upload. Arduino will try to compile the firmware, if any errors occur the process will stop and you will need to fix the errors before trying again. Once compilation is complete, the actual upload will start. This may take a minute for a large sketch.

Program Structure:

A basic Arduino sketch consists of two functions called setup() and loop().

Open the Arduino IDE and select File → Examples → 01.Basics → BareMinimum to see the two functions. These two functions now appear in a default new Arduino IDE window, so it is not necessary to open the BareMinimum example sketch in a new version of the IDE.

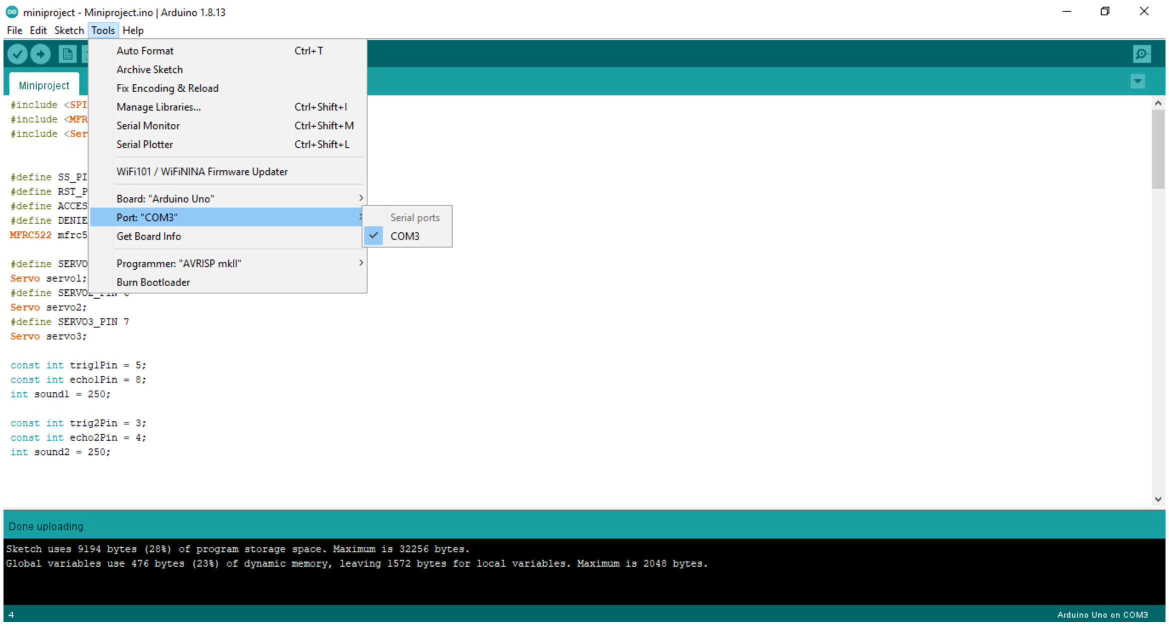

Figure: 5.1 Selecting the board

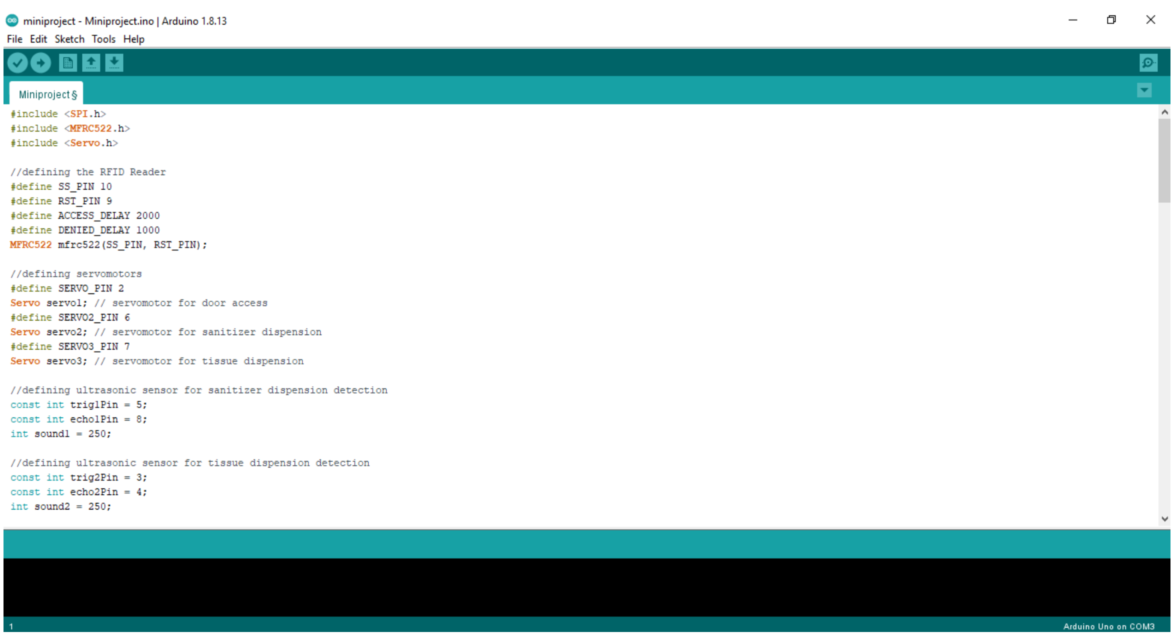

Figure: 5.2 Project code

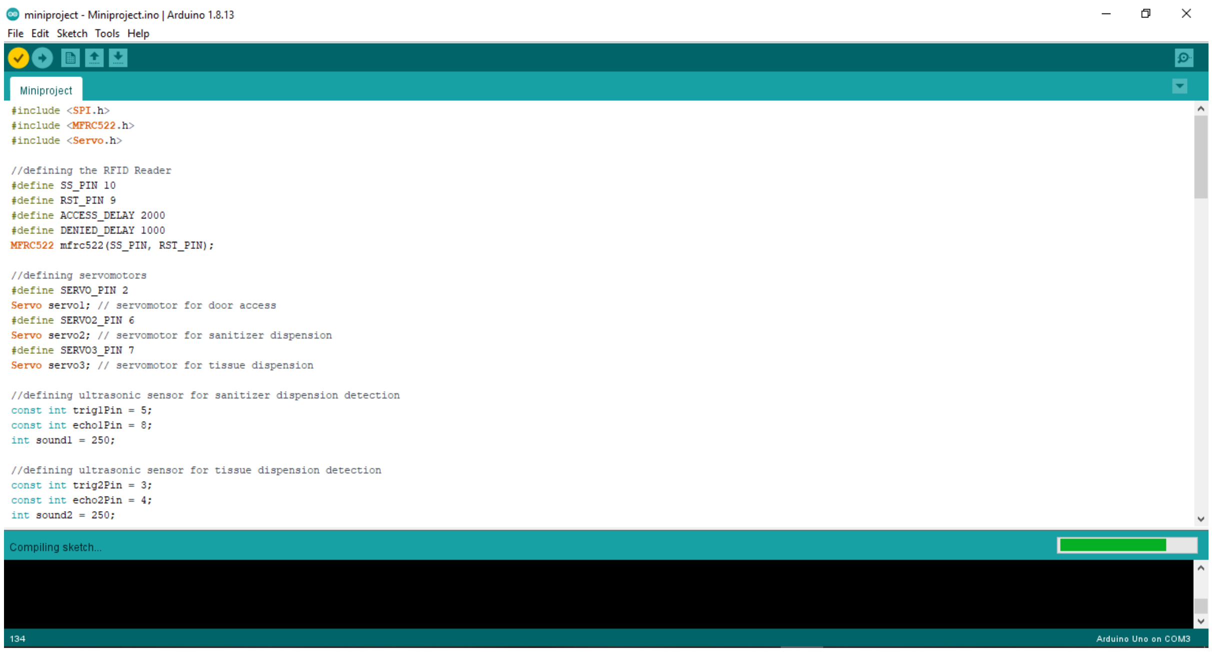

Figure: 5.3 Compiling the code

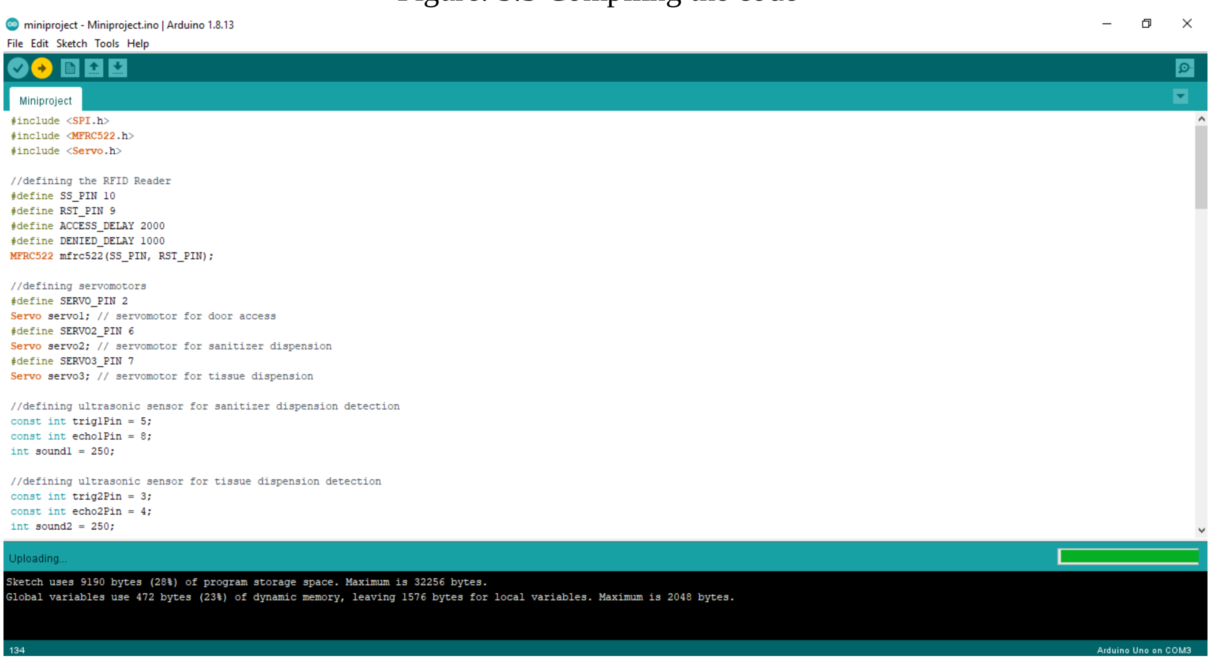

Figure: 5.4 Dumping the code

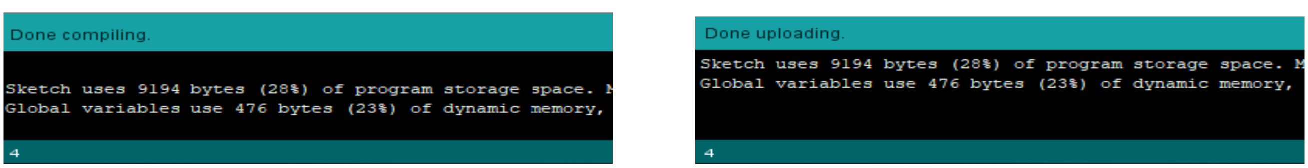

Figure: 5.5 Code uploaded successfully

CHAPTER-6

RESULTS

6.1 RESULTS

The proposed design used to control and provide the access to the people who are not affected by the virus and to keep out the people who are affected by it. The system majorly works on various components which are RFID Reader, MLX90614 temperature sensor, Ultrasonic Sensor, Servo Motors, Potentiometers and buzzers each of these components have their own participation to do and the results of these components will be displayed in the below images.

Access to the space is given by the RFID reader and then the Arduino sends a signal to the servo motor to open the doors, and the temperature sensor reads the body temperature value the Ultrasonic sensors activate the dispensers accordingly, the potentiometers along with the servo motors act as a robotic arm to deliver the products.

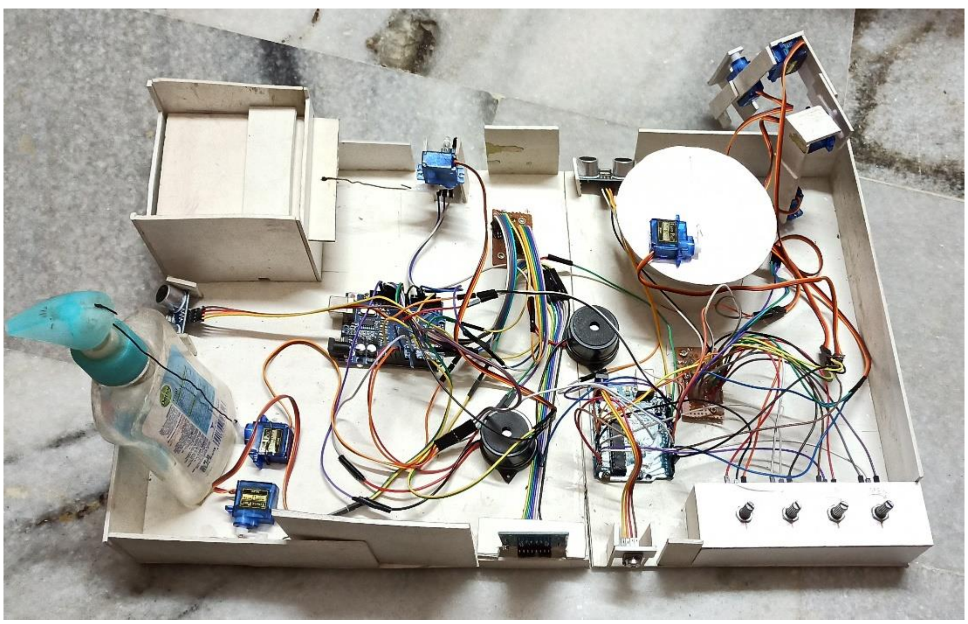

Figure: 6.1 Complete set up of the prototype

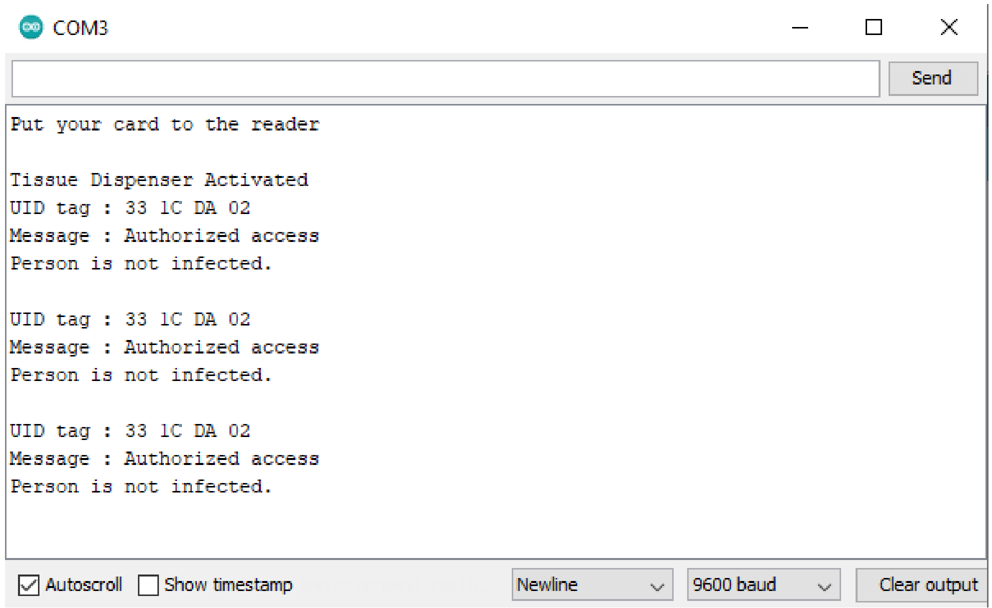

Figure: 6.2 Access Granted

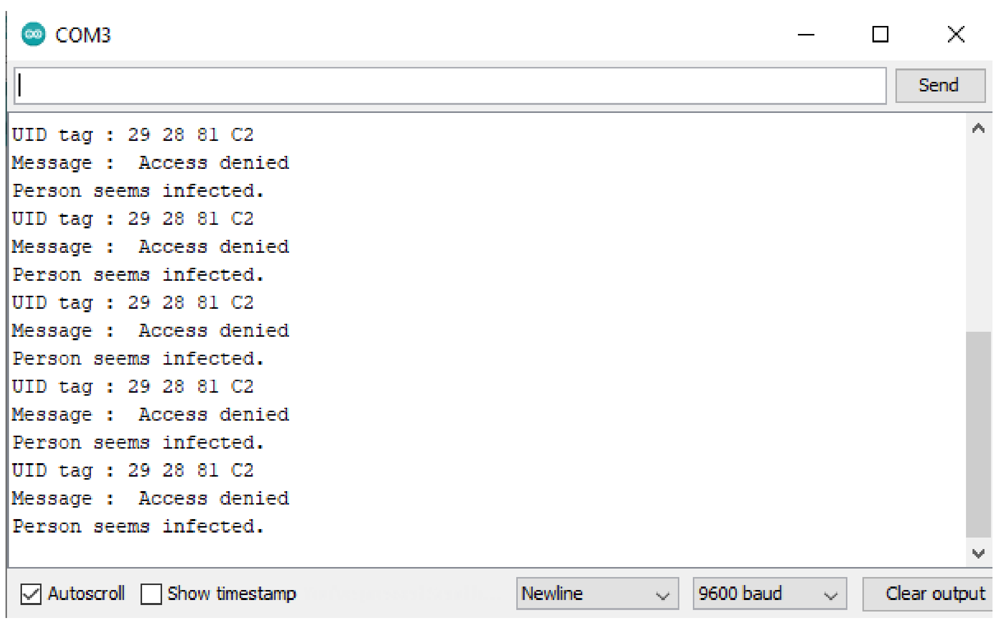

Figure: 6.3 Access Denied

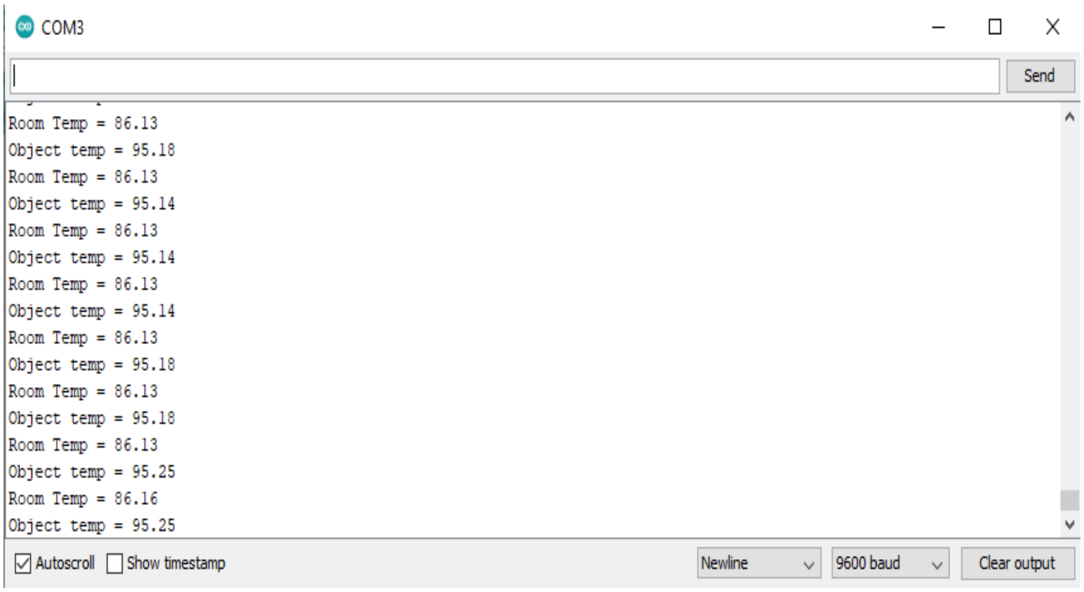

Figure: 6.4 Temperature Sensor Readings

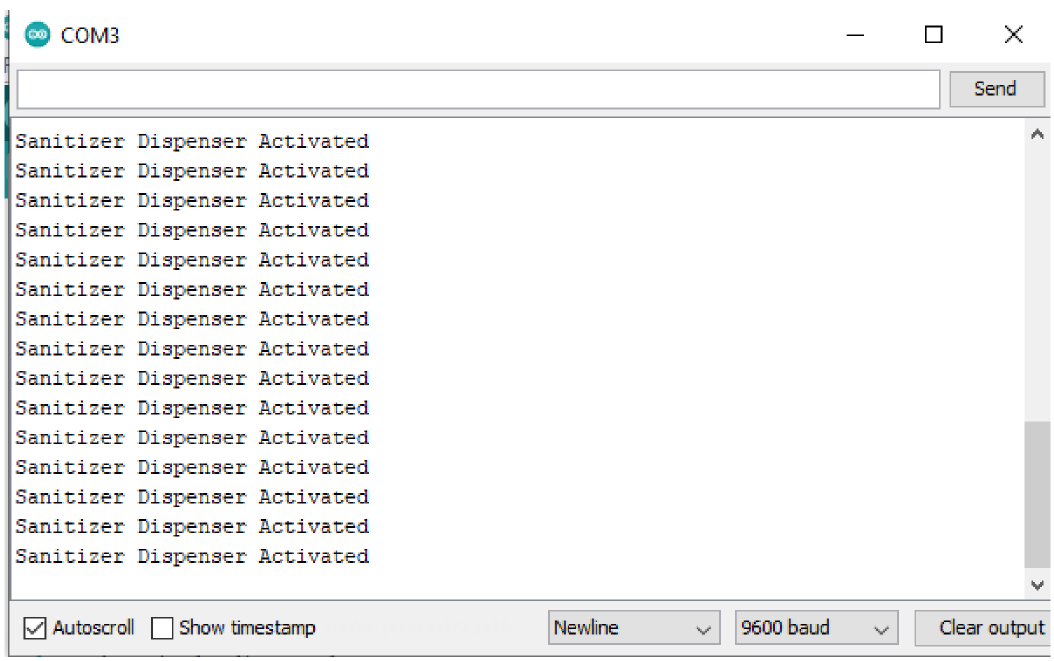

Figure: 6.5 Sanitizer Dispenser Activated

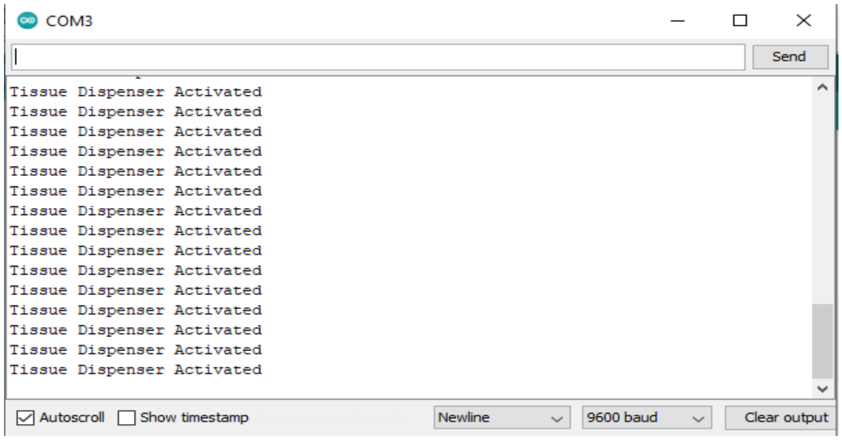

Figure: 6.6 Tissue Dispenser Activated

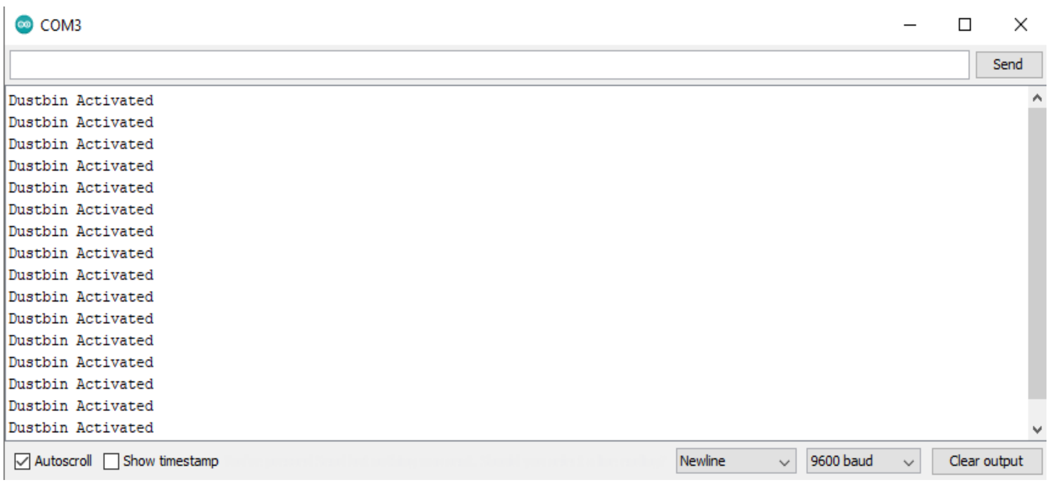

Figure: 6.7 Dustbin Activated

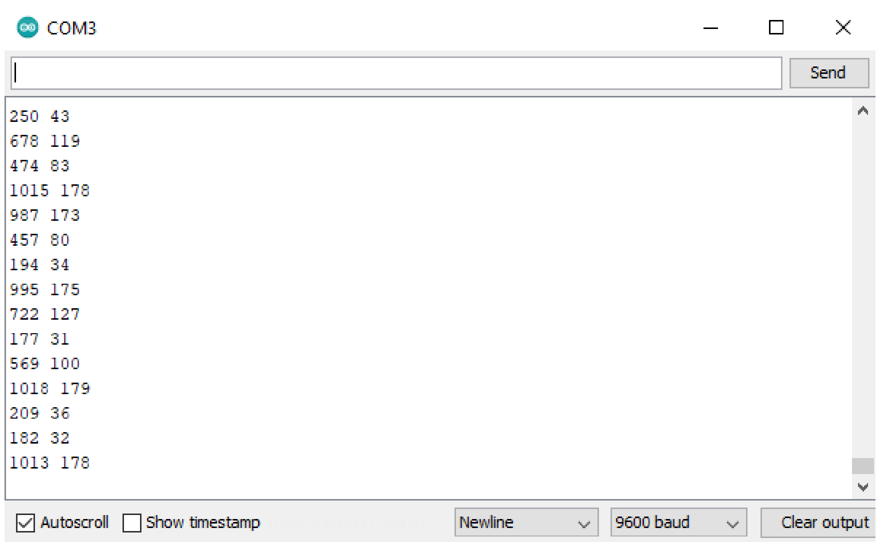

Figure: 6.8 Potentiometer and Servo Motor Readings

CHAPTER-7

ADVANTAGES AND APPLICATIONS

7.1 APPLICATIONS

- Automated door opening systems.
- Automated sanitizer dispensing.
- Safe entrance and access systems.
- Small scale organizations.

7.2 ADVANTAGES

- Contactless working.
- Safe entry into the space.
- Minimal hardware requirement.
- Decreases the spread of unwanted virus.
- Increases the security.
- Decreases human to human contact as much as possible.
- Reduces labor to provide access into the space, hence avoiding the person from being affected by the virus.

CHAPTER-8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

This project work is designed and developed successfully. For the demonstration purpose a prototype module is constructed for live demonstration, results are found to be satisfactory. Since it is a demo module a proto type module is constructed with less cost. While designing and developing this proto type module, we have consulted few experts those who are having knowledge in various fields, these professionals working at different organizations belongs to Hyderabad helped us while building the concept.

This project revealed that building a relatively low cost, efficient access systems, contactless hand sanitization controlling using Arduino controller. As mentioned earlier, the project has been carried out several times and the aim of this thesis is to provide safe entrances and sanitization and to get knowledge with fundamentals of Arduino and RFID reader along with other sensors incorporated into the system.

8.2 FUTURE SCOPE

- The performance of the system can be further improved in terms of the operating speed and instruction cycle period of the microcontroller by using other controllers. The number of channels can be increased to interface a greater number of sensors which is possible by using advanced versions of microcontrollers.
- Currently the system is mainly objected for organizational level and further developments can be made to increase the scale of the system to state level.
- This system can be used to provide access to other states so as to decrease the spread of the virus and contain the severity and the damage caused by the spread of unwanted virus.

BIBLIOGRAPHY

Text Books:

1. Basic electronics By: GROB
2. LINEAR INTERGRATED CIRCUITS – By: D. Roy Choudhury, Shail Jain
3. The concepts and Features of Micro-controllers - By: Raj Kamal
4. Digital Electronics By JOSEPH J.CARR
5. Electronic Circuit guide book – Sensors – By JOSEPH J.CARR
6. Introduction to Embedded System – By Sibhu K V

Published Papers:

1. Nath, S., Banerjee, P., Biswas, R. N., Mitra, S. K., & Naskar, M. K. (2016). Arduino based door unlocking system.

2. Bakht, K., Din, A. U., Shehzadi, A., & Aftab, M. (2019). Design of an Efficient Authentication and Access Control System Using RFID. 2019 3rd International Conference on Energy Conservation and Efficiency (ICECE).

3. Kassim, A. M., Jaafar, H. I., Azam, M. A., Abas, N., & Yasuno, T. (2013). Performances study of distance measurement sensor with different object materials and properties. 2013 IEEE 3rd International Conference on System Engineering and Technology.

4. Bhargava, A., & Kumar, A. (2017). Arduino controlled robotic arm. 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA).

Journals:

1. Electronic Design
2. Electronics for you
3. Electronics Text.
4. Practical Electronics

Websites:

1. https://www.arduino.cc/en/main/howto
2. https://components101.com/wireless/rc522-rfid-module
3. https://components101.com/servo-motor-basics-pinout-datasheet
4. https://components101.com/ultrasonic-sensor-working-pinout-datasheet

APPENDIX

CODE FOR MINI PROJECT

```
#include <SPI.h>

#include <MFRC522.h>

#include <Servo.h>


#define SS_PIN 10

#define RST_PIN 9

#define
ACCESS_DELAY 2000

#define
DENIED_DELAY 1000

MFRC522
mfrc522(SS_PIN,
RST_PIN);


#define SERVO_PIN 2

Servo servo1;

#define SERVO2_PIN 6

Servo servo2;

#define SERVO3_PIN 7

Servo servo3;


const int trig1Pin = 5;

const int echo1Pin = 8;

int sound1 = 250;


const int trig2Pin = 3;

const int echo2Pin = 4;
```

```
int sound2 = 250;
//buzzer
 int buzzer = A0;
void setup()
{
Serial.begin(9600);
SPI.begin();
mfrc522.PCD_Init();

servo1.attach(SERVO_P
IN);
servo1.write( 0 );
delay(7500);
servo1.write( 0 );
Serial.println("Put your
card to the reader");
Serial.println();

pinMode(trig1Pin,
OUTPUT);
pinMode(echo1Pin,
INPUT);
servo2.attach(SERVO2_
PIN);
servo2.write(0);
pinMode(trig2Pin,
OUTPUT);
pinMode(echo2Pin,
INPUT);
servo3.attach(SERVO3_
PIN);
```

```
servo3.write(0);

pinMode(buzzer,
OUTPUT);

}

void loop()

{

long duration, distance;

digitalWrite(trig1Pin,
LOW);

delayMicroseconds(2);

digitalWrite(trig1Pin,
HIGH);

delayMicroseconds(10);

digitalWrite(trig1Pin,
LOW);

duration =
pulseIn(echo1Pin,
HIGH);

distance = (duration/2) /
29.1;

if (distance < 10) {

Serial.println("Sanitizer
Dispenser Activated");

servo2.write(90);

delay(2000);

servo2.write(0);

}


long duration1,
distance1;

digitalWrite(trig2Pin,
LOW);
```

```
delayMicroseconds(2);

digitalWrite(trig2Pin,
HIGH);

delayMicroseconds(10);

digitalWrite(trig2Pin,
LOW);

duration1 =
pulseIn(echo2Pin,
HIGH);

distance1 = (duration1/2)
/ 29.1;

if (distance1 < 10) {

Serial.println("Tissue
Dispenser Activated");

servo3.write(150);

delay(2000);

servo3.write(0);

}

if ( !
mfrc522.PICC_IsNewCa
rdPresent())

{

return;

}

if ( !
mfrc522.PICC_ReadCar
dSerial())

{

return;

}

Serial.print("UID tag :");

String content= "";
```

```
byte letter;

for (byte i = 0; i <
mfrc522.uid.size; i++)

{

Serial.print(mfrc522.uid.
uidByte[i] <  0x10 ? " 0"
: " ");

Serial.print(mfrc522.uid.
uidByte[i], HEX);


content.concat(String(mf
rc522.uid.uidByte[i] <
0x10 ? " 0" : " "));


content.concat(String(mf
rc522.uid.uidByte[i],
HEX));

}

Serial.println();

Serial.print("Message :
");

content.toUpperCase();

if (content.substring(1)
== "33 1C DA 02")

{

Serial.println("Authorize
d access");

Serial.println("Person is
not infected.");

Serial.println();

servo1.write( 180 );

delay(4500);

servo1.write( 0 );

}
```

```
else
{
Serial.println(" Access
denied");
Serial.println("Person
seems infected.");
delay(DENIED_DELAY
);
digitalWrite(A0,HIGH);
delay(1000);
digitalWrite(A0,LOW);}
}
```

CODE FOR MAJOR PROJECT

```
#include <Servo.h>
#include <Wire.h>
#include
<Adafruit_MLX90614.h
>
Adafruit_MLX90614
mlx =
Adafruit_MLX90614();
double temp_amb;
double temp_obj;
int pot_pin_claw = A0;
int pot_pin_bottom =
A1;
int pot_pin_joint1 = A2;
int pot_pin_joint2 = A3;
int value_claw;
int value_bottom;
```

```
int value_joint1;

int value_joint2;


#define SERVO2_PIN 2

Servo servo2;

Servo servo_claw;

Servo servo_bottom;

Servo servo_joint1;

Servo servo_joint2;

//ultrasonic sensor

const int trig1Pin = 9;

const int echo1Pin = 8;

int sound1 = 250;

//buzzer

 int buzzer = 12;

void setup() {

 //setup for ultrasonic
sensor

pinMode(trig1Pin,
OUTPUT);

pinMode(echo1Pin,
INPUT);
servo2.attach(SERVO2_
PIN);

servo2.write(0);

pinMode(trig2Pin,
OUTPUT);

  pinMode(echo2Pin,
INPUT);

servo3.attach(SERVO3_
PIN);
```

```
  servo3.write(0);

  //setup for buzzer

  pinMode(buzzer,
OUTPUT);

  servo_claw.attach(4);

  servo_bottom.attach(7);

  servo_joint1.attach(3);

  servo_joint2.attach(6);

  Serial.begin(9600);

pinMode(12,OUTPUT);/
/ Connect buzzer at D8

  //Initialize MLX90614

  mlx.begin();

  digitalWrite(12,LOW);

}

void loop() {

//mlx.readAmbientTemp
F() ,
mlx.readObjectTempF()
)

  temp_amb =
mlx.readAmbientTempF
();

  temp_obj =
mlx.readObjectTempF()
+7;

  if(temp_obj > 102) {

  digitalWrite(12,HIGH);

}

  else{

   digitalWrite(12,LOW);

  }
```

```
  //Serial Monitor
  Serial.print("Room
Temp = ");
Serial.println(temp_amb)
;
  Serial.print("Object
temp = ");
 Serial.println(temp_obj);
 long duration1,
distance1;
  digitalWrite(trig2Pin,
LOW);
  delayMicroseconds(2);
  digitalWrite(trig2Pin,
HIGH);
  delayMicroseconds(10);
  digitalWrite(trig2Pin,
LOW);
  duration1 =
pulseIn(echo2Pin,
HIGH);
  distance1 =
(duration1/2) / 29.1;
  if (distance1 < 10) {
  Serial.println("Dustbin
Activated");
  servo3.write(90);
  delay(2000);
  servo3.write(0);
  }
  int pot_inputs[4];
  pot_inputs[0] =
```

```
analogRead(pot_pin_cla
w);
  pot_inputs[1] =
analogRead(pot_pin_bott
om);
  pot_inputs[2] =
analogRead(pot_pin_join
t1);
  pot_inputs[3] =
analogRead(pot_pin_join
t2);
  value_claw =
map(pot_inputs[0], 110,
800, 180, 0);
  value_bottom =
map(pot_inputs[1], 110,
800, 180, 0);
  value_joint1 =
map(pot_inputs[2], 110,
800, 180, 0);
  value_joint2 =
map(pot_inputs[3], 110,
800, 180, 0):
servo_claw.write(value_
claw);
servo_bottom.write(valu
e_bottom);
servo_joint1.write(value
_joint1);
servo_joint2.write(value
_joint2);
}
```