SMART GATE

Ahmed Aboulebdeh	**17700879**
Yasemin Keskin	**17000062**
Huddah Jannat	**17700359**
Eda Ozkonak	**15000081**
Volkan Oruc	**17000063**

Graduation Design Project Report

submitted to the

Department of Electrical and Electronic Engineering in

partial fulfilment of the requirements for the degree of

Bachelor of Science in

Electrical and Electronic Engineering Eastern

Mediterranean University

Approval of the Department of Electrical and Electronic Engineering

Prof. Dr. Rasime Uyguroğlu

Chair, Department of

Electrical and

Electronic Engineering

Authors: Ahmed Aboulebdeh
Yasemin Keskin
Huddah Jannat
Eda Ozkonak
Volkan Oruc

Supervisor: Assit.Prof. Dr. Davut Solyali

Jury Date: 28 June 2021

Jury Members:

1. Prof. Dr. Mustafa K. Uyguroğlu
2. Prof. Dr. Runyi Yu
3. Prof. Dr. Osman Kükrer
4. Prof. Dr. Hassan Amca
5. Prof. Dr. Hüseyin Özkaramanli

SMART GATE

by

Huddah Jannat

Ahmed Aboulebdeh

Eda Ozkonak

Volkan Oruc

Yasemin Keskin

Supervisor: Assit Prof. Dr. Davut Solyali

June 2021

ABSTRACT

This report discusses the design of a simple, and low-cost Smart Gate and aims to provide a solution for the problem of coming into contact with people whilst taking their temperature readings. With millions of cases and hundreds of thousands of deaths due to severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) infections in the world, screening people for SARS-CoV-2, the virus responsible for coronavirus diseases 2019 (COVID-19), has become a priority. In that fever is one of the earliest and most frequent manifestations of the illness, temperature screening has been a focal point of case detection during the pandemic. The purpose of this report is to give a clear and concise description of the smart gate design design , the sensors and controllers going to be used to operate it ,the method of and implementation of image processing used to detect and collect data which includes various car plates and their corresponding temperature readings in conjunction with the appropriate gate response. This report also explains the use of Arduino software going to be used to manipulate the feedback gotten from the temperature readings.

The report concludes that Temperature screening for SARS-CoV-2 is an integral component of containment efforts globally.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Ω	Ohm
AC	Alternating Current
DC	Direct Current
DOF	Degrees of Freedom
EEPROM	Electrically Erasable Programmable Read-Only Memory
ICSP	In Circuit Serial Programming
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IET	Institution of Engineering and Technology
IR	Infrared Radiation
I/O	Input/output
Li-Ion	Lithium Ion
MHz	Mega Hertz
OCR	Optical Character Recognition
PUMA	Programmable Universal Manipulation Arm
PWM	Pulse Width Modulation
RAM	Random-Access Memory
RPM	Revolutions Per Minute
SCARA	Selective Compliance Assembly Robot Arm
TTL	Transistor–Transistor Logic

Chapter 1

1. PROBLEM STATEMENT

With the alarming pace at which the corona virus spreads the need for effective foolproof measures to contain the pandemic have arisen. The first step in reducing the rate at which the virus spreads has been being able to identify possible victims through measuring their body temperatures especially in public settings with minimal physical interaction. In healthcare settings, the CDC defines fever as a forehead/hand temperature greater than or equal to (38°C). In screening persons for infections requiring quarantine in the nonhealthcare setting, the CDC defines fever as a forehead temperature (≥38.0°C) obtained with a noncontact infrared smart gate design. Handheld NCIT's are now being used to screen persons for possible SARS-CoV-2 infection in a variety of settings, of which airports, entrances of schools and hotels are of particular interest. However, these handheld NCIT's don't eliminate physical contact fully during mass screening of potentially infected persons and the close distance required to properly take a person's temperature represents a risk of spreading disease between the person using the device and the person being evaluated. This is where the Smart Gate comes in. Instead of humans coming into contact at all to risk contracting the Smart Gate will replace them and it will be controlled in a control room by a third party. One person will be present in the control room to preform maintenance and provide human intervention or respond accordingly with the given data collected. and this way. This smart gate design will be powered primarily by batteries and this will be effective for this purpose. We have realized that EMU institution as a whole has 2 personnel security staff just for that purpose at each gate, imagine how risky is it in terms of interaction with other potential covid holders and how costly will it be to have to hire 2 extra people at each gate of the university.

Chapter 2

2. PROJECT OBJECTIVES

The Smart Gate is very futuristic. Our objective is to make a smart gate design that will eliminate physical contact during the reading of temperatures, increase response time based on the feedback and data acquired from the whole process and enable a sizable number of readings to be undertaken in the shortest time possible whilst reducing the risk of infection as much as possible. This smart gate design will be primarily powered by a battery and we will build this design at an affordable price so that it can be used by anyone in who needs it. The smart gate design will be used in public settings that usually have human traffic like airports, schools, malls e.t.c and this will efficiently regulate and help in the containment of the COVID-19 virus. Furthermore, this project will also take pictures of car plate numbers by means of image processing and will feed in a database the time that the measurement of the temperature has been taken, the temperature, and the plate number, this project is aimed at assisting humans by providing quality work without necessarily eliminating human control. According to what temperature the person has it will decide whether to light up a green or a red LED light which will give a signal to the gate to be opened/remain closed accordingly “by means of using servo motor”. The project also boosts the employment sector by encouraging individuals to come up with even greater ideas that will benefit mankind. We made our project in such a way that it will benefit its users at the least possible cost. This project allows user control and all that is required is to run the program with the codes we will avail, once python and open CV, along with all the necessary libraries are properly installed on a computer.

Chapter 3

3. LITERATURE REVIEW

3.1 Temperature Sensors

Firstly, we will generally review the temperature sensors, they are various temperature sensor types which can be grouped in contact and contactless types. We can divide temperature measurement into contact and non-contact types. Contact measurement can only accurately measure temperature when the test object and sensor reach thermal equilibrium. This could mean longer response times and inaccuracies in reading balanced by ambient temperature. In contrast, non-contact measurement uses infrared radiation to measure temperature and does not require a direct touch. In addition, this measurement method can be read quickly and accurately. The examples include NTC is short for negative thermocouple. NTC is actually a resistor whose value changes depending on the temperature. As the temperature rises, its resistance decreases; In this way, it acts as a sensor.

3.2 Safety Limit Controllers

A safety limit controller is an off-off controller with a latching output. These devices are used in process control applications where independent limit switching for over or under temperature is required. They allow users to define upper or lower limits by entering setpoint values. If the pre-set temperature limit is exceeded, the output switches to ensure that the system is in a safe state. For example, the limit or monitor device can shut the system down or trigger forced cooling. Applications that often require limit and monitoring devices are ovens, furnaces, combustion plant, steam systems and food heating equipment, to name a few. When the output changes state it requires a manual reset to change it back. Safety limit controllers are typically used as redundant controllers, to shut down a process when undesirable limits are reached

Figure 3.1: Safety limit controller

3.3 Open CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library which was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. The library has over 2500 optimized algorithms, which include a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects and extract 3D models of objects, among others [10]. The library is also extensively used in companies, research groups and by governmental bodies. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications.

3.4 Arduino

Arduino was used in our project in order to code and program our sensors and micro servers. Arduino is a very flexible and user-friendly microcontroller . In addition, it is easy to differentiate Arduino boards from other microcontrollers due to its simple and user-friendly experience. It is easy for beginners to understand the software and hardware of Arduino. It can be used in advanced operations for example in robotics applications. Arduino based structure allows researchers to design, make improvements, makes the project less costly and allows remote control capability to the users. All these factors make Arduino a great choice for this project allowing us to program our non contact smart gate design and control it as we like while minimising the cost and making our lives easier. To obtain data, we read through various pertinent websites, books, journals and got assistance from our advisor.

3.5 Alternative designs

In this section, we are going to discuss the reasoning of how certain parts of our project work and how we could have done it differently.

An alternative design to build our project would have been with thermal camera using Arduino uno. We could have created a thermal camera system , generally with servo motors, non-contact temperature camera and Arduino. So the camera could check peoples temperature, and take photos if necessary. It could also have been possible to record a video to be utilized in the times when no person is physically present to monitor the camera feed. The system would be also connected to the door. So, after measuring people's temperature, the program can decide whether to grant access or not. However, the measured value may not be accurate for small temperature differences. It would costs a lot of money to implement this with good accuracy.

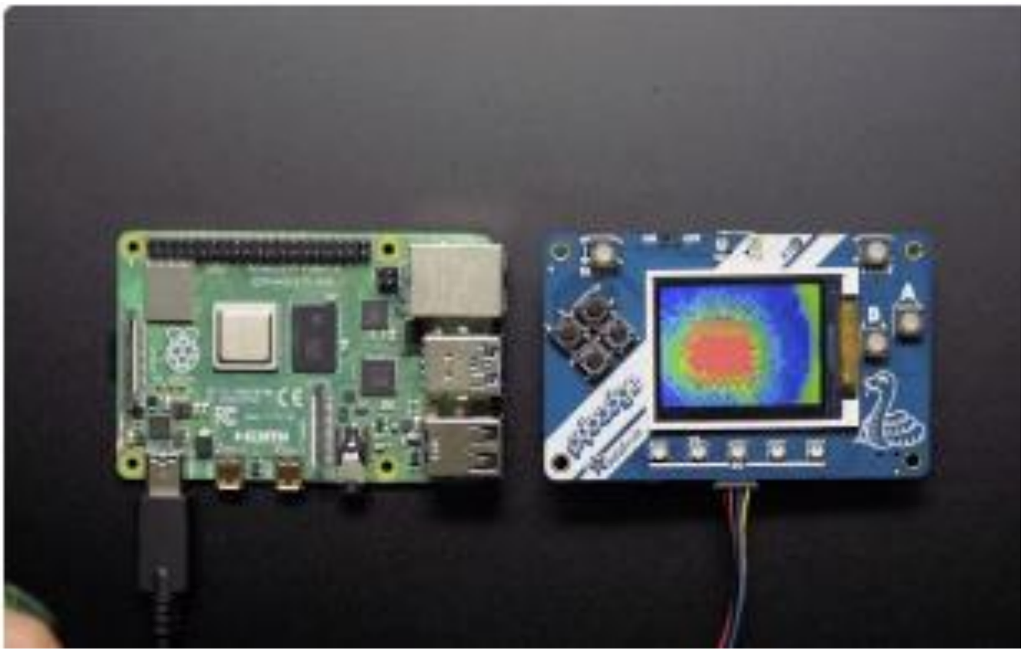


Figure 3.2: Thermal Camera Using Arduino

Secondly, a proximity sensor could have been used, a proximity sensor is a non-contact sensor that detects the presence of an object (often referred to as the “target”) when the target enters the sensor’s field. Depending on the type of proximity sensor, sound, light, infrared radiation (IR), or electromagnetic fields may be utilized by the sensor to detect a target. The two most commonly used proximity sensors are the inductive proximity sensor and the capacitive proximity sensor.

Unlike inductive proximity sensors, Capacitive proximity sensors, on the other hand, are not limited to metallic targets. These proximity sensors are capable of detecting anything that can carry an electrical charge. Capacitive sensors are commonly used in level liquid detection. Possible targets for capacitive sensors include but are but not limited to: glass, plastic, water, wood, metals, and a myriad of targets of other materials. But proximity sensors use more power and are more complex to program.

Furthermore, we could have implemented the gate movement using stepper motors. A stepper motor is an electromechanical device which converts electrical power into mechanical power. It is a brushless synchronous electric motor that divides its rotation into a number of steps, hence the name stepper motor. Since it completes its rotation in steps, it has smooth motion with excellent response. However, its less economical to use them as they are more costly compared to the micro servo motors.

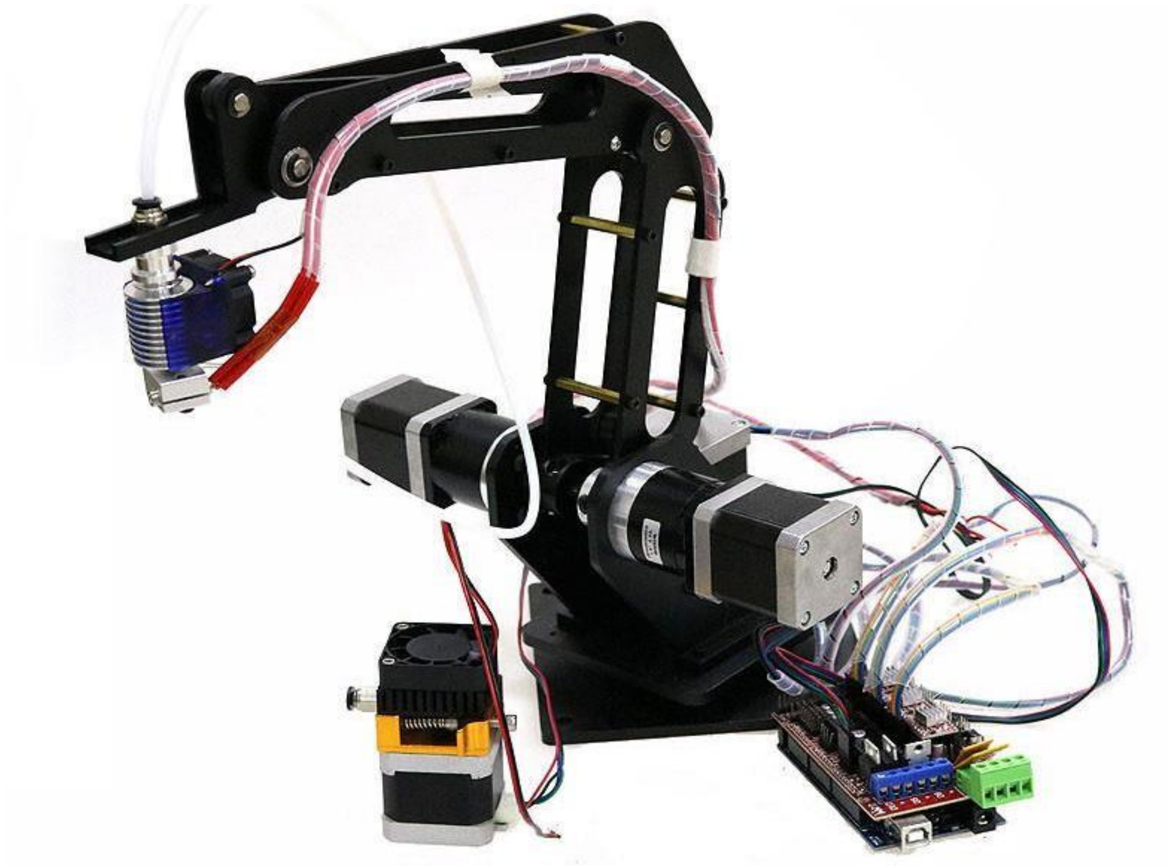


Figure 3.3: Stepper Motor Implemented device.

Finally, the material we used to construct our model housing was light wood An alternative would have been acrylic plastic or metal. Plastic is better in that it is less affected by the environmental factors. Metal is better in that it is more durable. However, we could not find a dense acrylic sheet and metal is heavier and can cause issues to the micro servo motors. With all these constraints, the design we chose was the optimum o

Chapter 4

4. METHODOLOGY

Our project is Arduino based. In our initial phase, we discussed on the type of materials to be used and the software applications to aid our design. In the development phase, we designed some parts of the model using the Fritzing application and simulated the smart gate design readings on MATLAB. We used an Arduino Uno, a servo motor, 1 breadboard, $4\times10K\Omega$ potentiometers, , remote control and IR sensor, $2\times3.7V$ 18650 lithium-ion batteries ,a camera, cars for testing, a power bank and plastic. Other pertinent materials are connecting wires, zip ties and glue/solder some parts of the model .

We chose this methodological approach because it minimizes the cost. This project has various advantages which include the use of Arduino which a very flexible and low-cost microcontroller however the disadvantage is that there are possibilities of malfunction.

4.1 Arduino Uno

The first task was to program the software which would be used to operate our smart gate which was Arduino uno to control the temperature and ultrasonic sensors and also the servo motors. Arduino is an open source programmable microcontroller which is used in various projects ranging from simple projects to even more complex ones. Since it is an open source, anyone can legally produce their own circuit board. It consists of a microcontroller which is programmable and is used for controlling and sensing objects. Arduino can respond to outputs connected to it by making use of sensors and inputs. It also consists of a software, Integrated Development Environment (IDE), in which the user can write the code on the computer and upload it to the board. The IDE uses a simple version of C++ programming language which is easier to learn. It is a microcontroller board that uses the AT mega-328 data. Fourteen input & output pins that are digital exist. From these pins, six are PMW outputs and six are analog inputs. In addition, there is an ICSP header, sixteen MHz ceramic resonator, a power jack with a USB port which is used to connect the microcontroller to the computer via USB cable to power it up and insert the code .

Figure 4.1: Arduino

Table 4.1. Arduino Uno Specifications

Type of microcontroller	**ATmega328P**
Supply voltage of microcontroller	5 V
Recommended power supply voltage	7- 12 V
The maximum power supply voltage of the board	6 – 20 V
Digital I / Os	14 (of which 6 support PWM)
PWM Modulation Output	6
Analog inputs	6
Permission current of digital output	20 mA
Permission output current 3.3 V	50 mA
The size of flash memory	32 KB (of which 0.5 KB is used by the loader)
RAM	2 KB
The volume of non-volatile memory (EEPROM)	1 KB
Clocking frequency	16 MHZ
Board length	68.6 mm
Board width	53.4 mm
Weight	25

4.2 MLX90614 Temperature Sensor

The reason we settled on this specific temperature sensor is because its small size, low cost, easy to integrate and Factory calibrated in wide temperature range:-40 to +85°C for sensor temperature -70 to +380°C for object temperature and also has a customizable PWM output for continuous reading among other reasons

The MLX90614 temperature sensor measures up to a distance of 40-50 cm without degradation and can measure the ambient temperature and also object temperature in front of it. It 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.

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.

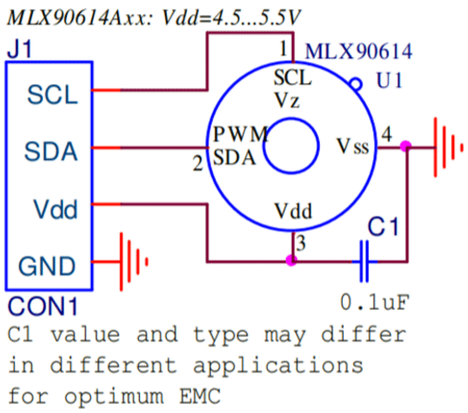


Figure 4.2: MLX90614 Temperature sensor 2D Model.

4.3 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 an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target). As simple as that, the ultrasonic gives us a result similar to the proximity sensor in terms of detecting objects in front of our “box” like kiosk.

4.4 Program code

Before setting up and tailor making the program code, its crucial to check the libraries required for the design that needs to be use. Hence we downloaded the libraries then installed Adafruits_MLX90614 to be able to use our temperature sensor on Arduino IDE. Also the servo library is needed but it's already included in Arduino IDE. After including the libraries we made the following specifications;

- Define green led to 9th pin as led_g and red led to 8th pin as led_g.
- Set servo motor as servo_3, mlx temperature sensor as mlx .
- To use the ultrasonic distance sensor , define a function named as getDistance. This function enables the distance to be determine from the ultra sonic sensor.
- The pinMode functionis used to determine output from sensor’s trigger pin and take input from sensor’s echo pin. But output from ultrasonic sensor is in microseconds unit. So conversion by dividing by 58 is pertinent . It's 58 because sound travels at approximately 340 meters per second. This corresponds to about 29.412 micoseconds per centimeter.
- To measure the distance the sound has travelled, the formula: Distance = (Time x Speedofsound) / 2 is used . It’s divided by 2 because sound has to travel back and forward. First the sound travels away from sensor after that it bounces off a surface and returns back. Finally the distance as a centimeters is read and recorded . We used Centimeters = ((Microseconds / 2) / 29) formula. So its microseconds/58. For example if it takes 500 microseconds for the ultrasonic sensor sound to bounce back, distance is 500/58 = 8.6 centimeters.

- In the setup part our servo motor is attached to the 3rd pin. Because it's needed to use Pulse-width modulation(PWM) pins such as 3rd pin in arduino uno. Then it's set servo to 0 degree using servo1.write() function.
- Using mlx.begin() function to start temperature sensor. Also using pinMode function the leds are set as output.

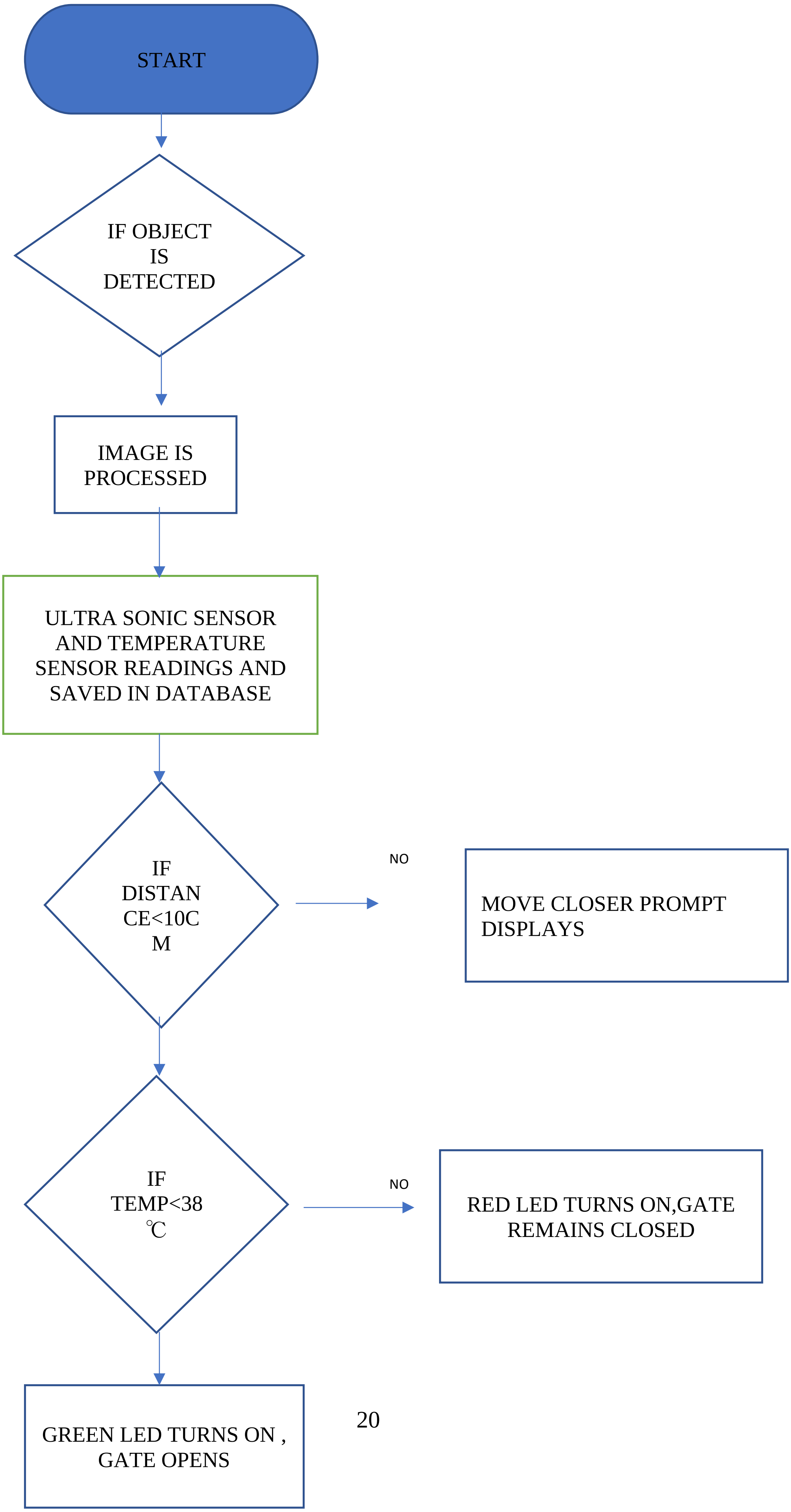
- In the loop section, first we define our getDistance function as distance and connect the pin to pin 7 and the echo pin to pin 6. Then use mlx.readObjectTempC() to read the temperature of the object and define it as the temperature. We then compare the distance to 10 centimeters in an if else statement. If it's less than 10 centimeters, we're comparing the temperature to 38 degrees Celsius in another if else statement. If it is lower than 38, the green led will turn on, the servo motor will move 90 degrees and the door will be opened.

```
void loop(){
 distance=getDistance(7,6);
 temperature=mlx.readObjectTempC();
 if(distance<10){
if((temperature)<(38)){
    digitalWrite(8,LOW);
    delay(15);
    digitalWrite(9,HIGH);
    servo_3.write(90);
    delay(15);
  }
  else{
    digitalWrite(8,HIGH);
    servo_3.write(0);
    delay(15);
    digitalWrite(9,LOW);
    delay(15);
    }
 }
 else{
    servo_3.write(0);
    delay(15);
    digitalWrite(9,LOW);
    digitalWrite(8,LOW);
 }
}
```

Figure 4.3: Arduino If-Else Condition.

- Setting servo.write(90) so servo will rotate 90 degree and the gate will be opened. If it does not meet with temperature condition gate will stay closed and red led will be turn on. If it does not meet with ultrasonic sensor condition, leds and gate will remain closed. It will not check temperature condition.

- We combined both sensors because temperature sensor is not accurate at far distance. We used digitalWrite(x,HIGH) to turn on the led and digitalWrite(x,LOW) to turn of it. Also using delay function, we added some delay between processes to avoid confusion with sensors and work properly.


START
IF OBJECT IS DETECTED
IMAGE IS PROCESSED
ULTRA SONIC SENSOR AND TEMPERATURE SENSOR READINGS AND SAVED IN DATABASE
IF DISTANCE<10CM
NO
MOVE CLOSER PROMPT DISPLAYS
IF TEMP<38℃
NO
RED LED TURNS ON,GATE REMAINS CLOSED
GREEN LED TURNS ON , GATE OPENS

4.5 Hardware

The following components were used to put together our device:

- Touch wires
- Arduino uno× 1
- Touch button × 1
- Resistance (5K ohm, 200 ohm) × 1
- MLX90614 sensor with capacitance or resistance × 1
- 9V battery × several
- GY-906 temperature sensor
- Breadboard
- Web camera

For putting together and connecting our hardware we utilisied the I2C protocol instead of the commonly used SPI. I2C was originally developed in 1982 by Philips for various Philips chips. The original spec allowed for only 100kHz communications, and provided only for 7-bit addresses, limiting the number of devices on the bus to 112 (there are several reserved addresses, which will never be used for valid I2C addresses). In 1992, the first public specification was published, adding a 400kHz fast-mode as well as an expanded 10-bit address space.

Each I2C bus consists of two signals: SCL and SDA. SCL is the clock signal, and SDA is the data signal. The clock signal is always generated by the current bus controller; some peripheral devices may force the clock low at times to delay the controller sending more data (or to require more time to prepare data before the controller attempts to clock it out). This is called "clock stretching"

I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

Like UART communication, I2C only uses two wires to transmit data between devices I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line).
Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master.

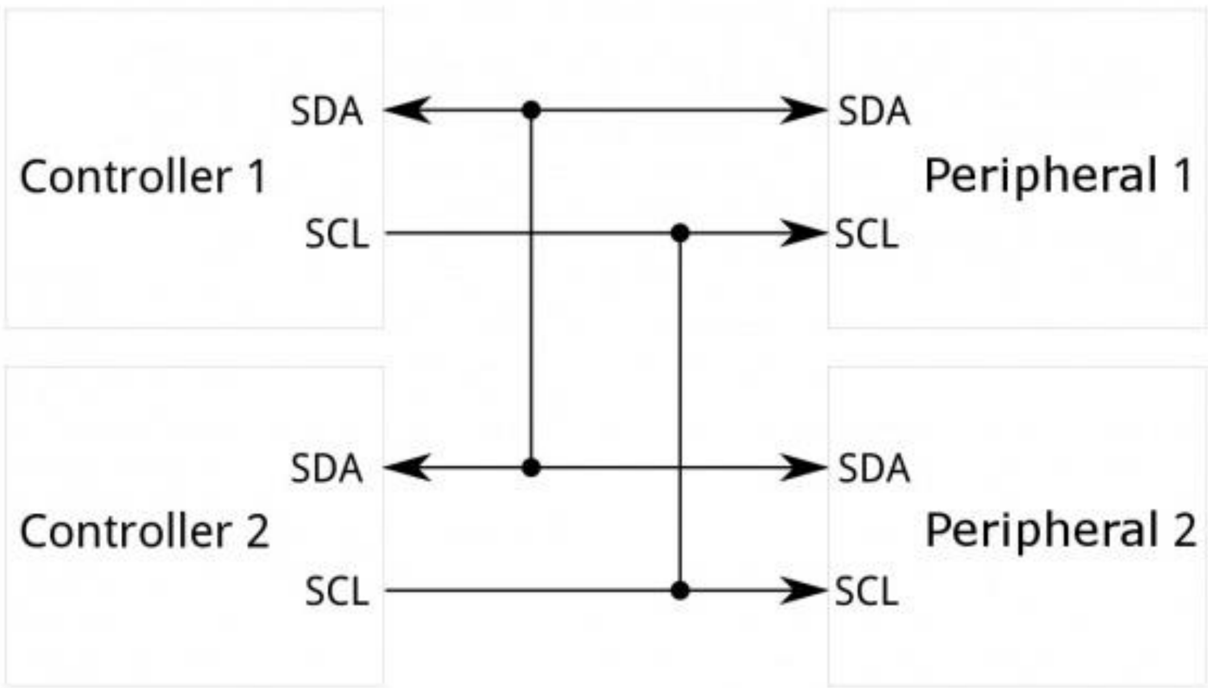


Figure 4.4: I2C Protocol

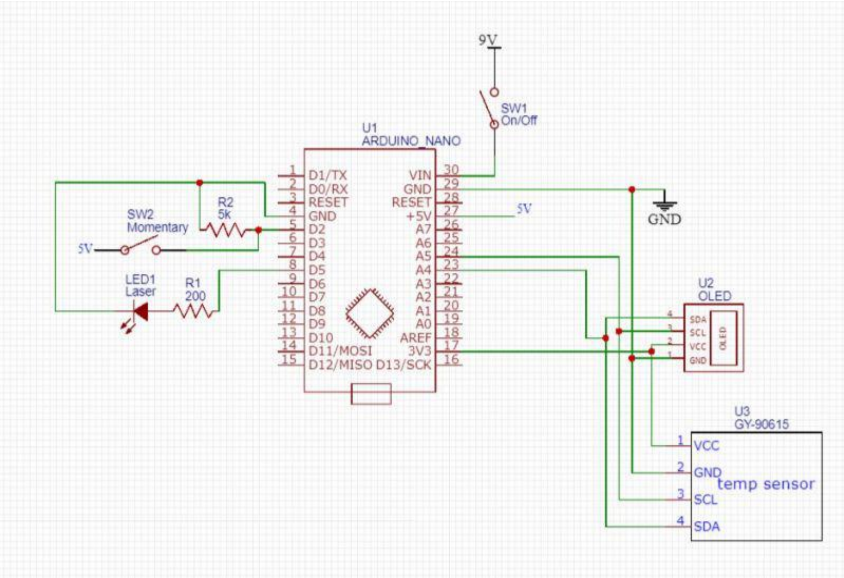


Figure 4.5: Circuit Diagram.

4.6 Servo Motor

This is a small rotary actuator with high power output. A micro servo can rotate about 180 degrees (90 degrees in each direction) and this is a useful factor in our project. It is very precise in terms of angular position, velocity and acceleration. Its precision surpasses that of other kinds of motors and will therefore be useful in our project in maintaining an almost perfect repeatability process. A servo motor is not a specific type of a motor as it contains both AC and DC motors and are used in the control system . The control signal is the input, which represents the final position and an encoder works as a sensor for position and speed feedback. The micro servo motor in our model will work in analogy to the servo motor in a real robotic arm.

Fig 4.6. Servo Motor

4.7 Other Materials

The micro servo motors are connected to each other by gluing them onto the light wood. We used the 3.7V 18650 Li-ion battery. They provide with high current and therefore high power. We used zip ties for our grip mechanism. We connected the potentiometers with the micro servo using the breadboard and jumper wires. We used light wood as our cover at the bottom and at the top for protection of the components.

4.8 Implementation

First, the components are prepared for use. The MLX 90614 temperature sensor is soldered since it comes separately from its pins. Then the ultrasonic sensor is placed on the breadboard and then the process of making connections begins. The Vcc pin is connected to the plus line on the breadboard with the help of a jumper cable. The Echo pin is connected to the 6th pin on the Arduino and the Trigger pin to the 7th pin. Finally, the gnd pin is connected to the minus line on the breadboard. Next, the MLX 90614 temperature sensor is placed on the breadboard and connected the Vin pin to the 3.3V pin on the Arduino, the gnd pin to the minus line, and the SCL and SDA pins to the SCL and SDA inputs on the Arduino, respectively.

Then the green and red leds are placed on the breadboard. The cathode legs which is short legs of the leds is added to the minus line of the breadboard, and the anode legs which is, the long legs, by adding 470 ohm resistors in front of them, and placed the green led on the 9th pin on the Arduino and the red led on the 8 pin on the Arduino.

Then the servo motor connections are made. The orange cable from the servo motor which is the communication pin is connected to the 3rd pin on the Arduino. The red cable which is the Vcc pin is connected to the plus line of the breadboard, and the brown wire the ground pin are connected to the minus line on the breadboard.

Finally, the jumper cable from the minus line of the breadboard is attached to the ground pin on the Arduino. The plus line is connected to the 5V pin on the Arduino, then the Arduino is connected to our computer with the help of usb cable.

The Arduino IDE had been previously installed is opened.

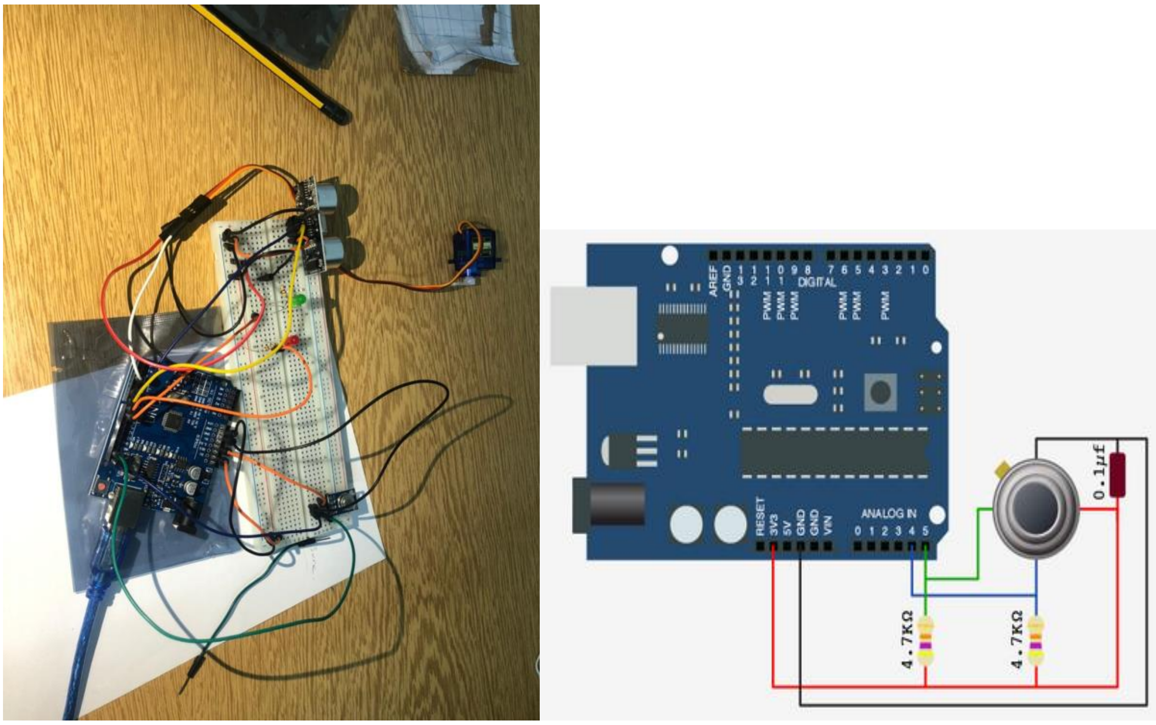


Figure 4.7: Smart Gate implementation

4.9 Car License Plate Recognition Using OpenCV

Car license plate recognition helps us to identify the cars that have entered through the gate and saves their license plate number into our database along with the body temperature value of the driver and the local time. This section includes, capturing the image from camera and adjusting the image using image processing, and detecting the license plate, then reading and saving the detected license plate into a text file.

OpenCV, Numpy, Pytesseract and Keras libraries are essential for car plate recognition in the written code. OpenCV is commonly used for applying image processing techniques to the image for better plate detection and character recognition. Numpy is used for working with ndimensional arrays, it can modify 3 layered (BGR) images with less effort compared to normal arrays. Pytesseract is used for license plate recognition using optical character recognition (OCR). Keras is used for implementing and executing the deep learning model for the car plate detection.

The car license plate recognition code can be split into two parts: license plate detection, license plate character recognition.

License Plate Detection:

Detecting the car license plate is the first thing that is focused on after taking the temperature of the driver. Using only OpenCV is not reliable to detect the car plate. The cars shape and color, daylight are the disturbing factors that requires adjustment of the image processing values. To prevent these issues, we used pre-trained Keras model that is called Wpod-Net (Warped Planar Object Detection). It detects the high probability position for the plate then wraps around the detected area as a rectangular object.

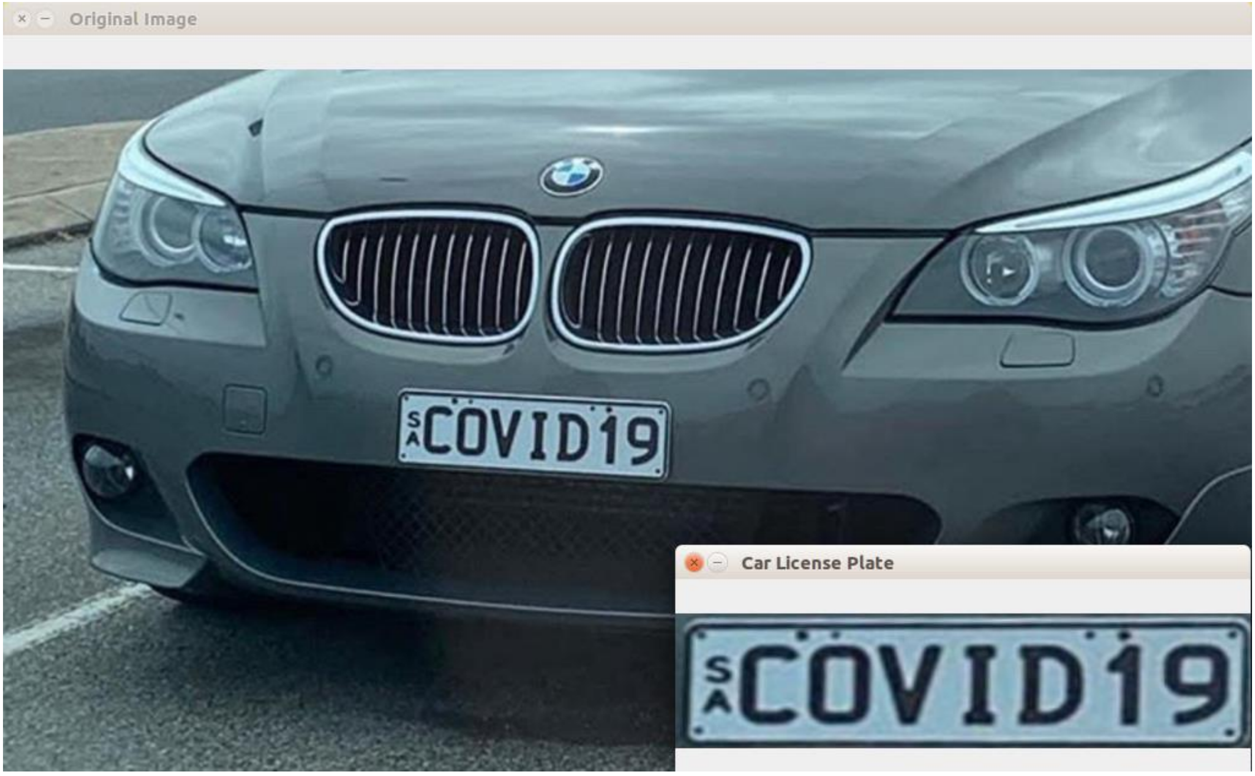


Figure 4.8: Car License Plate Detection

The Figure 4.8 above shows the result of the car license plate detection with Keras. As it can be seen, the output image is aligned perfectly horizontal even though the license plate is tilted in the original image. After cropping the image, the code continues with recognition of the characters in the plate.

License Plate Character Recognition:

In this section, OpenCV is highly used to adjust the image to make the characters readable by the pytesseract library. Pytesseract library is easier to implement in the code and runs faster compared to Keras models. So, it is efficient to use this library for license plate recognition.

Figure 4.9: Image Processing for Car License Plate

As seen in the Figure 4.9 above, firstly, the image is converted to grayscale since it is much manageable to work with one layered image. Then, Gaussian blur is used to reduce the noise from the image. After, threshold is applied to convert image to a binary format. Then the code finds the white areas that are separated from each other in this threshold image and labels each of them uniquely. So, it can be said that the characters are now separated and labeled.

Figure 4.10: Results of the Image Processing

The Figure 4.10 above shows the separated characters from the threshold image (left image). After, fine adjustments are done to clear the background (right image).

```

Car License Plate: COVID19

         Date/Time: 2021-06-22 04:52:29.740838

Car License Plate: COVID19

         Date/Time: 2021-06-22 04:52:54.560974
```

Figure 4.11: Saved Data Results

As a result, the data of the license plate and time is written in the text file as seen as above Figure 4.11.

Figure 4.12: Final Example of the Code

Another example from the code is shown above Figure 4.12.

Chapter 5

5. ENGINEERING DESIGN WITH REALISTIC CONSTRAINTS

5.1 Defining the Problem

In this section we will define the role of our Smart Gate. This helped us in the designing and implementation of a smart gate design that solves problems. We first tried to understand and specify situations in which our smart gate design will be working in. After this study, we will be able to answer the following questions:

- Operation: How will the smart gate design operate within the environment we are currently in? Are there any limitations with our system?
- Manipulation: How will the software be able to detect and tell the specific body temperature and detect the ideal one and the high one? Does it need to be combined with other applications in order to work perfectly?
- Accuracy: Is the margin of error for the results obtained in the data taken into account in our project?
- Which libraries are most suitable for the design?
- Is it efficient to use?
- Economic: Is the design economically feasible?
- Safety
- Time: How much time it takes to construct the model
- What development procedures are to be followed?

After answering these questions, we were able to determine the margin of error and the type of sensors and modifications to our code needed to optimize and increase the accuracy of the temperature readings. Furthermore the ideal placement needed for the smart gate design to enable cars and people at various heights to take their temperature readings.

5.2 Design

At this stage all data collected was utilized in building the smart gate design and some of the important parts namely the IR temperature sensor, the ultra-sonic sensor, microcontroller, Display, Display driver, web-cam and the Battery. Now our objective here was to reduce the cost and the most expensive material (at the time of documentation) was the temperature sensor itself. Sadly, though as a maker, there are not many options here that you can reach out quickly other than MLX90614 and MLX90615. On the other hand, if we were okay with using an Analog sensor, we would have had many cheaper alternatives but they wouldn't have been easy to build and calibrate our device. Furthermore the motors and sensors we used in the design were easier and faster and more efficient to program and use in conjunction with the Open CV image processing software and database.

5.3 The Construction of the Smart Gate

The smart gate design consists of four major parts which are the MLX90614 temperature sensor, the ultra-sonic sensor , the webcam and the Arduino Uno/ESP32. We set the base first followed by the first link. Successively, we tested the parts for accuracy and capability to withstand a specific temperatures and feed them into a database. This design procedure is useful because it enabled us to find replacement for faulty parts or components before its completion. We divided the design task amongst all group members in which other members were responsible for assembling the parts while others were responsible for post testing. We covered both the top and the bottom components of smart gate design using light wood for protection of the components and therefore, ensuring safety to the surroundings. We all contributed in writing the code and we uploaded to the Arduino at completion. Following this procedure, it was easier to assembly and manage the project without confusion.

5.4 Summary of procedure

In this section, we will summarise the steps through which we put together obtain . Firstly, we prepared the training data as explained earlier in this report, openCV 3.2.0 and NumPy. In addition we used other modules such as Adafruit_MLX90614 library.

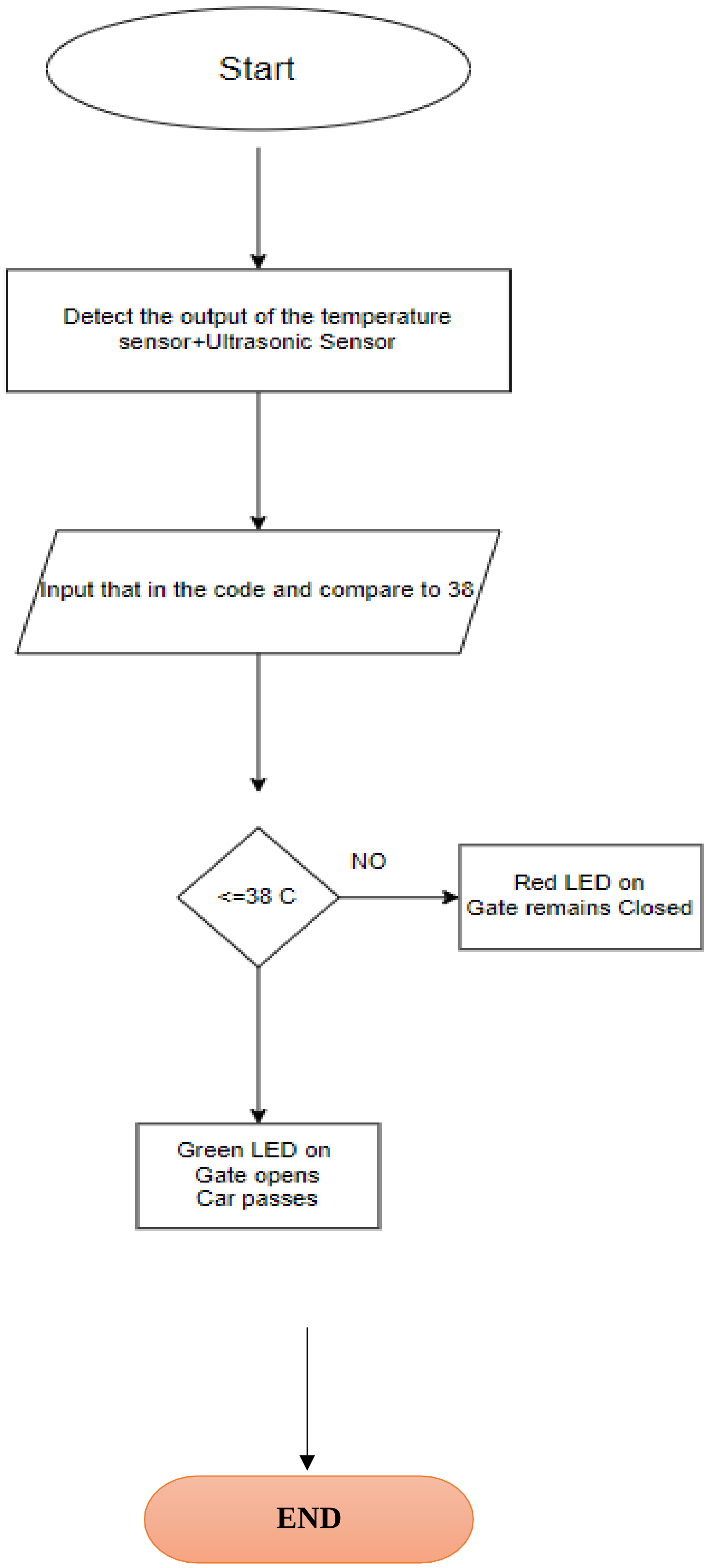


Figure 5.1: Smart Gate flow chart

5.5 Evaluating the project

In this section we will:

- evaluate the procedure
- evaluate the design

While assembling and programming work progressed, some group members carried out tests as mentioned earlier. When there was a failure in a joint, or any part of the design did not meet specifications and is not up to the IEEE standards of safety, we modified it. If the procedure followed failed to work, we also adjusted such that every member of the group does what they are well vested in. During programming of the code, the Arduino we were using is a clone, so the application could not find the Arduino. For this, we installed a driver that we downloaded from the internet and introduced the arduino to the computer. Then by choosing the port where we plugged the arduino. We transferred our code that we wrote via Arduino IDE to Arduino and ran it.

Chapter 6

6. ENGINEERING STANDARDS

According to European standards [14], “A standard is a document that provides rules, guidelines or characteristics for activities or their results, for common and repeated use”. Standards are created by considering all parties involved which include the engineers, consumers, inspectors, product or material and the process involved

The first fundamental Canon of the NSPE code of ethics for Engineers states that Engineers shall hold paramount the safety, health, and welfare of the public. To this end, our design model despite it’s use and operation being contactless, is enclosed in a box casing that is safe from electric shocks incase an object or person is to come into contact with it.

IEEE creates and oversees the international standards for telecommunications, information and technology and power generation [15]. However, the IEEE spans various industries which include aerospace, electronics and robotics and automation to mention but a few. These standards will ensure that the products developed are safe to the community. We ensured that our design meets the IEEE standards such it will be a useful design and pose no harm to the community. IEEE also sets the code of ethics which are useful in differentiating between wrong and right and applying certain principles to pass judgement to those respective situations.

IET standards were created to provide a uniform system that maintains the impeccable and high standards of our community [16]. These standards can depend on the field you are working in. for our case these standards are providing us a guide on how to design and implement a Smart Gate. These guidelines shed light on several key issues to consider when deciding the use of the design. Such us challenges in setting up and running the model, and how to effectively control it.

Chapter 7

7. TIME PLAN

To complete our project, we designed a time plan. Table.7.1 shows how research; data collection and documentation were done during the Fall 2020-21 semester and Table.7.2 states our time plan on how we will finish our project in the Spring 2021 semester.

Table.7.1 Fall 2020 time plan

		REASERCH AND DATA COLLECTION	**DOCUMENTATION AND REPORT WRITING**	**REPORT FORMATING AND EDITING**
Oct	Week 1			
	Week 2			
	Week 3			
	Week 4			
Nov	Week 1			
	Week 2			
	Week 3			
	Week 4			
Dec	Week 1			
	Week 2			
	Week 3			
	Week 4			

Table.7.2 Spring 2021 time plan

		CHOOSING SPECIFIC COMPONENTS	**ASSEMBLING THE PARTS TOGETHER**	**TESTING AND PROBLEM SOLVING**	**FINAL TESTING AND FINAL REPORT EDITING**
Mar	Week 1				
	Week 2				
	Week 3				
	Week 4				
Apr	Week 1				
	Week 2				
	Week 3				
	Week 4				
May	Week 1				
	Week 2				
	Week 3				
	Week 4				

Chapter 8

8. COST ANALYSIS

To accomplish this project, we purchased various materials and equipment. We purchased 2 Arduino Uno, 4 micro servo motors, 1 breadboard, 4×10KΩ potentiometers, 4 DC motors, L298N DC motor driver, remote control IR temperature sensor and ultrasonic sensor, 1×9V Duracell battery, 2×3.7V 18650 lithium-ion batteries and light wood.

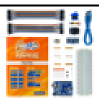
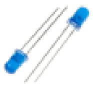
		Component	Quantity	Cost (in TL)
2		9V Battery Head	1	1,23
3		Jumper cable	1	14,37
4		Servo Motor	2	36,95
5		Push Button	1	0,62
6		Capacitor	1	0,21
7		9V Battery Holder	1	5,44
8		22uF 50V Capacitor	1	0,21
9		Green Push Button	2	2,46
10		3mm Red Led Package	1	2,05
11		Ardunio Breadboard Set	1	205,26
12		5 mm Blue Led Package	1	2,05

Table 8.1. Project’s Cost

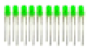
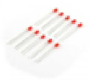
25		Push Button	1	1,44
26		5mm Transparent Red Led	1	2,05
27		Ultrasonic Distance Sensor	1	8,52
28		5 mm Green Led	1	2,05
29		Blue Push Button	2	2,46
30		3 mm Transparent Red Led	1	2,05
31		GY-906 Digital Infrared Thermometer Sens	1	69,9
32		A4 TECH PK-635G ANTI-REFLECTION WEBCAM	1	167,99
33			Compenents' Cost	799,6
34			Shipping Cost	43,8
35			Total Cost	843,199

Table 8.2. Project’s Cost

Chapter 9

9. CONCLUSION

In conclusion, This EMU graduates idea of the Smart Gate is an automated prototype designed to reduce or eliminate physical contact between people when taking temperature readings and storing the corresponding images of license plates or people in a data-base and use this data obtained in order to significantly lower the risk of infection during this COVID-19 global pandemic in an efficient and cost-effective manner. Especially when it is hoped that starting from the next academic year 2021/2022 the education will resume and continue as face to face partially, with online implementation too. This Project will enhance the safety and will make the environment more comfortable for the staff/student interaction.

This will keep humans safe and in addition it will enable a huge number of people to access public places in a way that is contained and up to health code guidelines and regulations.

Arduino and open CV image processing will be our great asset in this project and our primary hardware will be the temperature and an ultrasonic with a DC battery (can be laptop powered). We followed all the engineering and ethical principles whilst working on this proposal. We managed our time correctly and efficiently in order to stay productive while working on the proposal since good time management is a vital skill in engineering. Overall, a lot have been learnt off this research and project, and it helped get a very useful grasp on the field of automation and control systems, gaining knowledge about Arduino and image processing which is a very useful component. Learning outcomes on how to work with very useful hardware such as ultrasonic and temperature sensors.

This project includes image processing, Arduino programming, Wire soldering and connecting, UNO board, batteries so we are glad we could mix all electrical and electronics aspects into one, including control systems, robotics, etc.

9.1 Future Works

This project was made by us in hopes of serving the engineering world with a useful design. The implementation of this prototype is not that complicated, and it is very likely that it will get developed more in the future.

One future development plan can be to improve project by having a way back after entering the parking lot, university, etc. since there is a record of the time, we can have another booth on the way out with another camera that detects the plate numbers and synchronized with the other camera to detect the same car and find the row of that plate and extract the time it has entered and the time it left and find the difference and according to a pre-defined pricing it can produce an invoice receipt that can be used to charge people and collect funds automatically.

Another upgrade can be Voice recognition can be implemented and it will allow users to verbally control this automated smart gate design.

Currently, infra-red temperature guns are being used in the world to monitor the pandemic situation at most security checkpoints such as airports, entrances to corporate buildings, schools, hospitals etc. . But with this design, in the future, with an advancement in our codes, this technology can be improved and further developed it can also be widely and conveniently used on a small scale for homesteads for use in many other sectors in our daily lives . All as an automated system with minimum human intervention.

REFERENCES

[1] A, C. Soh., S, A. Ahmad., A, J. Ishak and K, N, A. Latif., "Development of an adjustable gripper for robotic picking and placing operation", International journal on smart sensing and intelligent systems, vol. 5, no. 4, December 2012.

[2] P, S. Ramaiah., M, V. Rao., G, V. Satyanarayana., "A Microcontroller Based Four Fingered Motor", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.2, April 2011.

[3] Line Follower with Circuit Daigram Explanation and Applications. (2019). Re- trieved 21 December 2019, from https://www.elprocus.com/line-follower-robot-ba- sics-controlling/

[4] A, A. Galadima. (2014, September). Arduino as a learning tool. In Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on (pp. 1-4). IEEE. Retrieved from: http://mypractic.com/lesson-2-arduino-uno-r3-board-description-characteristics/

[5] Arduino Uno R3 Boards (n,d) Retrieved from: https://www.electroniclinic.com/arduino-l298n-motor-driver-control-tutorial-speed-direction-pwm/

[6] n.a. (2016, May 27). Micro Servo motor SG90. Retrieved from https://wiki.eprolabs.com/index.php?title=Micro_Servo_motor_SG90

[7] Fahad, E. 2016: L298N Motor Drive Control, Pennsylvania College of Technology, USA

[8] Krishna, P. 2018: Infrared Receivers, Parde University Northwest, USA Retrieved from: http://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/ IR

[9] n.a. (n.d). Robotics design process. Retrieved from https://www.galileo.org/robotics/design.html

[10] Haaren, V. (2012, January 29). Design of a I2C circuit. Retrieved from https://www.techunited.nl/media/files/humanoid/IngridvanHaren_BEP2012_Design_of_a_robot_arm.pdf

[11] Yi, Z. K. Razali, Z. B. (2012). Design, analysis and fabrication of robotic arm for sorting of multi-materials. Retrieved from https://www.researchgate.net/publication/281642602_Design_analysis_and_fabrication_of_robotic_arm_for_sorting_of_multi-materials

[12] Face Recognition using openCV and Python: Abeginner's guide. Retrieved 3 August 2017, from https://www.superdatascience.com/blogs/opencv-face-recognition

[13] Python 3.6.0 (64-bit). Retrieved 23 December 2016, from https://www.filehorse.com/download-python-64/28830/

[14] n.a, (n.d). About openCV. Retrieved from https://opencv.org/about/

[15] Coronavirus disease COVID-19 pandemic. (2020, March 12). UNDP. https://www.undp.org/content/undp/en/home/coronavirus.html

[16] Non-contact infrared smart gate designs. (2020, April 23). U.S. Food and Drug Administration. https://www.fda.gov/medical-devices/general-hospital-devices-and-supplies/non-contact-infrared-smart gate designs

[17] Use of handheld infrared smart gate designs in COVID-19 pandemic for mass screening: Understanding its implications through a case report. (n.d.). PubMed Central (PMC). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7773056/

[18] n.a, (n.d). European Standards. Retrieved from https://www.cen.eu/work/products/ENs/Pages/default.aspx

[19] n.a, (n.d). Electrical and Electronics Engineers, Inc. (IEEE). Retrieved from https://ih-smarkit.com/products/ieee-standards.html

[20] n.a, (n.d). Electrical Installation. Retrieved from https://www.theiet.org/publishing/iet-standards/electrical-installation/

[21] Silva, S. M., & Jung, C. R. (2018). License plate detection and recognition in unconstrained scenarios. *Computer Vision – ECCV 2018*, 593-609. https://doi.org/10.1007/978-3-030-01258-8_36

[22] Bairwa, S. (2021, May 28). *Introduction to ANPR: Cropping a license plate from a car image [Part 2]*. Hacker Noon. https://hackernoon.com/introduction-to-anpr-cropping-a-license-plate-from-a-car-image-part-2-0s1c34us

[23] Dang, M. T. (2020, August 19). *Characters segmentation and recognition for vehicle license plate*. https://dangminhthang.com/computer-vision/characters-segmentation-and-recognition-for-vehicle-license-plate/

[24] *OpenCV: Image processing (imgproc module)*. (n.d.). OpenCV documentation index. https://docs.opencv.org/master/d7/da8/tutorial_table_of_content_imgproc.html

APPENDIX

A-1. Arduino Code

```
#include <Servo.h>
#include <Adafruit_MLX90614.h>

#define led_g 9
#define led_r 8
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double distance;
double temperature;

Servo servo_3;
Adafruit_MLX90614 mlx = Adafruit_MLX90614();

float getDistance(int trig,int echo){
  pinMode(trig,OUTPUT);
  digitalWrite(trig,LOW);
  delayMicroseconds(2);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  pinMode(echo,INPUT);
  return pulseIn(echo,HIGH,30000)/58.0;

}
void setup(){
  servo_3.attach(3);
  servo_3.write(0);
  delay(0.5);
```

```
  pinMode(led_g,OUTPUT);
  pinMode(led_r,OUTPUT);
  mlx.begin();

}
void loop(){
 distance=getDistance(7,6);
 temperature=mlx.readObjectTempC();
 if(distance<10){
if((temperature)<(38)){
    digitalWrite(8,LOW);
    delay(15);
    digitalWrite(9,HIGH);
    servo_3.write(90);
    delay(15);
  }
  else{
    digitalWrite(8,HIGH);
    servo_3.write(0);
    delay(15);
    digitalWrite(9,LOW);
    delay(15);
    }
 }
 else{
    servo_3.write(0);
    delay(15);
    digitalWrite(9,LOW);
    digitalWrite(8,LOW);
 }
}
```

A-2. Car License Plate Recognition Code

```python
from keras.models import model_from_json
from local_utils import detect_lp
import matplotlib.image as im
import numpy as np
import pytesseract
import functools
import datetime
import imutils
import cv2

# License Plate Detection

cap = cv2.VideoCapture(0)
ret, img = cap.read()
cv2.imshow('Original Image', img)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

with open('wpod-net.json' , 'r') as json_file:
    model_json = json_file.read()
model = model_from_json(model_json, custom_objects={})
model.load_weights('wpod-net.h5')
wpod_net = model

vehicle = img / 255
ratio = float(max(vehicle.shape[:2])) / min(vehicle.shape[:2])
side = int(ratio * 256)
bound_dim = min(side, 608)
_ , LpImg, _, cor = detect_lp(wpod_net, vehicle, bound_dim, lp_threshold=0.5)

img = np.float32(LpImg[0])
img = cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
cv2.imshow('Car License Plate', img)
im.imsave('plate.jpg', img)

# License Plate Character Recognition

img = cv2.imread('plate.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow("Grayscale", gray)

blur = cv2.GaussianBlur(gray, (5, 5), 0)
cv2.imshow("Gaussian Blur", gray)
print(type(blur))
thresh = cv2.adaptiveThreshold(blur, 255,
                               cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY_INV, 45, 15)
cv2.imshow("Threshold", thresh)
```

```python

_, labels = cv2.connectedComponents(thresh)
mask = np.zeros(thresh.shape, dtype="uint8")

total_pixels = gray.shape[0] * gray.shape[1]
lower = total_pixels // 70
upper = total_pixels // 20

for (i, label) in enumerate(np.unique(labels)):

    if label == 0:
        continue

    labelMask = np.zeros(thresh.shape, dtype="uint8")
    labelMask[labels == label] = 255
    numPixels = cv2.countNonZero(labelMask)

    if numPixels > lower and numPixels < upper:
        mask = cv2.add(mask, labelMask)

cv2.imshow("mask advanced", mask)

'''# uncomment this block to show character segmentation

cnts, _ = cv2.findContours(
    mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
boundingBoxes = [cv2.boundingRect(c) for c in cnts]

for c in boundingBoxes:
    x, y, w, h = c
    cv2.rectangle(gray, (x, y), (x+w, y+h), (255, 255, 255), 2)

cv2.imshow("mask blobbed", gray)
'''

text = pytesseract.image_to_string(mask, config='--psm 11')

f = open("data.txt", "a")
f.write("\nCar License Plate: %s\t Date/Time: %s\n" %(text, datetime.datetime.now()))
f.close()

cv2.waitKey(0)
```