RMIT UNIVERSITY

Technical Report

Covid19 Safety Protocol

Assessment 3

COSC2500 Intro to Computer Systems and Platform Technologies

Lecturer: Anna-Lyza Felipe Sanchos & Sam Goundar

Team:

Nguyen Hoang Khanh Duc - s3926243
Nguyen Thi Ha Thu - s3927104
Nguyen Nhat Hoang - s3926555
Bui Quang Kien - s3928848

Table of Contents

I, Group members' roles

No	Name	StudentID	Roles
01	Nguyen Hoang Khanh Duc	s3926243	1.Researching ideas for project 2.Composing technical report 3.Presenting project features
02	Nguyen Thi Ha Thu	s3927104	1.Researching ideas for project 2.Chronicling and supervising work progress 3.Composing technical report
03	Nguyen Nhat Hoang	s3926555	1.Researching ideas for project 2.Building the project 3.Presenting project features
04	Bui Quang Kien	s3928848	1.Researching ideas for project 2.Designing circuit schema 3.Assembling components 4.Presenting circuit board schema

II, Project Description

1. Overview

Project title: Covid19 Safety Protocol

Our project is a device that requires people to pass 3 checkpoints before entering the school. The 3 checkpoints include:

- Closeness detection – places visitor to the optimal distance for face-scanning
- Face mask detection – scans visitor's face for mask
- Hand sanitizer – dispenses hand soap to visitor

The checkpoints will be marked as completed when the user stood at the correct distance, wore a mask and used the hand sanitizer. When all the 3 checkpoints are fulfilled, the school's door lock will be opened.

2. Materials

No	Device & Components	Amount	Description	Roles in the project
01	Arduino Uno	01	The Arduino Uno is an ATmega328P based microcontroller board. It has 14 digital I/O Pins, 6 of which can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53R0,USB connection, power connector, ICSP header, and a reset button (Arduino 2021).	Connect all the other components of the projects to each other and allow them to exchange real-time data
02	HRC-S04 Ultrasonic sensor	02	The main parts of the HCSR04 ultrasonic distance sensor are the two ultrasonic transducers. One acts as a transmitter that converts electrical signals into 40 kHz ultrasonic sound pulses. The other - the receiver receives the transmitted pulse. When the receiver receives the transmitted pulse, it generates an output pulse, the width of which can be used to determine the distance traveled (Lastminuteengineers 2021).	1/ Check whether the person is standing close enough for the face mask detection 2/ Check whether the person's hand is close enough for the automatic hand sanitizer
03	LED	02	A light source that emits light when an electric current flows through it. To turn on the LED, the Arduino needs to send a HIGH signal to one of its pins. To turn off the LED, it needs to receive a LOW signal from the pin (Scott C 2021).	Throughout the project, the green light will flash to indicate that the checkpoint is completed and the red light will remain on when the checkpoint is not completed

04	Mini Servo	01	A servo is a component that has an output shaft. When a coded signal is called, the shaft can be shifted to a specific angular position (Seattle Robotics Society 2021).	The mini servo represents the school's door lock, which will be opened when 3 checkpoints are completed
05	ESP32-CAM	01	ESP32CAM is a small, low-power camera module based on ESP32. It comes with the OV2640 camera and has a built-in TF card slot (PiShop 2021).	Scan and detect whether the person is wearing a mask or not
06	USB to TTL CH340G	01	A small USB to TTL serial tool, using the CH340G chip. It can be used to connect some serial devices to PC via USB port	Connect the ESP32-CAM to computer to upload the code
07	Hand-sanitizer	01	A substance that can kill germs on the skin	The pump of the sanitizer will be pushed once the servo detected that there is a nearby object
08	LCD I2C 1602A	01	The LCD I2C included two separate components: the LCD and the I2C LCD adapter. The LCD is a 16x2 display with backlights that can show 16 characters in each row. The I2C adapter is an expander chip that transfers the I2C data from the Arduino to the data required by the LCD (Lastminuteengineer 2021).	The LCD helps display the address of the camera, the users' distance, and the mask scanning results. The I2C adapter helps reduce the number of pins that are required when connecting an LCD to Arduino.
09	Servo Futaba MG996	01	The Servo Futaba MG996 is an upgraded version of s MG995 in terms of speed, traction, and accuracy. The Servo Futaba's stall torque is 11kg (TowerPro 2021).	The servo will be used to press the hand sanitizer.

Developing environment

No	Name
01	Arduino IDE

3. Methods

3.1. The procedure

To come up with the ideas for the project, each of the members of the team researched Arduino-related projects and described the details on how to build them. After gathering everyone's input, we debated and chose the best idea for the project by evaluating which one is the most feasible to build. The two ideas that stand out the most were the "Face Mask Detector" and the "Automatic Hand Sanitizer" since they both suit the Covid-19 topic. Moreover, as some of our team members plan to minor in Artificial Intelligent we also wanted to choose an idea that uses AI. In the end, we decided to merge two ideas into one and build our project called Covid-19 Safety Protocol - a device that checks face masks and allows users to use hand sanitizer.

After coming up with the ideas, we started preparing for our project. We divided our teams into two groups - Kien and Duc were in charge of setting up and getting familiar with the circuit board, while Thu and Hoang were in charge of learning the code from Youtube tutorials. The first group set out and bought all the missing components - the ESP32 Camera, Ultrasonic sensor, the hand sanitizer, and the second group learned the basics of C/C++. During this process, we got the chance to acquire knowledge about Arduino and programming.

As soon as the components arrived, we did a test run according to the tutorial for our project, which did not work. We then wrote our own code to test individual components. Since we did not have much experience with Arduino, it took a long time for us to figure out where the errors came from. A lot of factors could be the root of the issues we faced: wrong syntax, semantic errors, wire defects, or damaged components. As a result of such, we had to prepare some more components since some of them are defective.

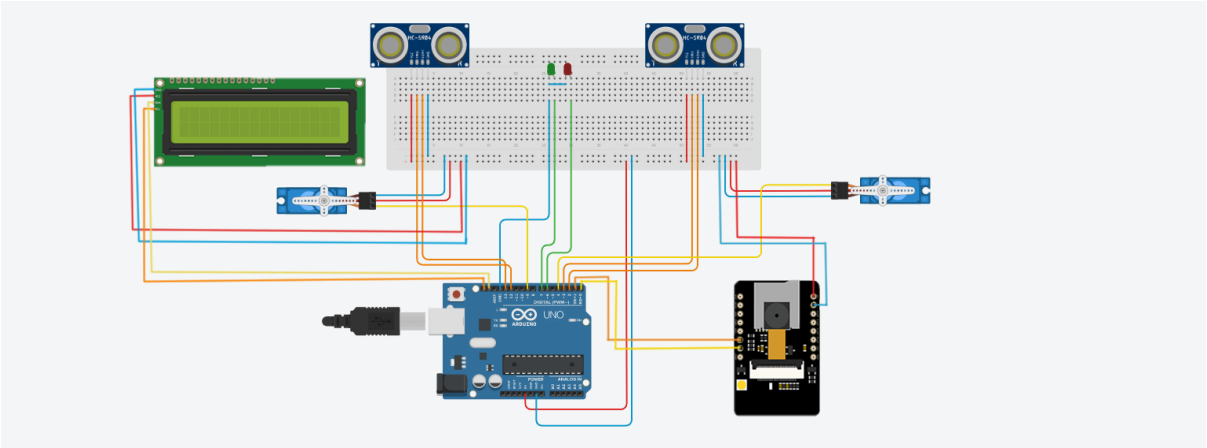
When it came to implementing our project, we spent a lot of time debugging and resolving all of the problems. In this phase, we set up each component individually and then connected everything together after making sure that the components work. The first component we worked with was the ESP32 Camera. With this component, we had to upload our face mask detection algorithm to the camera and then display the IP address to the LCD screen. We struggled with the camera not allowing us to compile and upload the code, and eventually, we found out that there was a problem with our board manager version installed on Arduino IDE. We had to downgrade our board manager version to 1.0.3, which only worked with Windows while some of our team members were programming on macOS. We managed to upload our code to the ESP32 successfully, but somehow the camera did not print out the IP address to

the serial monitor on the Arduino IDE. According to Arduino's documentation, it turned out that when we opened the serial monitor in the IDE, we had to select the same baud rate defined in the code. After fixing the issue and making sure that the face mask detection algorithm worked, we then connected the camera to the LCD so that it can display the IP address. The LCD screen was still not working then, so we were forced to connect our camera's and Arduino's Serial bus to the computer to assess them separately.

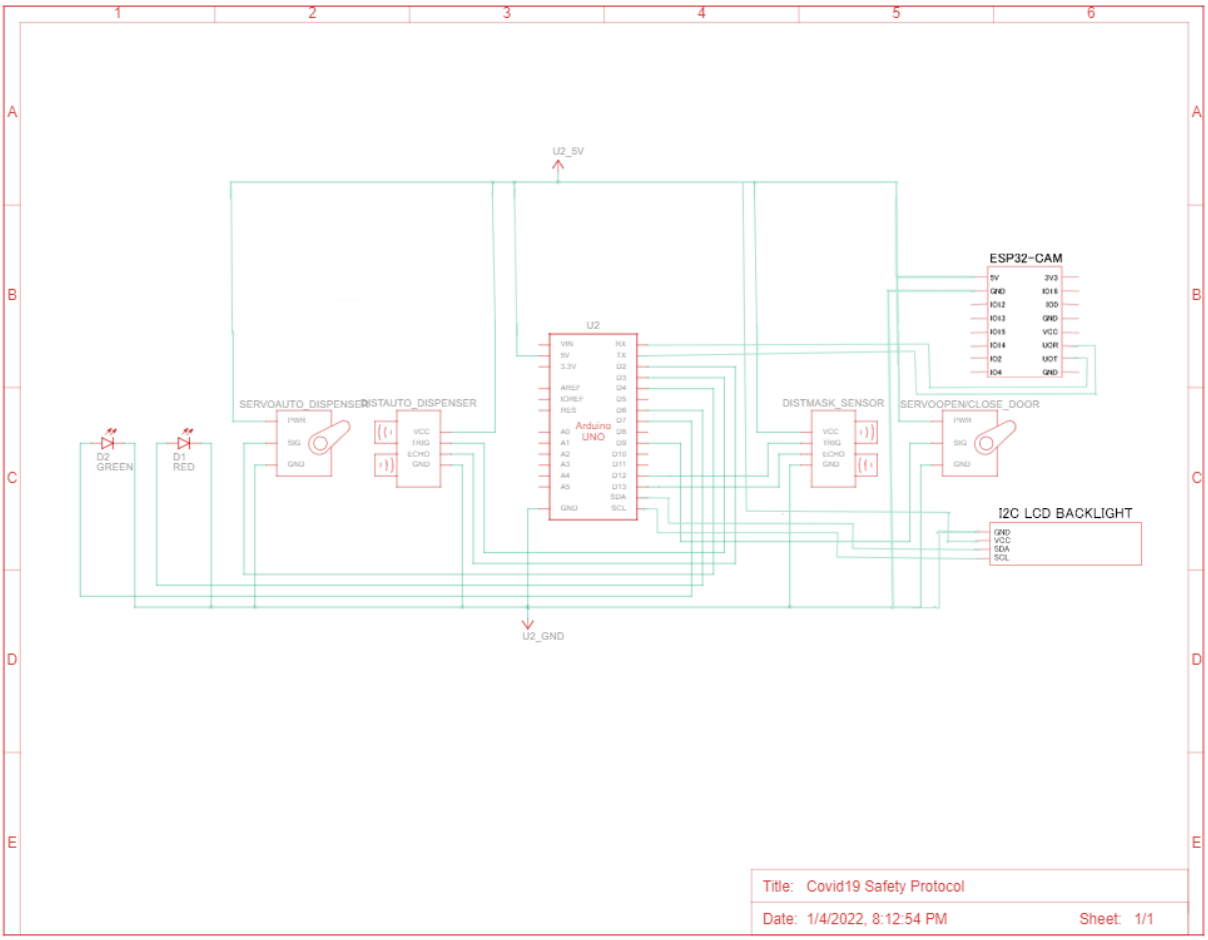
The LCD was glitchy and when the screen did connect, it did not display any content. We searched for our problem online and found out that for the output on the LCD screen to be stable, the I2C must be soldered into the LCD. We decided to buy a new LCD screen that already had the I2C soldered into it. To fix the display problem, we had to use the screwdriver to adjust the potentiometer, which helped change the LCD's brightness. After setting up the ESP32 and LCD, we moved on to the sensor to check whether the person was standing close enough to scan the face or not. During this part, we did not encounter any issues with implementing the sensor. The last component we worked with was the servo to press the hand sanitizer. The mini servo we had can only handle 1.7kg with its shaft, which was not strong enough. We then changed to the Servo Futaba MG996 that can handle up to 11kg.

As we had all the components running, we proceeded to combine all the codes and hardware of the components. For the coding part, in the beginning, we encountered many syntax errors which took time to debug and learn at the same time. The Arduino IDE was also challenging as it did not support autofill, shortcuts, and autosave. This results in an event where we forgot to save our one-week worth of code and had to redo everything again. We also met with a problem when uploading the code to Arduino. We were not aware that when uploading the code, we must unplug any components that are connected to TX and RX. Since they are used in microcontrollers for uploading the sketch, if we connect any components to the TR and TX during the upload, that component will also receive the data for the microcontroller. This causes the component and the microcontroller to misinterpret the data and stops the sketch from being uploaded to the Arduino, in some cases might even end up damaging the component. For the hardware implementation, we did face some issues related to managing the wires and the position of the components, but in the end, we succeeded in combining all the components into one working product.

3.2 Circuit Diagram & Schematic



Circuit Diagram (Made with Tinkercad and Adobe Photoshop)



Schematic (Made with Tinkercad and Adobe Photoshop)

3.3 Code

Link: https://drive.google.com/file/d/1QaZKUQ9_dtj5aeiuvvEpDVNv0kMwc7tw/view

III, Conclusion

Arduino is a great example of the Internet of Things and is suitable for the Covid 19 topic today since our product could be automated and could test to see if the person is following the Covid 19 procedure of protecting themselves, and help people to be aware of the protection needed in this dangerous situation. Since this is the automation function, human attention would no longer be necessary while the product could perform most of the essential tasks for the Covid checking.

Through this project, our group has learned a lot and has leveled ourselves up by having the opportunity to explore both technical and soft skills. In terms of technical skills, we learned how to deal with embedded systems by learning basic programming and algorithms with C/ C++ to code the Arduino system. Aside from programming skills, we also gained knowledge in basic electrical engineering and circuits designing which promoted our understanding of computer systems. Soft skills are the second things we have progressed so far. By developing our team working skills, we managed to build up the product and get through all the challenging parts. Time management skills are great contributing factors to our success of the project since we all have many different projects from other classes. Moreover, an opportunity to work on a computer system project is a good means to sharpen our critical thinking through discussing, debugging, and assembling the final project.

In conclusion, we are proud and satisfied with the effort that has been put into this project, and all the necessities relating to the technical part that is needed for our future career. If we had opportunities and more time for this project, we would like to improve our product by adding relevant functions, one of such can be the ability to test the human temperature to test whether that person would need to get a Covid-19 test since body temperature is the one of the most common Covid-19 symptoms.

VIII, References

Arduino 2021, ‘Arduino/Genuino UNO’, viewed 9 Jan 2022, <https://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardUno>

Lastminuteengineers 2021, ‘How HC-SR04 Ultrasonic Sensor Works & Interface It With Arduino’, viewed 9 Jan 2022, <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/>

Scott, C 2021, ‘GETTING STARTED WITH THE ARDUINO – CONTROLLING THE LED (PART 1)’, viewed 9 Jan 2022, <https://www.circuitbasics.com/arduino-basics-controlling-led>

Seattle Robotics Society 2021, ‘What’s a Servo?’, viewed 9 Jan 2022, <https://phas.ubc.ca/~kotlicki/Physics_319/servos>

PiShop 2021, ‘ESP32-CAM, Camera Module Based on ESP32’, viewed 9 Jan 2022 <https://www.pishop.ca/product/esp32-cam-camera-module-based-on-esp32/>

Lastminuteengineers 2021, ‘Interface an I2C LCD with Arduino’, viewed 9 Jan 2022, <https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/>

TowerPro 2021, ‘MG996R’, viewed 12 Jan 2022, <https://www.towerpro.com.tw/product/mg996r/>