Vehicle Fever Detection System

ECE 445 Design Document

Team 67 - Rahul Rajkumar, Vinayak Dhanawade, Vishnu Rathnam

ECE 445 Project Proposal - Spring 2021

TA: Dean Biskup

1. Introduction:

1.11 Objective:

A problem that is frequently encountered due to the COVID 19 pandemic is that of spread through rideshare services such as Uber and Taxis. This is primarily due to the fact that vehicles are a closed environment with relatively low ventilation and air circulation. This is even more true in taxis and rideshare vehicles because they are used by a large amount of different people every day. Our solution to this problem is to implement a system that can be mounted on the interior door of vehicles that extends an MLX90614 IR thermometer [1] out the window to check riders temperatures before they enter the vehicle. The system will then alert the driver via a digital display as to whether the current passenger has a fever and give the specific temperature. The system will also feature an additional IR thermometer near the driver just in case the passenger does not believe the reliability of the sensor so that the driver can check their own temperature to prove that the sensor works. The driver can then decide on a plan of action with this information in order to effectively limit the spread of COVID 19 and ensure both their safety and that of future passengers. This system also has applications in the future after the pandemic in that drivers can choose whether to accept riders that are sick in general based on if they have a fever, a common flu symptom.

1.12 Background:

The issue of COVID 19 being transmitted through rideshare and taxi services is pressing due to the fact that the drivers and patrons are put at a high risk of transmitting COVID 19 by being in close proximity to others in an enclosed space. This is exacerbated by the users having to touch surfaces that are frequently used and touched by many others. While masks suffice to prevent spread in open spaces, their effectiveness is severely limited in an enclosed space with limited ventilation like a car. According to a study by the Norweigan Institute of Public Health [2], Taxi and Rideshare drivers are among the highest risk groups for getting COVID 19 due to their constant proximity to many different people every day. Since this risk is so high we hope our solution will provide both those drivers and the passengers that interact with them a higher level of safety than that of simply wearing a mask.

1.2 Visual Aid:

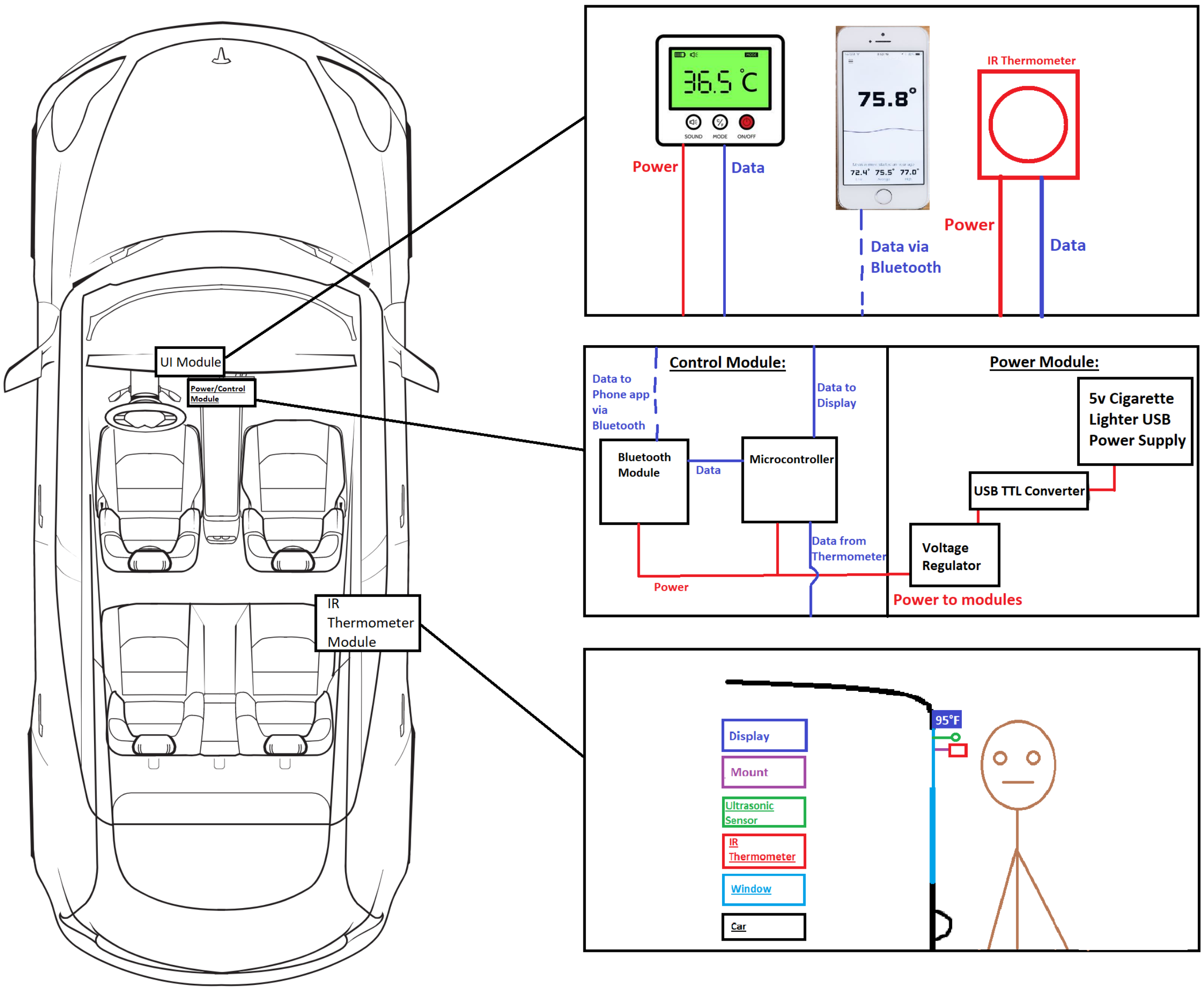

Figure 1: Visual aid for physical layout of components

1.3 High Level Requirements:

- The thermometer must be placed in a position in which it is accessible by the patron from a standing position with a maximum of 20° of leaning.
- The temperature readings must be at least 90% accurate with a +/- .5 degree celsius variance and provide info to the driver and the potential passenger on the

display on whether the passenger is at a dangerous temperature (above 37.5° C as defined by the national institute for health research [3]) for various infrared thermometers.

- Ultrasonic sensor must detect if the passenger is standing within valid distance from IR thermometer (4-6 inches) and must relay this information to the microcontroller to be displayed on LEDs to both the driver and potential passenger
- Bluetooth latency must be under 30 sec between the bluetooth module and the phone

2. Design:

2.1 Block Diagram :

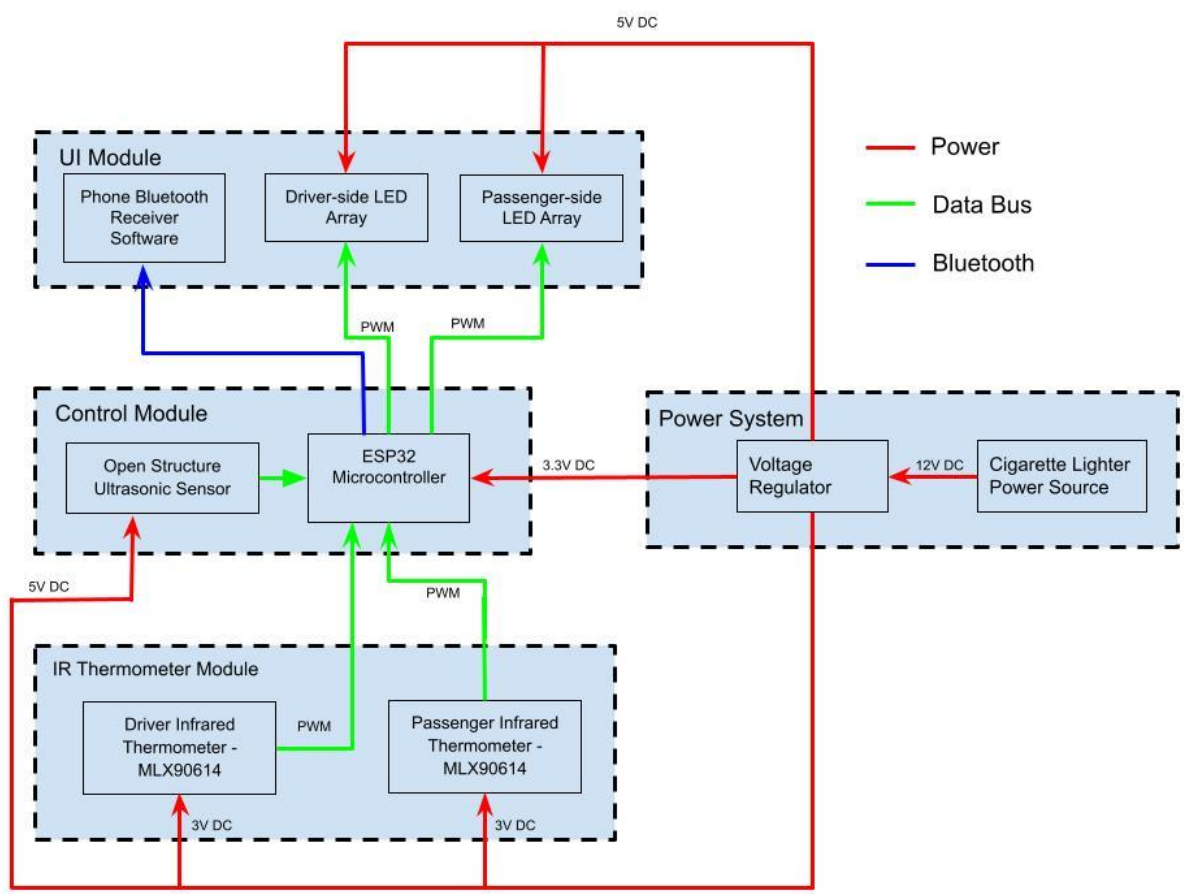

Figure 2 : Block diagram of entire system

2.3 Subsystem Descriptions / R&V Tables:

Power Subsystem : The power system of our project consists of two components, the cigarette lighter power source from the car, and the voltage regulator. The power source will be 12V DC, and this power will be delivered to the voltage regulator. The voltage regulator will bring this output voltage down to a manageable level and will route the power to the other components of our system. That is, it will power the UI , control, and thermometer modules.

UI Subsystem : The overall system must be able to deliver accurate results from the control module to the driver. These results must be delivered in the form of LEDs which represent the suggested course of action for the driver and passenger (the results will be displayed to both the driver and the passenger). While minimal, one of the LEDs will light up to recommend the driver to reject or admit the potential passenger, based on the data passed by the microcontroller. The driver will also receive this information via bluetooth on their phone. The data will be sent via bluetooth from the ESP32 microcontroller, to the driver's phone on receiver software. The passenger-side LEDs will notify the passenger if he/she is within the correct distance from the thermometer in order to get an accurate reading. Depending on the distance between the thermometer and the passenger, as measured by the ultrasonic sensor, the green LED will light up to represent correct positioning of the passenger, or the yellow/red LEDs will indicate if the passenger needs to move closer or farther.

IR Thermometer Module :
The thermometer module will contain two IR thermometers (MLX90614) [1] that will read the potential passenger's temperature as well as the driver's temperature and send that data to the microcontroller. The passenger thermometer reading will be used by the control module to determine if the passenger should be allowed into the vehicle, while the driver side thermometer will be to confirm that the reading for the passenger is accurate.

Control Module :
This module will house the ESP32 microcontroller that will process the data from the thermometer and produce results to be displayed on the UI subsystem for the driver and passenger to view. The microcontroller will be powered by the 3.3V output from the voltage regulator in the power module. This module also contains an open structure ultrasonic sensor that will detect if the passenger is too close or too far or properly positioned from the IR Thermometer in order to get an accurate reading. The control module overall serves as a middle man, as well as the core processing unit for the overall system. It links all three main components of the system together, and will

generate the recommended course of action (either to allow the passenger into the vehicle or to turn them away) for the driver. It will also generate the recommended change in positioning for the passenger to get an accurate temperature reading.

Thermometer Module : This module consists of the two IR thermometers

Requirement	Verification
1. IR Thermometers measure temperature with at least 90% accurate with a +/- .5 degree variance	Equipment : IR thermometers, arduino or redboard, USB power supply Conduct the following steps for both thermometers : 1. a. Connect arduino or redboard to USB power supply. Use wires to connect 3.3V power and 0V from arduino to VDD and VSS pins of IR thermometer, respectively. b. Connect SDA and SCL pin of arduino to SDA/PWM and SCL pins of IR thermometer, respectively, using 4.7kΩ Resistor to pull up to 3.3V c. On the Arduino software, run the MLX90614_Serial_Demo to be able to observe the temperature readings from the sensor in Fahrenheit d. Gather 15 people or cuts of meat (subjects), place them one by one in front of the IR thermometer for 10 seconds each and record in a table the average measurement from the 10 seconds of data gathered for each subject. e. Perform the previous step using a traditional armpit/under-tongue thermometer instead of the IR thermometer. It is not necessary to take the average of the measurements over 10 seconds for this thermometer as the final temperature can be found in about 5 seconds.

	f. Compare the IR thermometer reading to the traditional thermometer reading for each test subject to ensure that the 90% accurate with a +/- .5 degree C variance holds throughout.

Table 1 : Thermometer module requirements and verifications

Power Module : This module handles the power delivery to each of the other components. It derives the system's power source from a car's cigarette lighter, and regulates it to be sent to the other components.

Requirement	Verification
1. The voltage regulator must deliver power within a 3.3 +/- .5% range, from a 12V DC cigarette lighter source at 100 μA 2. The current delivered is between 0-150 mABig	1 a. Attach 33 kΩ resistor as load b. Attach oscilloscope across load c. Supply regulator with 12V DC d. Ensure output voltage remains between 3.3 +/- .5% range 2 a. Connect the output from the voltage regulator to the input voltage (VDD) of a constant-voltage test circuit b. Alter resistance values in this circuit until a consistent nonzero current at most 150mA is reached c. Once this current is reached, ensure that the voltage is within 5% of the 3.3-5V range using an oscilloscope.

Table 2 : Power module requirements and verifications

UI Module : This subsystem contains the deliverables for the driver. That is, it displays the temperature reading to the driver and the recommended course of action (either to allow or reject the potential passenger).

Requirement	Verification
1. The LEDs display the temperature results within 500ms +/- 250ms of a subject moving into position.	1. a. Using an ESP32 dev kit, a breadboard, an LED, a 220 ohm

2. The passenger-side LEDs display the status of the passenger's positioning immediately, within 500ms +/- 250ms of changing position. 3. Phone accurately (90% accuracy with a +/- .5 degree variance) receives temperature data via bluetooth and displays it within 30 sec.	resistor, and a 4.7k ohm resistor, connect the ESP32 dev kit to a computer running Arduino IDE and wire all of the components as shown in figure 4. b. As Figure 4 employs a different thermometer device than us, replace the depicted device with the MLX90614 device. To do so, connect the right and left most pin buses of the figure's thermometer to the VDD and VSS pins of the correct IR thermometer. Connect the middle bus (GPIO4) to the SDA pin of the thermometer. Connect the SLC pin of the thermometer to the CLK pin of the dev kit. c. Run code given in [6] such that when the temperature measured by the thermometer passes the threshold of 37.5° C, the LED lights up, and if the threshold is not passed, the LED stays off. d. Take 10 objects/people with temperatures above the threshold record their temperatures, and time the switching of the LED when presenting and removing these objects to/from the thermometer. e. Record these timings in a table and ensure that they stay within the desired range around 500ms 2. a. Using an ESP32 dev kit, an ultrasonic sensor, and an Arduino IDE, build the circuit depicted in figures 5 and 6 and plug the dev kit into a computer running the IDE b. Connect an LED to GPIO 2 and a resistor from the LED to power as shown in part of Figure 4 c. Upload the code found in [7] to the dev-kit, but modified such that once a once the distance measured by the ultrasonic sensor

	within the range of 4-6 inches, the LED turns on. d. For 10 different sized objects and using a ruler, place the object just outside the 4-6 inch range, and move it into range whilst starting a timer. Record the time taken for the LED to be triggered on in a table and ensure the timings fall within the desired latency. 3. a. Set up a circuit as described in the Control Module Verification #2, but without the LED. b. Update the code used in this verification to use BluetoothSerial.h library to open up bluetooth communication and relay the IR thermometer measurements to a connected device. c. After uploading the code, open the Serial Monitor at a baud rate of 115200. Press the ESP32 Enable button to enable pairing, and pair to the dev kit from the phone. d. On the phone, open the “serial bluetooth terminal” app. This is where the data will be received and seen from the dev kit. e. Gather 10 objects or people of known varying temperatures and present them to the thermometer one by one. f. For each object, record in a table the known temperature. Once the object is presented to the thermometer, start a timer and record the time taken for the results of this measurement to show on the phone. Record the temperature received from the dev kit in the phone receiver app. Ensure that these values fall within the desired accuracy standards and latency standards.

Table 3 : UI module requirements and verifications

Control Module : This subsystem contains the core processing unit of the overall system. It contains an ultrasonic sensor to control movement of the thermometer module, as well as a microcontroller that will process measurements, determine results and transmit them via bluetooth and LEDs.

Requirement	Verification
1. The ultrasonic sensor accurately (within 1 inch) detects whether passenger is 4-6 inches [5] +/- 0.5 inch away from the IR thermometer in order to get accurate reading or inform passenger to reposition 2. The ESP32 microcontroller must accurately determine the recommended course of action to the driver such that if the passenger's body temperature is above 37.5° C +/- .1° C, the passenger should not be allowed into the car.	1. a. Wire the ultrasonic sensor to an arduino according the Figure 3, and hook up the arduino to a power source (USB) b. Run code found in [5] but with the distance conditional values set to 4 and 6 inches such that if the distance detected between the sensor and object is between 4 and 6 inches, the green LED lights up, else, the red LED lights up. c. Gather 10 objects and/or body parts, and using a ruler, position each object 3, 3.75, 4.25, 5, 5.75, 6.25, and 7 inches away. Record the distances measured by the ultrasonic sensor via the arduino code output for each object in a table. d. Verify that for all test subjects, the sensor accurately measures the distance within the variation stated above 2 a. Using an ESP32 dev kit, a breadboard, an LED, a 220 ohm resistor, and a 4.7k ohm resistor, connect the ESP32 dev kit to a computer running Arduino IDE and wire all of the components as shown in figure 4. b. As Figure 4 employs a different thermometer device than us, replace the depicted device with

	the MLX90614 device. To do so, connect the right and left most pin buses of the figure’s thermometer to the VDD and VSS pins of the correct IR thermometer. Connect the middle bus (GPIO4) to the SDA pin of the thermometer. Connect the SLC pin of the thermometer to the CLK pin of the dev kit. c. Run code given in [6] such that when the temperature measured by the thermometer passes the threshold of 37.5° C, the LED lights up, and if the threshold is not passed, the LED stays off. d. Take 15 objects/people with varying temperatures, record their temperatures, and record the behavior of the LED when presenting these objects to the thermometer, ensuring that the LED switches appropriately.

Table 4 : Control module requirements and verifications

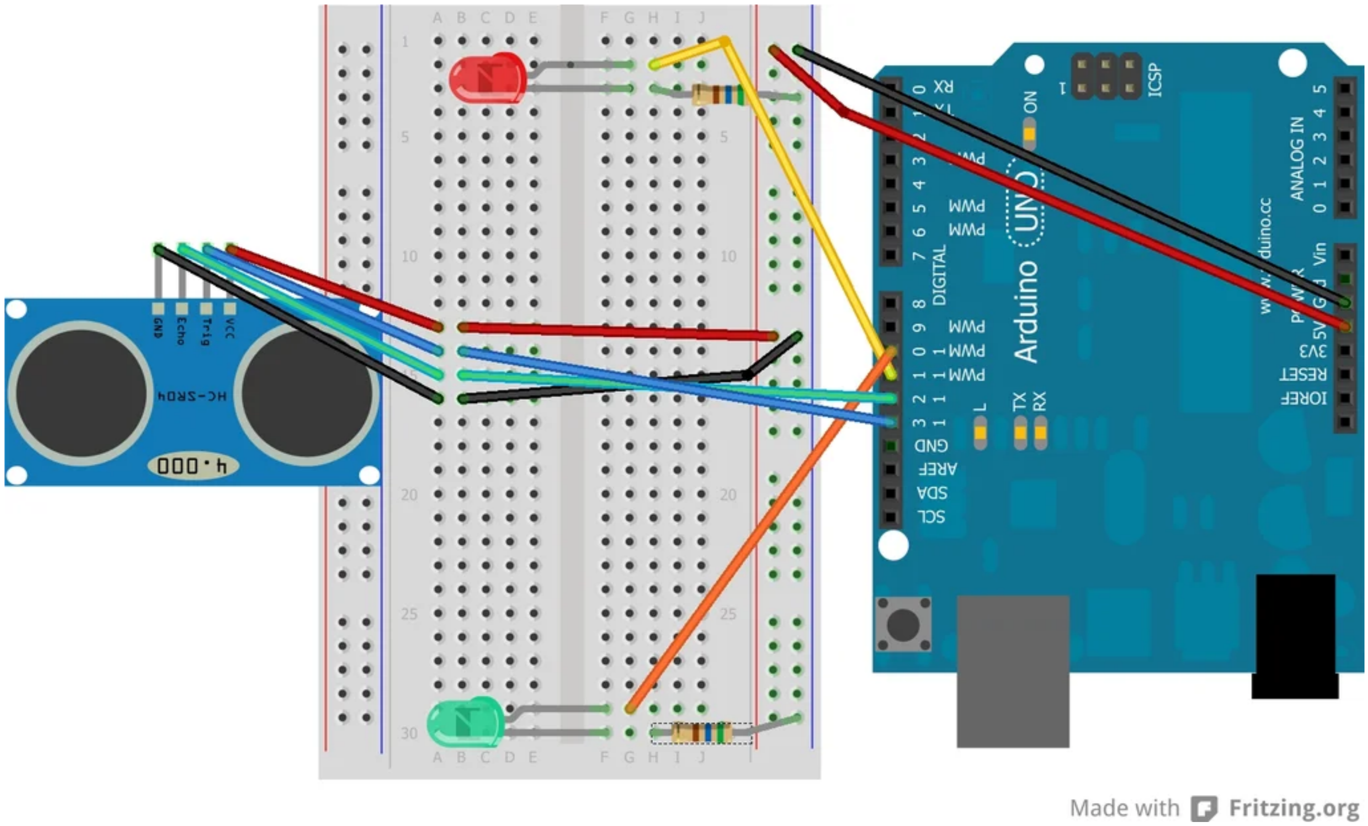

Figure 3 : Setup for control module verification #1 [5]

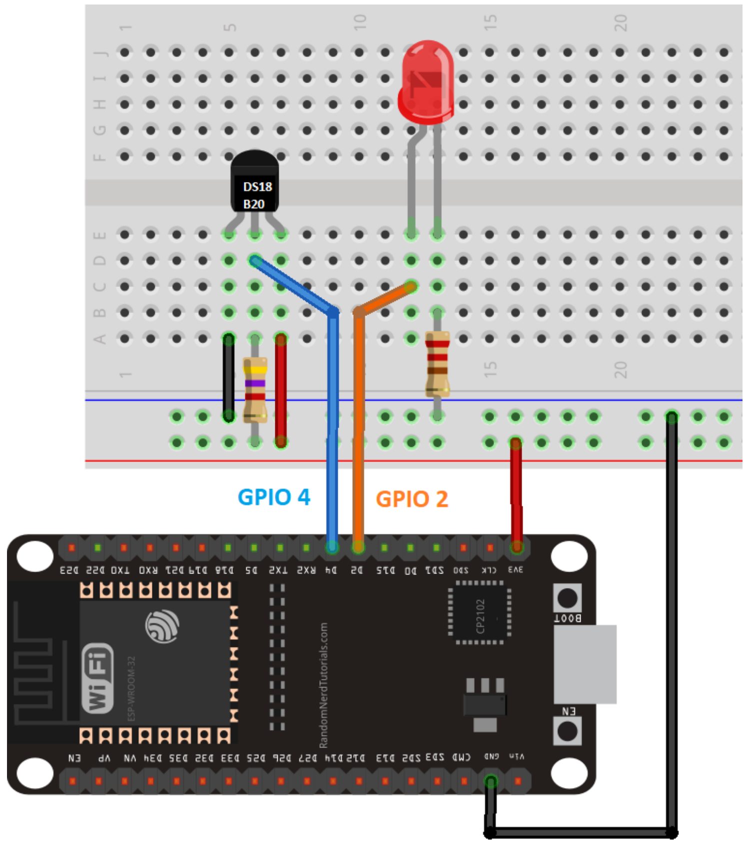

Figure 4 : Similar setup to that used in control module verification #2 [6]

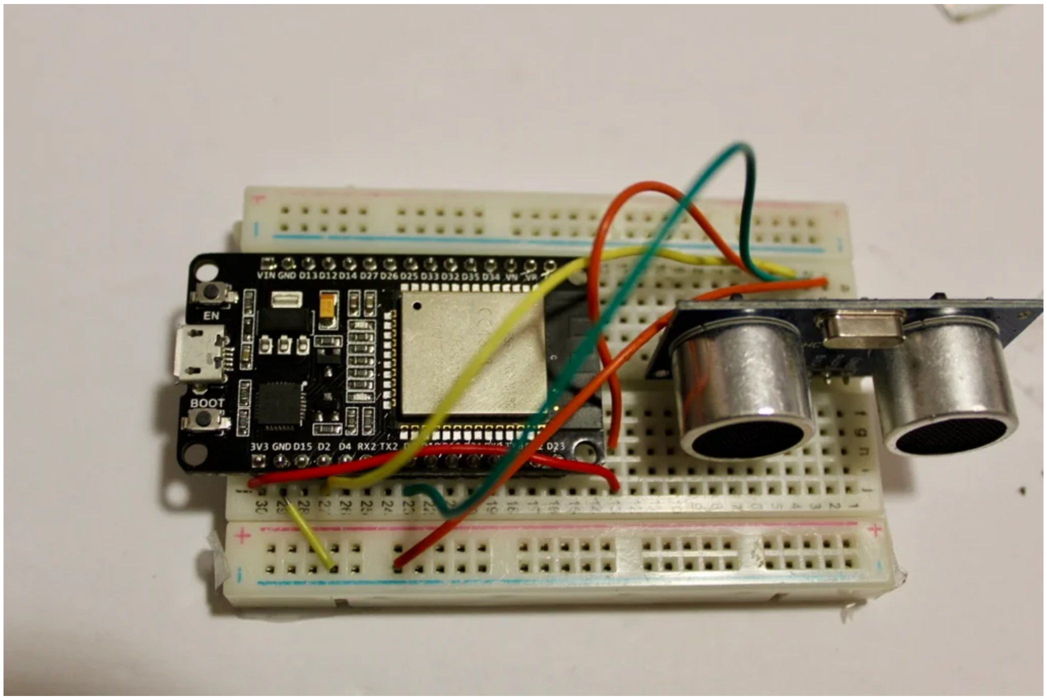

Figure 5 : Partial setup for UI Module Verification #2 [7]

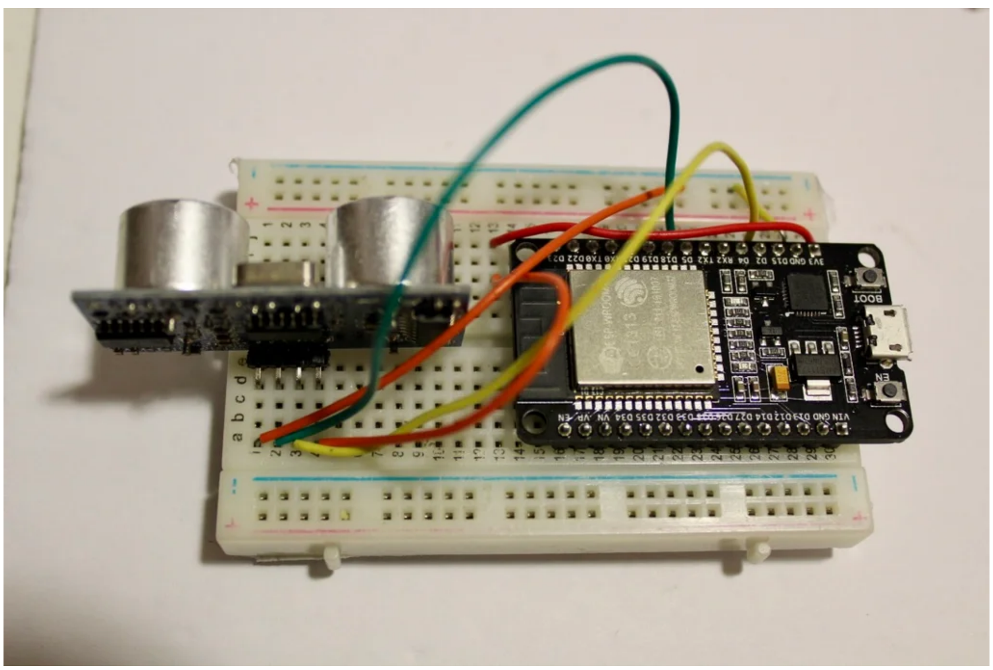

Figure 6 : Rear-view of partial setup for UI Module Verification #2 [7]

Points Summary :

Module Name	Requirement	Points
Thermometer Module	1. IR Thermometers measure temperature with at least 90% accurate with a +/- .5 degree variance	10
Power Module	1. The voltage regulator must deliver power within a 3.3 +/- .5% range, from a 12V DC cigarette lighter source at 100 μA 2. The current delivered is between 0-150 mABig	10
UI Module	1. The LEDs display the temperature results within 500ms +/- 250ms of a subject moving into position. 2. The passenger-side LEDs display the status of the passenger's positioning immediately, within 500ms +/- 250ms of changing position. 3. Phone accurately (90% accuracy with a +/- .5 degree variance) receives temperature data via bluetooth and displays it within 30 sec	15
Control Module	1. The ultrasonic sensor accurately (within 1 inch) detects whether passenger is 4-6 inches [5] +/- 0.5 inch away from the IR thermometer in order to get accurate reading or inform passenger to reposition	15

	2. The ESP32 microcontroller must accurately determine the recommended course of action to the driver such that if the passenger's body temperature is above 37.5° C +/- .1° C, the passenger should not be allowed into the car.	
Total		50

Table 5 : Demo point distribution summary

2.5 Tolerance Analysis:

One of the tolerance factors we will test is the bluetooth modules. We tested both the receiver and transmitter using over-the-air link testing. A bluetooth connection is established between the tester and DUT and the tester conducts a series of tests and measurements while maintaining connectivity. The main points we test are latency, power output, data rate and range. For our connection to be successful, the data has to be sent to the receiver phone under 500ms. The range of the device has to be at least 3m and the power output must not exceed the maximum power draw of 5V. The ESP32 microcontroller we are using has a built in bluetooth module and our results for are displayed below.

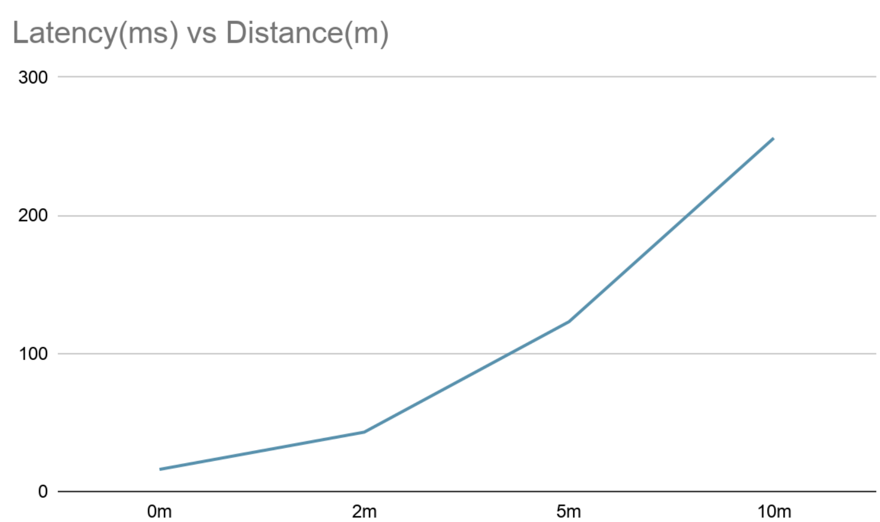

Figure 7: Latency in ms vs Distance in meters of the connection of the bluetooth connection

```
void loop() {
  unsigned long currentMillis = millis();
  // Send temperature readings
  if (currentMillis - previousMillis >= interval){
    previousMillis = currentMillis;
    sensors.requestTemperatures();
    temperatureString = String(sensors.getTempCByIndex(0)) + "C  " +  String(sensors.getTempFByIndex(0)) + "F";
    SerialBT.println(temperatureString);
  }
  // Read received messages (LED control command)
  if (SerialBT.available()){
    char incomingChar = SerialBT.read();
    if (incomingChar != '\n'){
      message += String(incomingChar);
    }
    else{
      message = "";
    }
    Serial.write(incomingChar);
  }
  // Check received message and control output accordingly
  if (message =="led_on"){
    digitalWrite(ledPin, HIGH);
  }
  else if (message =="led_off"){
    digitalWrite(ledPin, LOW);
  }
  delay(20);
}
```

Figure 8: Snippet of code we use to test the connection between the transmitter and receiver bluetooth devices

Our IR sensor is another tolerance factor. A paper from an IOP conference [9] shows the error and tolerance analysis of an ultrasonic sensor (HC SR-04) and an IR sensor (SHARP GP2Y0A21YKOF). Both these errors fall within our acceptable tolerance range. These sensors are also able to calculate the full range of our temperature reading use case within a reasonable amount of tolerance [10]. The testing of the MLX90614 IR sensor we are using also yielded similar results as shown below.

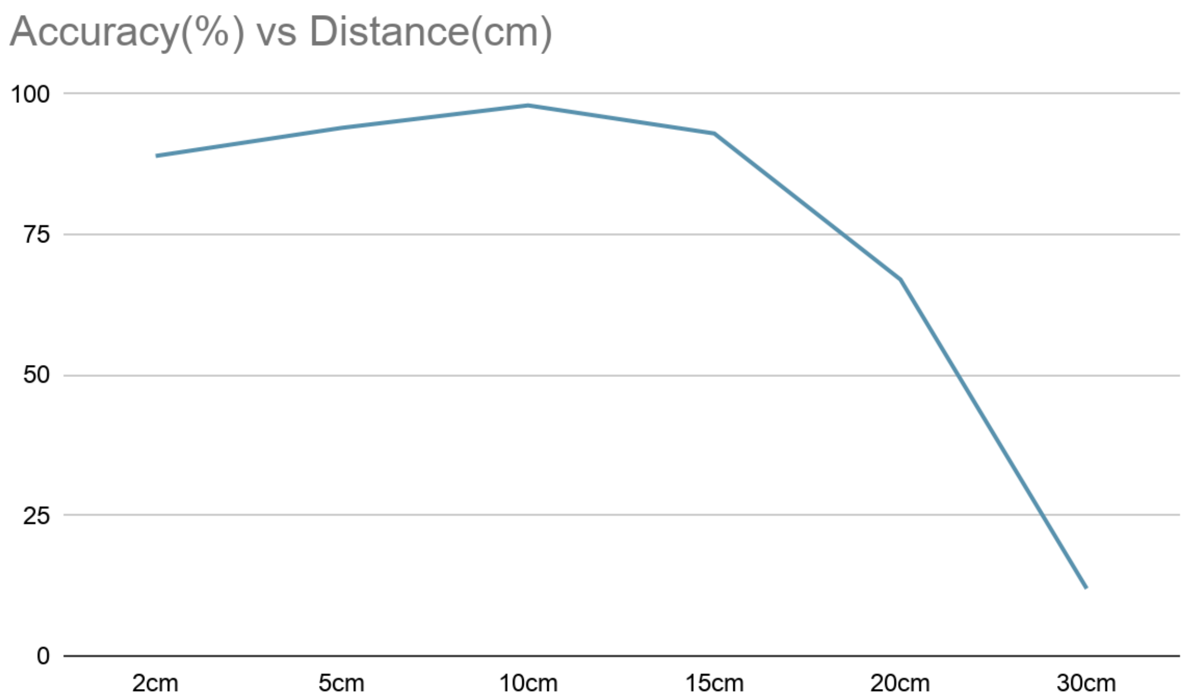

Figure 9: Accuracy of IR Thermometer vs Distance Measured From

We currently do not have an ultrasonic sensor on hand, but we will use similar testing to the IR sensor, calculating the true value beforehand as the control and continue to test readings from the sensor at different distances. The formula for accuracy we use is

$$\frac{actual - measured}{actual}$$

3 Costs & Schedule:

3.1 Cost Analysis:

We calculated our labour costs at approximately $30/hr for 3 graduate engineers over the course of 16 weeks working 10 hrs/week.

$$3 \times 30 \times 10 \times 16 \times 2.5 = \$36,000$$

Part	**Cost**
HC05 Bluetooth controller	$8.80
2x 7-Segment Display - 20mm (White) COM-11409 ROHS	$6.50
ATMEGA328P-AUR	$2.18
2x Infrared Thermometer - MLX90614	$59.90
Voltage Regulator - 3.3V COM-00526 ROHS	$1.95
USB DC 5V to 3.3V DC/DC Step-Down Converter Power Supply Module	$1.37
12V Car Cigarette Lighter Socket Splitter Dual USB Charger Power Adapter	$7.95
Total	**$88.65**

Total Parts + Labour = $36,000 + $48.13 = $36,088.65

3.2 Schedule:

Week	Rahul	Vinayak	Vishnu
3/7	Finalize design for PCB	Start developing software to connect sensors to display	Finalize parts list and place orders
3/14	Place order for PCB. Validation tests for Thermometer module	Validation tests on UI modules	Test and confirm sensor reliability
3/21	Validation tests for control module	Begin testing UI connection through bluetooth	Validation tests for power module

3/28	Assemble control module	Assemble UI module	Assemble power and thermometer modules
4/4	Unit test control module	Unit test UI module	Unit test Power and thermometer modules
4/11	Connect modules and verify basic operation	Connect modules and verify connectivity	Connect modules and verify power delivery
4/18	Bug fixing and final modifications and adjustments	Stress testing, particularly with UI module and bluetooth connection	Begin working on presentation and final paper
4/25	Fix any lingering issues and begin work on report and presentation	Fix any lingering issues and begin work on report and presentation	Fix any lingering issues and begin work on report and presentation
5/2	Finalize and wrap up presentation and paper	Finalize and wrap up presentation and paper	Finalize and wrap up presentation and paper

Ethics and Safety:

To follow the ECE 445 Safety Guidelines while working on this project, we will have at least 2 people in the lab working at all times, have completed the mandatory online safety training and have certification, have proper planning and additional safety training completed when working with high voltages, and follow safe battery usage guidelines. We will wear safety glasses while working on the project and proper hand protection while soldering. We will also be careful of electrical circuitry by grounding ourselves and the circuit properly before turning on the power.

In order to follow all of the Code of Ethics as determined by the IEEE [4], we will not identify a specific individual with their temperature results and will not store any temperature data. Also, we will disclose to the patient that their temperature results will be shared with the driver but not stored or shared with anyone else. All our wiring will be enclosed in insulation and routed safely throughout the vehicle, preventing any entanglement, static hazards, or loose/hazardous electrical connections. Our unit will also be stress tested for durability and will be able to withstand travelling at high speeds without detaching from the car door and being a potential debris hazard. All this combined ensures that we will be following all ethical and safety guidelines set by OSHA [5].

References:

[1] Sparkfun, ‘Infrared Thermometer - MLX90614’ 2015 [Online]. Available: https://www.sparkfun.com/products/9570

[2] Norweigan Institute of Public Health, ‘More COVID-19 in some occupational groups’ 2020 [Online]. Available: https://www.fhi.no/en/news/2020/more-covid-19-in-some-occupational-groups/

[3] National Institute for Health Research, ‘Non-contact infrared thermometers’ 2013 [Online]. Available: https://www.community.healthcare.mic.nihr.ac.uk/reports-and-resources/horizon-scanning-reports/hs-report-0025#:~:text=Two%20studies%20defined%20fever%20as,%2Dglass%20thermometer%20(17).

[4] Grainger, ‘Choosing and Using an Infrared Thermometer’, 2020 [Online]. Available: https://www.grainger.com/know-how/equipment-information/kh-370-infrared-thermometers-qt#:~:text=Clinical%20non%2Dcontact%20IR%20thermometers%20are%20designed%20to%20be%20used,about%20two%20to%20six%20inches.

[5] Instructables, ‘Simple Arduino and HC-SR04 Example’, 2020 [Online]. Available: https://www.instructables.com/Simple-Arduino-and-HC-SR04-Example/

[6] RandomNerdTutorials, ‘ESP32/ESP8266 Thermostat Web Server – Control Output Based on Temperature’, 2020 [Online]. Available:
https://randomnerdtutorials.com/esp32-esp8266-thermostat-web-server/

[7] Instructables, ‘ESP32: Pocket Size Distance Measuring and Logger’, 2020 [Online]. Available :
https://www.instructables.com/Pocket-Size-Ultrasonic-Measuring-Tool-With-ESP32/

[8] IEEE code of ethics. (n.d.). 2021 [Online] Available:
https://www.ieee.org/about/corporate/governance/p7-8.html

[9] Department of Labor logo United Statesdepartment of labor. (n.d.).2021 [Online] Available: https://www.osha.gov/laws-regs

[10] Murata, ‘Basic knowledge about ultrasonic sensors: Features of each type of ultrasonic sensor’ 2014 [Online]. Available:
https://www.murata.com/en-us/products/sensor/ultrasonic/library/basic/feature

[11] EDGEFX, ‘How Bluetooth Module Interfacing with Microcontrollers’ 2017 [Online]. Available:
https://www.edgefxkits.com/blog/interfacing-hc-05-bluetooth-module-with-microcontroller/#:~:text=The%20microcontroller%20can%20communicate%20with,information%20to%20other%20Bluetooth%20devices.

[12] HC-05 Latency Test [Online] Available:
https://www.youtube.com/watch?v=AnqPpt8dsbE&ab_channel=GautamM

[13] S Adarsh et al 2016 IOP Conf. Ser.: Mater. Sci. Eng. 149 012141 [Online] Available:
https://iopscience.iop.org/article/10.1088/1757-899X/149/1/012141/pdf

[14] GP2Y0A21YK Data Sheet [Online] Available:
https://www.sparkfun.com/datasheets/Components/GP2Y0A21YK.pdf