ELEC3848 Final Report

Bentulan, J Dylan Fourier Gauss A. P. (3035573276)

Cendra, Fernando Julio (3035556216)

Bachelor of Engineering in Electronics Engineering

December 4, 2020

1. **Project Abstract**

Our proposed application acts as a surveillance rover that will check each allocated space using a coordinate system and check whether or not users are in their respective areas through facial recognition systems. If users are not in their respective areas, an alert will be given out through a buzzer.

The black line sensor is used in determining the coordinates and the OLED display is used in displaying these coordinates. We made use of a similar route to that of required function 1 as the route of the rover, stopping at each green paper to perform the security check.

1. **System Description**

**Required function 1:**

For the first required function, we created a function where the car can move forward and turn right and left depending on the circumstances i.e. when the car detects a red color, it will turn left and when it detects green color it will turn right. Moreover, the car can also display its current coordinates in the OLED display. The end goal of this required function is that the car can arrive at certain coordinates where there will be some turning mechanism during the process.

So, for the implementation we reused the interim task 3 program where it enables the car to go forward, detects the coordinates and we also reused the interim 2 program to use the color detection program. From the high level view of our loop program, we first call the color detection function that will check whether it detects the red or green color papers and it returns the number corresponding to the color detected (can be red, green, or black). Then this number is going to be used as an input argument for the motor control function (you can refer to the sample code given by the teaching assistants). Then the motor control function uses the input argument to check whether the car has to go forward or turn right / left. At the same time, because we know that everytime the car turns, the face of the car direction will be changed. Thus, we created a global list of face directions (i.e. north, east, south, and west. But, those directions are in integers) and updated it by changing the current face direction when the car turns during the motor control function. Going back to the main loop function we also created the coordinate system where when both real black lines sensors detect a black line, then there will be a switch statement that will update the coordinate depending on the current face direction. For example, when the both sensors detect the black line and the current face direction is east, then the x axis will be updated i.e. x++. Going back to the main loop function, after updating the coordinates using the coordinate system, we also created the coordinate display function where the function will be called when there is a new updated coordinate.

**Required function 2:**

For the required function, we had to create a system where the car will go to a predefined coordinate where it is going to charge itself wirelessly using a wireless coil at that predefined coordinate. We are using the same code as the required function 1, but we add this function called the charging system function.

The main idea is that when the car arrives at a certain point then it will use it’s ultrasonic sensor to act like the guidance for the car to determine whether the car is near the wireless charging station or not. For example, the predefined coordinates is (2,4) where the first number corresponds to the x-axis and the second number corresponds to y-axis. The function will utilize the current coordinates to check whether the car is near the wireless charging system. When the function detects the coordinate to be (2,3), then instead of looping around inside the main loop function, it will be redirected to the wireless charging function. Inside the function, there is a while loop function that will utilize the ultrasonic sensor to check the distance between the car and the wireless charging station. While the distance is still not in the range where the wireless charging function can work, it will make the car keep moving. When it already achieved the desired distance obtained from the ultrasonic sensor, then the while loop breaks and there will be another while loop where it makes the car stop and updates the voltage value obtained from the wireless coil to the OLED display.

**Proposed function:**

For the proposed function as mentioned from the project abstract, we utilize two microcontroller i.e. Arduino Mega and Raspberry Pi where Arduino Mega is in charge for the motor system while the Raspberry Pi is in charge for the face recognition system. Moreover, we also created a booking system mobile application that utilizes bluetooth connectivity to send the data to Arduino Mega.

For the implementation, there are three rooms and two people in charge for that area. These two people can be assigned anywhere but they cannot be in the same room. Inside the mobile application, we can simply type the person's name to each room and there is going to be a room where there is no person assigned to that room. We assume that room can be utilized by any person where no alert is going to be called inside the room. When we want to send the data through bluetooth, there is a while loop inside the Arduino Mega setup function that waits for it. In addition there is a green LED that turns on while the data is not received yet. When the data has been received in integer, the integer corresponds to the number of combinations that the persons and the rooms can make i.e. 6 combinations. The combination value then is encoded by the 4 digital pins from Arduino Mega to a HIGH and LOW digits combination that is going to be sent using digitalWrite() to Raspberry Pi (From Arduino Mega digital pins to Raspberry Pi GPIO pins).

Before we created the face recognition program, we had gathered the face datasets for both users and we have already trained the model. The model will be used in the face recognition system. For the face recognition program, it will first wait for the incoming data to arrive in the GPIO inputs. To know when the combination values has come, we have already predefined the combinations of HIGH and LOW voltages to be not all LOW voltages i.e. (0,0,0,0). Thus, inside the face recognition system program, it will check the incoming inputs. If the inputs have come, then there will be a decoding function that returns the combinations value in integer. And from the combinations value, we can make it as an index corresponds to the list of combinations of rooms and the persons assigned. For the next step, it will stay in a while loop and wait for another incoming value coming from Ardunio Mega. this incoming value will only come when the car has arrived to the predefined room.

Going back to the Arduino program, similar to the first required function, we add another function that acts as a switch statement that will check whether the car has arrived to the predefined rooms or not. For example, when it arrives at room A then it will first trigger the Raspberry Pi by sending a HIGH voltage value to it. Moreover, the car will be stopped and wait for a response sent by Raspberry Pi. When the Raspberry Pi receives the inputs, it knows that the first room is going to be room A. Moreover, it also knows which person is assigned to room A. then the Raspberry Pi program will check the person’s face and returns two values. The first value is used to indicate that the Raspberry Pi has done its job to recognize the person’s face and the second value is used to indicate whether the person inside the room is the right one. After sending these two values, the Raspberry Pi goes to the previous state (waiting state) but now it knows that for the next input sent by Arduino Mega, the Raspberry Pi will check the next room and so on. Going back to the Arduino Mega program, when it receives those two values, if the second value is HIGH, the Arduino program turns on the green LED for a certain seconds. However, if the second value is LOW, then the Arduino program will turn on the red LED and also the buzzer that acts as an alert for a certain seconds. When it's done, it will continue to go to the next room and do the same thing until it reaches the final room.

1. **Changes to Proposal**

There were various changes that had to be made to our proposed application to increase its overall efficiency.

One change was that the mobile application had to communicate with the Arduino Mega first instead of the Raspberry Pi. We found this change to be more logical and efficient since we would have more control over the Arduino program. Moreover, we could use the Raspberry Pi to solely focus on the facial recognition aspect of the application. By allowing both microcontrollers to specialize in a specific task for the application, the power distribution will also be more balanced.

Another change we had to implement was the two-way low-level communication protocol (LOW and HIGH input combinations) between the Arduino Mega and the Raspberry Pi. We planned to use the serial communication from the Raspberry Pi to Arduino Mega by connecting the serial port of the Arduino to the Pi, however it was not that simple because of the difference in voltage and power levels between the Arduino and the Pi. This led to another change that we had to implement to make the data communication between the two possible: the use of a voltage converter. As mentioned earlier, there is a difference in the voltage levels between the Arduino (5V) and the Pi (3.3V), thus a voltage converter had to be implemented.

We also changed the use of the OLED display for our application. Initially, we wanted it to be used as an indicator of whether or not the person booked the room, however since we changed to using the coordinate system for the space allocation for the user, we found that it would be much easier to debug our program if we used it to display the coordinates instead. Besides, we already have the LED lights and the buzzer as our indicators, so it would be redundant and unnecessary to use the OLED display as an indicator as well.

1. **Reflection of Project**

Required Function 1:

One difficulty we encountered for this required function was the colour calibration using the colour sensor. For the initial position of our colour sensor, we noticed that the readings for the red and green colours were not consistent (i.e. changing every day). Thus, we had to change the position of the colour sensor to a more stable and suitable location to give more consistent readings.

Another difficulty we encountered was the turning of the rover. With the sample code,

we noticed that the rover was not turning properly; at times the rover would stop at the coloured paper, the rover would continuously turn for more than 360 degrees or it would turn a little bit (about 10 degrees) and keep moving forward. We overcame this problem by creating our own code to turn it properly as stated by the required function.

We also noticed that the position of the coloured paper affected or the position at which the rover detects the coloured paper played a major role in turning the rover properly, which was another difficulty we had to overcome. We had to do several trial-and-error methods with the positions of the papers and the rover to complete the path as in the required function.

Required Function 2:

One difficulty we encountered for this required function was finding out the most suitable position for the wireless receiver. Initially, the wireless receiver was positioned at the rear end of the rover, but we noticed that it would be easier to position it to the front end of the rover since we would not have to consider turning the rover to face its rear end to the wireless transmitter.

Another difficulty was determining the distance using the ultrasonic sensor. Because the speed of the rover approaching the wireless transmitter is not constant, it was difficult to control the delay at which the ultrasonic sensor processes the distance and stops, thus it was difficult to control the trade off between received voltage of the wireless receiver and the distance of the rover from the wireless transmitter. We overcame this difficulty by setting a suitable delay function in the program that does not affect the smooth movement of the rover

but helps in slowing down the rover in order to give more time to the ultrasonic sensor to process the distance.

Another difficulty was figuring out how to connect the 50-ohm resistor to the rover. With so many connections already within the rover, connecting the resistor was not so simple; we had to solder the resistor into the connection, and since it was my first time soldering it took me some time to complete it properly.

Proposed Application:

One difficulty we encountered for our proposed application was learning the logistics behind the communication protocol between the Arduino Mega and the Raspberry Pi. We had to take into account the delay between transmitting data from the mobile application via bluetooth communication protocol and receiving the data from the Arduino to the Pi.

Another difficulty encountered was making the app itself. It was our first time using an app inventor (i.e. MIT App Inventor), so we also had to learn the logistics behind its programming logic within a relatively short period of time.

On the software side, we had encountered various difficulties. Firstly, we had a problem with the facial recognition function. We initially used the OpenCV library installation for the facial recognition function, however it turned out that there was a version mismatch, so we had to find and install a suitable version of the OpenCV library to run the facial recognition function properly. Moreover, we had to create a virtual environment to run and test the facial recognition function.

Another difficulty on the software side was the overall control system. For instance, debugging our program, especially the python program, was not easy because it was difficult to determine the current state of the python program as we made use of the SSH connection. In addition, because of this SSH connection, we also encountered several network connection errors and issues.

Limitations and improvements:

One limitation of our proposed application at the moment is that it only accounts for 24 possible combinations. In order to make it more practical for work environments, more combinations have to be accounted for. This can be achieved by making use of more digital pins on the Arduino Mega.

Another limitation is that there is no proper cooling system for both microcontrollers. A proper cooling system must be implemented in order for the rover to work for long hours. This can be achieved by installing cooling fans or using other cooling materials such as thermal paste.

Moreover, another limitation regarding its working hours is its power constraints. We believe that using lithium-ion batteries instead of a powerbank is more suitable for long working hours because the batteries are scalable and thus cheaper.

In order for the rover to be working in an environment such as the one we proposed (i.e. construction industry), proper fabrication is needed for the rover’s outer case. Currently, the rover cannot operate at a workplace with all the microcontrollers and connections exposed. We were thinking of using materials such as fiber glass for fabricating the rover’s case since it is not too heavy. If the materials are too heavy, it might affect the speed of the rover.

Future developments:

For future developments of our proposed application, we were thinking of making use of object and image recognition for safety measures. For instance, it can detect whether an object at the workplace may fall off or can be in a harmful state. Moreover, it can be used to detect faulty equipment, so as to prevent workers from using faulty equipment and potentially harming themselves.

Another future development is that instead of using the Raspberry Pi for facial or image recognition, we will make use of the NVIDIA Jetson Nano. It is a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts. Because it is GPU-enabled, it is more suitable for machine-learning tasks such as facial or image recognition.