

ISTANBUL MEDIPOL UNIVERSITY

SCHOOL OF ENGINEERING AND NATURAL SCIENCES

INTRODUCTION TO COE AND EEE – FALL 2019

FINAL REPORT

**[TEAM MEMBERS]**

MEHMET EMRE AYAR

KEMALETTIN KOÇYİĞİT

SENANUR DEMİRCİ

BUSE NUR FİDAN

**[GROUP NAME]**

EECO

**[PROJECT LEADER]**

MEHMET EMRE AYAR

BLOCK DIAGRAM FOR THE AUTONOMOUS CAR

POWER MODULE

DESCRIPTION OF COMPLETED PARTS

To make the hardware of an autonomous car, fundamental materials that needed are two DC motors, one power source, two wheels, one ball caster, one ultrasonic distance sensors, one IR sensor, an L298 motor driver module, a control card for the IR sensor and Arduino UNO.

**Motor Driver Module**: The motors are foundation of the autonomous car. The motors are secured to the body by means of screws to chassis.

**Sensors**: Sensors must be installed for the autonomous car to receive data from the environment. These sensors include an ultrasonic distance sensor and an IR sensor. IR sensors are placed as close to the ground as possible so that they could receive the data properly. IR sensors are connected to the mainboards. In this way, the data from the sensors is transmitted to the Arduino microprocessor

**Power Module**: The cables from the power supply and the motor are connected to the powerbank with terminal input. This apparatus is connected to the Arduino. In this way, the power from the battery goes to both motors and feeds the Arduino. Lastly, all connections are checked for accuracy and hardware part of the autonomous car is created.

**Camera Connection**: Beside these, ESP32 Camera module is required to receive input from the environment. As mentioned in the mid-report, the autonomous car has been modularised to operate without camera connections. Arduino was installed on the computer used to integrate the camera into the autonomous car. However, to work with Arduino, their library needs to be added. So, added libraries supplied with links below to work the ESP32 CAM module.

<https://dl.espressif.com/dl/package_esp32_index.json> <http://arduino.esp8266.com/stable/package_esp8266com_index.json>

Camera connections were made by following the directions in the lab manual. ESP32 is a Wi-Fi module. When using ESP-32, mobile hotspot was opened from mobile phone and it was entered hotspots information’s on the code. The code which for retrieving images was uploaded. After this step, the cables which between IO0 and GND were took out and serial monitor was opened and reset button on the ESP-32 CAM was pushed. After this step, IP address was created, and it was appeared on the screen. This IP address was entered from the browser and it was started to show images, and stream was watched.

**Sensor processing:**

Arduino was used for sensor operations. In the C programming language, codes were written to activate the software of the sensors and to adjust the distance of the sensors.

**Autonomous Car Control:**

The commands of the vehicle were developed with the help of the Mbot3 program. With the help of these commands, the wi-fi module was used to remotely control the vehicle. The vehicle and the control center of the vehicle were connected to the same network and vehicle control was achieved.

**Image Processing:**

In the image processing section, the vehicle had to interpret the image taken from the ESP-32 camera and determine the color in the image.

First, we investigated the RGB values of the colors. In Python, we have defined the RGB value ranges for red, blue and green. Thus, when the tool sees a color, it will first calculate the RGB value range and, according to Python code, then it will determine that color if it matches the value range of the colors we define (red, green, blue), and the car continues to move.

ROLE OF TEAM MEMBERS

HARDWARE:

* Creating circuits using Arduino
* Connection of camera module
* Modularizing the tool

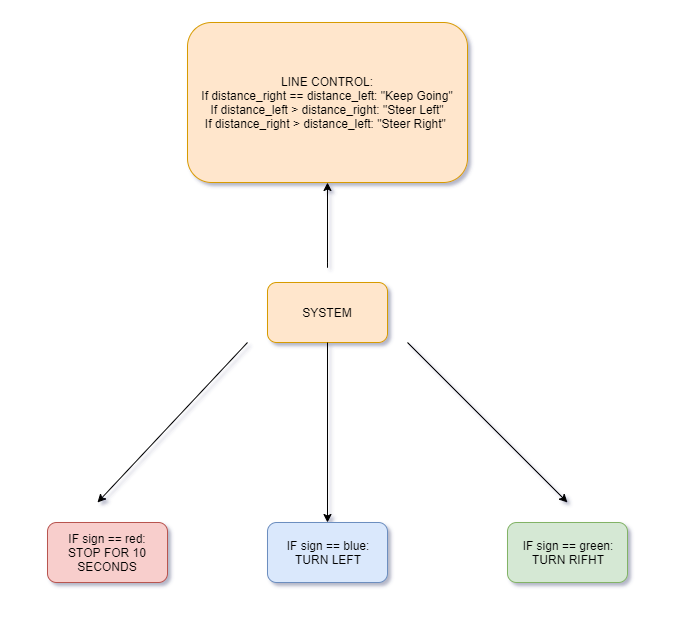
This part was completed by Buse Nur Fidan and Senanur Demirci

SOFTWARE:

• Developing the car's software using Arduino

• Retrieving images from the environment with the camera and processing them with Python.

This part was completed by Kemalettin Koçyiğit and Mehmet Emre Ayar.

MATHEMATICAL MODELLING

The commands passed in the software to enable the movement of the car were given to the engine.,

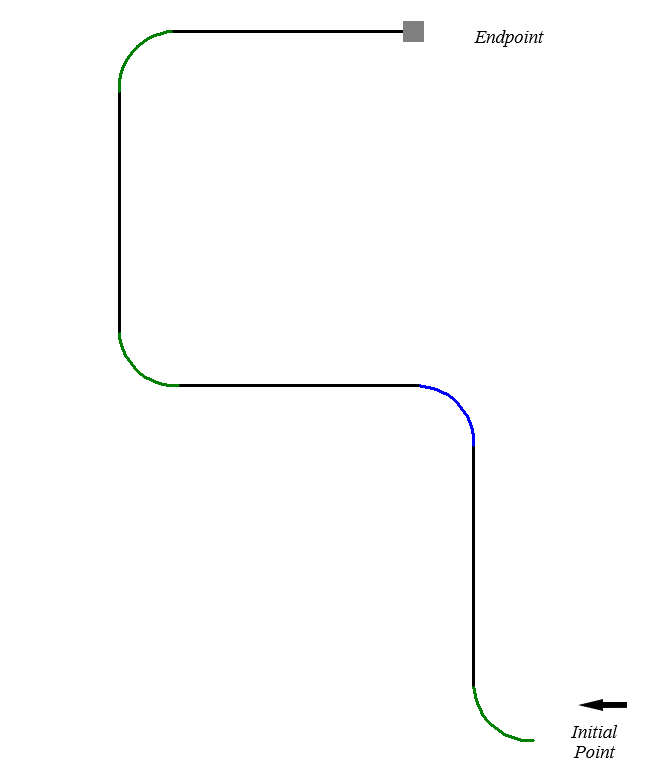
If the sign == red power of two engines is cut for 10 seconds.

If the sign == bluepower the engine on the right and stop the engine on the left.

If the sign == green power the engine on the left and stop the engine on the right.

For line tracking, the car must follow the yellow lines on the black ground. That's why the car should detect the yellow colour. To do this, software commands should be given to the camera. The camera should be running along with the engine as the car turns right and left. The engine works as mentioned above while the camera detects yellow lines.

SIMULATION/DEMO RESULTS



It was thought that in order to simulate for an autonomous car, the road which by using to the car would have to be known. However, as the road to which the autonomous car will go has not been determined, it could not be determined with certainty whether the simulation would work correctly in practice.

**CHALLENGES**

It was difficult to integrate hardware and software. For this purpose, the task distribution between the members of the group was made to try to better understand the ultrasonic sensor, image processing, and Open CV.

The other difficult part of the project was the final stage of the project, the image processing part. Because the data obtained from the camera of the autonomous car had to be interpreted with the help of python and the car had to be controlled. In addition, the red, green and blue plates had to be interpreted with the camera to command the car. However, distinguishing and defining these colors from reflections and other colors in the environment was a difficult part of the project.

**Some problems and solutions during project construction:**

**Problem 1 (Autonomous car control):**

The most important part of adjusting the control of the vehicle was the right and left turns. When the vehicle was in motion, when it saw the green plate, the vehicle had to rotate 90 degrees (in a quarter circle) to the right in a safe distance (not to make a fast and sharp turn) at a distance from the plate.

**Solution**: On the right turns, we first calculated the path that the right engine would take per unit time. We then calculated the path the left engine would take per unit time. With these data, we obtained the linear speed of the motors and entered these speeds into the Mbot3 program. Thus, we have realized the safe rotation of the vehicle.

We have given the same command to Mbot3 so that the vehicle can turn left when it sees the blue plate.

**Problem 2 (Image Processing):**

Determining the color of the image We investigated the RGB ranges and defined these values into Python. However, when a red plate was placed in front of the vehicle, the vehicle perceived it as 'blue'. Because there was a blue cupboard standing behind the red plate. Although the cabinet was remote, the RGB values of the image were very close to blue. As a result, other colors in the environment caused us to misinterpret the image.

**Solution:** The vehicle was standing 10 cm before the red, blue and green plates came and it was beginning to interpret the color. According to the image taken from the camera, 60 percent of the image had a plate. We first tried to focus the camera on this 60 percent area, and then we wrote a code on it. According to this code, the vehicle was stopping when it sees the plate. It was scanning the area formed by the plate in the image and only was interpreting the color of the plate.

**Problem 3(Image Processing):**

In this part, we had a problem with the camera.

The fps of the image from the ESP-32 was too low. Regardless of the quality of the image, it was difficult to get a picture from the camera. The fps value of the live image obtained from the camera was too low. Therefore, the vehicle could not interpret the image taken from the environment.

**Solution:** According to the values we applied in the laboratories, we needed to give the car's camera a 3.5 Volt energy. Then we tried to increase this value to 5 V. Thus, the fps of the image are up to 5-6 times higher quality. At these values, despite the vehicle was tested for 2 hours, there was little heat in the camera.

CONCLUSION

The main aim of the assignment is to transfer the theoretical knowledge learned in labs to practice and to make concrete an autonomous car in line with this knowledge. In general, the information in the lab manual was used to make the hardware part of the car. Hardware part involves making the autonomous car's engine connections, mounting the sensors, integrating Arduino into the autonomous car, Arduino and main cart connections and power connections. The software part of the project, different programming languages ​​(C, Python) and different programs (Mbot3, some camera applications) were used. The software parts were constantly updated and the project was completed successfully. In addition to these, the last part of the project followed the directions in the lab manual and the camera connection was made. As a result, the software and hardware were integrated as desired and the project was successfully completed

As the social aspect, while doing this project, it was learned how to work efficiently with people who have different branches of engineering. The workload of each group member working on the project was eased by effectively allocating tasks. Technically, it had the opportunity to study and observe with Arduino. Also, it was worked with the ESP32 camera module. The importance of modularization and obtaining an efficient product using less material that is optimization was understood and while the project was being carried out.