Zewail city of science and technology
Embedded Systems - FALL 2020

Students Names	Students IDs
Ahmed Eid Abdullah	201701264
Khalied El Abyad Abdelnabi	201700500
Ahmed Kamal El Melegy	201700295

CIE 408 – Project Report

Zewail city of science and technology
Embedded Systems - FALL 2020

Table of content :-

Problem definition

We have a car in a maze , and we want the car moves with avoiding each obstacle. and the car escape the maze .

Hardware structure

- The main controller is : ESP32 but after many trials we changed the controller to Arduino uno as we faced problems with the ESP32
- Three ultrasonics sensors
- Motor driver
- Three batteries of 3.7 Volt

Problem we faced and their solution

- The car itself is funtioning with no problems ; we tested it first with Bluetooth controller with phone :

https://drive.google.com/file/d/19sZ4L7zB4LSF0jqqAFnS8YgZthq_G9be/view?usp=sharing

- The code of the car should work perfectly but we face some problems also .

- We found that for PWM writing with ESP32 , there is no direct way to output a PWM signal. However, that was not the problem; as we had to define the channel for the PWM and so I can not have 2 pins having PWM at the same time with the same channel, but when defining two channels we faced problems, so we chose to use the other simpler controller which is Arduino uno

- The 2 motors are not with same speed , mostly because they are old. Hence it was hard for the car to be stable.

We tried to solve this problem by changing the pwm signal fed to the motor driver for each motor by assigning one with 150 and other with 140 and changed it multiple times but it was the best choice for PWM value.

- Another problem is : The car is so fast , which in return make it hard to know the right direction to move to.

We tried to get less voltage with only 2 batteries but still it is fast , we then tried to make it work with less speed controlled with PWM but it does not function well with low PWM as it hardly moves. and if one motor moves , the other motor does not move.

So the only solution is to make it work with enough PWM but make it stops every time to choose to move forward , left or right. So , it can take enough time to calculate the distance and chooses which path to take.

- We then faced another problem with the calculation of the distance with ultrasonic as it sometimes does not read anything.

We tried to solve this problem with applying median filter and take many distances so if some one is missing it won't be such a big trouble. we then faced problems as it did not work as expected.

We then tried to implement a code to filter the distances from the ultrasonics but we got a problem with the timing; the more values we take the more time it stops and hangs.

We solved this problem by taking few values from the ultrasonic and get the average of them.

- The last problem is that the route for the car itself is not good , it is home made with some wood and books .

Finally, the car worked and succeeded in solving the maze but still with problems from ultrasonics (may be they are not positioned perfectly) and the motors (they still have different speed so one motor would be faster than the other which makes the car turns left and that would make the problem with the ultrasonic much bigger).

We assume that with better components and better route , we would do it more successful.

But finally , we couldn’t ask for more with all the above problem we faced :D

Zewail city of science and technology
Embedded Systems - FALL 2020

Hardware videos

The Video when car move to left “as we mentioned in problem we faced section” :

https://drive.google.com/file/d/1A4Ynebo_hFksnWBbm4IJTizxZDKjyu93/view?usp=sharing

The Video when car move as well as possible :

https://drive.google.com/file/d/19sZ4L7zB4LSF0jqqAFnS8YgZthq_G9be/view?usp=sharing

Zewail city of science and technology
Embedded Systems - FALL 2020

Software code

There are 5 files :-

- Car.h : Header File responsible for deine every thing in car , such as left motor , right motor , right motor pwm and left motor pwm. It contains the headers of the car's function such as stop car , move forward , turn right and turn left.

- Car.cpp : defining the paramters of the car and define the functions of car. Here we assigned thepins output. Stopcar function is a void function that stop the car by making all cars parameters aslow such as the motors and motors pwm are low.

- sonicSensor.h : Header File responisble for define every thing in sensor. Here there is an mportant function called get distance , this function responsible for avoiding the obstalces in the maze.

- sonicSensor.cpp : File responsible for defining the get distance function and define the input and output pins of the sensors.

- MazeRobot.ino : The arduino code that assign the sensors pins and the car motors values.