POZNAŃ UNIVERSITY OF TECHNOLOGY

FACULTY OF CONTROL, ROBOTICS AND ELECTRICAL ENGINEERING

« TERM DESIGN »

ALPHABOT2 – AR

RAM ELLANKI, 139783

RAM.ELLANKI@STUDENT.PUT.POZNAN.PL

MOHAMED MAJDOULINE, 142064

MOHAMED.MAJDOULINE@STUDENT.PUT.POZNAN.PL

INSTRUCTOR:

MGR INŻ. JAN WIETRZYKOWSKI

JAN.WIETRZYKOWSKI@PUT.POZNAN.PL

18 JUNE 2021

Table of Contents

INTRODUCTION

The Aim of the project is to traverse a maze using AlphaBot2 – Ar. We used additional sensors (Ultrasonic sensor) to make it easy to the bot to detect the walls and to traverse the maze. This project also includes various versions and algorithms to confirm which suits best for the bot and building various prototypes of mazes including the final moderate level testing maze. The best likely suitable algorithm for our alphabot is wall algorithm using one additional senor but with unique setup.

BASE PLATFORM (ALPHABOT2 – AR)

AlphaBot2-Ar robot kits includes a chassis (AlphaBot2-Base chassis) and an adapter board AlphaBot2-Ar. The robot supports Arduino with AlphaBot2-Ar adapter board. The platform comes with everything we needed including wheels with ball supports on the bottom section of the bot and thanks to this modular design it was very easy to assemble the whole bot without any additional wiring or soldering unless we want to use soem additional sensors other than on – board. It also comes with various options and chips for communicating or connecting the additioanl sensors which we actually didn't used most of them. The complete description of all the base platform can found in AlphaBot2 – Ar.

ADDITIONAL SENSORS

To make it easier to traverse the maze we added an additional ultrasonic sensor (*HC-SR04*) to the alphabot. Thanks to this sensor we were able to implement the wall – following algorithm much easier to the AlphaBot2 base platform.

2.1 ULTRASONIC SENSOR (HC-SR04)

Ultrasonic ranging module HC-SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit.It works with a 5V power supply.

Figure 1 : HC-SR04 Ultrasonic Distance Sensor.

2.2 PLACEMENT OF HC-SR04.

The Additional sensor we added on the base platform AlphaBot2 – Ar is responsible to detect the walls on its right side and follow it and at the same time to detect the obstacles or a freeway between the bot and the walls. The base platform already consists of same sensor on – board in the front which we used it to detect the obstacles and move forward. We placed the additional sensor on the top left of the bot but with a twist where the sensor actually faced towards right direction and follows the walls on the right side. We applied this concept here so that the sensor can work properly without any minimum distance range issues. To connect this additional sensor, we removed the jumpers of DOUT and ADDR on the bot and used the pins 11 and 12 from the Arduino to connect the sensor's ECHO and TRIG pins.

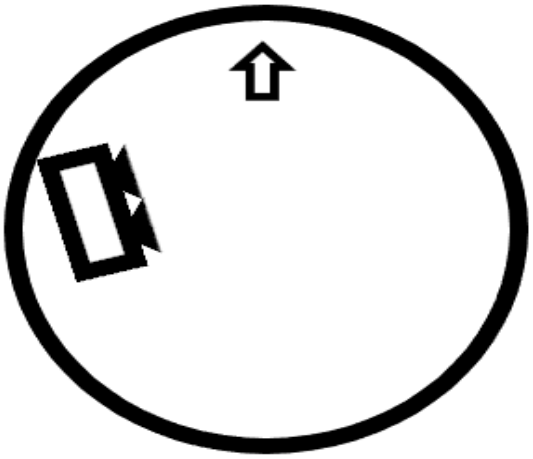

Figure 2 : Sensor Placement on the AlphaBot2 - Ar.

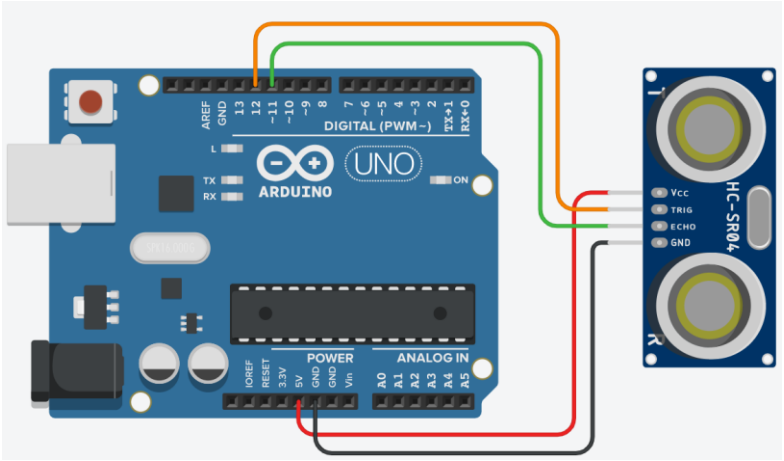

Figure 3 : Connection of Additional Sensor to Arduino.

MAZE

To test the behavior of AlphaBot2 – Ar we build a special maze starting with a simple maze followed by a complex maze. The construction is fully made with cardboard. But for the final test we used one of our colleagues maze which was made with wood.

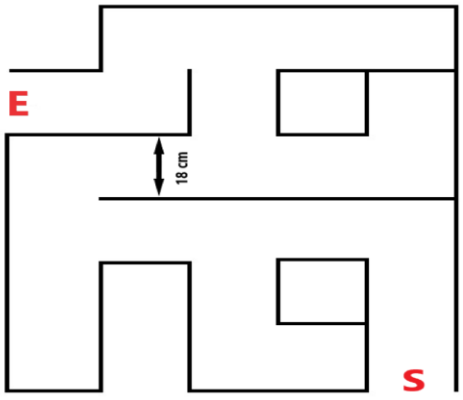

Figure 4 : Schematic of Final Maze.

SOFTWARE AND HARDWARE IMPLEMENTATION

4.1 SPECIFICATION

The platform is supposed to follow the wall on its right side. Based on the signal received from the sensors it should detect whether the wall is on front or right side or a free way to change the lanes.

4.2 IMPLEMENTATION

Several Arduino compliant C++ libraries were used for the purpose of motor and sensor handling. Some of the code is provided in the examples from the manufacturer. All the library files that are used in our code are uploaded in our GitHub repository to which the instructor has full access. In the listing below, the algorithm that we used in our bot is presented.

Listing 1 : ***maze_final.ino***

```
const int trigPin1 = 3;       // Front sensor

const int echoPin1 = 2;

const int trigPin2 = 12;      // Right faced left sensor

const int echoPin2 = 11;

const int in1 = A0;           // Right Forward

const int in2 = A1;           // Right Backward

const int in3 = A3;           // Left forward

const int in4 = A2;           // Left Backward

const int enA = 6;           // Right motor PWM Speed pin

const int enB = 5;           // Left mo0tor PWM Speed pin



#define PWM 50               // Both Motor's Speed (between 0 to 255)
int DIS = 8,                 // Front sensor minimum distance
    DIS2 = 9;                // Right faced left sensor minimum distance

boolean statusfront = false,          // Required Booleans to drive the bot in order
to solve maze
        statusrightdone = false,
        statusleftdone = true,
        righttimer = false,
        lefttimer = false;

int duration1,                        // required variables to drivethe bot in order
to solve maze
    cm1,
    cm2,
    duration2;

void setup() {
  pinMode(trigPin1, OUTPUT);          // Declares the trigpin as OUTPUT and Echo as
input of 2 sensors

  pinMode(echoPin1, INPUT);

  pinMode(trigPin2, OUTPUT);
```

```
  pinMode(echoPin2, INPUT);

  pinMode (in1, OUTPUT);              //Declare all motor pins as OUTPUT

  pinMode (in2, OUTPUT);

  pinMode (in3, OUTPUT);

  pinMode (in4, OUTPUT);

  pinMode (enA, OUTPUT);

  pinMode (enB, OUTPUT);

}

void loop() {

  frontdis();                       // Runs the void frontdis(); in order to get
distance from front sensor stored in cm1 variable
  leftdis();                        // Runs the void leftdis(); in order to get
distance from left sensor stored in cm2 variable



  if (cm1 < DIS) {                   // If front sensor detects wall
    statusfront = false;             // Statusfront sets to false
  }
  if (statusfront == true) {         // But if nothing is in front of front sensor,
the statusfront stays true and does the next loop in the {} bracket
    if (cm2 < DIS2)                  // If nothing is in front of front sensor but
left sensor detects wall near than the limited distance then :
    {
      digitalWrite(in1, HIGH);       // Only the right motor goes forward until the
sensor comes to its perfect distance to make the bot straight
      digitalWrite(in2, LOW);

      digitalWrite(in3, LOW);

      digitalWrite(in4, LOW);

      analogWrite(enA, PWM);

      analogWrite(enB, PWM);
    }
    else
    {
      forward();                     //if the bot is straight then it goes forward
until next wall comes
    }
  }


  if (statusfront == false && statusrightdone == false) {
    // if the bot detects obstacle in front of it then it firstly goes to right 90
degrees and sets in memory that it already had turned right

    if (righttimer == false) {  //Runs the 90 degrees right turn loop only once
      turn_right();
      delay(270);
      stop();
      righttimer = true;
    }
    else if (righttimer == true) {  // Turns right until the front sensor detects
blank space to go on
      turn_right();
```

```
    }

    if (cm1 > DIS) {                // if front sensor finds blank sapce to go
forward
      // Left timer starts to turn left at next blockage
      statusfront = true;
      statusrightdone = true;
      statusleftdone = false;
      righttimer = false;
    }
  }
  if (statusfront == false && statusleftdone == false) {    // Same as right turn,
it turns left and sets in memory that the next turn will be turned right side
    if (lefttimer == false) {
      turn_left();
      delay(270);
      stop();
      lefttimer = true;
    }
    else if (lefttimer == true) {
      turn_left();
    }

    if (cm1 > DIS) {
      statusfront = true;
      statusrightdone = false;
      statusleftdone = true;
      lefttimer = false;
    }
  }

}
void frontdis()                                          // Function to measure
distance
{


  digitalWrite(trigPin1, LOW);                           // Send tiny pulses
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);

  duration1 = pulseIn(echoPin1, HIGH);                  //Calculate the time till
it gets the pulse reflected
  cm1 =  duration1 * 0.034 / 2;                         //Calculate the distance
from the travelling time of previous pulse
}
void leftdis()                                          //Same as front sensor,
left sensor detects distance of it
{


  digitalWrite(trigPin2, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2, LOW);

  duration2 = pulseIn(echoPin2, HIGH);
  cm2 =  duration2 * 0.034 / 2;
}
void forward ()

{

  digitalWrite(in1, HIGH);
```

```
  digitalWrite(in2, LOW);

  digitalWrite(in3, HIGH);

  digitalWrite(in4, LOW);

  analogWrite(enA, PWM);

  analogWrite(enB, PWM);

}

void turn_left ()

{
  delay(270);
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);

  digitalWrite(in3, HIGH);

  digitalWrite(in4, LOW);

  analogWrite(enA, PWM);

  analogWrite(enB, PWM);

}


void turn_right ()

{
  delay(270);

  digitalWrite(in1, HIGH);

  digitalWrite(in2, LOW);

  digitalWrite(in3, LOW);

  digitalWrite(in4, HIGH);

  analogWrite(enA, PWM);

  analogWrite(enB, PWM);

}

void reverse ()

{

  digitalWrite(in1, LOW);

  digitalWrite(in2, HIGH);

  digitalWrite(in3, LOW);

  digitalWrite(in4, HIGH);

  analogWrite(enA, PWM);

  analogWrite(enB, PWM);

}

void stop()
```

```
{
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  analogWrite(enA, LOW);
  analogWrite(enB, LOW);
}
```

4.3 CONCLUSION

The bot was able to solve the maze although the obtained results are not perfect but satisfactory. The reason it is not perfect because it is getting stuck at the edges of the maze while traversing due to sharp edge of the alphabot around its body. Because of this issue the bot needs a little back support sometimes when it is stuck in this scenario. Other than this issue out maze successfully was able to traverse the maze in first try without any back support which is a great progress.

SUMMARY

We successfully finished this project although with little drawbacks. Thanks to this project which helps us to gain more knowledge about different types of bots and their nature of working. After finishing this project we can say that it is more interesting in this field and gaining knowledge more about Robots. The following images are taken in the process of whole project.

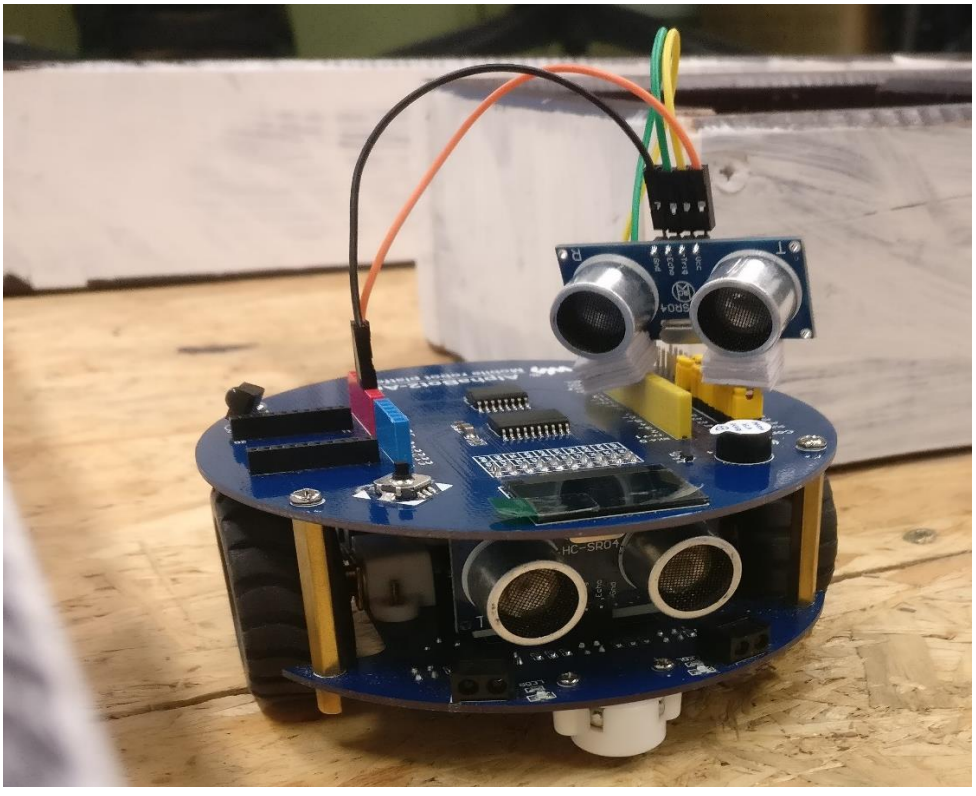

Figure 5 : AlphaBot2 - Ar traversing Maze.

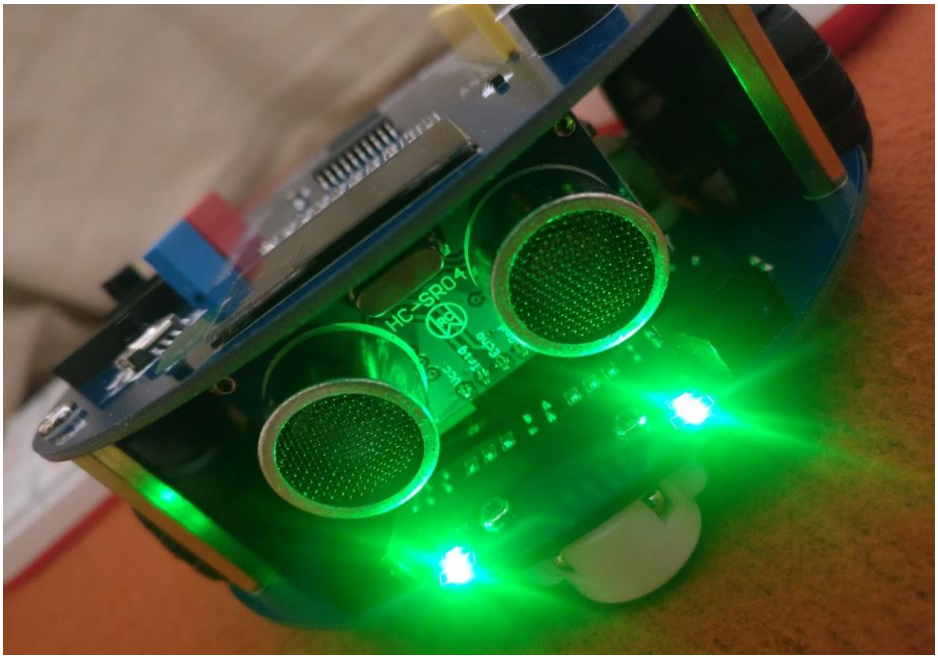

Figure 6 : AlphaBot2 - Ar.

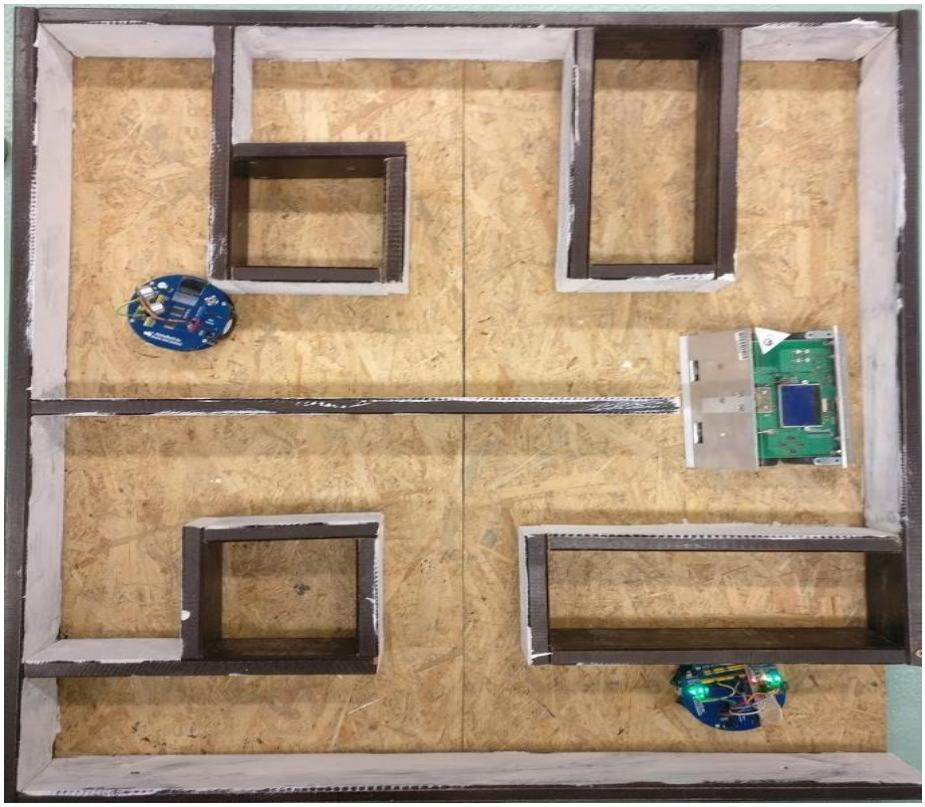

Figure 7 : Bot traversing Final Maze with other bots.

REFERENCES

1. https://www.waveshare.com/wiki/AlphaBot2-Ar
2. https://www.mouser.com/pdfdocs/Alphabot2-user-manual-en.pdf
3. https://github.com/RamEllanki/Alphabot