Tyler Thompson

2/25/23

tjt5498@psu.edu

**Lab 11**

**Objective**

Install line sensors onto the robot and demonstrate that the software and hardware function properly. Perform different tests to ensure accuracy. After it is confirmed to be working edit the code to enable other functionalities to the robot like stopping when detecting an object while following the path. Ports may have to be changed in code to allow for everything to function correctly. Materials include the line sensors as well as an obstacle course to follow and demonstrate results.

**Results**

|  |  |
| --- | --- |
| Picture of line sensor installed  A picture containing indoor, cluttered  Description automatically generated | Here we can see the line sensor properly installed onto the robot. It was an easy installation and did not require taking the robot apart. After the wiring was complete it was clear that once the code had been set up some pins would have to be switched around in the code to allow it function properly. After all that the software and hardware were ready to function correctly. |

|  |  |
| --- | --- |
| Picture of the various paths the robot completes  A picture containing floor, building, indoor, tile  Description automatically generated | Here we can see the multiple paths the robot will complete. The robot will demonstrate that it can complete these properly and accurately. Once the robot has proved it can replicate the same result new code will be constructed to allow the robot to stop when an object in front of it is detected. This will be demonstrated on a path and the results will be noted. |

**Trails At Slow Speed**

|  |  |
| --- | --- |
| Robot doing sharp turn to the right at slow speed  <https://share.icloud.com/photos/0bdq1NwtZUP3Lo5qAmUQfZjow> | The robot is moving at 150 speed here and follows the line all the way along the path. We can see the robot oscillating quite frequently. The robot reaches the 45 degree angle to the right and it takes it nicely and continues forward along the path. |
| Robot doing sharp turn to the left at slow speed  <https://share.icloud.com/photos/05dxZ5uyTo99FQ7D9GuO8YI9g> | The robot is moving at 150 speed here and it follows the line all the way along the path. The robot is oscillating quite frequently. The robot once again reaches the turn which is a 45 degree angle to the left and takes the turn nicely and continues along the path. |
| Robot taking curve to the right at slow speed  <https://share.icloud.com/photos/02ddvxKmvfhF8IcepYgMzLHcg> | The robot is moving at 150 speed and follows the path and once it reaches the curve it stays in line to the right and follows it nicely. The robot is oscillating quite frequently. |
| Robot taking curve to the left at slow speed  <https://share.icloud.com/photos/058WTln7q8rQBrXaC6zxM3_pw> | The robot is moving at 150 speed and follows the path and once it reaches the curve it goes to the left nicely. The robot is oscillating quite frequently. |

**Trails At Fast Speed**

|  |  |
| --- | --- |
| Robot doing sharp turn to the right at fast speed  <https://share.icloud.com/photos/058hcUINk9zSNa1neoyjbUQhQ> | The robot is moving at 250 speed here and follows the line all the way along the path nicely. The robot is seen to be oscillating even more frequently here. The robot reaches the 45 degree angle to the right and take it well and continues along the path. |
| Robot doing sharp turn to the left at fast speed  <https://share.icloud.com/photos/0ecL7EBTcW9slJdQfFbNhqoUw> | The robot is moving at 250 speed here and follows the line all the way along the path nicely. The robot is seen to be oscillating even more frequently here. The robot then reaches the 45 degree angle to the left turn and takes it well and continues along the path. |
| Robot taking curve to the right at fast speed  <https://share.icloud.com/photos/0aavvfPYbZBlc4lfin5qqrv_g> | The robot is moving at 250 speed here and follows the path and reaches the curve. The robot is moving really fast and still handles it really nicely. The robot is oscillating quite frequently. |
| Robot taking curve to the left at fast speed  <https://share.icloud.com/photos/00dCoFcKT7gl9e-2RXytzgCxA> | The robot is moving at 250 speed here and follows the path and reaches the curve. The robot is once again moving really fast and handles the curve nicely. |

**Trail With Object**

|  |  |
| --- | --- |
| Robot follows the line while encountering an object in the path.  <https://share.icloud.com/photos/0bcGOpmB7r9SPjP0IJci5eosQ> | The robots speed here is set to 150 to allow it to successfully traverse the path. The robot follows the path until it encounters an object 6 inches away from the robot. The alarm is set, and the robot is stopped. The robot waits until the object is removed then continues following the path. The whole time the robot is oscillating while moving and it is staying within the line correctly. The robot then gets to the end of the path where it completely stops since there is no longer a path to follow. |

**System Block Diagram**

Line Sensor

Control Software

Arduino

Sonar Device

Motor Control H Bridge

Front Left Motor

Back Right Motor

Front Right Motor

Back Left Motor

**Pseudocode**

auto\_tracking {

 sensorval = read\_sensor\_values

  Serial.println (sensorval)

  // This makes sure there is no object in front of the robot.

  if (watch < 10) {

    stop

set alarm

  }

  else {

    if (   sensorval=="10000" ) {

      //The black line is in the left of the car, need  left turn

      go left

set motor speed (fast speed, fast speed)

    }

    if (sensorval=="10100"  or sensorval=="01000" o sensorval=="01100" or sensorval=="11100"  or sensorval=="10010" or sensorval == "00100" or sensorval=="11010") {

    go forward

set motor speed (0, fast speed)

    }

    if (    sensorval=="00001"  ){ //The black line is  on the right of the car, need  right turn

      go right

set motor speed (fast speed, fast speed)

    }

    if (sensorval=="00011" or sensorval=="00010"  or sensorval=="00101" or sensorval=="00110" or sensorval=="00111" or sensorval=="01101" or sensorval=="01111"   or sensorval=="01011"  or sensorval=="01001")

    {

      go forward

set motor speed (fast speed, 0)

    }

    // Stops if no line is encountered.

    if (sensorval == "00000") {

      stop

      set motor speed (0, 0)

    }

    if (sensorval=="11111"){

      stop

      set motor speed (0, 0)

    }

    // This slight delay gives the robot time to detect the different inputs.

    delay 90

  }

**Code**

// This program allows the robot to follow a line as well as detect if an object is in front of it.

// If an object is detected the robot will stop until the object is moved away. The robot will

// also stop if there is no longer a line to follow.

/\*Declare L298N Dual H-Bridge Motor Controller directly since there is not a library to load.\*/

//Define L298N Dual H-Bridge Motor Controller Pins

#include <Servo.h>

/\*From left to right, connect to D3,A1-A3 ,D10\*/

#define LFSensor\_0 A0  //OLD D3

#define LFSensor\_1 A1

#define LFSensor\_2 A2

#define LFSensor\_3 A3

#define LFSensor\_4 A4  //OLD D10

#define FAST\_SPEED 250

#define MID\_SPEED 200

#define SLOW\_SPEED  150

#define speedPinR 5    //  RIGHT PWM pin connect MODEL-X ENA

#define RightMotorDirPin1  12    //Right Motor direction pin 1 to MODEL-X IN1

#define RightMotorDirPin2  11    //Right Motor direction pin 2 to MODEL-X IN2

#define speedPinL 6    // Left PWM pin connect MODEL-X ENB

#define LeftMotorDirPin1  7    //Left Motor direction pin 1 to MODEL-X IN3

#define LeftMotorDirPin2  8   //Left Motor direction pin 1 to MODEL-X IN4

#define SERVO\_PIN     9  //servo connect to D9

#define Echo\_PIN    4

#define Trig\_PIN    10

#define BUZZ\_PIN     13

// This all has to do with the sonar and servo motor.

int leftscanval, centerscanval, rightscanval, ldiagonalscanval, rdiagonalscanval;

const int distancelimit = 30; //distance limit for obstacles in front

const int sidedistancelimit = 30; //minimum distance in cm to obstacles at both sides (the car will allow a shorter distance sideways)

int distance;

int numcycles = 0;

const int turntime = 250; //Time the robot spends turning (miliseconds)

const int backtime = 300; //Time the robot spends turning (miliseconds)

int thereis;

Servo head;

/\*motor control\*/

void go\_Advance(void)  //Forward

{

  digitalWrite(RightMotorDirPin1, HIGH);

  digitalWrite(RightMotorDirPin2,LOW);

  digitalWrite(LeftMotorDirPin1,HIGH);

  digitalWrite(LeftMotorDirPin2,LOW);

  analogWrite(speedPinL,200);

  analogWrite(speedPinR,200);

}

void go\_Left(int t=0)  //Turn left

{

  digitalWrite(RightMotorDirPin1, HIGH);

  digitalWrite(RightMotorDirPin2,LOW);

  digitalWrite(LeftMotorDirPin1,LOW);

  digitalWrite(LeftMotorDirPin2,HIGH);

  analogWrite(speedPinL,200);

  analogWrite(speedPinR,200);

  delay(t);

}

void go\_Right(int t=0)  //Turn right

{

  digitalWrite(RightMotorDirPin1, LOW);

  digitalWrite(RightMotorDirPin2,HIGH);

  digitalWrite(LeftMotorDirPin1,HIGH);

  digitalWrite(LeftMotorDirPin2,LOW);

  analogWrite(speedPinL,200);

  analogWrite(speedPinR,200);

  delay(t);

}

void go\_Back(int t=0)  //Reverse

{

  digitalWrite(RightMotorDirPin1, LOW);

  digitalWrite(RightMotorDirPin2,HIGH);

  digitalWrite(LeftMotorDirPin1,LOW);

  digitalWrite(LeftMotorDirPin2,HIGH);

  analogWrite(speedPinL,200);

  analogWrite(speedPinR,200);

  delay(t);

}

void stop\_Stop()    //Stop

{

  digitalWrite(RightMotorDirPin1, LOW);

  digitalWrite(RightMotorDirPin2,LOW);

  digitalWrite(LeftMotorDirPin1,LOW);

  digitalWrite(LeftMotorDirPin2,LOW);

}

/\*set motor speed \*/

void set\_Motorspeed(int speed\_L,int speed\_R)

{

  analogWrite(speedPinL,speed\_L);

  analogWrite(speedPinR,speed\_R);

}

// Turns the buzzer on.

void buzz\_ON()   //open buzzer

{

  for(int i=0;i<100;i++)

  {

   digitalWrite(BUZZ\_PIN,LOW);

   delay(2);//wait for 1ms

   digitalWrite(BUZZ\_PIN,HIGH);

   delay(2);//wait for 1ms

  }

// Turns the buzzer off.

}

void buzz\_OFF()  //close buzzer

{

  digitalWrite(BUZZ\_PIN, HIGH);

}

// Sets the buzzer on quickly then turns it off.

void alarm(){

   buzz\_ON();

   buzz\_OFF();

}

/\*detection of ultrasonic distance\*/

int watch(){

  long echo\_distance;

  digitalWrite(Trig\_PIN,LOW);

  delayMicroseconds(5);

  digitalWrite(Trig\_PIN,HIGH);

  delayMicroseconds(15);

  digitalWrite(Trig\_PIN,LOW);

  echo\_distance=pulseIn(Echo\_PIN,HIGH);

  echo\_distance=echo\_distance\*0.01657; //how far away is the object in cm

  //Serial.println((int)echo\_distance);

  return round(echo\_distance);

}

void setup()

{

  pinMode(RightMotorDirPin1, OUTPUT);

  pinMode(RightMotorDirPin2, OUTPUT);

  pinMode(speedPinL, OUTPUT);

  pinMode(LeftMotorDirPin1, OUTPUT);

  pinMode(LeftMotorDirPin2, OUTPUT);

  pinMode(speedPinR, OUTPUT);

  stop\_Stop();//stop move

  pinMode(BUZZ\_PIN, OUTPUT);

  digitalWrite(BUZZ\_PIN, HIGH);

  pinMode(Trig\_PIN, OUTPUT);

  pinMode(Echo\_PIN,INPUT);

  digitalWrite(Trig\_PIN,LOW);

  /\*init servo\*/

  head.attach(SERVO\_PIN);

  head.write(90);

  Serial.begin(9600);   // initialize serial for debugging

}

boolean flag=false;

void loop()

{

  auto\_tracking();

} //end of loop

char sensor[5];

 /\*read sensor value string, 1 stands for black, 0 starnds for white, i.e 10000 means the first sensor(from left) detect black line, other 4 sensors detected white ground \*/

String read\_sensor\_values()

{   int sensorvalue=32;

  sensor[0]= !digitalRead(LFSensor\_0);

  sensor[1]=!digitalRead(LFSensor\_1);

  sensor[2]=!digitalRead(LFSensor\_2);

  sensor[3]=!digitalRead(LFSensor\_3);

  sensor[4]=!digitalRead(LFSensor\_4);

  sensorvalue +=sensor[0]\*16+sensor[1]\*8+sensor[2]\*4+sensor[3]\*2+sensor[4];

  String senstr= String(sensorvalue,BIN);

  senstr=senstr.substring(1,6);

  return senstr;

}

void auto\_tracking(){

 String sensorval= read\_sensor\_values();

  Serial.println(sensorval);

  // This makes sure there is no object in front of the robot.

  if (watch() < 10) {

    stop\_Stop();

    alarm();

  }

  else {

    if (   sensorval=="10000" ) {

      //The black line is in the left of the car, need  left turn

      go\_Left();  //Turn left

      set\_Motorspeed(FAST\_SPEED,FAST\_SPEED);

    }

    if (sensorval=="10100"  || sensorval=="01000" || sensorval=="01100" || sensorval=="11100"  || sensorval=="10010" || sensorval == "00100" || sensorval=="11010") {

      go\_Advance();  //Turn slight left

      set\_Motorspeed(0,FAST\_SPEED);

    }

    if (    sensorval=="00001"  ){ //The black line is  on the right of the car, need  right turn

      go\_Right();  //Turn right

      set\_Motorspeed(FAST\_SPEED,FAST\_SPEED);

    }

    if (sensorval=="00011" || sensorval=="00010"  || sensorval=="00101" || sensorval=="00110" || sensorval=="00111" || sensorval=="01101" || sensorval=="01111"   || sensorval=="01011"  || sensorval=="01001")

    {

      go\_Advance();  //Turn slight right

      set\_Motorspeed( FAST\_SPEED,0);

    }

    // Stops if no line is encountered.

    if (sensorval == "00000") {

      stop\_Stop();

      set\_Motorspeed(0, 0);

    }

    if (sensorval=="11111"){

      stop\_Stop();   //The car front touch stop line, need stop

      set\_Motorspeed(0,0);

    }

    // This slight delay gives the robot time to detect the different inputs.

    delay(90);

  }

}

**Conclusion**

The combination of this hardware and software demonstrates the power of line tracking. The robot is able to understand a path the is created which is really practical in the real world. This means software can easily be created to follow these types of paths and the programmer would never have to know what the path looks like. We can see there a different methods when it comes to the robot following the line and in this example there was a lot of oscillating with the robot since there is a lot of factors when it comes to writing PID involved code. We could see that the robot behaved exceptionally well when handling the curves and sharp turns and the robot never deviated away from the path. The robot handled the paths correctly and any given speed although it was observed the oscillation was greater with higher speeds. With all this in mind combining the line tracking technology along with the sonar device allowed the creation of a powerful program. The robot was able to demonstrate that it could follow a path all while keeping note of any objects it might encounter. Once again, the applications of this are endless in the real world and this can be expanded on immensely. Overall, the robot is becoming very versatile and can complete many tasks that it may be given and it becomes even more powerful when combining all the hardware together.