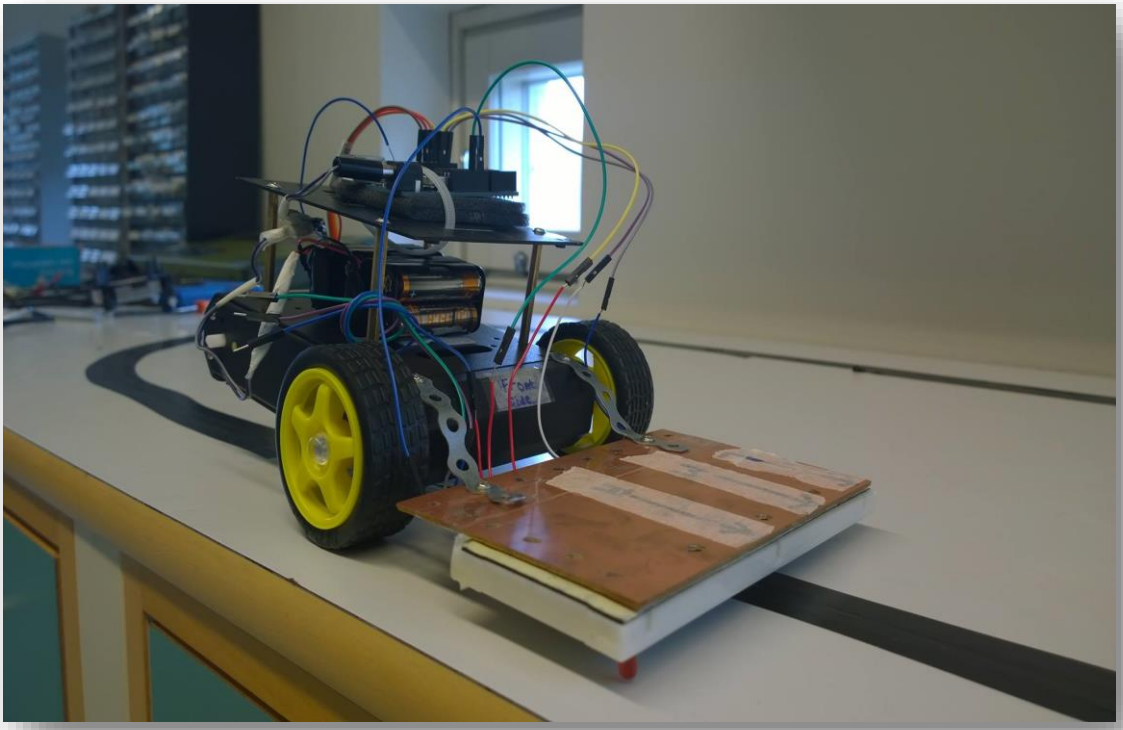

MetroBot

Line Following And Light Following Robot

Amrit Gautam
Hailu Dibaba
Karar Mehdi Habib

15th May 2014

Helsinki Metropolia University of Applied Sciences

Contents

1 Introduction

There are several types of robot navigation. Sensor based and vision based systems are the two major categories of them. Sensor based approach uses various kinds of sensors such as IR sensors and ultrasonic sensor. IR sensors are generally used for measuring the difference in reflectivity of surfaces depending on the properties like color, roughness. Ultrasonic sensors are used to measure the distance to an object. Robots are usually built to navigate using these outputs according to the application.

The objective of the project was to build an obstacle avoiding robot using the Beagle-Bone Black microcomputer. We were unable to find the BeagleBone Black microcomputer so we deviated from our conceive plan to the line following and light following robot using Arduino microprocessor. Arduino microprocessor is readily available and cheap. Line following robot is a machine that can follow a black line. Sensing a line and maneuvering the robot to stay on course while constantly correcting wrong moves using feedback mechanism forms a simple yet effective closed loop system. On the other hand, light following robot can follow a bright light source. Building such kind of robot is like giving it a human-like property of responding to stimuli.

First of all, we would like to thank Sir Tero Nurminen, Sir Joseph Hotchkiss and Sir Timo Lepola for their great support, help and guidelines on making this robot. It would not be possible to accomplish the project without their guidelines.

2 Team Members

There were three members in our team. The names are as follows-

- Amrit Gautam
- Hailu Dibaba
- Karar Mehdi Habib

3 Goals

According to the conceive plan, our aim was to build a robot that can avoid obstacles in its way. Due to the shortage of BeagleBone board, we changed to our plan to the line

following and light following feature on the robot along with the obstacle avoidance feature. At last we were successful to integrate two features on the robot:

1. Line following
2. Light following

We could not find a way for the obstacle avoidance feature for not having much time. So, we had to stick on with these two features.

4 Components Used

The line following and light following robot is made by using Arduino microprocessor. The main reason of choosing Arduino is because of its availability and easy user interface so that we can work again on it when we want to upgrade it. The hardware components that are used in making the robot are listed below:

1. Arduino Uno
2. Three IR sensors
3. Two LDRs
4. Two DC motors
5. Chasis
6. Breadboard
7. Resistors
8. One Castor wheel
9. Two 6V batteries
10. Wires

5 Project Timeline

The *Figure 1* below is an overall timeline of the distribution of our work in this project which shows the different steps that we took while building the robot.

17 March 2014 - 24 March 2014	Searched Beagle Bone Black (BBB) Board but could not found anywhere.
25 March 2014 - 10 April 2014	Studied about arduino to get familiar to the platform.
11 April 2014 - 18 April 2014	Driving DC motors using H-Bridge.
19 April 2014 - 26 April 2014	Integrating IR sensors for line following feature.
27 April 2014 - 2 May 2014	Integrating LDR sensors for light following feature.
3 May 2014 - 5 May 2014	Testing the robot.

Figure 1 Project TimeLine

6 Light Following Robot

6.1 Introduction:

MetroBot Light following robot is designed to find out the brightest source of light around its surrounding and move towards that light source. The robot uses Light Dependent Resistors (LDR) to find out the light source. It works on the principle that the resistance of a LDR which changes on the amount of light that falls on it. In general, the resistance of LDR goes minimum when the maximum amount of light falls on it and maximum when no light falls on it.

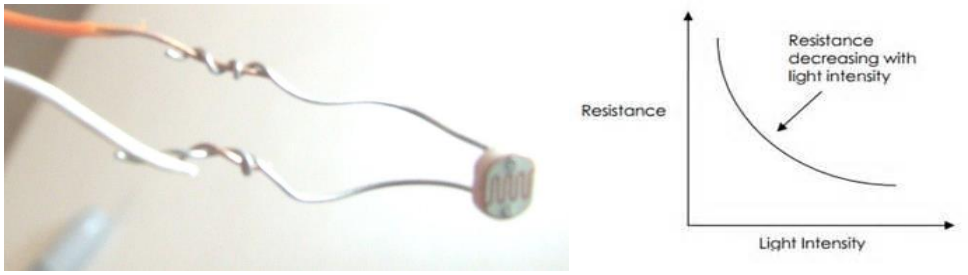

Figure 2 Light Dependent Resistor and Graph (Resistance vs Light) [1]

6.2 Working:

Two LDRs attached to the left and right side of the robot differentiate the dark and bright source of the light. If the difference between the intensity of light is large enough which falls on the LDRs then the robot makes it turns towards the bright light source (High Intensity of Light).

The circuit diagram for connecting the LDR with the Arduino is shown below:

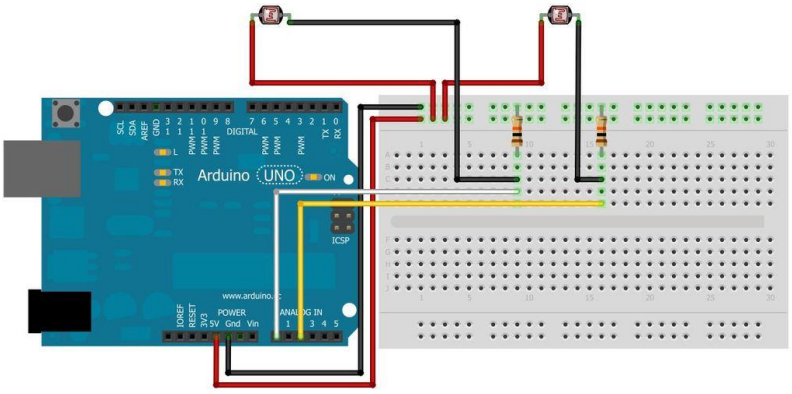

Figure 3 Circuit connection of Arduino and LDRs [2]

The programming code for the light following robot is given in appendix 1.

7 Line following Robot:

7.1 Introduction:

MetroBot line following robot is a robot that can follow a certain line of specified color on the floor. It can be programmed to follow a black line on a white surface or a white line on a black surface. The robot gets information about the different colors on the surface from the Infrared sensors (TCRT5000 IR Sensors) mounted on the robot chassis facing the floor. Based on the color information from the sensors, the microcontroller varies the speed and direction of the motors on each side of the robot.

Metrobot has two 6V DC geared front motor wheels and one rear castor wheel. When both motors run in the same direction with the same speed, the robot will move straight. When there is speed variation between the two motors or either of the motors is turned OFF while the other is running, the robot will turn to the side running at slower speed.

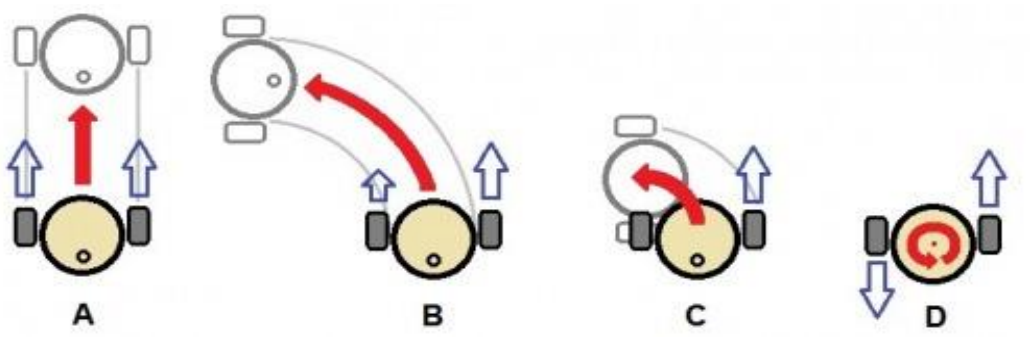

Figure 4 Robot turning based on speed variation. [3]

At the heart of all the processes is the Arduino Uno microcontroller. The Arduino Uno microcontroller drives the IR sensors; receives sensor outputs and controls the wheels accordingly.

7.2 Working:

The Dual H-Bridge Motor Driver:

The motor driver used in the MetRobot is the L298N dual h-bridge based DC motor driver module. It is a high voltage, high current dual full bridge motor driver that can run two motors in different modes. It can control the speed and direction of rotation of each motor.

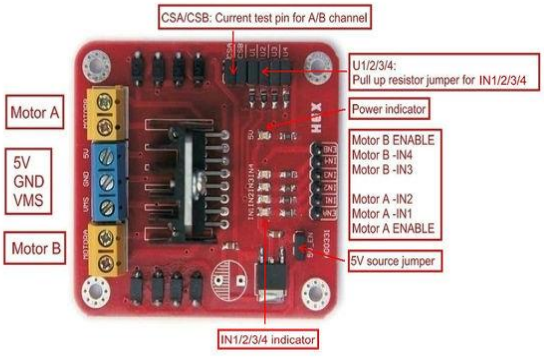

Figure 5 L298N Motor driver module [3]

- CSA (CSB): are current sensing pins used to measure the amount of current each motor is drawing.
- 5V: Power input for the logic circuit on the board.

- EA/EB: is the enable pin for the two motors A/B, the motor speed also can be controlled by the PWM of this pin.
- IN1/2/3/4: are the pins to be connected to the Arduino digital pins. Both motors will take values for these inputs from the Arduino.

So the motors are controlled using the four inputs from the Arduino. Different combinations of these values rotate the motor in different orientation.

Table 1 Input combinations for the motors:

Motor A

Input		Function
EA=H	IN1=H, IN2=L	Forward
	IN1=L, IN2=H	Reverse
	IN1 = IN2	Fast Motor Stop
EA=L	IN1=X, IN2=X	Free Running Motor Stop

L = Low H = High X = Don't Care

The same rule applies for Motor B.

TCRT 5000 reflective sensor:

The TCRT5000 is a reflective optical sensor which includes an infrared emitter and phototransistor. The emitter emits lights of wavelength of 950nm.the phototransistor is the detector part of the package. Based on the nature of the surface it's facing, the phototransistor's collector voltage varies.

When a black object is faced, much of the incident rays are absorbed, so there will not be much reflected rays falling back on the phototransistor. The digital sensor value to the Arduino in this case will be HIGH or 1.When a white object is faced, much of the incident rays are reflected. These reflected rays will fall on the phototransistor. The digital sensor value to the microcontroller in this case will be LOW or 0.

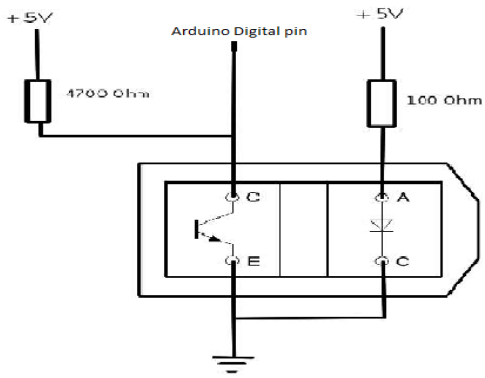

Figure 6 IR pin connection.[3]

Three IR sensors have been used in the MetRobot and placed with equal spacing between them across the robot chassis; one on the left, one in the middle and the other one on the right side of the robot.

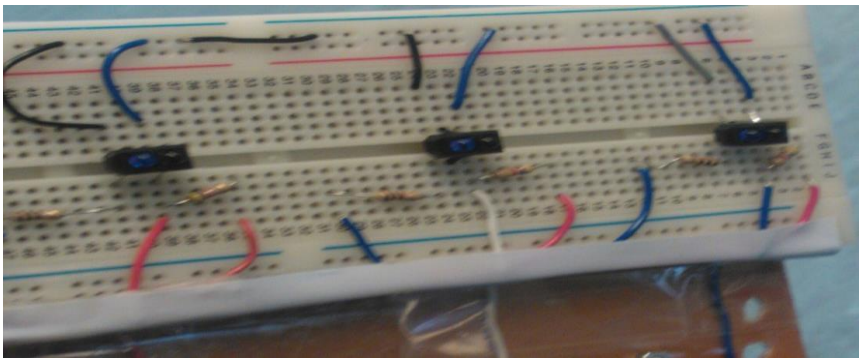

Figure 7 IR sensors underneath the chassis

The black line has been made in such a way that it will be facing the IR sensor in the middle. When the robot is following the black line, the IR sensors on the left and right will not detect any black line. So when the middle IR sensor detects a black line, both wheels should rotate at the same speed and in the same direction to keep track of the black line without a need for turning.

When the sensor on the left detects a black line, it means that the robot is moving away from the black line to the right. So the Arduino will turn OFF the left wheel's motor and keep the right wheel's motor ON, causing the robot to turn to left until the IR sensor in the middle detects the black line.

When the sensor on the right detects a black line, it means that the robot is moving away from the black line to the left. So the Microcontroller will turn OFF the right wheel's motor and keep left wheel's motor ON, causing the robot to turn to right until the IR sensor in the middle detects the black line

.

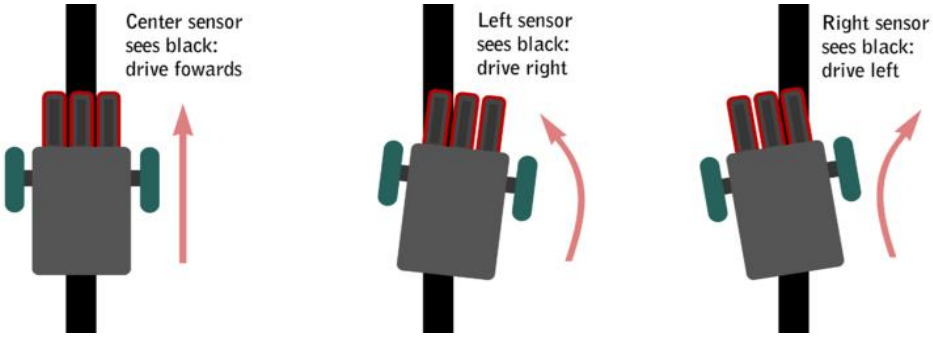

Figure 8 Robot orientation based on IR sensors [4]

The programming code for the light following robot is given in appendix 2.

8 Problems and solutions

There were many problems that our team faced while making the robot but the number of majorities were three. The first problem that we faced was finding the Beagle Bone Black Board (BBB) which we could not manage from any shops or internet shops after a long search. As a result, we had to postpone our project because of switching our plan of making the robot using Arduino about which we had to study for a period of time to get familiar with it.

Secondly, the two motors that are situated at the back of the robot were not rotating according to our plan. So, we had to replace them with a Castor. The last major problem that we faced was that the noises from the motors were affecting the inputs of Ultrasonic sensors which are related to the obstacle avoiding feature. As a result, we had to skip working on the obstacle avoiding feature of the robot for not having much time because of the deadline.

9 Conclusion

After two months of hard work, my team finally paved the way to the final destination of the project solving various difficulties along the way. We deviated from our set plan due to lack of hardware materials but also we were able to complete the project as per schedule.

Doing this project with the international students was real fun and challenging. In the beginning of the project, it was difficult for us to explain our ideas and views. As the time passed by, we begin to understand and implement our ideas and viewpoint. We applied the knowledge that we learned from our lecturers in the practical way through this project. Group work, proper time division, problem solving abilities are increased among all my team members through this project.

References

1. Light following robot [online]; 10 April 2014
   URL:http://cdn.instructables.com/FBY/X3OS/H82UEKAN/FBYX3OSH82UEKAN.LARGE.jpg
2. Light following robot [online]; 10 April 2014
   URL: http://www.instructables.com/file/FRS8CXSH82UEKAH
3. Line following robot [online]; 18 April 2014
   URL: http://letsmakerobots.com/node/37225
4. Line following robot[online]; 18 April 2014
   URL:http://www.robotc.net/wiki/Tutorials/Arduino_Projects/Mobile_Robotics/VEX/Using_the_line_tracker_to_follow_a_line
5. TCRT5000 datasheet [online]: 1 April 2014
   URL: http://www.vishay.com/docs/83760/tcrt5000.pdf
6. L298n dual full bridge motor driver datasheet [online]: 2 April 2014
   URL: https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

Appendix 1

Arduino code for Light following robot:

The following code is written and programmed in the AT-mega 328p micro-processor which is located in inside Arduino-board. This code drives the robot to follow the bright light source.

```
#define IN1 2   //IN1 on pin 2 controls one side of bridge A
#define IN2 4   //IN2 on pin 4 controls other side of A
#define IN3 6   //IN3 on pin 6 controls one side of bridge B
#define IN4 7   //IN4 on pin 7 controls other side of B
const int SENSORr = A0;
const int SENSORl = A2;
int LeftSensor = 0;
int RightSensor = 0;
int SENSORDIFFERENCE = 0;

void setup(){
Serial.begin(9600);
}
void loop() {
LeftSensor= analogRead(SENSORl);
RightSensor= analogRead(SENSORr);
SENSORDIFFERENCE = abs(LeftSensor - RightSensor);
Serial.println("Left Sensor");
Serial.println(LeftSensor);
Serial.println("Right Sensor ");
Serial.println(RightSensor);
if (LeftSensor > RightSensor && SENSORDIFFERENCE >75) { //Turns
Left as left sensor detects Bright light
Serial.println("RightTurn");
TurnLeft();

}
if (LeftSensor < RightSensor && SENSORDIFFERENCE > 75) {
//Turns Right as Right sensor detects Bright light
Serial.println("LeftTurn");
```

Appendix 1

Arduino code for Light following robot:

```
(....continue)

TurnRight();
}
else if (SENSORDIFFERENCE <75){
Serial.println("Forward");
Forward();
}
}
void Forward(){                          //Moves Forward
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW );
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW );
}
void TurnRight(){                      //Turns Right side
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW );
}
void TurnLeft(){                       // Turns Left side
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW );
}
```

Appendix 2

Arduino code for Line following robot:

The following code is written and programmed in the AT-mega 328p micro-processor which is located in inside Arduino-board. This code drives the robot to follow the black line.

```
int inputPin1 = 2;
int inputPin2 = 3;
int inputPin3 = 4;
int val1 = 0;
int val2 = 0;
int val3 = 0;
#define IN1 5  //IN1 on pin 2 conrtols one side of bridge A
#define IN2 6  //IN2 on pin 4 controls other side of A
#define IN3 7  //IN3 on pin 6 conrtols one side of bridge B
#define IN4 8  //IN4 on pin 7 controls other side of B
void setup() {
  pinMode(inputPin1, INPUT);     // declare pushbutton as input
  pinMode(inputPin2, INPUT);
  pinMode(inputPin3, INPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
void loop(){
  val1 = digitalRead(inputPin1);  // read input value
 if (val1 == HIGH ) {            // check if the input is HIGH
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(50);
  }
    val2 = digitalRead(inputPin2);  // read input value
  if (val2 == HIGH ) {             // check if the input is HIGH
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
```

Appendix 2

Arduino code for Line following robot:

```
(...continue)
digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(50);
  }
  val3 = digitalRead(inputPin3);  // read input value
  if (val3 == HIGH ) {            // check if the input is HIGH
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
    delay(50);
  }
}
```