International Journal of
Science and Engineering Investigations vol. 5, issue 53, June 2016
ISSN: 2251-8843

An Intelligent Line Following Robot with Obstacle Detection

MD. Abdul Awal Ansary[1], Mir Mehedi Hasan[2]
[1]Assistant Professor, Dept. of Computer Science and Engineering, Sylhet International University, Bangladesh
[2]IT Officer, Shafi Consultancy Bangladesh Ltd, Sylhet, Bangladesh
([1]awal_sust@yahoo.co.in)

Abstract- The Line following robot is a mobile machine that can detect and follow a line drawn on the floor. Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted color or it can be invisible like a magnetic field. Therefore, this kind of Robot should have sense to detect the line with its Infrared Ray (IR) sensors that installed under the robot. After that, the data is transmitted to the processor by specific transition buses. Hence, the processor is going to decide the proper commands and then it sends them to the driver and thus the path will be followed by the line following robot. It can also detect any obstacle in front of it while it is running on the line and provides a signal about the obstacle. This line following robot designed and tested for increasing skill and knowledge on robotics. But it encounters some technical and mechanical problems. In this paper, we have illustrated the process of design, implementation and testing a line following robot. The technical and mechanical issues and problems also have investigated.

Keywords- *Robot, Obstacle, Comparator, IR, LED, DC Motors, Microcontroller*

I. INTRODUCTION

The line following robot, operates as the name specifies. It is programmed to follow a dark line on a white background and detect turns or deviations and modify the motors appropriately. The optical sensor is an array of commercially available IR reflective type sensors. The core of the robot is the PIC 16F72 microcontroller.

The H-Bridge motor driving/control chip takes the signals and translates it into current direction entering the motor armature. The motors require separate power supply for operation. The differential steering system is used to turn the robot. In this system, each back wheel has a dedicated motor. To move in a straight line, both the motors are given the same voltage (same polarity). To manage a turn of different sharpness, the motor on the side of the turn required is given lesser voltage. To take a sharp turn, its polarity is reversed.

The sensor is an array of 4 IR LED-Phototransistor pairs arranged in the form of an inverted V. One is used to obstacle detection and the remaining three are used to move forward and turn toward right or left. The output of each remaining three sensor is fed into an analog comparator with the threshold voltage (used to calibrate the intensity level difference of the line with respect to the surface). These three signals (from each photo-reflective sensor) is given to a priority encoder, the output of which to the microcontroller. The control has six modes of operations, turn left/right, move left/right, and drift left/right. The actual action is caused by controlling the direction/speed of the two motors (the two back wheels), thus causing a turn. The actual implementation is a behavior based (neural) control with the sensors providing the inputs.

A. *Objectives:*

The objectives of our Line Following Robot are

- This will follow a black line path on the floor.
- It will stop within a safe distance of an object when any obstacle is in front of on its path.
- This will give a signal of warning or notification that there is an obstacle or object on its path.
- This will automatically continue when the obstacle is moved.

B. *Equipments:*

- **Microcontroller:** PIC 16F72 is a small and effective microcontroller. This microcontroller is used as a core of this robot. Everything is maintained by this microcontroller.
- **Battery:** This robot is powered by a battery. One 9V battery is enough to perform the process. For more usages, four 9V batteries may be required.
- **6V DC motor:** If a DC power is passed on a DC motor, it will produce torque. The torque created will lead to the rotation of the wheels. It will only operate on the direct current. Here, two 12V DC motors are used.
- **Two plastic wheels:** The plastic wheels will be connected to the DC motors. As soon as they create the torque, these wheels will help the robot to move.
- **IR sensor circuit:** This process requires 4 IR sensor circuits. An IR sensor circuit will include an IC LM358N, potentiometer, IR receiver, and IR transmitter for sensing the black lines and obstacle.

- **IC LM358N:** It is an operational amplifier or comparator used to evaluate the voltage current. The high amount of voltage gained will be considered as the output. It is one of the main components of an IR sensor circuit.
- **IC 7805:** The output given from the battery to the circuit is 9V. In this process, 5V of power supply is sufficient to operate. For that reason, an IC 7805 voltage regulator is used to control the high amount of voltage.
- **IC L293D:** It allows the DC motor to run in both front and back directions. It consists of up to 16 pins.
- **Wires (four-core & two-core):** Four core wires are implemented for connecting the motor drives, and the two core wires are used to connect on the breadboard. Two meters of each wire will be required.
- **LED:** The glowing of the LED makes sure that the current is flowing in the circuit. If the LED does not glow, it is considered that there is some problem with the connection.

II. SCHEMATICS

The schematic of the "Line following robot with obstacle detection" is shown in the figure. The main component is the PIC 16F873 microcontroller. Due to page limitations, the schematic is divided into two sections; one the Sensor Array Board, and the other the motor-control or main board.

The main features incorporated into the hardware are given below:

- The PIC 16F873 microcontroller
- The voltage regulator and supporting components
- Crystal oscillator (20MHz)
- Voltage Comparator IC (LM324)
- The H-bridge motor control IC (L293D)
- Motors, with coupled reduction gears
- 6V, 1.2AH Lead-Acid battery
- IR interrupt sensor, modified to be a reflective sensor
- A POT to calibrate the reference voltage
- Connectors to join the different boards to form one functional device

A. Block Diagram:

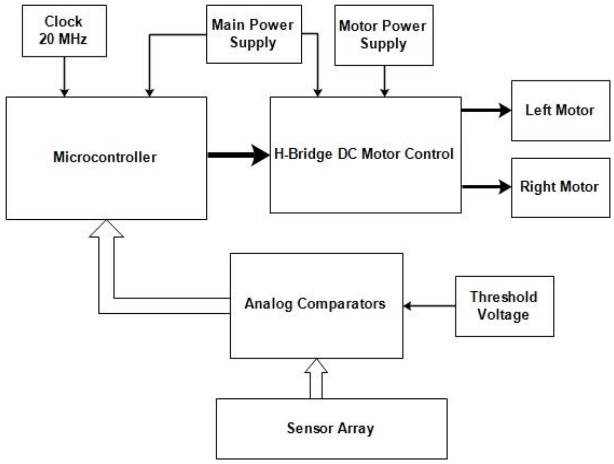

Figure 1. Block Diagram of the Line Following Robot with obstacle detection

B. DC Motors:

DC motors are widely used, inexpensive, small and powerful for their size. Reduction gearboxes are often required to reduce the speed and increase the torque output of the motor. Unfortunately more sophisticated control algorithms are required to achieve accurate control over the axial rotation of these motors. Although recent developments in stepper motor technologies have come a long way, the benefits offered by smooth control and high levels of acceleration with DC motors far outweigh any disadvantages. Several characteristics are important when selecting DC motors and these can be split into two specific categories. The first category is associated with the input ratings of the motor and specifies its electrical requirements, like operating voltage and current. The second category is related to the motor's output characteristics and specifies the physical limitations of the motor in terms of speed, torque and power.

C. Motor driver L293D:

In a line following robot, usually the motor is powered by a source from the main circuit so that the motor will move faster and be more powerful. This IC works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. Voltage needs to change motor's direction for being able to rotate the motor in clockwise or anticlockwise direction. Hence H-bridge IC is ideal for driving a DC motor.

In a single l293D chip, there are two H-bridge circuits inside the IC which can rotate two DC motors independently. Due to its size, it is very much used in robotic application for controlling DC motors. Given below, the pin diagram of a L293D motor controller.

There are two Enable pins on l293D. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge you need to enable pin

1 to high. And for right H-bridge you need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend working.

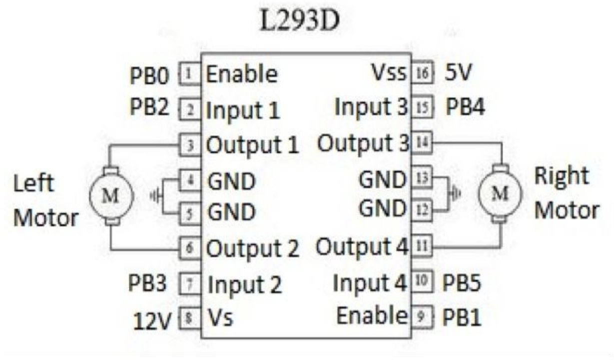

Figure 2. L239D Block Diagram

D. The PIC 16F72 Microcontroller:

This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 28-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. The PIC16F72 features 5 channels of 8-bit Analog-to-Digital (A/D) converter with 2 additional timers, capture/compare/PWM function and the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus. All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

TABLE I. SPECIFICATIONS

Parameters Names	Values
Program Memory Type	Flash
Program Memory (KB)	3.5
CPU Speed (MIPS)	5
RAM Bytes	128
Digital Communication Peripherals	1-SSP(SPI/I2C)
Capture/Compare/PWM Peripherals	1 CCP
Timers	2 x 8-bit, 1 x 16-bit
ADC	5 ch, 8-bit
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	2 to 5.5
Pin Count	28

E. IR sensors:

For this project, we have used three IR sensors which are attached to the bottom of the robot. These three sensors will be classified as left sensor, middle sensor and right sensor. A view of the placement of the sensors is as below:

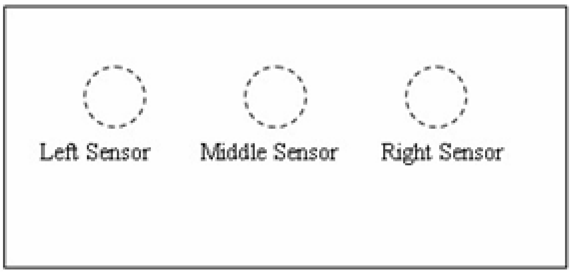

Figure 3. IR Sensors

The distance between 2 sensors depends on the width. The sensor should be placed in such a way that maximum distance of two sensors is equal to the width of the line as shown in figure below.

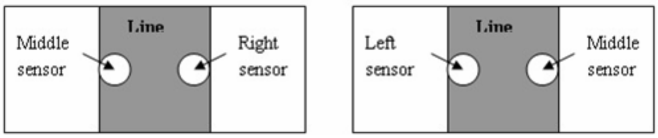

Figure 4. Distance between left and right sensors

To teach the robot for line detection, one may follow the following steps:

1. Adjust the robot so that the center infrared sensor is on top of white floor; make sure the wheels and castor of the robot touches the floor properly.

2. Use the screw driver to adjust the preset of center sensor until indicator LED (center) light ON.

3. Now adjust the robot to move the center sensor towards the black line where the reflection of infrared is poor.

4. At this point, make sure the indicator LED is OFF. If the LED is still on, it means you have over tune the preset. Tune it back so that the indicator is OFF.

5. Repeat step 1 To 4 for a few time and make sure the indicator LED ON and OFF correctly at the right spot.

6. Repeat step 1 To 5 to "teach" left sensor and right sensor.

7. Now the robot has been "taught".

Let’s see how this robot can follow line based on the response from each sensor. In our discussion, we will bypass the comparator by assuming that this robot has been taught. Once a particular sensor sensed black line, it will trigger the PIC controller. Example of how the sensors function when the robot follows a black line on a white floor:

TABLE II. ROBOT'S RESPONSE REGARDING SENSORS SENSES

Sensor				Response
Obstacle Sensor	**Left**	**Middle**	**Right**	
Yes	W	B	W	Stop
Yes	B	W	W	Stop
Yes	W	W	B	Stop
No	W	B	W	Go Straight
No	B	W	W	Turn Left
No	W	W	B	Turn Right

Where, W = White, B = Black

This is the concept of a line following robot. We can see that when the middle sensor detects the line, the robot will move forward if the obstacle sensor status is No. This is because the line is in the center of the robot. But when the left sensor or the right sensor detects the black line, this means that the robot have strayed from the line. If the robot strayed to the right, the left sensor will detect the line and the brain will react by turning left to go back to the line. As for the right sensor, it reacts in the opposite way. Above process is depended on the obstacle sensor.

III. PROGRAMMING CODE

```
// Line Follower Robot

// Sensor Input Must be B0 to B3
// Motor Output Must be C0 to B6

// Description:
// B0 = Input from Left Sensor.
// B1 = Input from Middle Sensor.
// B2 = Input from Right Sensor.
// B3 = Input from Obstacle Sensor.

// C0 = Output for Left Motor Enable.
// C1,C2 = Output for Left Motor PWM.
// C3 = Output for Right Motor Enable.
// C4,C5 = Output for Right Motor PWM.

 // Robot Motions
//    Stop 0b00000000
//    Left 0b00000011
//    Right 0b00011000
//    Forward 0b10011011


unsignedint j[1];

voidob()
   {      PORTC = 0b00000000;      // Stop}

void main() {

  // Port Configure
  TRISA = 0b11111110;   // Port A All output define
  TRISB = 0b11111111;   // Port B All Input Define
  TRISC = 0b00000000;   // Port C All output Define

  // All output make zero/ Low First
  PORTA = 0b00000000;
  PORTB = 0b00000000;
  PORTC = 0b00000000;
Delay_ms(5000);   // wait 5000ms= 5 second

while(1)
  {if ((PORTB.F0==0 && PORTB.F1==0 && PORTB.F2 ==0) ||
(PORTB.F0==1 && PORTB.F1==1 && PORTB.F2 ==1))   // Left, Middle
& Right = Low.   && PORTB.F3==0
        {if(PORTB.F3 == 0)
          {              PORTC = 0b10011011;      // Go Forward
j[1] =  0b10011011;          }
else
          {ob();          }  }
else if (PORTB.F0==0 && PORTB.F1==0 && PORTB.F2 ==1)   // Left,
Middle = Low, Right = High    && PORTB.F3==0
        {if(PORTB.F3 == 0)
          {              PORTC = 0b00000011;      // Turn Left
j[1] =  0b00000011; }
else
          {ob();}}
else if (PORTB.F0==1 && PORTB.F1==1 && PORTB.F2 ==0)   // Right,
Middle = High, Left = Low    && PORTB.F3==0
        {if(PORTB.F3 == 0)
          {              PORTC = 0b00011000;      // Turn Right
j[1] =  0b00011000;}
else
          {ob();}}
else if (PORTB.F0==1 && PORTB.F1==0 && PORTB.F2 ==1)   // Left,
Right = Height & Middle = Low && PORTB.F3==0
        {if(PORTB.F3 == 0)
          {              PORTC = 0b10011011;      // Go Forward
j[1] =  0b10011011;  }
else
          {ob();   }}

else if (PORTB.F0==1 && PORTB.F1==0 && PORTB.F2 ==0)   // Right,
Middle = Low, Left = High    && PORTB.F3==0
        {if(PORTB.F3 == 0)
          {              PORTC = 0b00011000;      // Turn Right
j[1] =  0b00011000; }
else
          {ob();          }         }

else if (PORTB.F0==0 && PORTB.F1==1 && PORTB.F2 ==1)   // Left,
Middle = Low, Right = High    && PORTB.F3==0
        {if(PORTB.F3 == 0)
          {              PORTC = 0b00000011;      // Turn Left
j[1] =  0b00000011;           }
else
          {ob();          }         }
else
        {          PORTC = j[1];         }   } // end while}
```

IV. APPLICATIONS

- Industrial automated equipment carriers.
- Entertainment and small household applications.
- Automated cars.
- Tour guides in museums and other similar applications.
- Second wave robotic investigation operations.

V. RESTRICTIONS

- Choice of line is made in the hardware abstraction and cannot be changed by software.
- Calibration is difficult, and it is not easy to set a perfect value.

- The steering mechanism is not easily implemented in huge vehicles and impossible for non-electric vehicles (petrol powered).
- Few curves are not made efficiently, and must be avoided.
- Lack of a four wheel drive, it is not suitable for a rough terrain.
- Use of IR even though solves a lot of problems pertaining to interference, makes it hard to debug a faulty sensor.
- Lack of speed control makes the robot unstable at times.
- If the sensors detect any others IR source it will be not working properly.
- It has to be battery powered and be able to operate for up to 2 hours without having to be recharged.

VI. CONCLUSION

Finally the Line Following Robot with Obstacle Detection is completed. A lot of effort was put into the design, implementation and days of toil in front of the computer, writing and debugging the code. The robot was finally running with a few glitches here and there which were sorted in the later revisions of the firmware. The line following robot with obstacle detection still has a few short-comings but we achieves most of the objectives.

VII. FUTURE SCOPE

The robot can be further enhanced to let the user decide whether it is a dark line on a white background or a white line on a dark background. The robot can also be programmed to decide what kind of line it is, instead of a user interface. The motor control could be modified to steer a conventional vehicle, and not require a differential steering system. The robot could be modified to be a four wheel drive. Extra sensors could be attached to allow the robot to bypass obstacle and get back to the line. In other words, it must be capable predicting the line beyond the obstacle. Speed control could also be incorporated using PWM (Pulse width modulation) technique. Position and distance sensing devices could also be built in which can transmit information to a mother station, which would be useful in tracking a lost carrier.

VIII. ACKNOWLEDGEMENT

The acknowledgement should go to the Center for Research, Testing and Consultancy (CRTC), Department of Computer Science and Engineering (CSE), Sylhet International University (SIU) for providing the total funding to develop this project including the Laboratory facilities.

REFERENCES

[1] Digital Logic and Computer Design, by M. Morris Mano, 4th edition, publication- Prentice – Hall of India PVT limited

[2] Digital Systems Principles & Applications, by Ronald J. Tocci, Sixth Edition, publication - Prentice – Hall of India PVT limited

[3] Design with PIC Microcontrollers, by John B. Peatman – PEARSON Education

[4] PIC Micro Mid-Range MCU Family Reference Manual, by MICROCHIP

[5] PIC Robotics, A beginner's guide to robotics projects using the PICmicro, by John Iovine – McGraw Hill

[6] Programming and Customizing the PIC microcontroller, By Myke Predko Second edition – McGraw Hill