

A Line Follower Robot from design to Implementation: Technical issues and problems

Abstract— the Line follower robot is a mobile machine that can detect and follow the 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 sense 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 follower robot.

Keywords—Line Follower; Problems and solutions; Circuit; Actuator; Programming.

I. INTRODUCTION

Generally, the line follower robot is one of the self-operating mobile machines that follows a line drawn on the floor. The path can be a visible black line on a white surface (reverse). The basic operations of the line follower are as follows:

- Capturing the line position with optical sensors mounted at the front end of the robot. Most are using several numbers of photo-reflectors. Therefore, the line sensing process requires high resolution and high robustness.
- Steering the robot to track the line with any steering mechanism. This is just a servo operation; actually, any phase compensation will be required to stabilize tracking motion by applying digital PID filter or any other servo algorithm.
- Controlling the speed according to the lane

○

condition. The speed is limited during passing a curve due to the friction of the tire and the floor.

This kind of robot can be used for military purposes, delivery services, transportation systems, blind assistive applications. Moreover, there are many annual line follower robots competitions organized by universities or industries around the world. They usually ask robotic teams for building a small robot with specific dimensions and weight according to the competition rules.

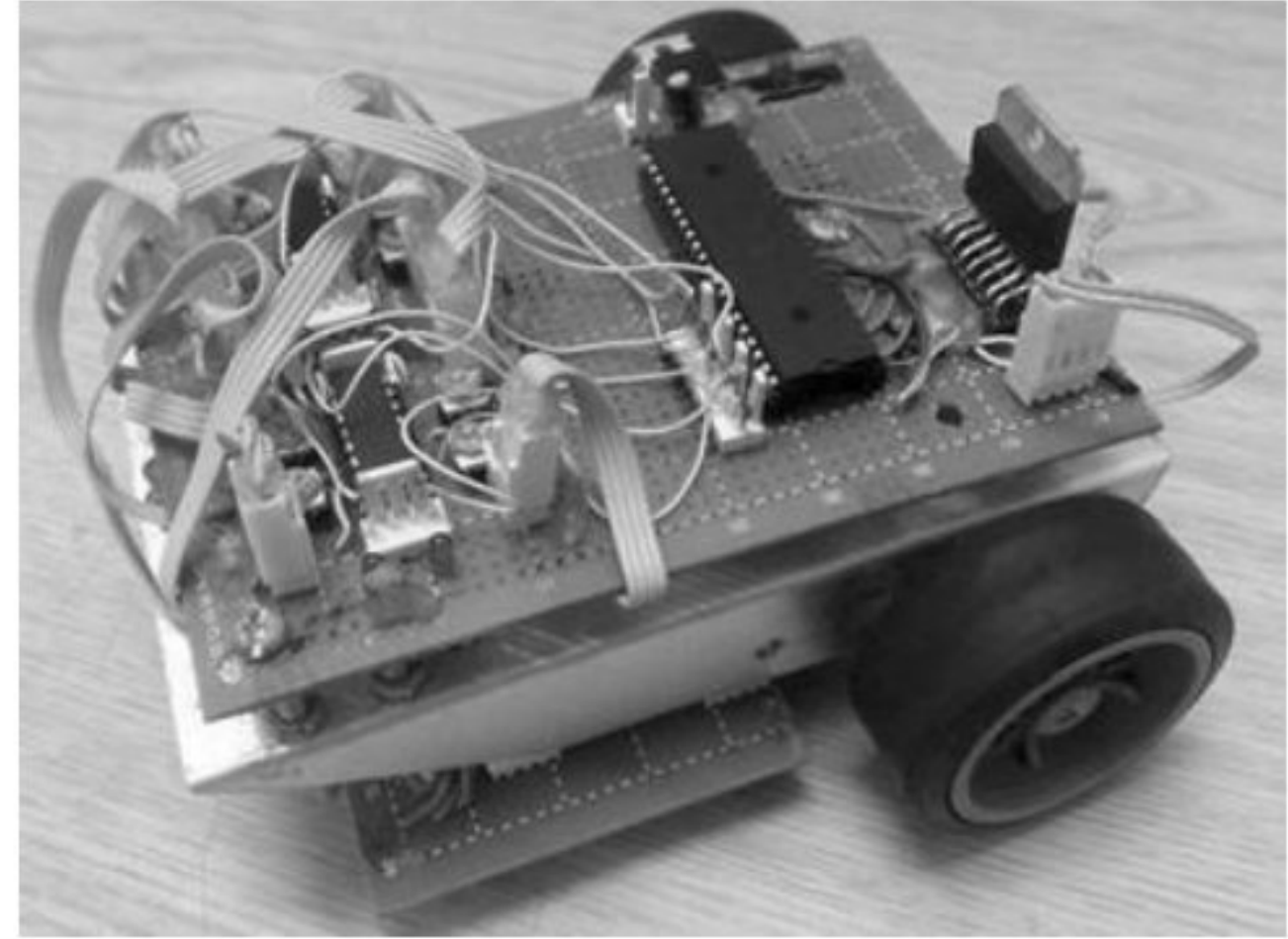


Figure 1. The Designed Line follower robot

In order to attend at Tabriz line follower robots competition, Tabari Institute of Babol had functional support for design, implementation and testing TABAR, a small line follower robot as can be seen in Fig. 1.

In this paper, we intend to share our experiences. Therefore, the line follower robot structure and architecture issues and challenges along with their technical issues and problems will be discussed in section 2 and the programming subjects will be explained in section 3 and the algorithms and solution of passing the lines will be illustrated in section 4.

II. LINE FOLLOWER ROBOT STRUCTURE

This robot can be divided into several parts:

- Sensors
- ADC (Analog to Digital Converter) and sensor circuit
- Processor
- Driver
- Actuators (Motors and wheels)
- Chassis and body structure

The electrical circuit of some line follower robots can compare the analog signal received from sensors and then transmit the result to the processor in digit '0' or '1' and some of them send the analog signals to the processor directly and the processor ought to convert them to digital form. Anyway, the analog signals must be converted to the digital form and then the processor can process them. [2]

The main elements of electrical structure are sensors, Analog to Digital Converter, Processor and Motor Driver.

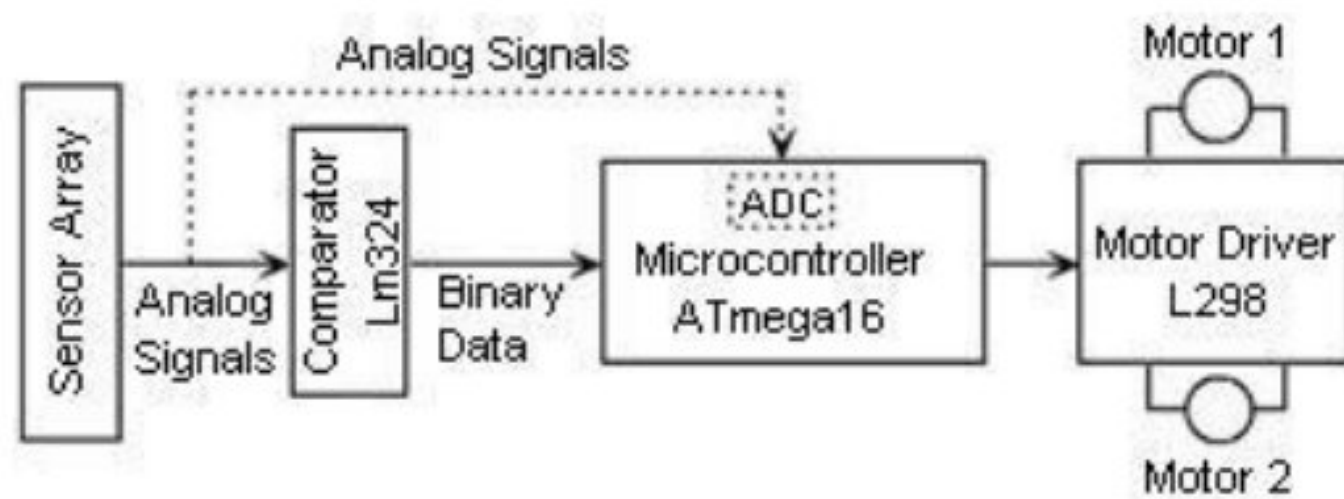


Figure 2. Line follower block diagram

A. The Sensors

This kind of robot uses Infrared Ray sensors to find the path and direction; Infrared Ray sensors contain a match infrared transmitter and infrared receiver pair. White surfaces generally reflect well, but while black surfaces reflect poorly. If the receiver receives the reflection ray, it means that the robot is on white and if it cannot receive it, so the robot is on black. These devices work by measuring the amount of light that is reflected into the receiver. Because the receiver also responds to ambient light, the device works best when well shielded from ambient light, and when the distance between the sensor and the reflective surface is small (less than 10mm). IR reflectance sensors are often used to detect white and black surfaces. White surfaces generally reflect well, but while black surfaces reflect poorly.

Hence, the distance between sensors and ground surface is important and it is more important that how we put sensors near each other. The distance between sensors and ground surface must be 2 to 10 mm and the distance between each sensor is dependent on the line width. [1]

In the designed robot, we have used eight sensors and they have suitable distance by each other as can be seen in Fig. 3 and our competition line width was 18 mm.

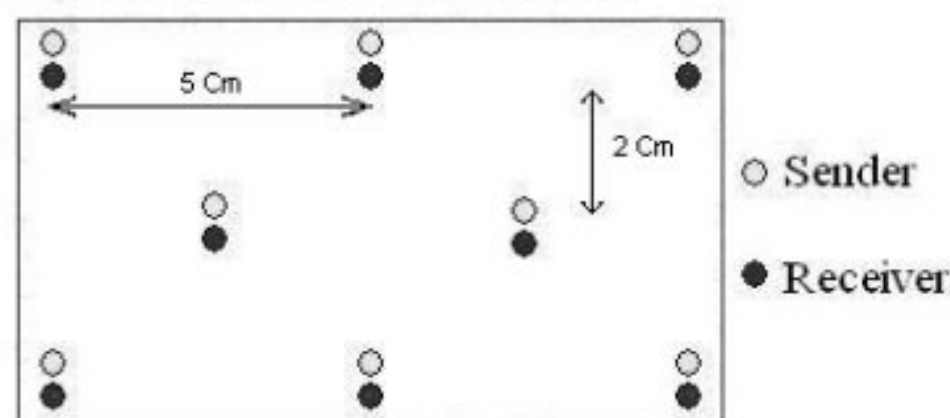


Figure 3. Sensors location method

If the line width is thin, the distance between sensors must be reduced; otherwise, while curving the line, the robot will not be turned on time.

B. The ADC and sensor circuit

Generally, the received signals from the sensors are analog and must be converted to the digital form. Therefore, the circuit can be designed to send the sensors' signals to the processor, directly. Hence, the processing time can be managed just by using an external ADC. LM324 is good ADC that we have used in our project. Two LM324 can support eight sensors.

The resistance of the receiver sensor is decreased when infrared is radiated on it.

A good sensor will have near zero resistance in the presence of the ray and have a very large resistance in the absence of the ray. We can use the property to form a potential divider. [2]

You can see schematic of sensor circuit in Fig. 4.

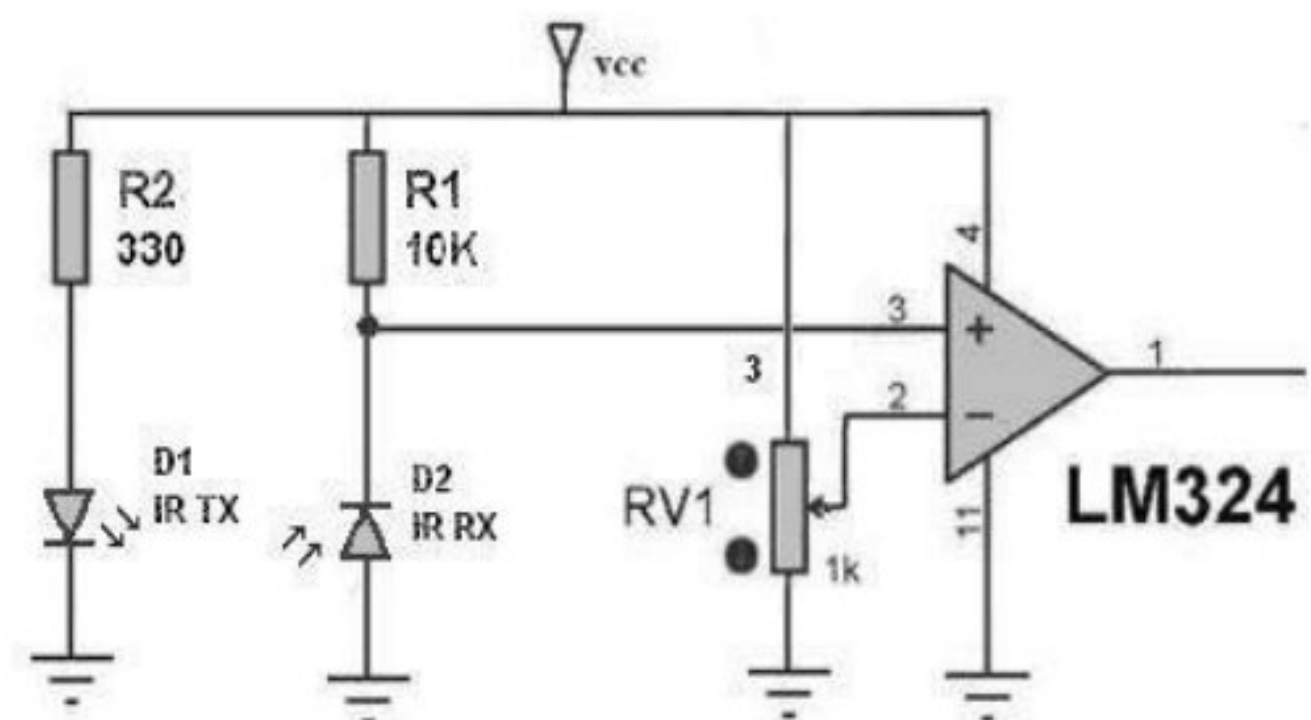


Figure 4. Schematic of a IR sensor

C. The Processor

For very small line follower that has 2 to 4 sensors, we can use electrical gate such as AND/OR to decide and send simple instruction to the driver but it hasn't any precision and its speed for processing is very low. Moreover, it can not follow bad curves or cycloid lines. It is momentous to use a processor, undoubtedly.

We have used the Atmel's AVR microcontroller "At Mega 16" in our project. The Atmel's AVR microcontrollers have a Reduced Instruction Set Coding (RISC) core running single cycle instructions and a well-defined I/O structure. Internal oscillators, timers, UART, SPI, pull-up resistors, pulse width modulation, ADC, analog comparator and watch-dog timers are some of the features you will find in AVR devices.

AVR instructions are turned to decrease the size of the program whether the code is written in C or Assembly. With on-chip in-system programmable Flash and EEPROM, the AVR is a perfect choice in order to optimize cost. [5, 3]

One of the best AVR is "At Mega 16" which has four ports for I/O and 16 MIPS speed in 16 MHz. The microcontroller power is 5V and it is better to use the 7805 regulator.

D. The Driver

Robot needs a driver IC for controlling and giving power to the motors. The microcontroller sends a signal to the driver which acts as a switch. The microcontroller only sends a signal to the switch and then the switch must give required voltage to the motors. If the received signal by the driver is high, it will rotate the motors.

A good motor driver is IC L298 which can be used to control two motors. It is an integrated monolithic circuit in 15-lead Multiwatt and Power SO20 packages. It is a highvoltage, high-current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC, and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. L298 has 2 amperes per channel current capacity and it can support up to 45 volts for outputting. Moreover, L298 works happily up to 16 volts without any heat sink.

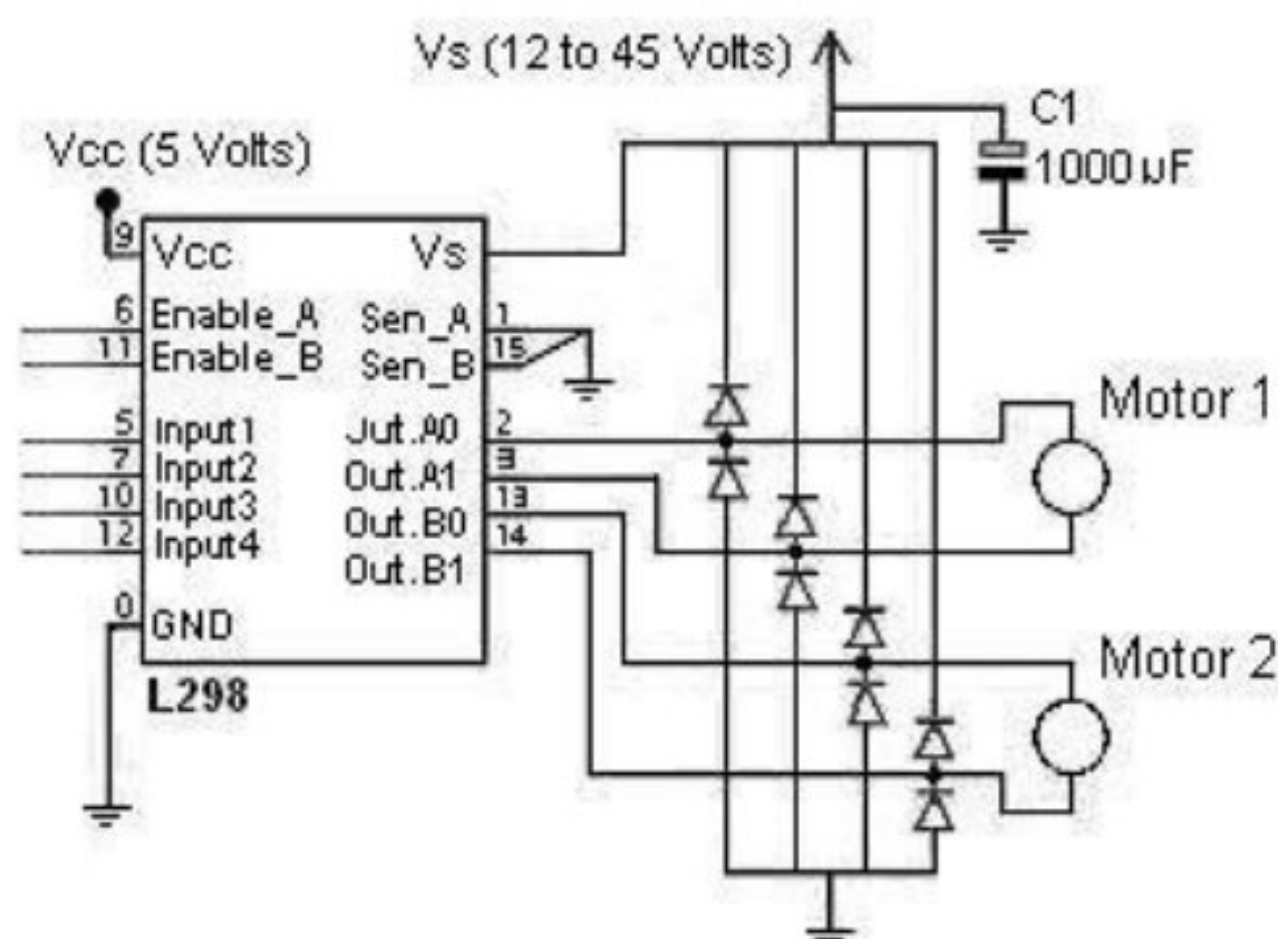


Figure 5. Driver and motors circuit

The microcontroller sends instructions to the driver after processing the data coming from sensors part. The driver gives voltage to the motors according to the inputs. Actually the driver gives positive voltage to one of the motor pins and gives negative voltage to another one which there is five states:

1. Both of the motors are turn on and rotate forward simultaneously. (Move Forward)
2. The right motor is turn on and the left motor is turn off. (Move Left)
3. The left motor is turn on and the right motor is turn off. (Move Right)
4. The right motor rotates forward and the left motor rotates backward. (Move Left Fast)
5. The left motor is rotate forward and the right motor rotates backward. (Move Right Fast)

• Usually two states are not practical in this kind of line follower robots:

1. Both of the motors are turn off.
2. Both of them rotate backward.

E. The Actuators (Motors and wheels)

The movement system is an important part of a robot and its objective is how to move robot from one point to another point. This system has some details which show us how to use motors and wheels. There are many kinds of motors and wheels. Our choice is dependent on the robot function, power, speed, and precision. [4]

At the beginning of the project, at first, we wanted to use two step motors for gaining the best speed and a remarkable control but it was too hard and sometimes impossible to write a good program for two step motors because each step motor has at least four inputs, and moreover we must use two motor drivers, L298, for getting the required voltage to the motors.

Actually, it is better to use gearbox motors instead of common DC motors because it has gears and an axle and its speed does not change towards the top of a hill or downhill. Pay attention that the more speed is, the less precision will be; thus it is better to choose a motor that has authentic RPM. Eventually, we used two DC gearbox motors.

In general, a programming technique is used in this project to control the speed of motors and it will be explained in the programming part, section 3.

Usually there are two movement systems for robots:

1. Wheel
2. Tank system

It is better to use wheels for line follower robots. We can use three wheels. Two of them are joined to the motors and installed rear the robot and the other wheel is free and installed front of the robot as a passive caster. Like Fig. 6 which can be seen in below.

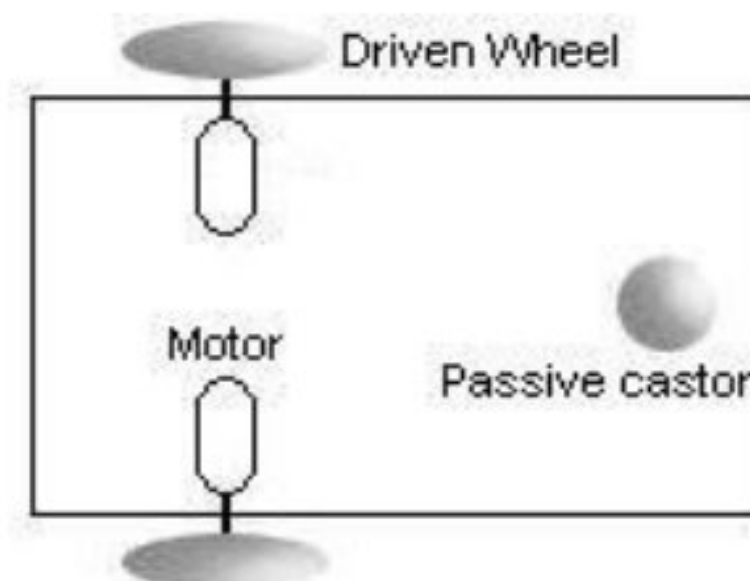


Figure 6. Front and rear wheels

F. The Chassis and Body

There are some good materials for designing robots such as wood, plastic, aluminum and brass alloys. We must pay attention to the resistance, weight and mechanical ability for choosing one of them. There are some agents that we can use them to choose a good body, ability to perforate, incision, flexibility and etc. [4]

In the designed robot, aluminum has been used for chassis because of its lightweight and being strong enough for our project.

All components can be installed on the circuit fiber to decrease the weight. For example, motors can be fixed under the fiber and other segments can be installed on the fiber.

The appearance of robot, definitely, is not important. It is noted that the performance is much more important than other things. We have reached to this point that it is better not to use any types of glue for installing components. Therefore, the components can be installed just by screws.

III. PROGRAMMING

Every microcontroller has own special compiler which we can write program with the language C or Basic for them. After compiling, we must send the compiled program to the microcontroller with programmer. A programmer is a device that can be connected with a computer and we must put the microcontroller on the programmer and then the programmer sends the program into the microcontroller's ROM. There are a lot of programmers and we must use a programmer which can support our microcontroller. [2, 5]

A program has been written for our robot with the language C in Codevision and the summary of the program has been shown below. In this project, we use "At Mega 16" that has four ports. Port A and port B have been used for inputs and outputs, respectively. The data the perform from sensors after Analog to Digital Converting will sends to the microcontroller in port A and after processing with microcontroller the instruction will send to the driver from port B. Below we offer our program to you.

For writing the program easily, we define new name for port A's pins and port B's pins. We use the renamed pin in our

L3	C1	R3
L1		R1
L2	C2	R2

```
#include <mega16.h>
#define LM1 PORTB.1
#define LM0 PORTB.2
#define RM1 PORTB.3
#define RM0 PORTB.4
#define R3 PINA.7
#define R2 PINA.6
#define R1 PINA.5
#define C1 PINA.4
#define C2 PINA.3
#define L1 PINA.2
#define L2 PINA.1
#define L3 PINA.0
void main (void)
{
while (1)
{
    If (C1)
    {
        LM1 =1;
LM0 =0;
        RM1 =1;
        RM0 =0;
    }
else if (R1 || R2)
    {
        LM1 =1;
        LM0 =0;
        RM1 =0;
        RM0 =1;
    }
else if (L1 || L2)
{
    LM1 =0;
        LM0 =1;
        RM1 =1;
        RM0 =0;
    }
    if (C1 && C2 && R1 && R3 && !R2 && !R1 && !L2
&& !L3)
```

Scanned with CamScanner

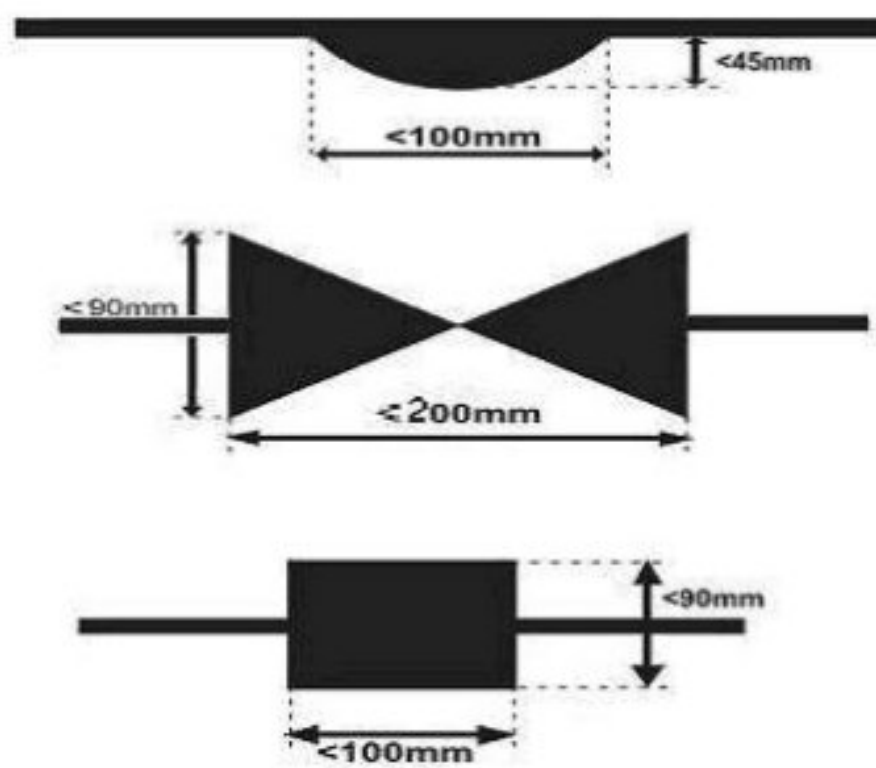


Figure 9. Unwanted change in the line width

- Some parts of the path are cycloid or broken. So, it is better to pay attention to the Fig. 10.

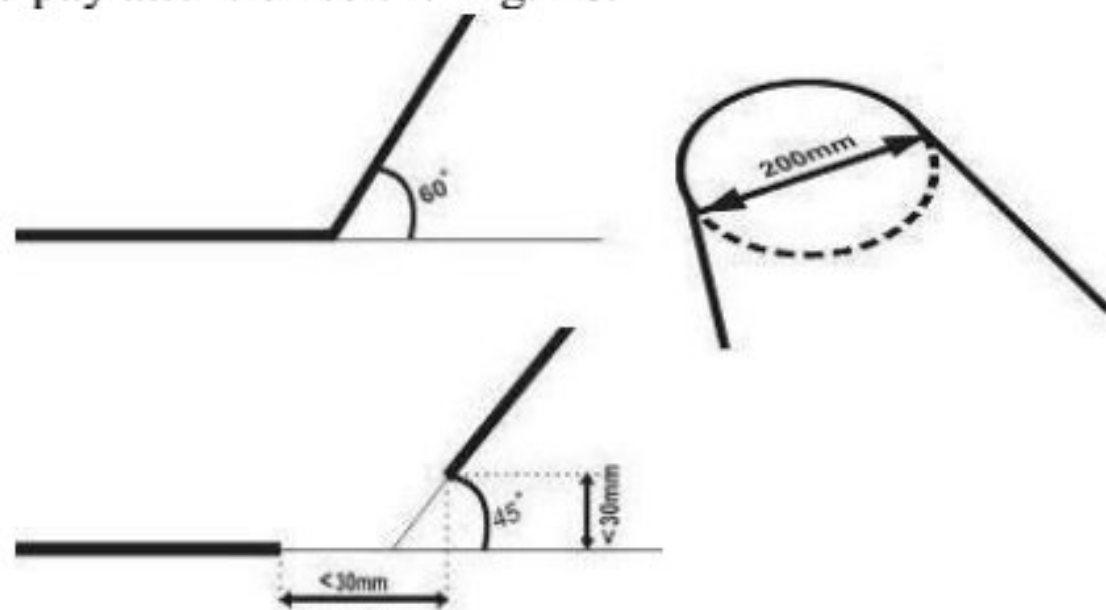


Figure 10. cycloid and break in way

- Maybe the line has some cycles or bad curves. Pay attention to Fig. 11.

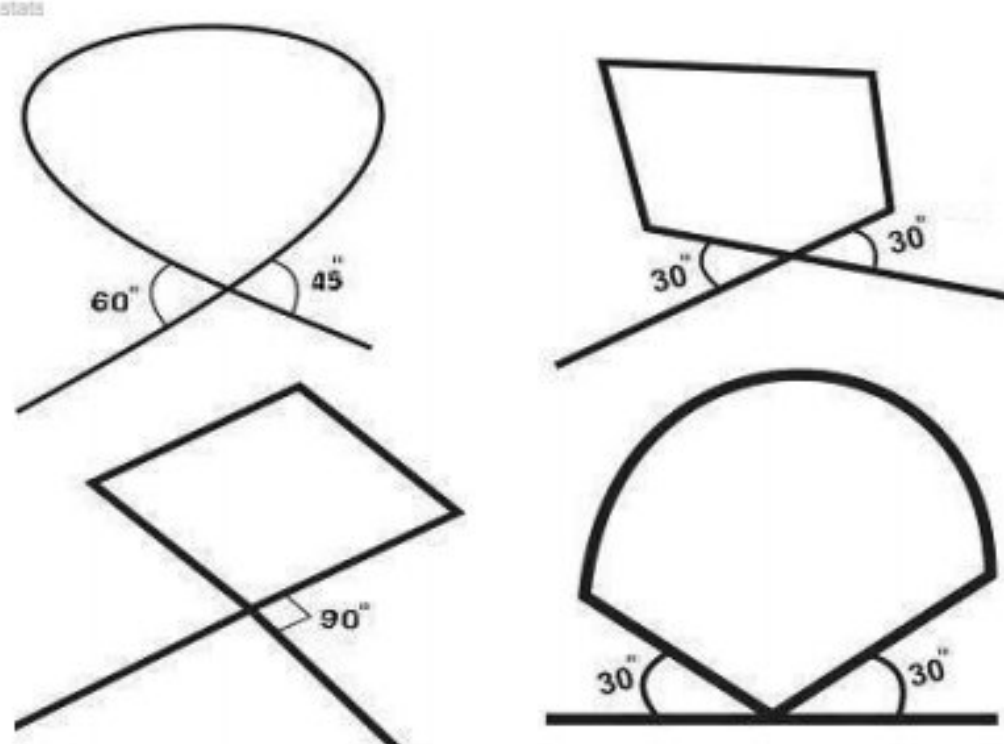


Figure 11. cycle

CONCLUSION

In its current form robot is enough capable. It can follow any curve and cycle. We must build a robot that has light weigh and high speed because points are awarded based upon the distance covered and the speed of the overall robot. Therefore, we used two high speed motors and high sensitivity sensors circuit.

The body weight and wheels radius have effects on speed, too. The weight of the designed robot is around 300 gram and it can be lighter. To get better maneuver, we must build a robot that uses two motors and two wheels on the rear and a free wheel on the front. The power supply is 12 with regulator.

The designed robot has eight infrared sensors on the bottom for detect line. Microcontroller ATmega16 and driver L298 were used to control direction and speed of motors. The robot is controlled by the microcontroller.

In performs change the motor direction by giving signal to driver IC according to receives signals from sensors.