Line Follower Robot

Robotics Workshop
Currents 15th march 2008
EEE Department
NIT Trichy

Developed By:

Mayur Agarwal
Prashant Agrawal
Krishna Nand Gupta
Hitesh Meghani

To Our Readers

We are glad to have had an opportunity to share our knowledge with interested robotics enthusiasts. In this book we have attempted to provide a brief compilation of our experiences in robotics (participating and winning in Technical Festivals all over the country), extending over last three years.

The prospect of practically implementing engineering concepts is the hallmark of robotics. By reading the basics in this book you will gain a significant insight into various tools employed in shaping a robot. However to participate in technical festivals with ever changing problem statements you will be required to apply these basics concepts and come up with innovative algorithms and superior designs.

Do not expect this book to be a panacea for all robotic problems, rather you will have to sit and work for hours to get a functioning robot. Transform each failure into a stepping stone instead of stumbling over it. We appreciate the beauty of diamond but little do we wonder how it became so bright? Its perseverance extending thousands of years transformed it into its present sparkling state.

We encourage you to plunge further into the field of robotics with dedicated perseverance, make your own mistakes and gain valuable experience from them.

ALL THE BEST

Your valuable suggestions and inquisitive doubts are welcome. You can contact us at

Mayur Agarwal (mayurmagic.nitt@gmail.com)

Krishna Nand Gupta (kng_sweet94@yahoo.com)

Prashant Agarwal (pras_nitt@yahoo.co.in)

Hitesh Meghani (h.nittian@gmail.com)

Introduction

The line follower is a self operating robot that detects and follows a line that is drawn on the floor. The path consists of a black line on a white surface (*or it may be reverse of that*). The control system used must sense a line and maneuver the robot to stay on course, while constantly correcting the wrong moves using feedback mechanism, thus forming a simple yet effective closed loop System. The robot is designed to follow very tight curves.

Sample Paths

The path is a black line on a white background with width of 3 cm (except at bends where a little variation may be present). It may contain paths laterally displaced by a around 3 cm and also gap of at most 5 cm. (*All these specifications may vary one competition to other*).

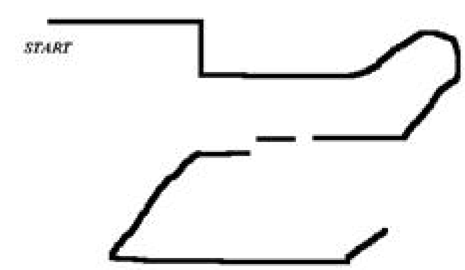


Figure 1: Basic Sample Arena

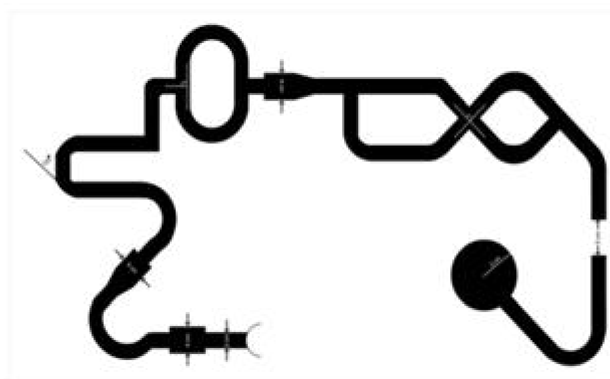


Figure 2: Sample Arena of Kurukshetra'08 (Anna University)

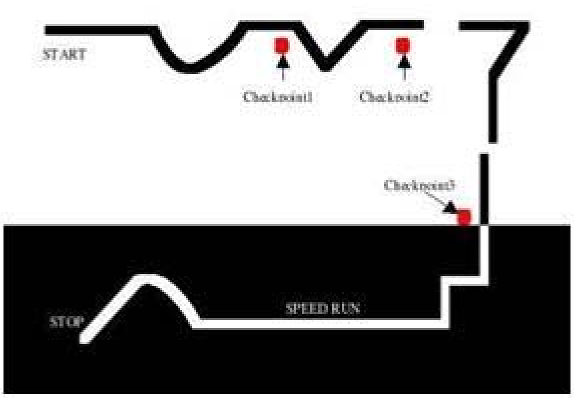


Figure3: Sample Arena of Pragyan'07 (NIT Trichy)

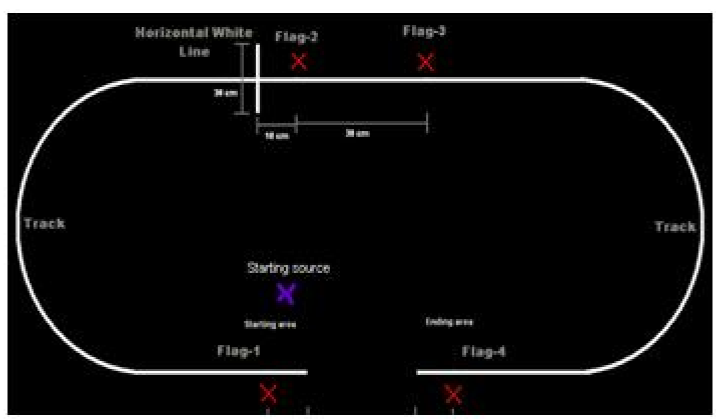


Figure4: Sample Arena of Robo-Relay of Kishitij'08 (IIT KGP)

From above images we can conclude that in most of line follower competitions, we have to perform some additional task apart from following the line.

Basic design and requirements

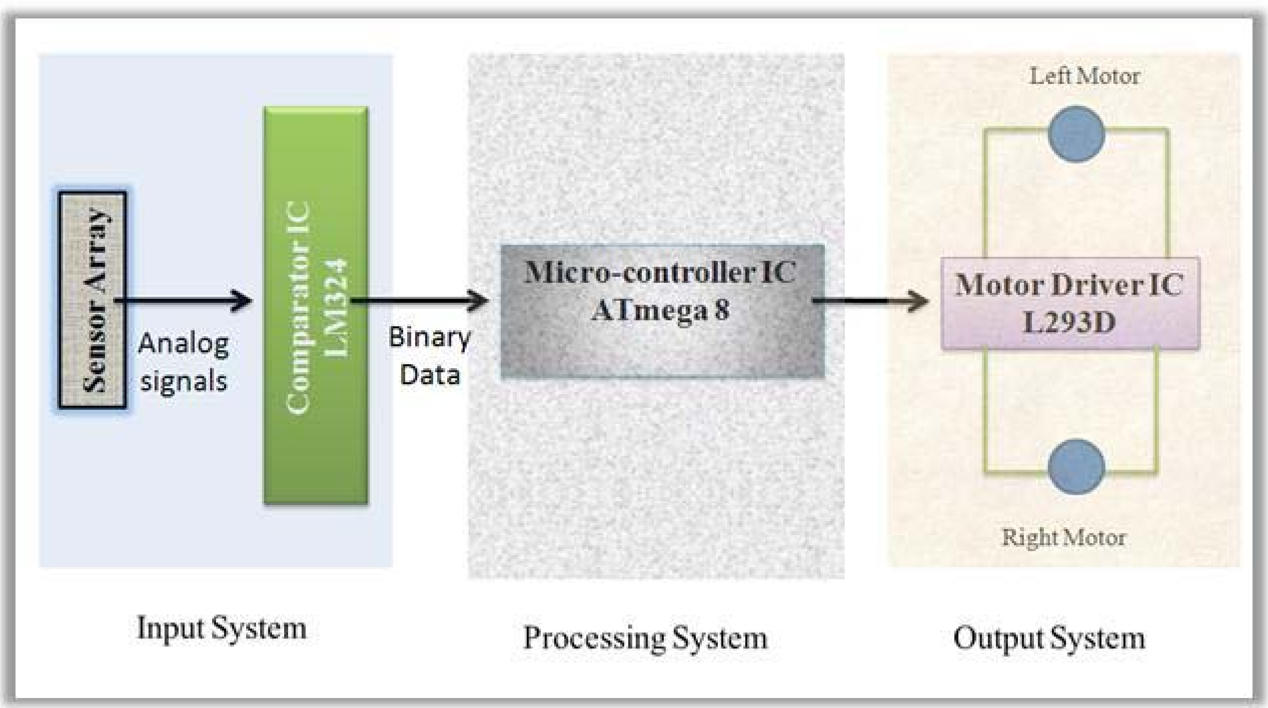
The robot is built with ATmega8L, L293D,IR sensors,LM324 , platform consisting of a toy car chassis (or hand made Al sheet chassis). The robot is designed using two motors controlling wheels. It has infrared sensors on the bottom for detect black tracking tape .It captures the line position with the help of these optical sensors called opto-couplers mounted at front end of the robot. (*Each opto-coupler consists of an IR LED and an IR Sensor*) when the sensors detect black surface, output of comparator, LM324 is low logic and for white surface the output is high. It reports to the microcontroller for accurate control and steering of motors. Microcontroller ATmega8L and Motor driver L293D were used to drive the motors.

Basic operation

The basic operations of the line follower are as follows:

1. Capture line position with optical sensors mounted at front end of the robot. For this a combination of IR LED's and Photo Transistor called an opto-coupler is used. The line sensing process requires high resolution and high robustness.
2. Steer robot to track the line with any steering mechanism. To achieve this we use two motors governing wheels motion.

Block Diagram



Let's see all the system in detailed manner.

INPUT SYSTEM

Sensors

IR reflective sensors have one emitter (IR LED) and one receiver (Photo-Transistor or photo diode.

If we have white surface it reflects the light and it will sensed by the receiver, similarly if we have black surface it absorbs the light and receiver can not sense light.

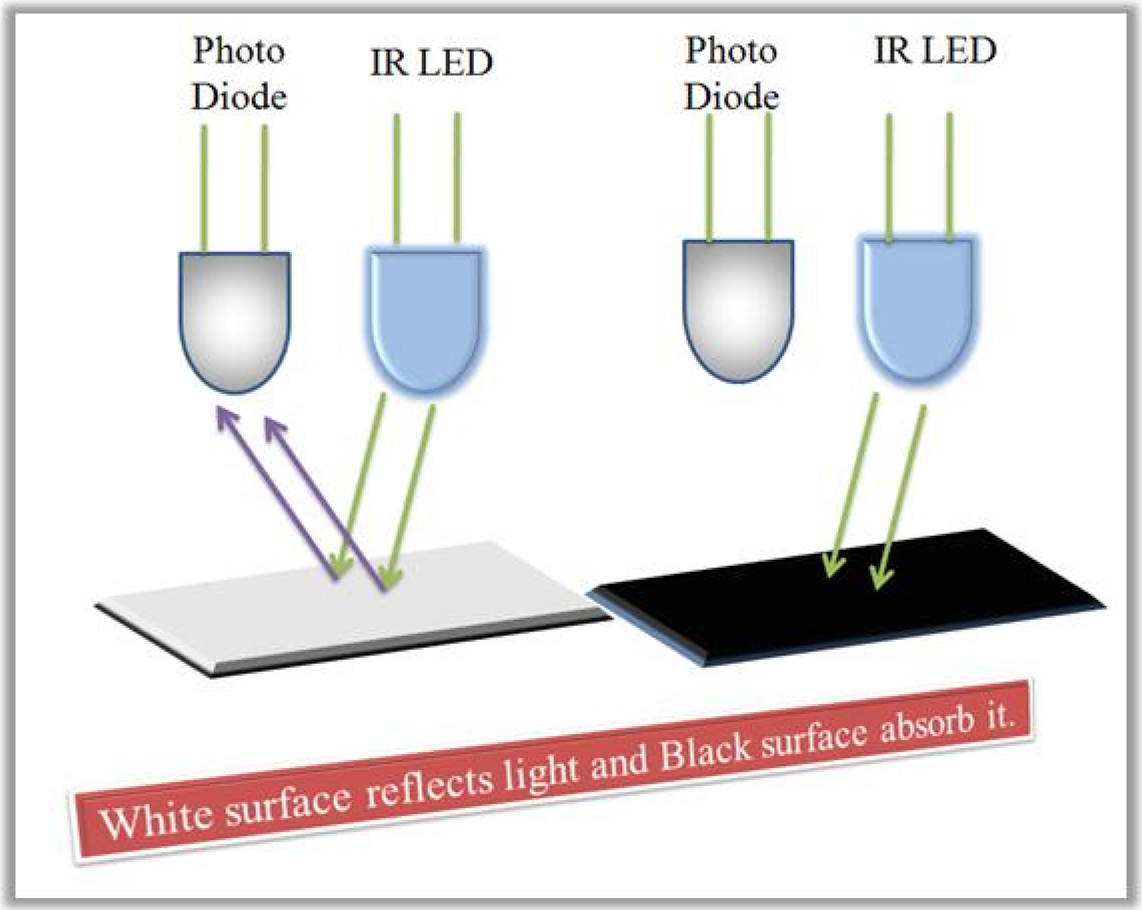
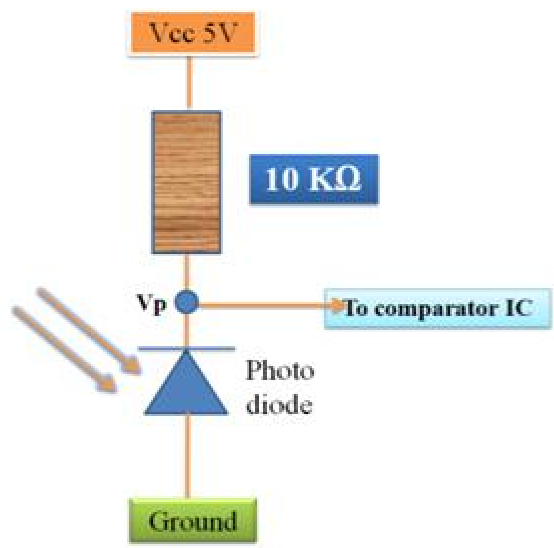


Photo diode has property that if IR light fall on it its electrical resistance comes down (i.e. its comes down from 150kΩ to 10kΩ if no noise present).For sense the change in resistance we use voltage divider circuit (as shown in figure below).



<u>Sample Calculation:</u>
Say Receiver has resistance-
Rs=150kΩ without light (on black surface)
Rs=10kΩ with light (on white surface)
The voltage that goes to comparator

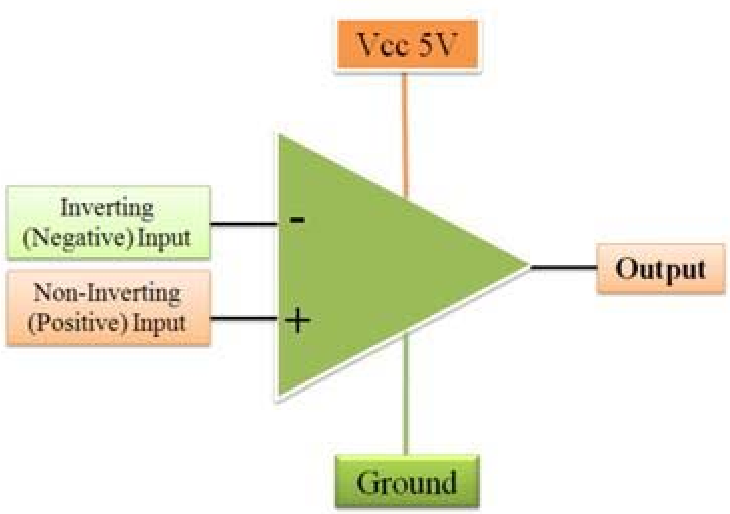
Without light: (on black surface)
$Vp=\frac{Rs}{Rs+R} * Vcc = \frac{150}{150+10}$*5 V=**4.6875 V**

With light: (on white surface)
$Vp=\frac{Rs}{Rs+R} * Vcc = \frac{10}{10+10}$*5 V=**2.500 V**

Thus we get variation of voltage that is sensed by comparator IC (LM324). This gives logical high or low according to input.

Comparator

Comparator is a device which compares two input voltages and gives output high/low.In circuit diagram it is normally represented by a triangle having-Inverting (negative) Input (-),Non-Inverting (positive) Input(+), Vcc, Ground, Output.

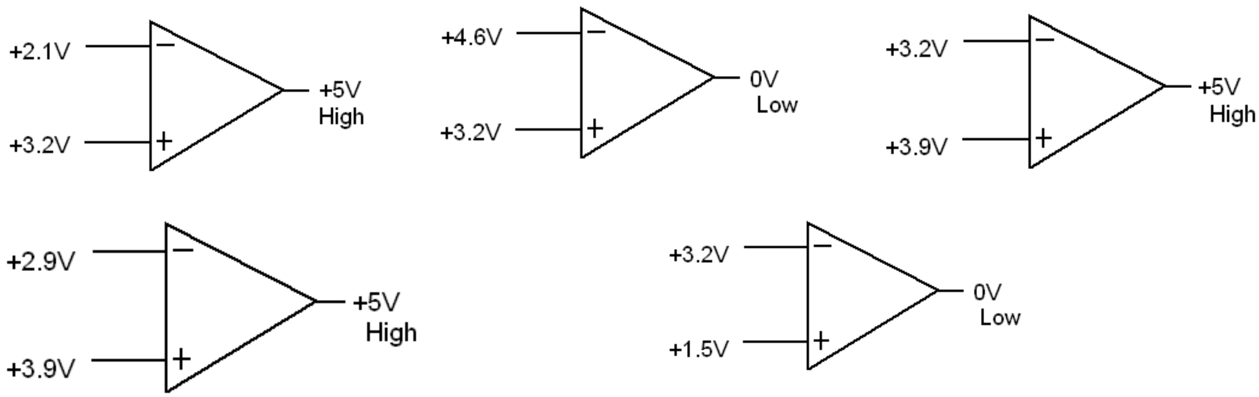


Properties of comparator:

- If V+ > V-
  then Vo=Vcc
  (Digital **High 1** output)

- If V+ < V-
  then Vo=0
  (Digital **Low 0** output)

Let's see some examples



Use of comparator in IR sensor

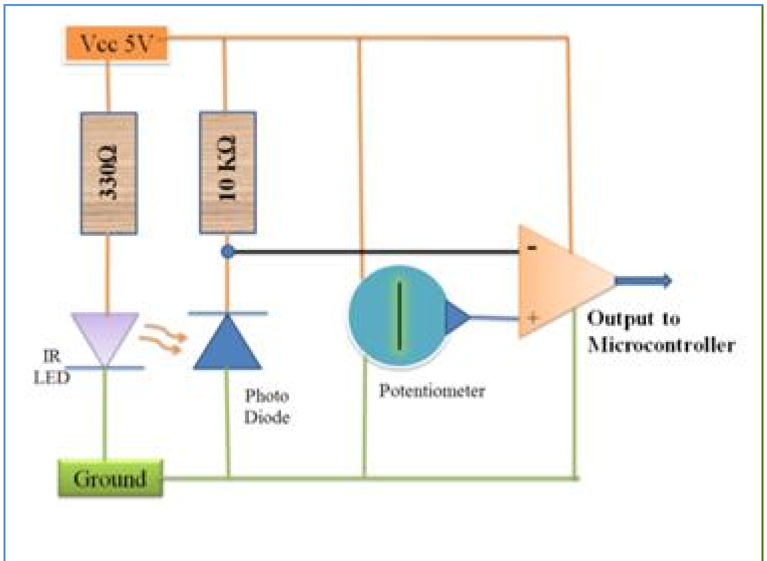
As above we see that two inputs are required for comparator. One input is from photo-receiver (like photo-diode), other is generated by us using potentiometer. The second voltage is also called as reference voltage for that sensor.

Setting of reference voltage (Vref)

We can vary reference voltage by using potentiometer, such that it can vary from 0V to Vcc. We set reference voltage as mean value of the sensor inputs measured with and without light.

Eg. From above example Vref $= \frac{4.675+2.500}{2} = \mathbf{3.5875\ V}$

Lets connect Inverting Input of Comparator to photo- receiver, Non-Inverting Input to potentiometer (as shown in figure) and output goes to micro controller.



<u>Sample Calculation:</u>

Let V+ = **3.5875 V**

<u>With light :(on white surface)</u>

V- = **2.500 V**

Thus ***V+>V-*** and Vo= Vcc = 5 V

Thus we got digital **HIGH** output.

<u>Without light:(on black surface)</u>

V- = **4.6875 V**

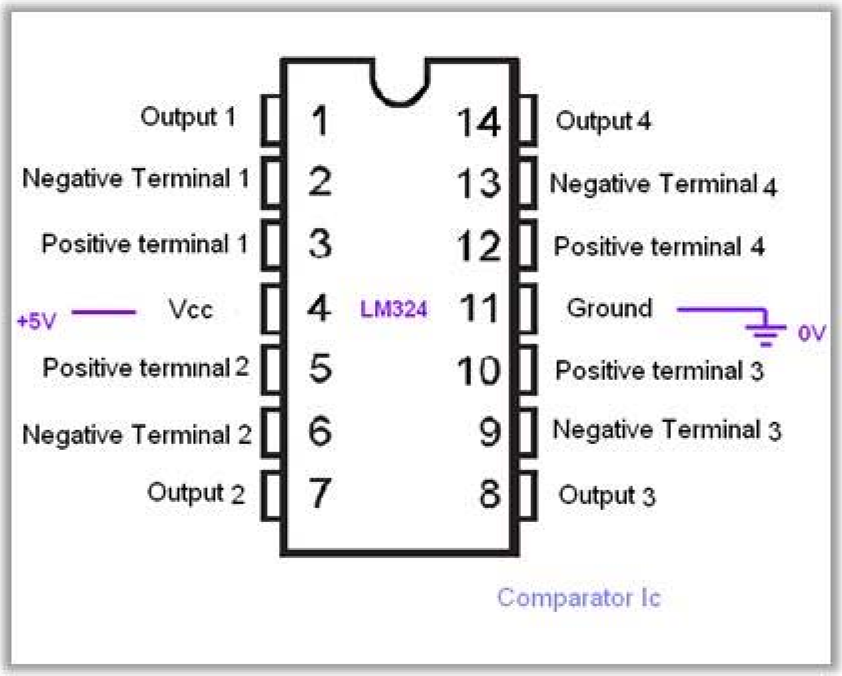
Thus ***V+<V-*** and Vo = 0 V

Thus we got digital **LOW** output.

Note: If we connect Inverting Input of Comparator to potentiometer and Non-Inverting Input to photo- receiver, the only difference observed is that at white surface we will get Low output and for black surface we will get High output.

IC LM324

IC LM324 contains four comparators.



You can see the datasheet of that IC for more details.

Arrangement of Sensors

An array of sensors arranged in a straight row pattern is bolted under the front of the robot. It is used to locate the position of line below the robot.

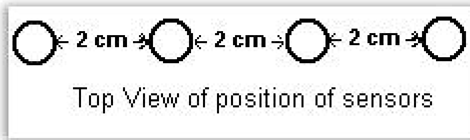
We can use any **number of sensors**. If we have lesser number then our robot movement is not smooth and it may face problems at sharp turns. If we used higher number of sensors robot movement will become smooth and reliable for sharp turns, only drawbacks it requires complex programming for micro-controller and requires more hardware. Thus we must choose optimum number of sensors.

The **distance between each sensors** depend on

1. No of sensors used
2. Width of straight line (distance between sensors should be less than width of line).

3. Distance between sensors may not be constant (it depends on the logic).

Sample figure:

Left  Right

Tips for Input System

- *Significant problems are faced in input system. So make reliable connections, use printed PCB.*
- *Set potentiometer value for each sensor, because each sensor may behave differently and may give different voltage on the same surface.*
- *You have to check potentiometer value for new surface, as reflectivity depends on roughness of surface.*
- *IR LED draws high current (high power), so you have to connect a good quality battery (high current rating) for sensors.*
- *The advantages of IR sensors is that they less effected by ambient light. Suppose if we use LDR as photo receiver, their sensitivity will also depend on ambient light.*
- *To get a good voltage swing, the value of R1 must be carefully chosen, we are working on 10kΩ.*
- *Proper orientation of the IR LED and Photo diode is must so as to have good voltage swing. Also IR rays of one opto-coupler should not disturb the other opto-coupler.*
- *If a opto-coupler is not working then first check IR LED supply a standard IR LED has 1.1-1.2 volt drop across it then check voltage variation then reference voltage*

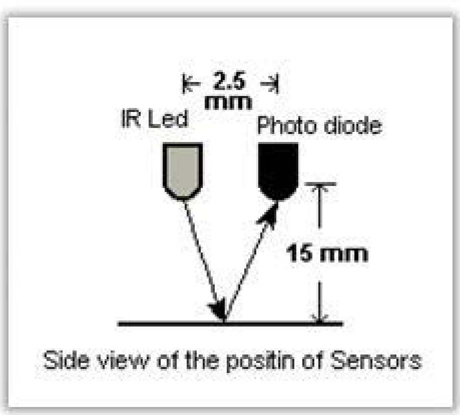


Figure 2: Alignment of Sensor in an opto-coupler

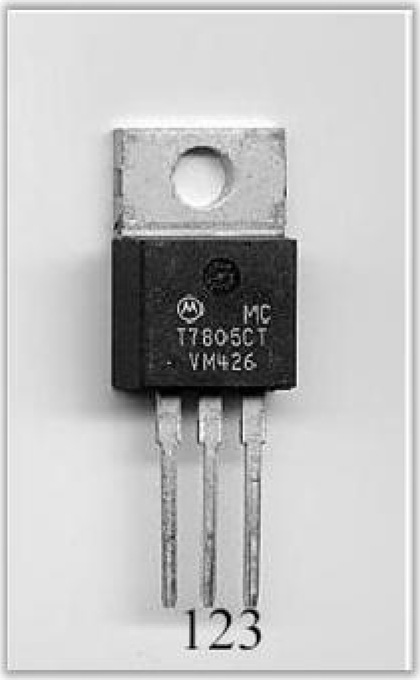
Voltage Regulator 78xx

Voltage regulators convert fixed DC output voltage from variable DC. The most commonly used ones are 7805 and 7812. 7805 gives fixed 5V DC voltage if input voltage is in between 7.5V to 20V.

They help to maintain a steady voltage level despite varying current demands and input voltage variations.

If input voltage is <7.5 V then regulation won't be proper i.e. if input is 6V then output may be 5V or 4.8V, but there are some parameters for the voltage regulators like maximum output current capability, line regulation etc.. , that won't be proper.

To identify the leads of the 7805, you have to keep the lead downward (Fig a) and the writing to your side, (see the figure below). You can see the heat sink above the voltage regulator.**(1-input,2-gnd,3-output).**



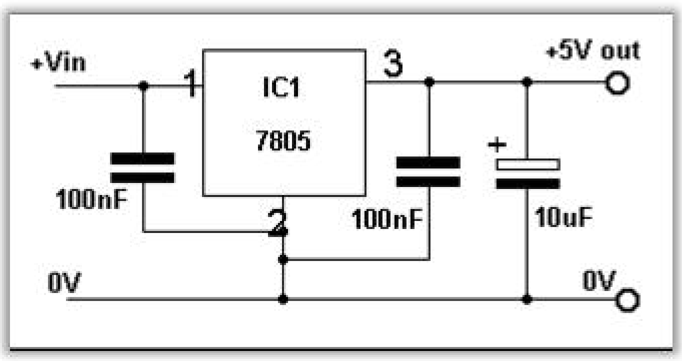


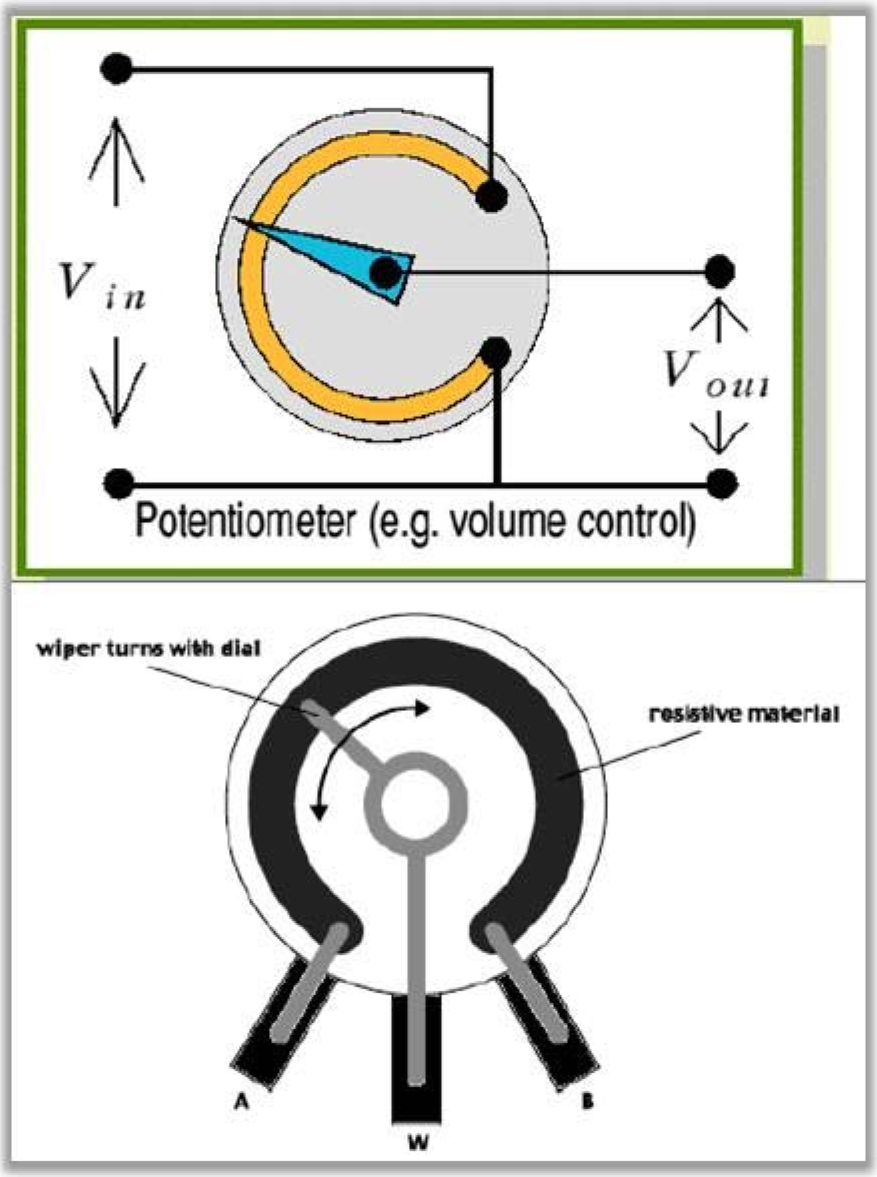
Fig a

Fig b

Fig b shows how to use 7805 voltage regulator. Here you can see that coupling capacitors are used for good regulation. But there is no need for it in normal case. But if 7805 is used in analog circuit we should use capacitor, otherwise the noise in the output voltage will be high. The mainly available 78xx IC's are 7805, 7809,7812,7815,7824.

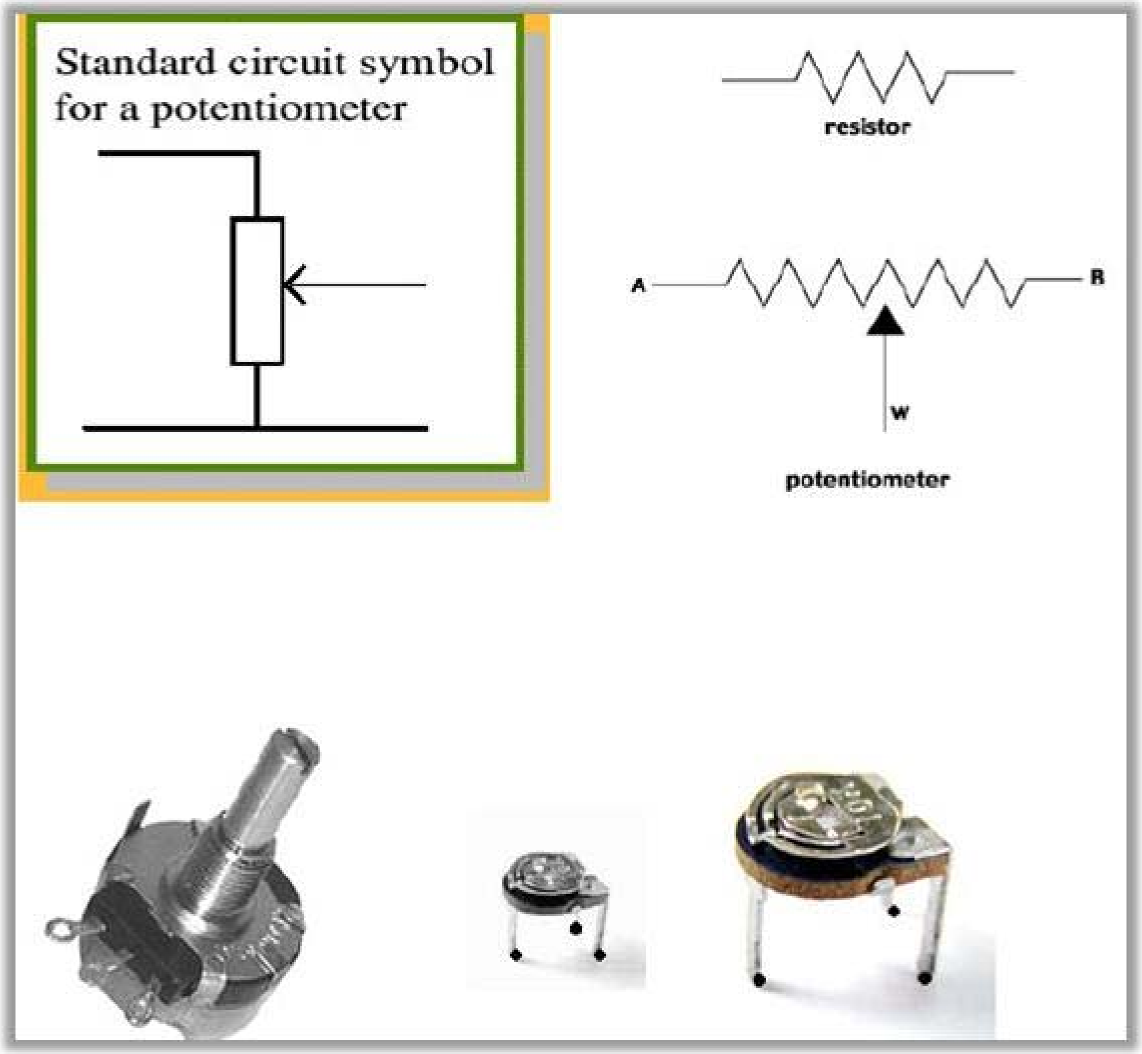
POTENTIOMETER ('POT ')

Potentiometer is a variable resistor which is used to vary the resistance by rotating the shaft. Potentiometers are available from 100 ohm to 470Kohm (or more).



Potentiometer is a voltage divider. If we connect Lead A to Vcc and Lead B to ground then you can get voltages from 0 to Vcc by at LeadW. Mainly potentiometers are used to generate reference voltage for LM324.

E.g. if we couple potentiometer to the shaft of a motor, then we can measure the angle moved by shaft by connect the output of Leads W and Lead B to an ADC to get a digital reading of angle. i.e a shaft encoder, but there is a limitation, we can't get rotation >270 degree and also number of rotations since potentiometer shaft can only move from A to B.



Above figure shows different types of potentiometers available. Second and third potentiometers are mainly used when you only want to change the value of resistance occasionally while the first one is used when we have to vary resistance frequently. Second and third one are easy to be inserted in breadboard and PCB they remain fixed. Second and third type of potentiometers also called **Preset**. Resistance is varied by rotating the shaft in the body of the potentiometer.

PROCESSING SYSTEM

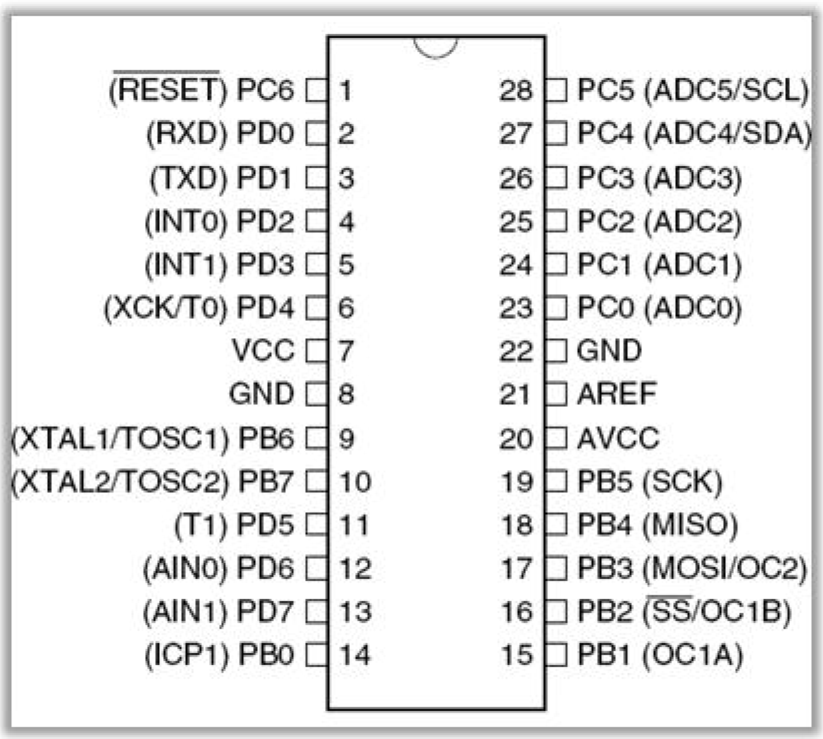
Processing system acts as the Brain of robot, which generates desired output for corresponding inputs. For that we use microcontrollers. In present days, there are several companies that manufacture microcontrollers, for example ATMEL, Microchip, Intel, Motorola etc.We will be using ATmega8L microcontroller in our robot. It is an ATMEL product. It is also called **AVR**.

Why ATmega8L

Line follower robot requires simple microcontroller as it uses simple algorithms. We can use any microcontroller for that. But we use ATmega8, because it has following extra features:

1. It is an **ISP (In System Programmable)** device. It means programming (Burning) of ATmega8 IC can be done without removing it from the system.
   Note: *Programming (Burning) of a microcontroller means transferring the code from computer to microcontroller. We will explain burning later.*
2. It has on chip **PWM (Pulse Width Modulation)** circuit at three pins (Pin 15, 16 and 17). We have explained PWM in another tutorial.
3. It has an inbuilt RC **oscillator**. (Oscillator is a clock generator circuit).
4. It consumes low power than other microcontrollers.

A) Hardware Details



Basic hardware connections of ATmega8

<u>Pin 1 (Reset):</u>

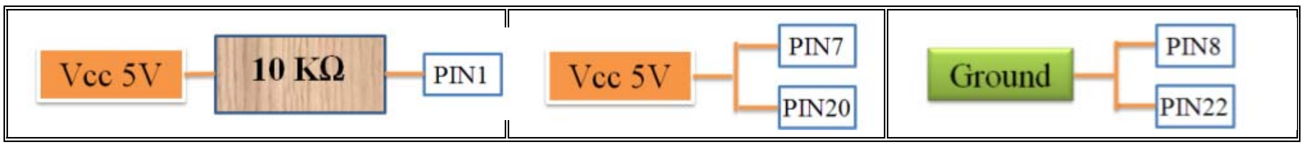
The use of reset pin to reset the ATmega8 microcontroller. This can be done by connecting this pin to ground. But in normal mode of execution it should have at least **2.7V**. Thus it is connected to +5 volt through 10k ohm resistance .We should make sure it should be above 2.7 for proper execution of code.

<u>Pin 7 and 20</u> <u>(Vcc):</u>

Pin 7 and 20 should be connected to Power supply. (2.7 to 5.5 volt for ATmega8L)

<u>Pin 8 and 22</u> <u>(Ground):</u>

Pin 8 and 22 should be connected to Ground. This ground should be common to for the entire circuit.



Make sure Ground and Vcc does not get interchanged and Vcc should not exceed 5.5 V. If you connect supply wrongly, ATmega8 will suffer permanent damage.

Input and Output Ports

- In ATmega8 we have three I/O (input/output) ports viz. Port B, Port C, Port D.
- One can configure any pin of all these ports as input or output pin by software.

We are using Port B pins (PB0 to PB3) as output pins because at pinPB1 and PB2 we have on-chip PWM output that can control the speed of motors.

Pin	PORT	Connection	PWM
14	PB0	Negative of right	NO
15	PB1	Positive of right	YES
16	PB2	Positive of Left	YES
17	PB3	Negative of Left	NO

Thus we can use Port C or Port D or remaining Pins of Port B as input. But for the sake of simple hardware connection we choose Port D pins as input pins.

Burner (or programmer)

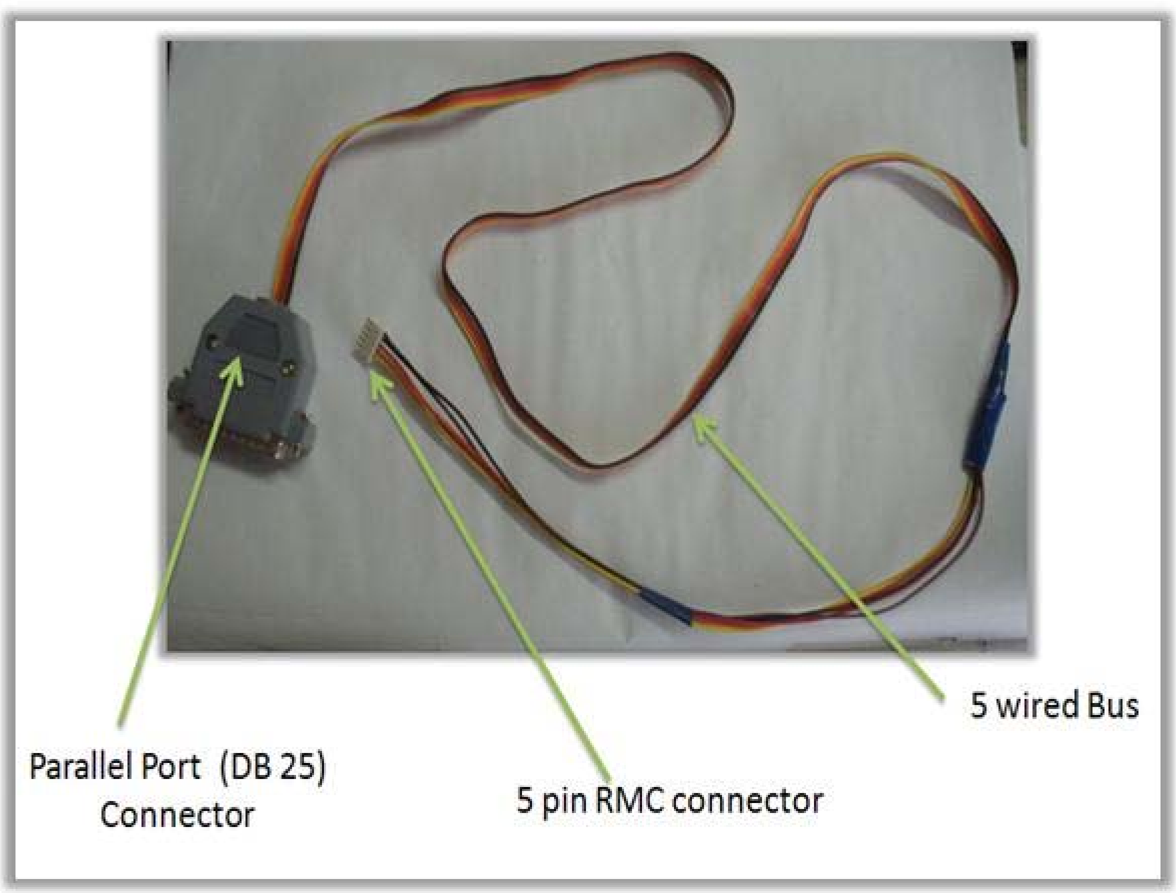
Burner (or programmer) is the circuit used to transfer the code from computer to microcontroller IC. For programming AVR there are different types of burners available Eg stk200, stk500, jtag2 etc.
We are using **stk200** programmer.

stk200 Programmer:

Requirements

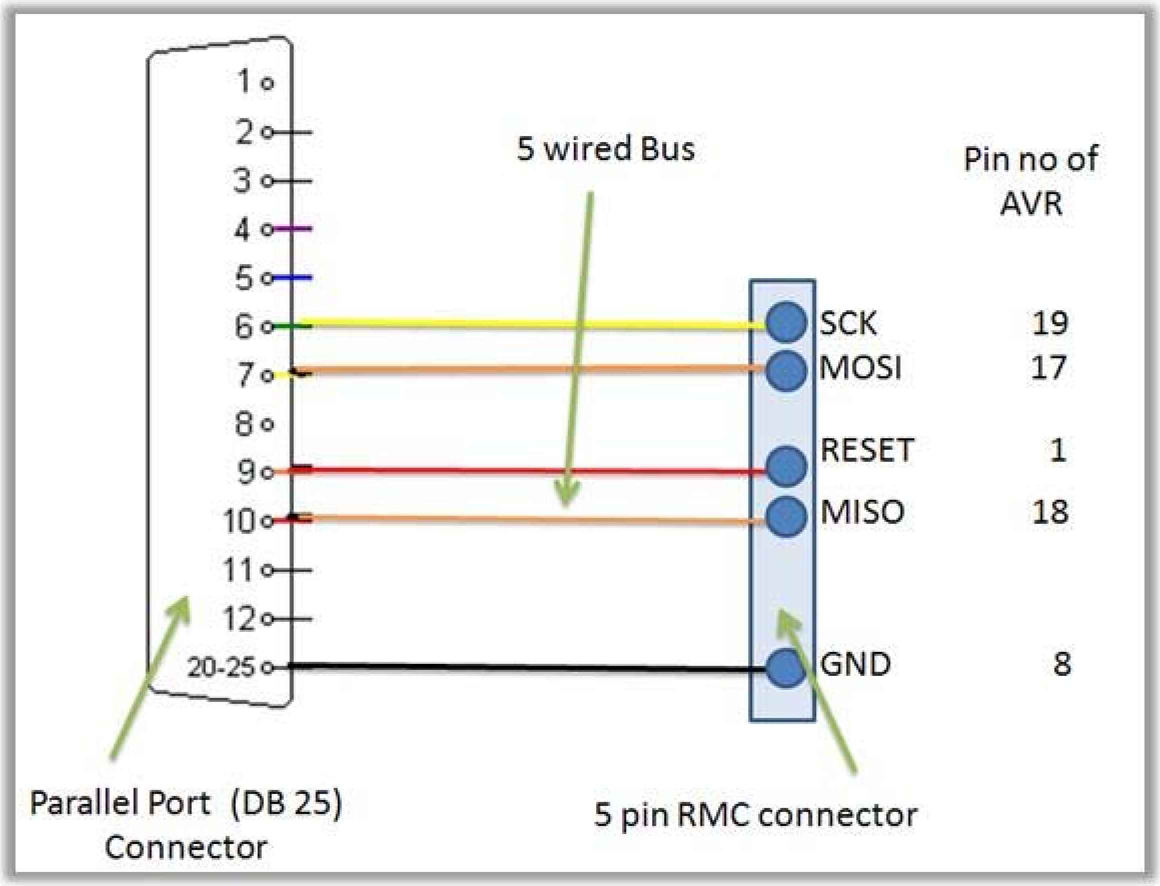
- Parallel Port DB 25 connector (connects to the computer)
- 5 Pin RMC connector goes to AVR PCB.
- 5 wired Bus

They are shown in following figure.



Connections

Connections are given by the following figure.



Explanations of Pins of ATmega8

SCK	SPI Bus Master clock Input)
MOSI	Master Output/Slave Input)
RESET	Reset pin
MISO	Master Input/Slave Output
GND	Ground

B) Software Details:

Programming and Simulating

The program code acts as the decision-maker embedded in the micro-controller i.e. it decides what will be the outputs for particular set of input combination.

Programs for the AVR series of microcontrollers can be written in assembly (AVR ASM), C and BASIC. AVR Studio, WinAVR etc. are some free development softwares for programming the AVR Microcontrollers. We are using winAVR for programming and AVR Studio for simulating (*Simulation means debugging the code on software, one can virtually give the input and check the output for that code*).

In winAVR programmers Notepad we write our C code, after compilation it generates **'.hex'** file that is a hardware level code.

Sample Code: To blink a LED connected at pin 6 (PD4) of ATmega8.

```
#include <avr/io.h> //header file to include input output port
#include <util/delay.h>//header file to include delay function
#define LED PD4

int main(void)
{
 DDRD = (1 << LED);
 /* DDR=Data Direction register... its to define PD4 OUTPUT pin
 rest bits of DDRD can be 0 or 1 does not make any significance */

 while (1)
 {
  PORTD = (1 << LED);    // switch on
 _delay_ms(200);
  PORTD = (0 << LED);    // switch off
       _delay_ms(200);
 }
return 0;
}
```

MOTOR OUTPUT SYSTEM:

For moving a robot we have two dc motors attached to wheels gears.

Why DC motors

DC motors are most easy to control. One dc motor requires only two signals for its operation. If we want to change its direction just reverse the polarity of power supply across it. We can vary speed by varying the voltage across motor.

Use of gears

The DC motors don't have enough torque to drive a robot directly by connecting wheels in it. Gears used to increase the torque of dc motor on the expense of its speed.

Mathematical interpretation:

Rotational power(Pr) is given by:

Pr=Torque(T)   Rotational Speed (ω)

Thus

$$\boxed{T = \frac{\mathrm{Pr}}{\omega}}$$

Pr is constant for DC motor for a constant input electrical power. Thus torque (T) is inversely proportional speed (ω).

$$\boxed{T \ \alpha \ \frac{1}{\omega}}$$

Thus to increase the value of torque we have to loose speed.

Note:

- In toy car, there is a gear box that contains several combinations of gears.
- Geared motor has gears box at its front.

Simple DC Motor of Toy Car — Gear Box of Toy Car — Geared DC Motor — Gear Box of Geared DC Motor

Why two motors

By using two motors we can move our robot in any direction. This steering mechanism of robot is called as **differential drive**.

Let's check how it works

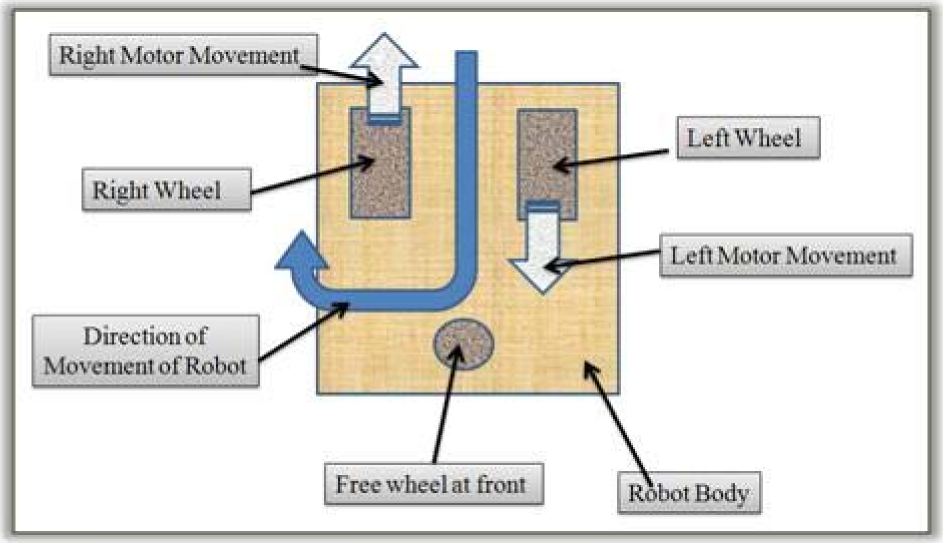


Figure. Description of various parts

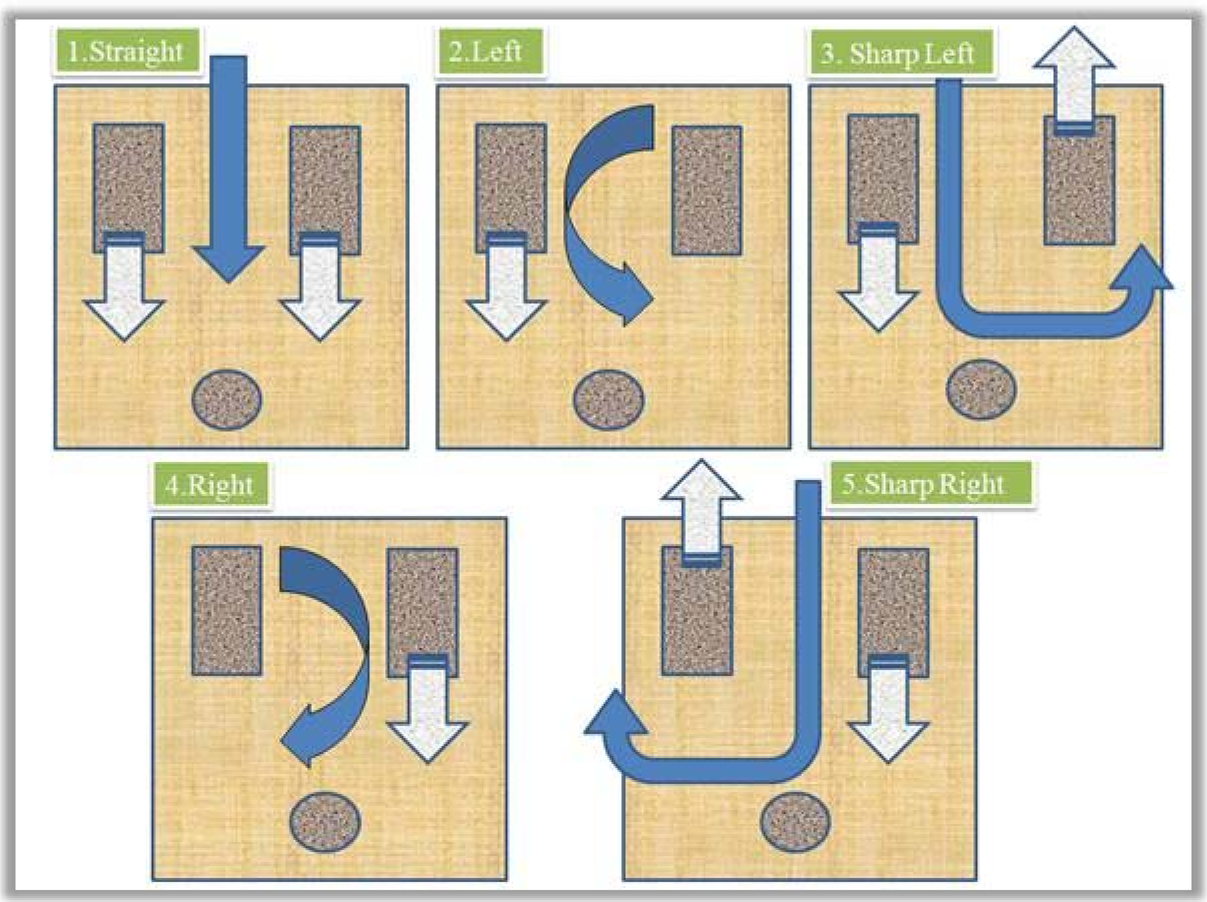


Figure. Different Types of Movement of Robot

The table:

Left Motor	Right Motor	Robot Movement
Straight	Straight	Straight
Stop	Straight	Left
Reverse	Straight	Sharp left
Straight	Stop	Right
Straight	Reverse	Sharp Right
Reverse	Reverse	Reverse

Use of Motor Driver

From microcontroller we can not connect a motor directly because microcontroller can not give sufficient current to drive the DC motors. *Motor driver is a current enhancing device, it can also be act as Switching Device.* Thus we insert motor driver in between motor and microcontroller.

Motor driver take the input signals from microcontroller and generate corresponding output for motor.

IC L293D

This is a motor driver IC that can drive two motor simultaneously. Lets see how we use this IC.

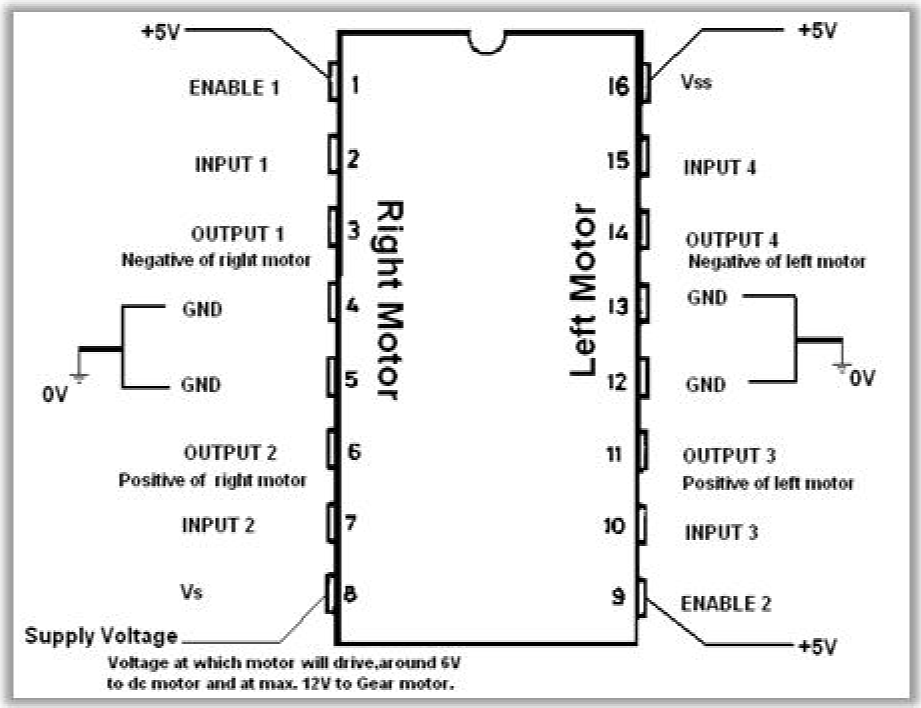


Figure. Pin Details of L293D

Points regarding L293D

- ❖ Supply voltage (Vss) is the Voltage at which we wish to drive the motor. Generally we prefer 6V for dc motor and 6 to 12V for gear motor, depending upon the rating of the motor.
- ❖ Logical Supply Voltage will decide what value of input voltage should be considered as high or low .So if we set Logical Supply Voltage equals to +5V, then -0.3V to 1.5V will be considered as Input Low Voltage and 2.3 V to 5V will be considered as Input High Voltage.
- ❖ L293D has 2 Channels .*One channel is used for one motor*.
    - Channel 1 - Pin 1 to 8
    - Channel 2 - Pin 9 to 16
- ❖ Enable Pin is use to enable or to make a channel active .Enable pin is also called as Chip Inhibit Pin.
- ❖ All Input (Pin No. 2, 7,10and 15) of L293D IC is the output from microcontroller (*ATmega8*).

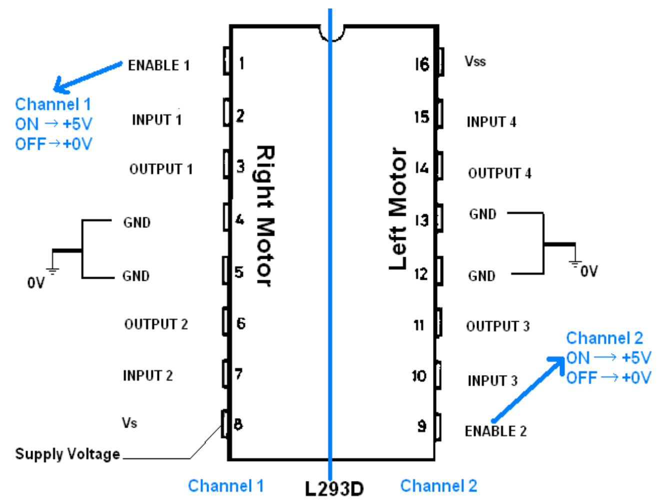
  E.g.-*We connected (Pin No. 2, 7, 10 and 15) of L293D IC to (Pin No. 14, 15,16and 17) of ATmega8 respectively in our robots*, because on pin 15 and 16 of ATmega8 we can generate PWM.
- ❖ All Output (Pin No. 3, 6,11and 14) of L293D IC goes to the input of Right and Left motor.

Output Connections

- OUTPUT 1 (Pin No 3) --- Negative Terminal of Right Motor
- OUTPUT 2 (Pin No 6) --- Positive Terminal of Right Motor
- OUTPUT 3 (Pin No 10) --- Positive Terminal of Left Motor
- OUTPUT 4 (Pin No 14) --- Negative Terminal of Left Motor

For one motor:

Positive Terminal	**Negative Terminal**	**Motor Output**
0	0	No movement
Vss	0	Straight
0	Vss	Reverse
Vss	Vss	No Movement



- One channel corresponds to one motor.
- Enable pin should be high for activating the corresponding channel.
- Input 1 corresponds to Output 1.

If Enable 1=High (1)
Input1 =High (1), Output1=Vss
Input1 =Low (0), Output1=0

If Enable 1=Low (0)
Input1 =High (1), Output1=0
Input1 =Low (0), Output1=0

Means if Enable pin low, the output will be at 0 always. If its high output depend on input

Let's check the sample outputs for some sample inputs (when both the Enable pins are high):

Input 1	***Input 2***	***Input 3***	***Input 4***	***Output 1***	***Output 2***	***Output 3***	***Output 4***	***Motors Output***		***Movement***
								Right	***Left***	
Low	High	High	Low	0	Vss	Vss	0	Straight	Straight	Straight
Low	High	Low	Low	0	Vss	0	0	Straight	No mov	Left Turn
Low	Low	High	Low	0	0	Vss	0	No mov	Straight	Right Turn
Low	High	Low	High	0	Vss	0	Vss	Straight	Reverse	Sharp Left
High	Low	High	Low	Vss	0	Vss	0	Reverse	Straight	Sharp Right
High	Low	Low	High	Vss	0	0	Vss	Reverse	Reverse	Backward

You can see the datasheet of that IC for more details.

Algorithm for Line Follower

There are various line follower algorithms. The main use of algorithm is to move the robot on the line in a very smooth fashion. Apart from the task, the algorithm also depends on hardware including number of sensors, type of motors, chassis etc.

Same problem can be approached by different algorithms depending on the thinking of an individual. Readers are encouraged to develop their own algorithms.

Appendix:

1. Printed Circuit Board

When making the circuit with the electronic parts of the resistors, the capacitors, the transistors, the ICs and so on, it is necessary to connect the lead line of each part appropriately. Also, each part must be fixed, too. The printed circuit board is used to do the wiring among the parts and the fixation of the part.

Advantages:

- Reliability and durability due to compact nature of the circuit especially needed for mobile applications like robotics.
- Easy debugging.
- Large number of circuits can be made with a greater accuracy and at a cheap cost.
- Chances of loose connections get eliminated.

Disadvantages

- Once the circuit is made on a PCB its layout cannot be changed.
- PCB leads can burn at higher values of current rendering it useless for further use
- PCB designing can be a tedious and time consuming process.

PCB layout that will be given in Workshop

Bottom View:

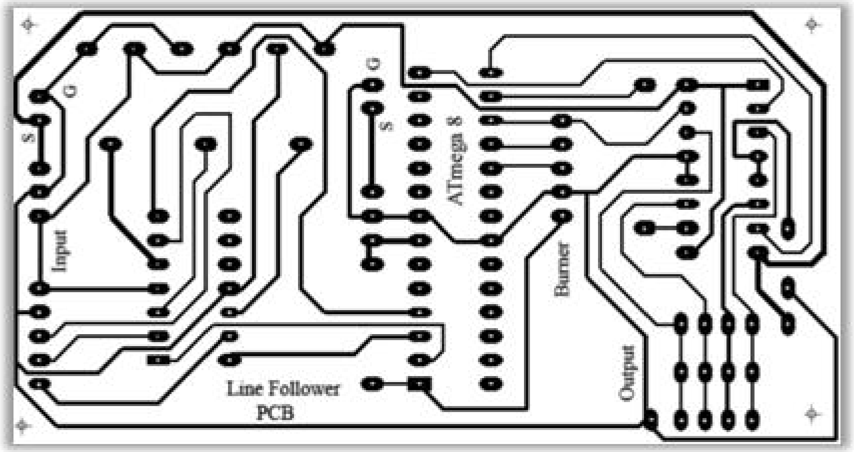


Figure: PCB Layout designed by Software

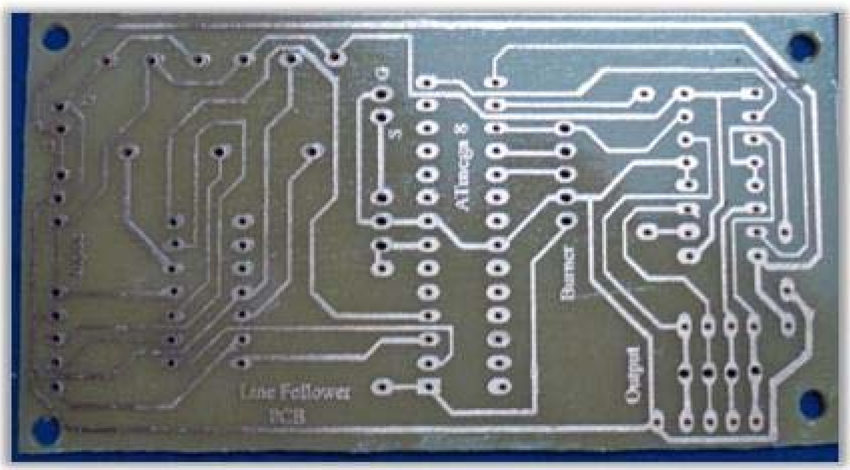


Figure: Photo taken by digital camera

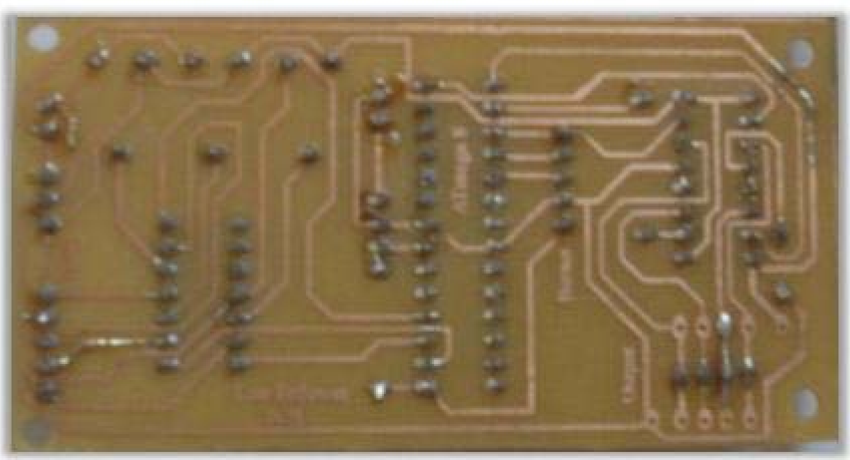


Figure: After Soldering

Top View:

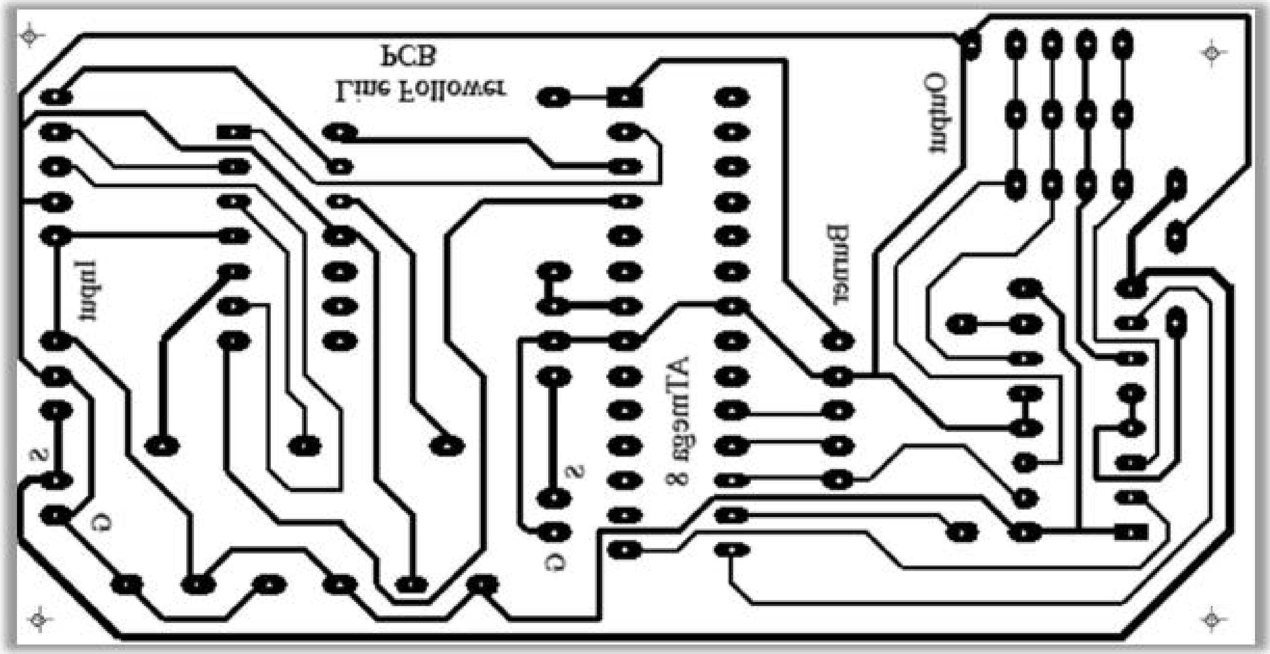


Figure: Mirror image of Bottom Layout

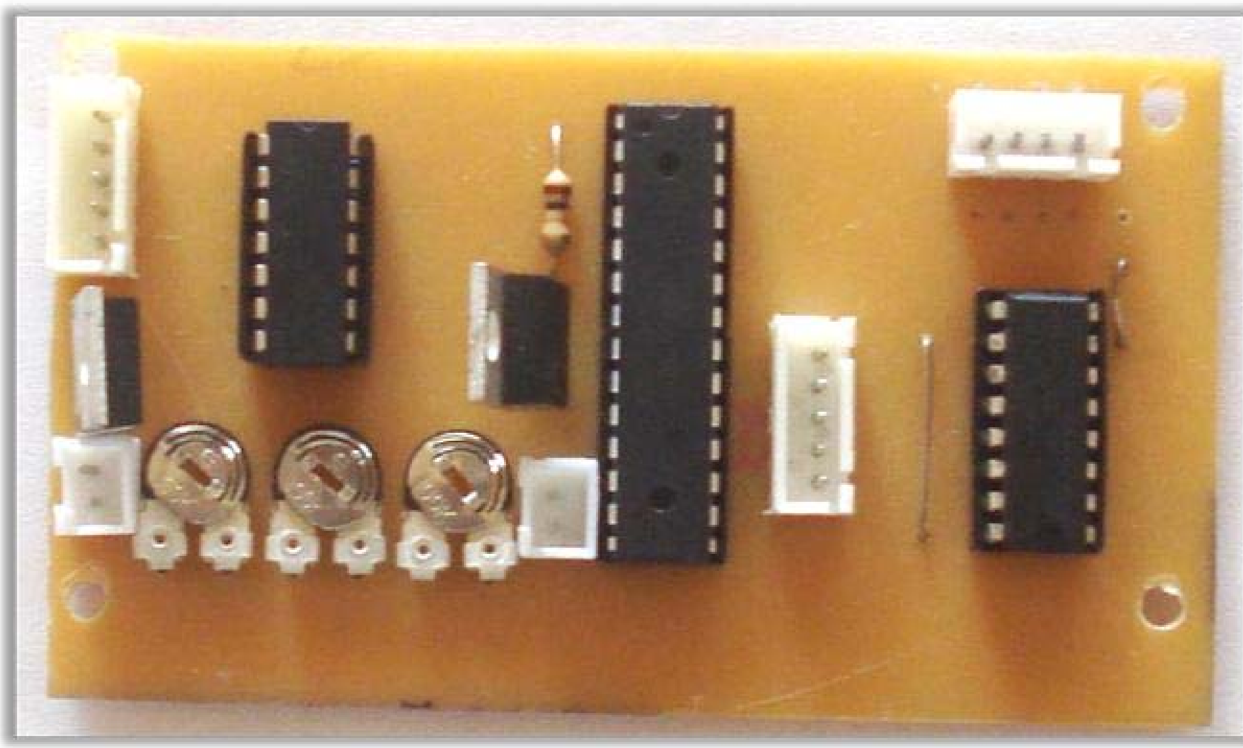
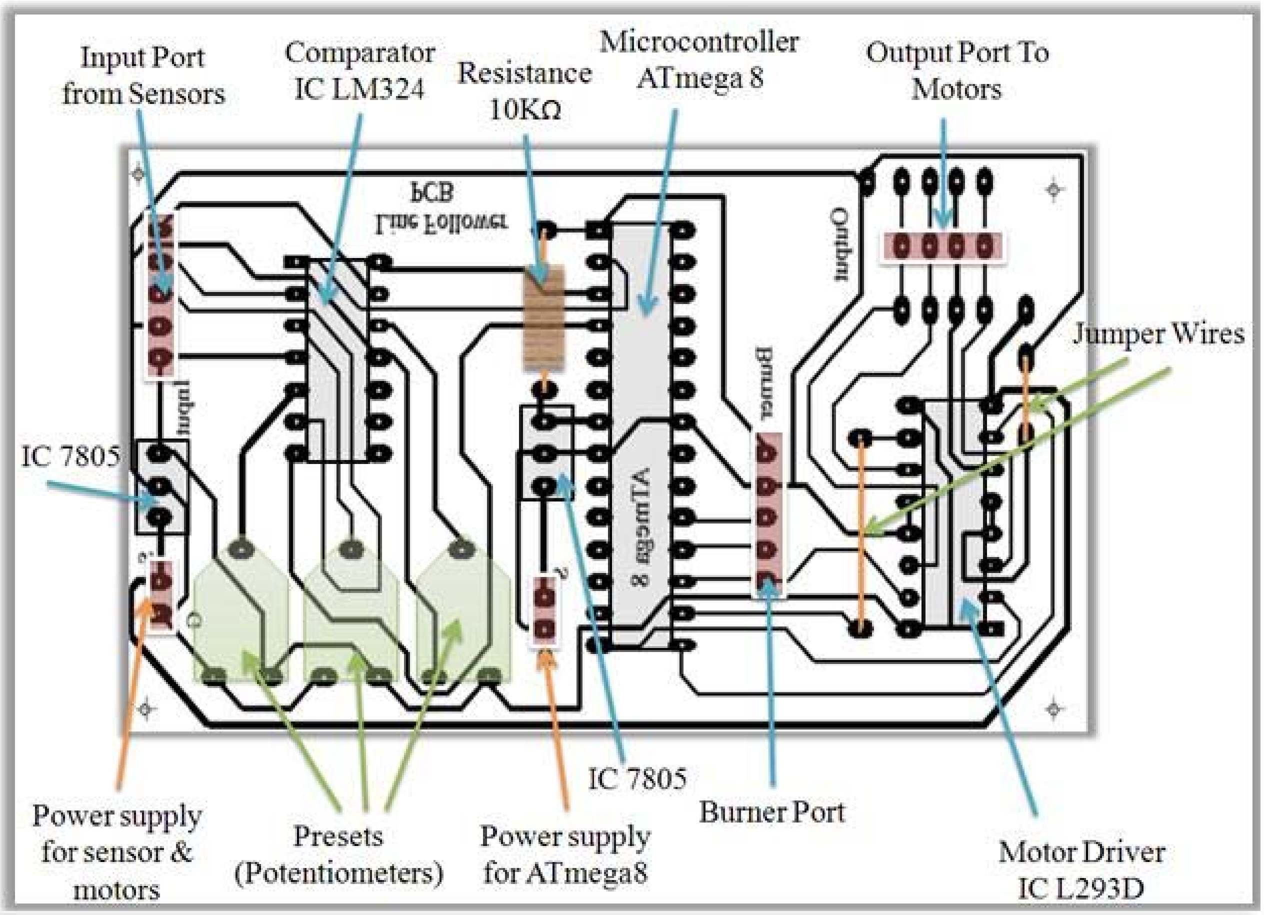
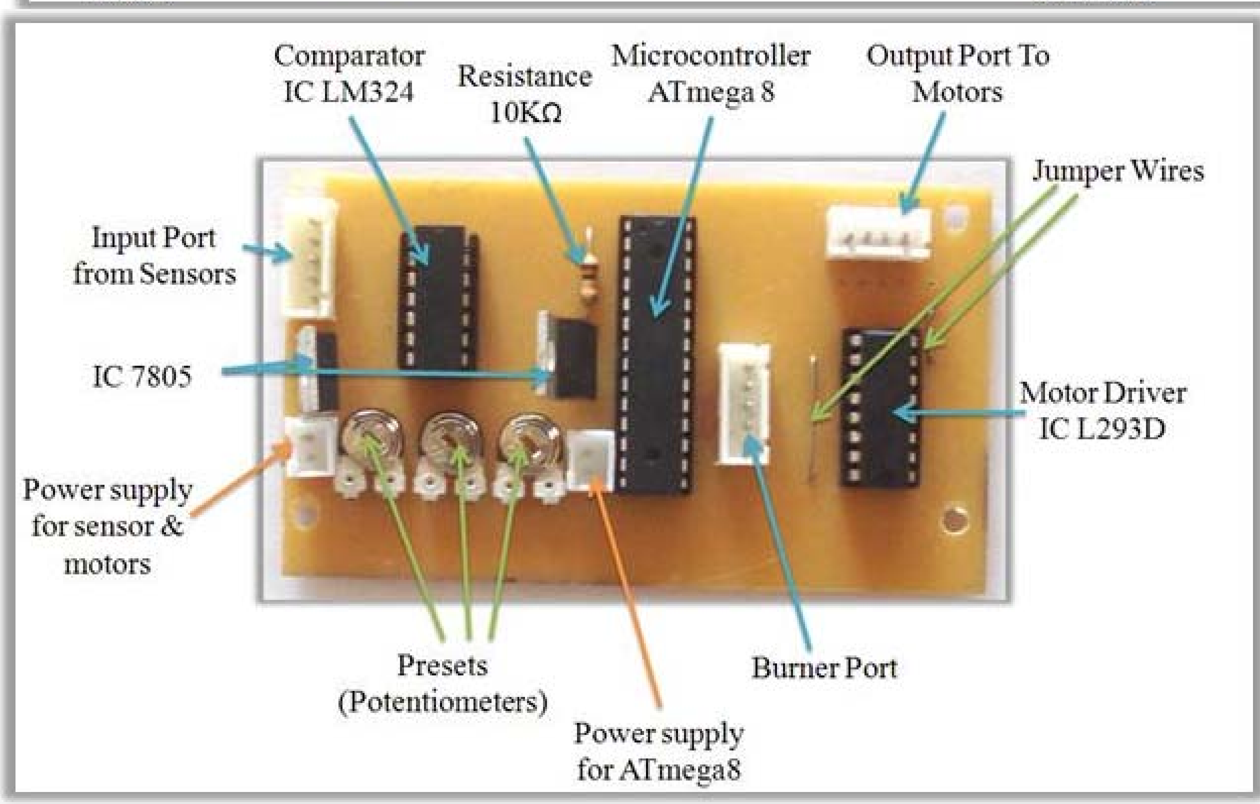


Figure: Photo taken after placing all Components

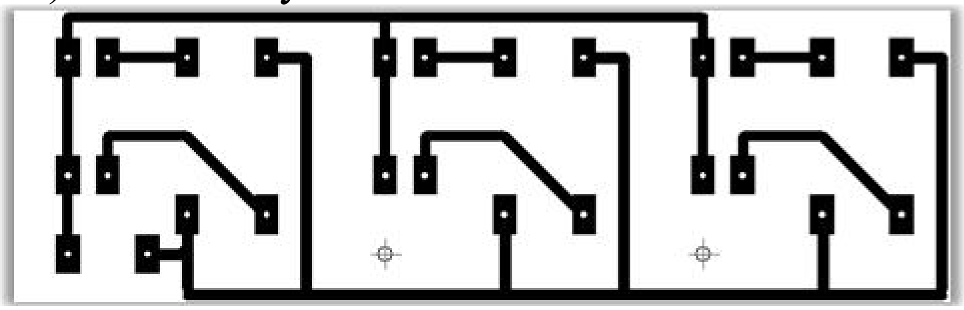
Explanation of all components:





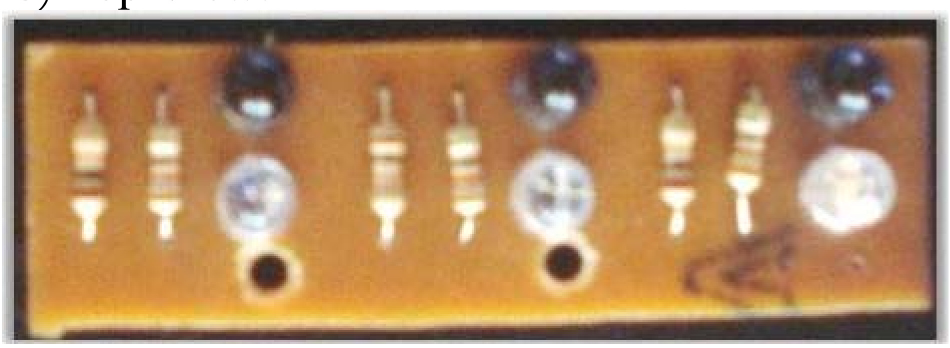
Sensor PCB

1) Bottom Layout:

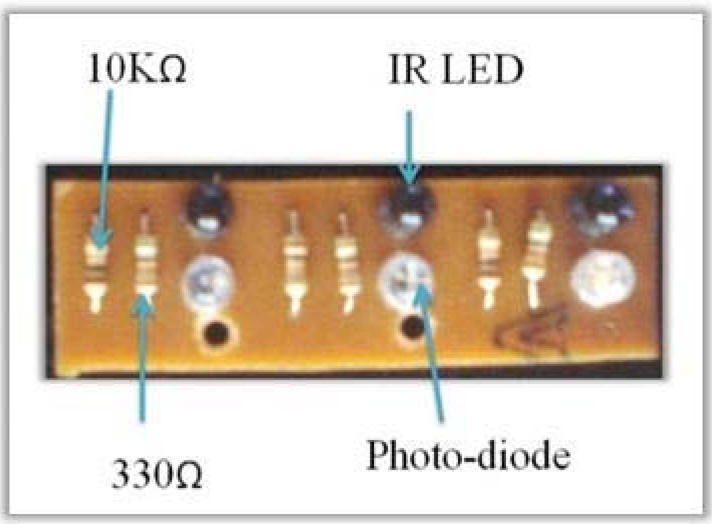


2) Bottom View:

3) Top View:



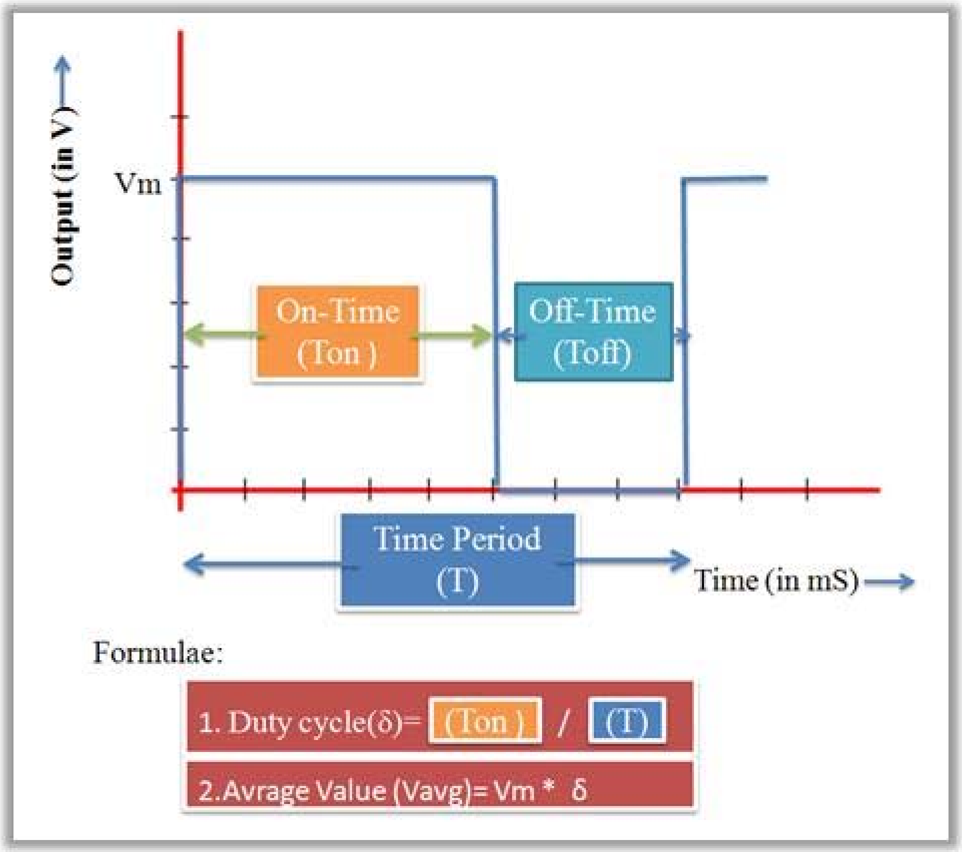
Explanation:



PWM

Pulse Width Modulation (PWM) is a method to generate Periodic Square Wave of various frequencies (time period) or duty-cycles.

The periodic square wave has two levels (high or low), with some constant frequency and duty-cycle.



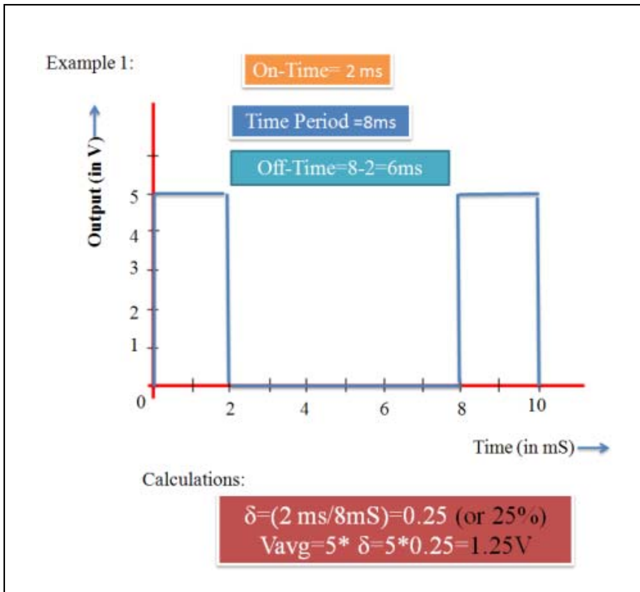
Definitions:

From above figure we can define some terms.

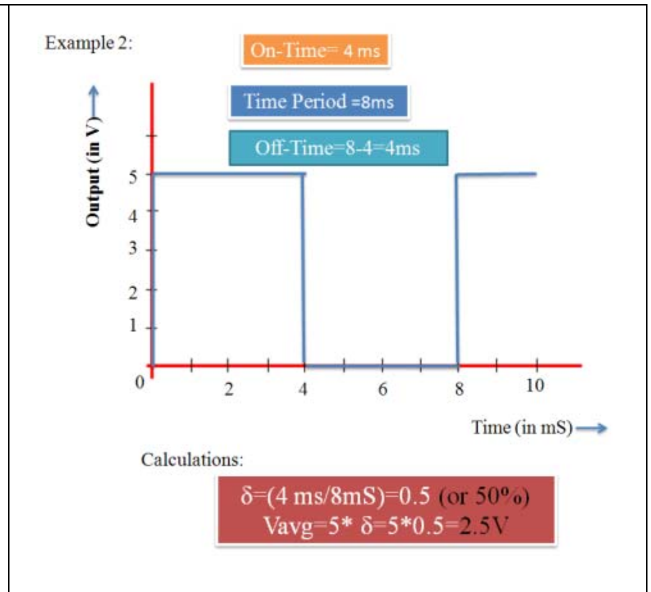
- **Time period (T):** The minimum time after witch function repeats itself is called as Time period.
- **Frequency (f):** =1/T
- **On Time (Ton):** Time for which function is at high voltage.
- **Off Time (Toff):** = T-Ton
- **Duty Cycle (δ):** =(Ton)/T
- **Average Value(Vavg):** =Vm*δ

  (Vm is value of high voltage)

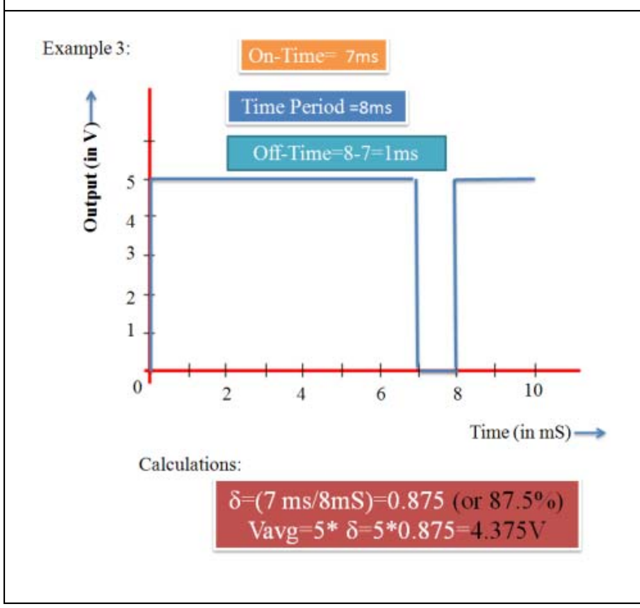
Examples



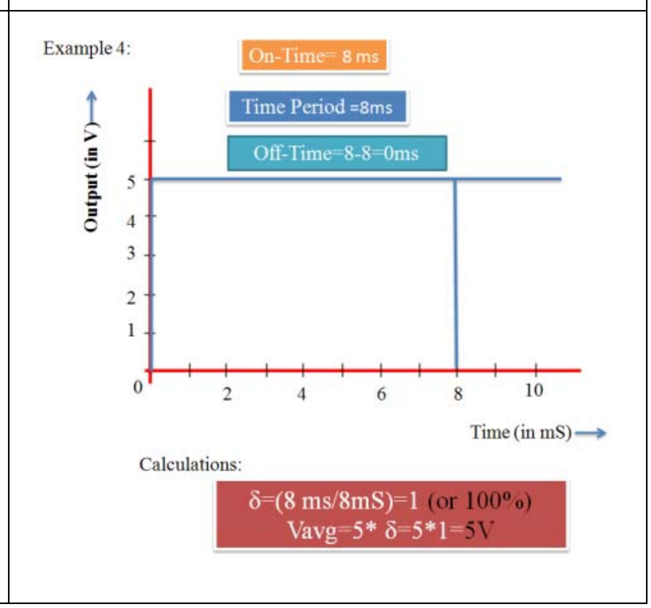
Example 1:
On-Time= 2 ms
Time Period =8ms
Off-Time=8-2=6ms
Output (in V)
5
4
3
2
1
0
2
4
6
8
10
Time (in mS)
Calculations:
δ=(2 ms/8mS)=0.25 (or 25%)
Vavg=5* δ=5*0.25=1.25V



Example 2:
On-Time= 4 ms
Time Period =8ms
Off-Time=8-4=4ms
Output (in V)
5
4
3
2
1
0
2
4
6
8
10
Time (in mS)
Calculations:
δ=(4 ms/8mS)=0.5 (or 50%)
Vavg=5* δ=5*0.5=2.5V



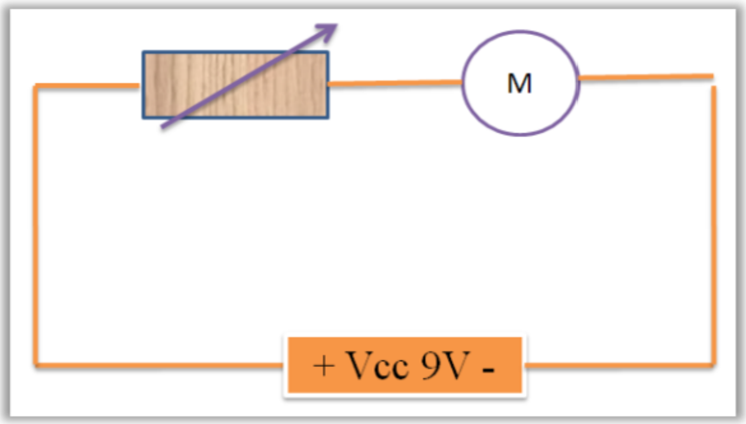
Example 3:
On-Time= 7ms
Time Period =8ms
Off-Time=8-7=1ms
Output (in V)
5
4
3
2
1
0
2
4
6
8
10
Time (in mS)
Calculations:
δ=(7 ms/8mS)=0.875 (or 87.5%)
Vavg=5* δ=5*0.875=4.375V



Example 4:
On-Time= 8 ms
Time Period =8ms
Off-Time=8-8=0ms
Output (in V)
5
4
3
2
1
0
2
4
6
8
10
Time (in mS)
Calculations:
δ=(8 ms/8mS)=1 (or 100%)
Vavg=5* δ=5*1=5V

USE of PWM in DC Motor

For example: If a person wants to drive a DC motor with variable speed but he has a constant Voltage supply. As an alternative for PWM he can add a variable resistance in series with the motor (As shown in figure).



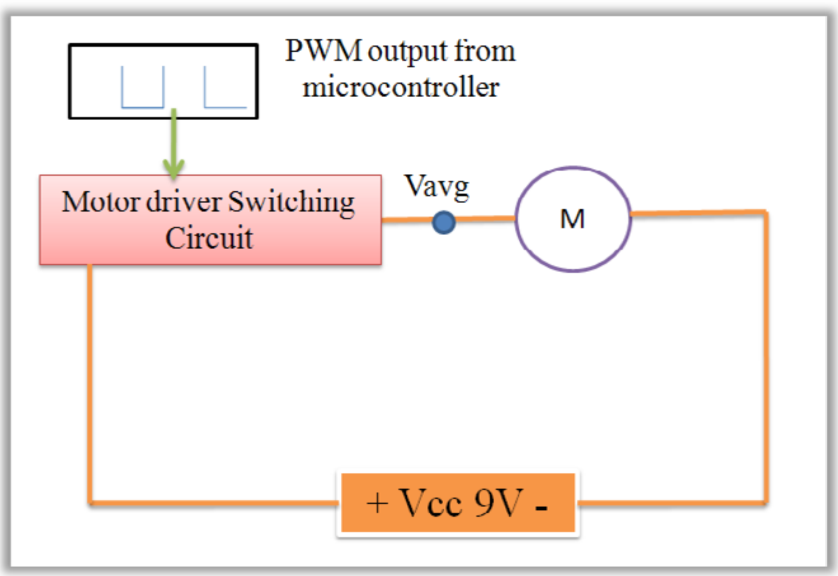
The drawbacks of this connection are:

1. The resistance value cannot be varied dynamically (automation is difficult).
2. There will be unnecessary power loss across the resistor.

Now "By adjusting the duty cycle of the signal (modulating the width of the pulse, hence the 'PWM') i.e., the time fraction for which it is "on", the average power can be varied, and hence the motor speed."

In other words by varying the duty cycle we are getting different values of average voltage (Vavg) across the motor resulting in different speeds.

We can design the system as shown in the following figure:



Now

$$\textbf{Vavg} \;\; \alpha \;\; \boldsymbol{\delta}$$

And

$$\textbf{DC Motor Speed} \;\; \alpha \;\; \textbf{Vavg}$$

Thus

$$\textbf{DC Motor Speed} \;\; \alpha \;\; \boldsymbol{\delta}$$

Advantages of PWM:

1. The is either on or off, not partly on as with normal regulation, so less power is wasted as heat and smaller heat-sinks can be used.
2. Since no resistor is used, there is no power loss.
3. Can be easily automated by programmable control.

Disadvantages:

1. We require extra circuitry to implement PWM (in AVR we have in built PWM-circuitry on chip).
2. Some authorities claim the pulsed power puts more stress on the fan bearings and windings, shortening its life.

Implementation of PWM:

For developing PWM, we require two properties:

1. Time Period (T)
2. On-Time Period (Ton)

Implementation of PWM in microcontroller:

In microcontroller we use clock of several Mega Hz. Thus time of one clock

$$\mathbf{T_{clk} = 1/(clock\ frequency)}$$

E.g. In ATmega 8 clock frequency is near about 1 Mega Hz.

$$T_{clk} = 1/(1M\ Hz) = 1\ \mu s$$

For implementation of PWM in microcontroller we require these variables:

- **Nt**= Number of clock cycles for one time period of PWM

  Nt=T/Tclk i.e., T*frequency of clk= Nt

- **OCR**= Number of clock cycles for On-Time of PWM

  OCR= Ton/Tclk

- **Ni**= It is a index of counter, that counts from Nt to zero and Zero to Nt in each cycle.

When if Ni $\geq$ OCR then PWM Output=Low

And if Ni < OCR then PWM Output=High