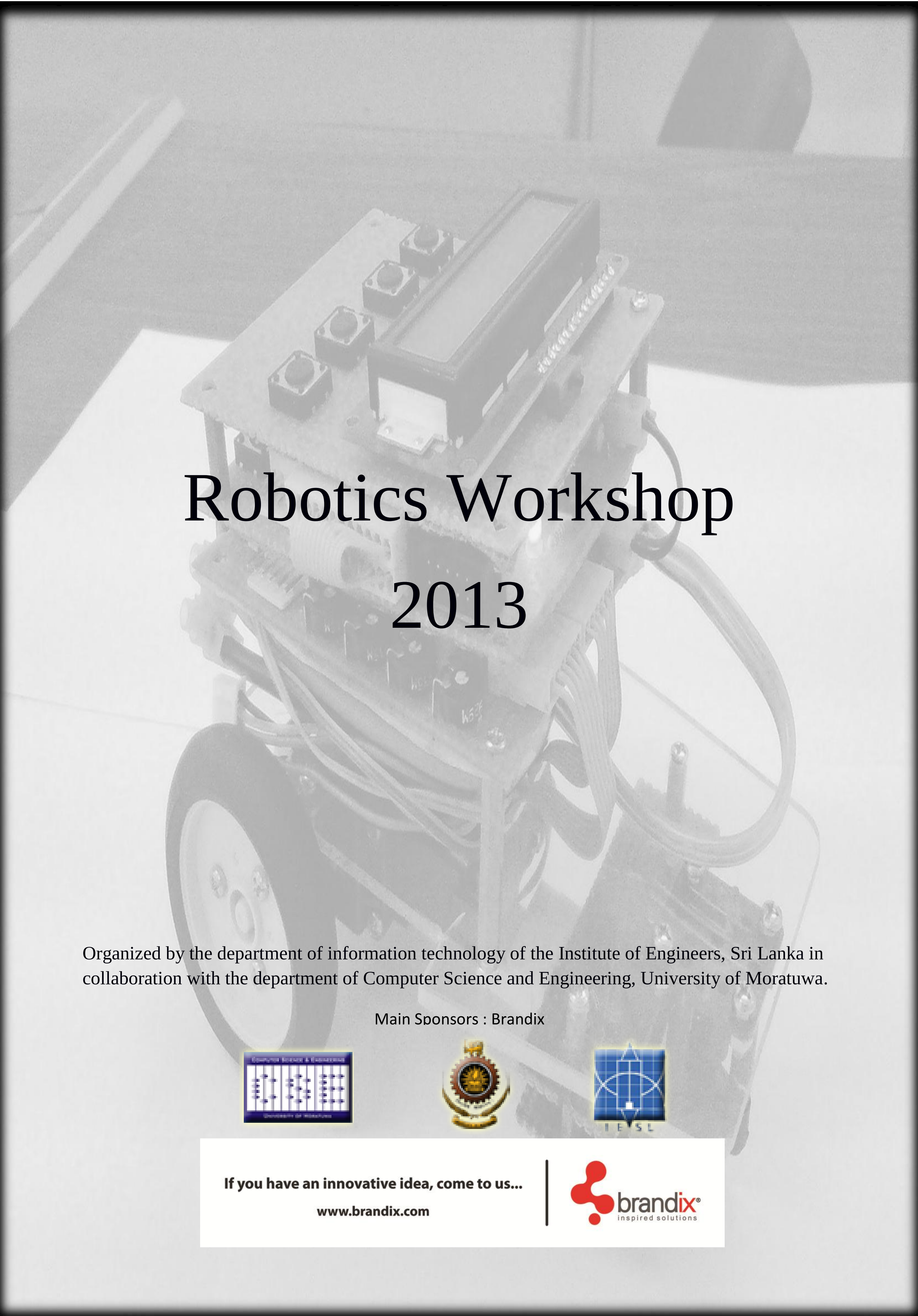


# Robotics Workshop 2013

Organized by the department of information technology of the Institute of Engineers, Sri Lanka in collaboration with the department of Computer Science and Engineering, University of Moratuwa.

Main Sponsors : Brandix

# Table of Contents

## Introduction

The purpose of this workshop is to give you the basic knowledge in making a robot that can follow a black line on a white background. This is a very basic step towards robotics. Our goal is for you to create your own robot, building on this knowledge which will show its colors at “IESL RoboGames 2013”.

The basic elements of automation can be identified as perception, communication, controlling and moving. Robotics helps you study most of them.

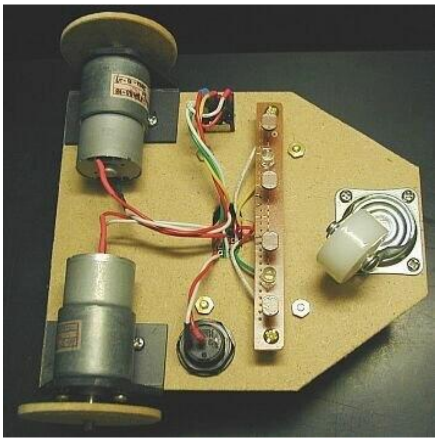
By following this booklet you can learn

- how to identify a path using sensors
- how to propagate (move along) the path using actuators ( motors)
- how to establish control to keep the robot on the path and to control the speed

## Basics of Sensors and Motors

The goal of this document is to give you the basic understanding about the basic components required to make a “line-following robot”. But we will not go into details about the circuitry or method of building. However we will discuss about the types of sensors and motors required in the following sections.

The expectations from you, in this competition, to build a robot that can follow a black line on a white background . First of all, your robot should be able to recognize the path. The usual method used for this is to emit radiation that belong to a specific region in the electromagnetic spectrum at the background and identify the reflected radiation from different colors on the background. As you have learned in your Ordinary Level Science lessons, white backgrounds reflect radiation (or light) very well when black backgrounds do not do so. We shall use this principle to identify the path correctly.



For this you must place sources that emit electromagnetic radiation and sensors to catch the reflected underneath your robot. Two or more of these source-sensor pairs (here, we will use eight of them) should be placed so, and by the strength of the reflected light that each pair receives we can know under which pair the black line (the path) is located. If the received reflected radiation is lower towards the middle then the robot is on-track. Otherwise the turning of the robot should be done to get the robot back on the track.

We can set these sensors such that the output will be 0 V when they receive a strong beam and +5 V when they receive a weak beam by using a transistor-circuit. We will discuss the detailed circuits later. These voltage signals can be fed to a micro-controller which can then be coded to control the speed of the motors at will.

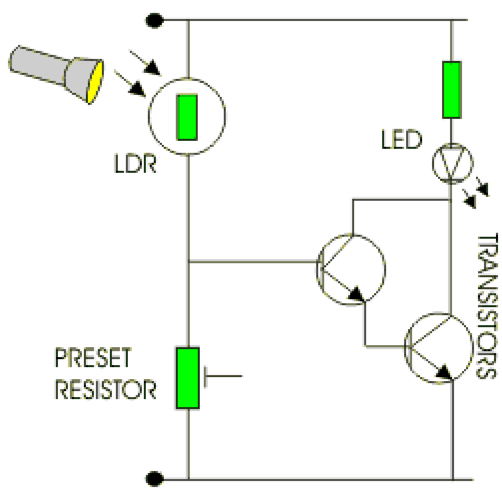
We need to use a micro-controller to control the motor-speeds according to the inputs of the sensors. A micro-controller is a small computer which is a compact collection of many equipment. In addition to a processor, it contains a memory along with many other apparatus, so that they can be activated with a minimal external circuit and, inputs and outputs. They are relatively cheap and are very useful for work with simple inputs and outputs as we do now. We can buy an adequate micro-controller for about RS 500 for our purpose. There are various different micro-controllers and you should choose one with enough equipment and enough input/output terminals. The circuit should be designed according to the micro-controller that you choose and then the micro-controller should be programmed according to the inputs, so that they will give the right outputs.

You can get a sequence of detailed articles on the use of micro-controllers from the following address, which were published in the "Vidusara" paper. They contain details about micro-controller programming, circuits used and so on.

Follow: http://www.ent.mrt.ac.lk/web/knowledgebase/index.xml

## Electromagnetic sources and sensors

### LDR (Light Dependent Resistors)




Anyone who has had an interest in electronic circuitry must have used LDR(Light Dependent Resistors). LDRs change the resistance between the terminals according to the intensity of the light that falls on them. When the intensity is high, the resistance is in the range of 400 Ω (Ohms) and can be several Megohms (MΩ) when in dark. The figures shows a circuit using an LDR. The LED will light up when the LDR is in the light. When LDR is in the light, its resistance lowers and the voltage across the preset resistor will increase. Because of this the base of the transistor receives a high voltage and the transistors are forward biased which lights the LED.

If you are using LDRs for your robot, you should use Light Emitting Diodes (LEDs) as the source of electromagnetic radiation.

### The operation of an LDR

This contains a semiconductor. When light (photons) are incident on it the electrons will be released from quantum energy levels and transmit electricity. This increases the concentration of electrons of the semiconductor and reduces the resistance.

### IR sensors

The basic operation of an IR sensor is the same as an LDR except for the fact that this detects Infra-Red radiation which has a longer wavelength than visible light.

If you use IR sensors as the sensors for your robot, you must use IR-LEDs as your sources. However, since the source and receiver are available in pairs it is not problematic in practical usage.

You cannot check if they work or not when you are buying them. However, you can check them using a simple camera in your mobile phone.

### Comparison of Sensors

When we consider the two options (IR sensors and LDRs), we can see that the main weakness of LDR is that it is sensitive to external light sources, which could cause the robot to not function as expected. It also reduces the quality of the robot due to the display of light. However they are the cheaper of the two options.

## DC Geared-Motors

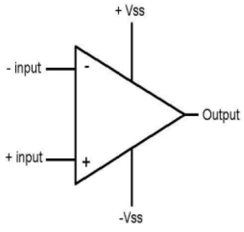
You need motors to provide the power to propagate the robot. If you use a simple average motor, you will require a lot of additional apparatus to obtain enough power. Because of this, you need a set of gears to get a higher power and a lower speed. And also, even if the signal to move forward is stopped, it will travel a bit further due to inertia, which makes the control of the robot even more difficult.

To overcome these issues we can use “geared-motors”. These motors have an in-built gear system which removes the need for an external gear system. And the technology used in them is such that when the speed increases, it reduces the torque of the motor, which acts as a self-regulation of speed.

The operation of these motors is the same as any other DC (direct current) motor, except for the internal gear system. And some of the motors are built with 'brakes' to stop the motor instantaneously when the signal is stopped.

# Analog to Digital Conversion

## Operational Amplifiers (Op-amps)

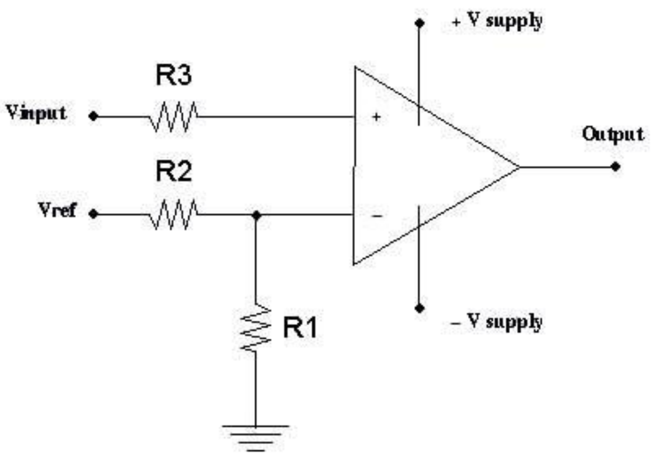



An operational-amplifier is a simple integrated circuit that can be used as a power amplifier that is used frequently in electronics. Like a transistor, this can be used for many purposes. Often they are used as amplifiers or voltage comparators.

### Identifying the terminals

There are two input terminals- namely, +input and -input (Figure 1). The output pin is also indicated in the figure. The voltages , +Vss and -Vss are the voltages that are required to operate the integrated circuit.

### Usage as a Comparator




An Op-amp can be used as a comparator as illustrated in the figure. The reference level must be given to the $V_{ref}$ terminal and the voltage that you want to compare must be given to the $V_{input}$ terminal. The Op-amp then compares the voltages and if Vinput < Vref , Vout will be given as zero, whereas if Vinput>Vref , the output will be given as +5V.

Note: Here, the -input is used as the Vref and +input is used as the Vinput. If the two terminals are changed, the operation will be reversed.

We shall not go into details about other operational states of an op-amp because they are not required.

## IR Sensor Circuit- Practical Experiment

What we hope to do here is to see how we can build a circuit suitable for the input of a microcontroller. We will use a sensor that contains an IR emitter and receiver (see the following figure).

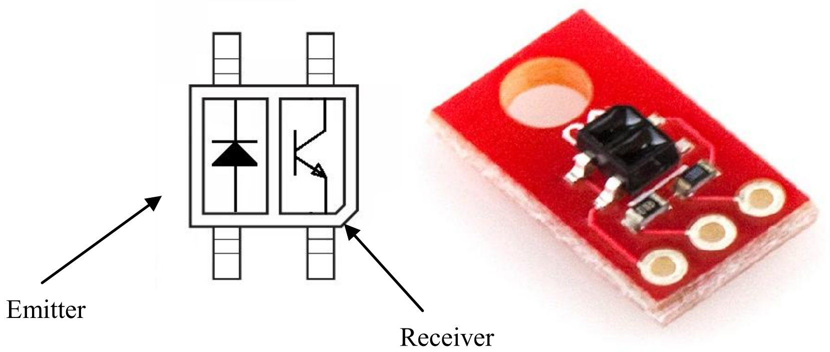
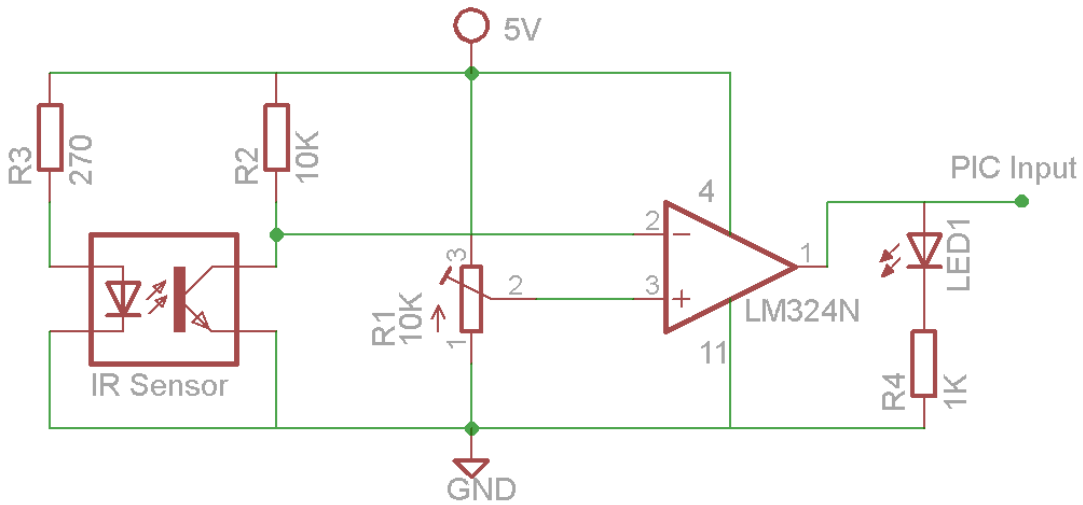



Figure : IR Sensor

The current that can flow through the IR receiver depends on the light incident on it. Through a normal receiver, a minimum current flows in darkness and a maximum in light. Therefore, by using a resistor in series with it, we can convert it to a difference in potential. The aim of this circuit is to identify black or white objects. The logic voltage of the microcontroller that we use is 5V. Therefore let us get a 5V output from this circuit for a white background and 0V output for a black background. The complete schematic of the circuit is shown below.




## Operation

A complete understanding is not required to make a robot. This section is only included for the sake of your knowledge. You will need a basic understanding of the function of a transistor which will be explained in our workshops.

An op-amp is used to convert the analog voltage of the receiver into a digital signal (0V, 5V). If the (+) terminal of the op-amp has a low potential compared to the (-) terminal the output will be 0V and 5V if it is higher. The potential between the two terminals of the photo-transistor of the receiver will decrease when a light is incident on it, which gives a low voltage on the (-) terminal of the op-amp. Similarly, in the absence of light, it gives a high voltage. By adjusting the R1 resistor, we can control the photo-sensitivity of the circuit. We have used an LED to check the output of the circuit. By changing the (+) and (-) ends of the op-amp, we can make the LED go off when the light is incident on the sensor. When connecting to the microcontroller we must remove the LED. In this circuit, we have used an LM324 IC for the op-amp. It contains 4 op-amps in one casing, so one of them can be used for 4 sensors. The 4$^{th}$ pin of the IC acts as the power supply for all four op-amps so all you have to do is give it +5V (+ terminal) and 0V(-terminal) to the 11$^{th}$ pin.

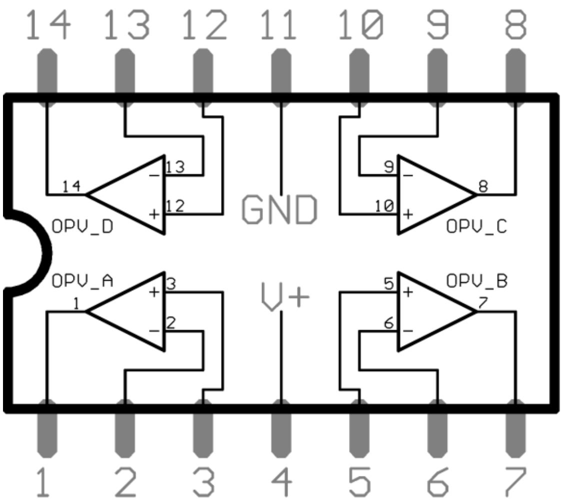


Figure : Integrated Circuit - LM324

## Apparatus

1. Infra-red sensor
2. LM324 IC
3. 10K variable resistor
4. 270 Ohm resistor
5. 10K resistor
6. 1K resistor
7. LED
8. 8.Circuit wires
9. 5V supply

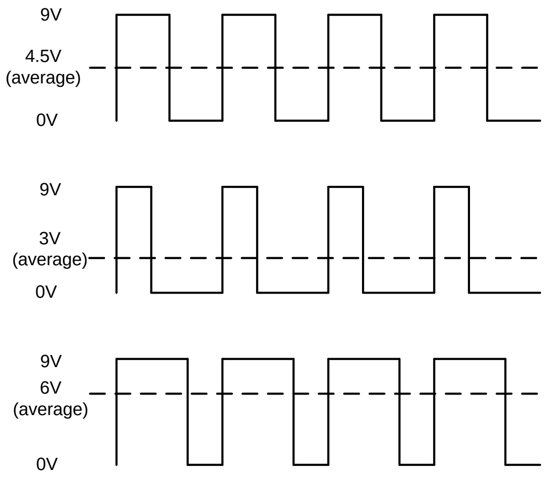
## Experiment

Build the circuit on a bread-board to observe the functionality. Use four 1.5V batteries or a 6V power supply as the power source. Aim the sensor at a white background and adjust the variable resistor so that the LED will remain lit when the circuit is switched on. When aimed at a black background the light should go off. However, further adjustments to the sensitivity will have to be made according to the ambient light.

## Using Pulse Width Modulation (PWM)for Speed Controlling

The fundamentals on Pulse Width Modulation (PWM) which is used to control the rotation of a DC (Direct Current) Motor is considered in this section. Usually we control the rotational speed by changing the voltage which is applied to a Motor. There are two disadvantages in this method. The first one is the controlling of the speed of the motor is limited to a small time period. The reason for that is the rapid decrease of the torque with the voltage. Which means the torque is not sufficient to rotate in the required speed. The second disadvantage is the outputs of the Micro controller which we're suing here are digital. Because of that reason if we need to apply a variable voltage we're supposed to use a Digital to Analog Converter.

In PWM method a constant voltage (the average voltage of the motor)is applied as a pulse rather than applying a varying voltage. Let's consider the following pulse patters.




According to the first diagram one half of the wave is 9V and the other half of the wave is 0V. In this scenario the average value of the voltage which is applied to the motor is 4.5V. The ration between the time period where 9V is applied and the Period of the wave is called Duty Cycle. Hence the duty cycle of the first wave is 50%. In the second pulse pattern 9V has been supplied within a 1/3 duration of the period of the wave. Therefore the duty cycle of the wave is 33.33% and the average voltage applied to the motor becomes 3V. The duty cycle of the 3rd pulse is 66%. By changing the time where the voltage is applied and not the average voltage applied could be changed. The frequency of the pulses is quite important where it should be neither too much or too low. For a usual motor 10KHz is sufficient in general. The micro controller we use have two PWM modules and the way they are used will be discussed later.

# Model Circuit

## Basic Requirements

This circuit was developed to create a line following robot and give facilities to allow future developments. It has the ability to control two DC motors by controlling the rotation and plug the sensors and input output devices whenever necessary in various events.

The removing and plugging of the micro controller to the programmer, when programming the micro controllers is a common problem. But this circuit is designed to reduce that possibility. Therefore the micro controller is existed with RJ11 connector which has 6 pins and which enables the use of ICD3 In-Circuit Debugger Tool.

A serial connection is provided as the next option because the above method is expensive. A micro controller with a boot loader can download the program code to the micro controller itself by connecting it with the computer. The boot loader should have been inserted to the chip previously using some other method (Using an external programmer) and the beginning of the program code should be clearly mentioned. If not the program code you write may replace the boo loader and the reprogramming of the boot loader using an external programmer may be required. The sample code on how to use the boot loader is given in our examples list.

## Understanding the circuit

By this moment you should know the following points

- Pulse Width Modulation - To control the speed of motors
- IR Sensor Experiment - To sense the path
- Basic knowledge on Micro controllers

It has the capability to execute the commands stored inside using a computer

- The ways to write commands
    - Using the Assembly language. This is a low level programming language. MPLAB software is used for assembly coding.
    - The variations of C language. PIC C, Mikro C, Hitech C etc.
    - The further details about these softwares can be acquired through following official web sites.

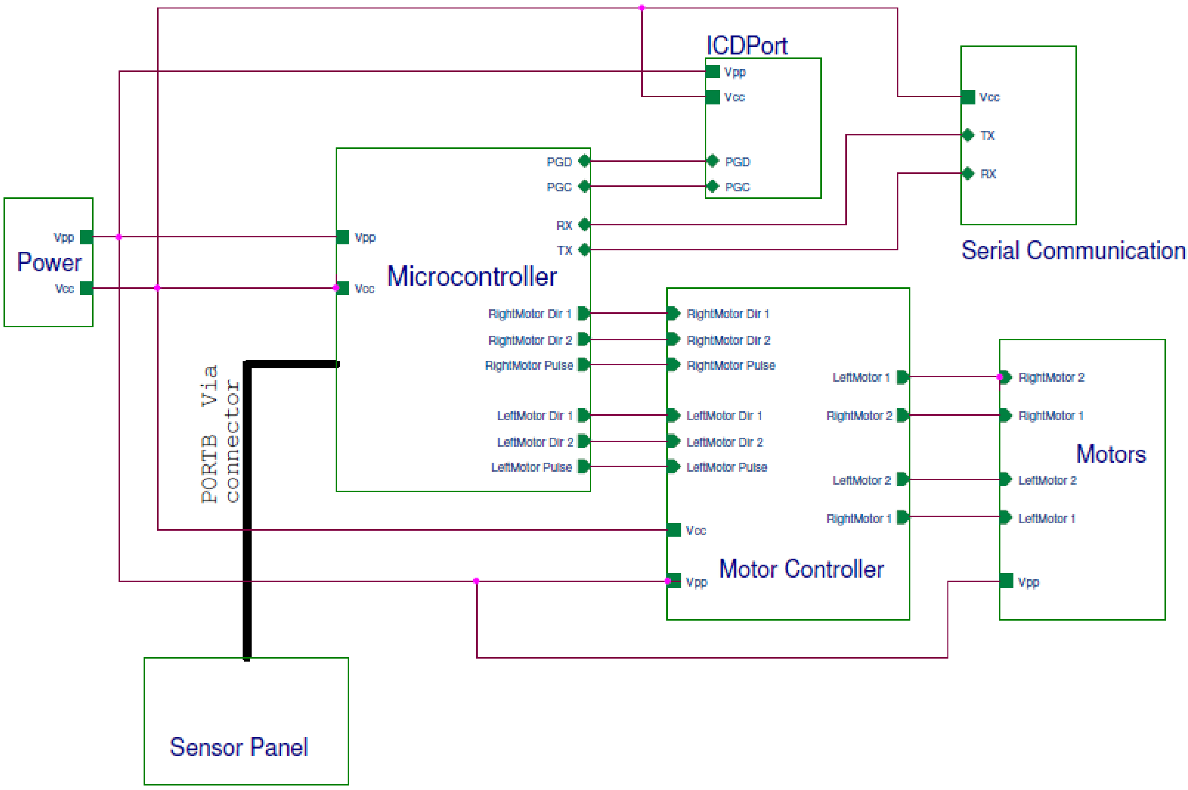
- PIC C - http://www.ccsinfo.com/content.php?page=compilers
- MikroC - http://www.mikroe.com/eng/products/view/7/mikroc-pro-for-pic/
- HitechC - http://www.htsoft.com/ (Supporting Documents are given with the MPLAB software)

- The examples given in this tutorial are written by PIC C
- Whichever method is used you will get a .hex file at the end.
- The methods to copy programs to chips
  - Use a serial programmer (JDM Programmer). Here the code is copied to the chip by removing it from the circuit and plugging into another circuit which is connected with the computer. Though this is a different task at the experiment level it is the most cost effective method. ICProg, WinPic800 softwares control this downloading mechanism.

  - Using a serial connection without removing it from the computer. Here a bootloader (a special software to copy the program code )should be available inside the chip. As the bootloader is essential the programmer is only a secondary usage,because a serial programmer to copy the bootloader is needed at some point. But in other cases the code can be directly dowloaded to the chip while in the circuit. Availability of a serial port in the computer is essential. The PICdownloader software controls the copying of the program code.

  - Use ICD/PICkit programmer software. This can be done very easily through USB while the chip is in the circuit. The copying of the code is controlled by MPLAB software

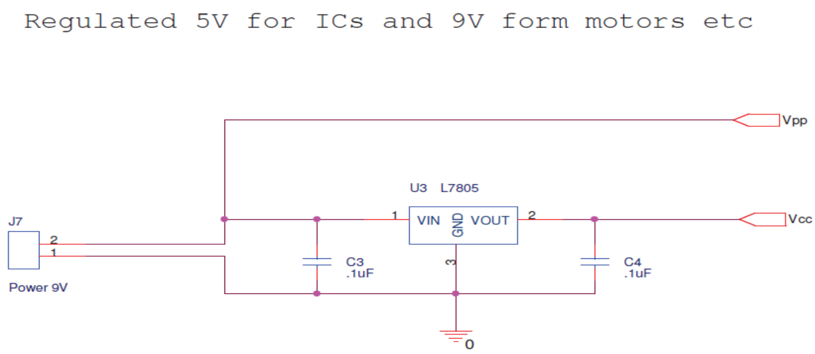
## The elements of the circuit



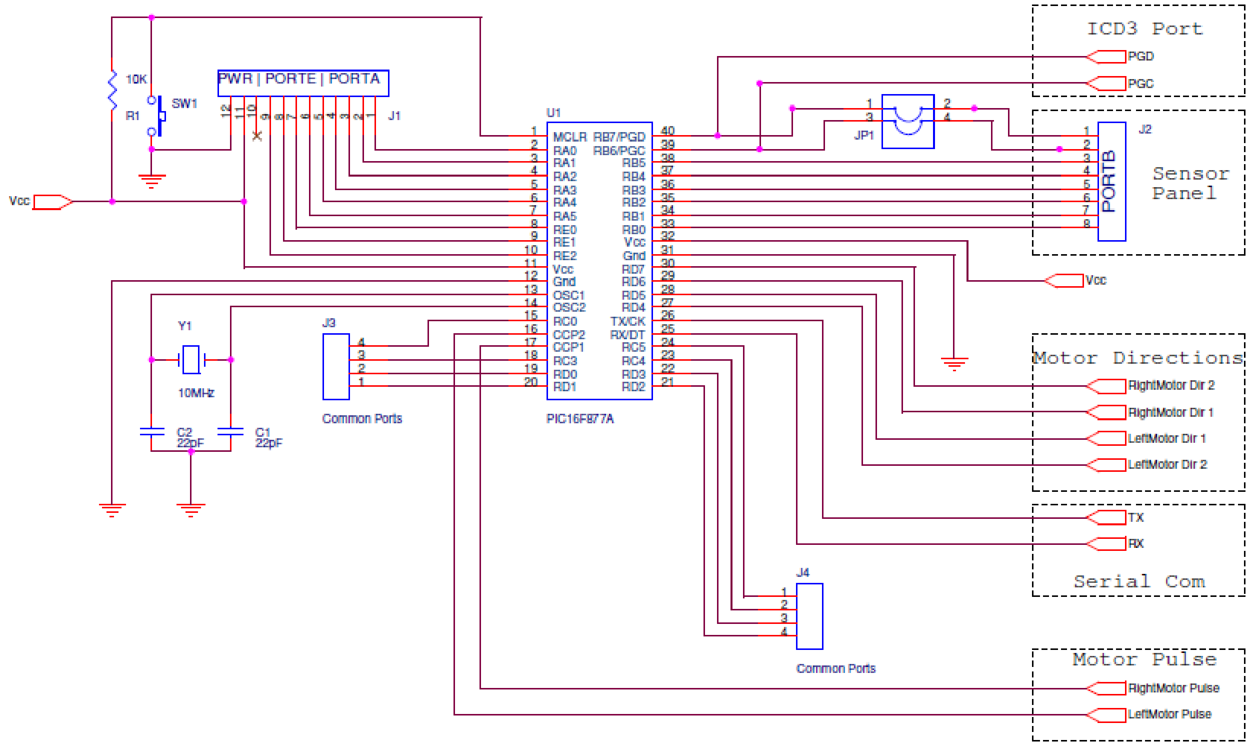

The above diagram shows the arrangements of the circuit. The detailed diagrams of each block are given below separately.

### Power Supply Circuit

The 9V voltage received from the battery is converted to a 5V supply which can be given to the Micro controller by this circuit. In addition to that to supply the 9V supply to the motor the 9V is directly given to the L 298 circuit. The battery voltage need not to exactly to be 9V. It can be a voltage between 7V-18V which can be bearable by the motor.

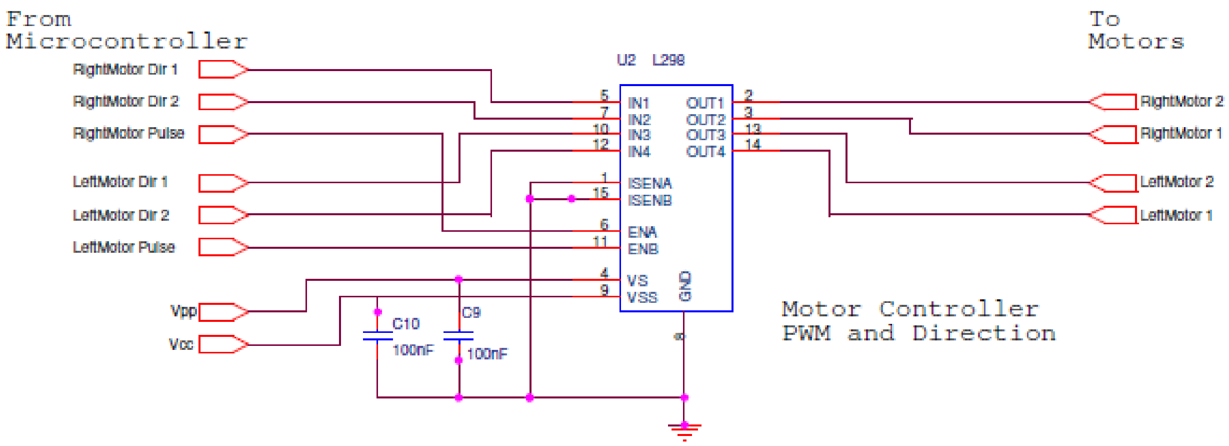


## The main Micro Controller Circuit



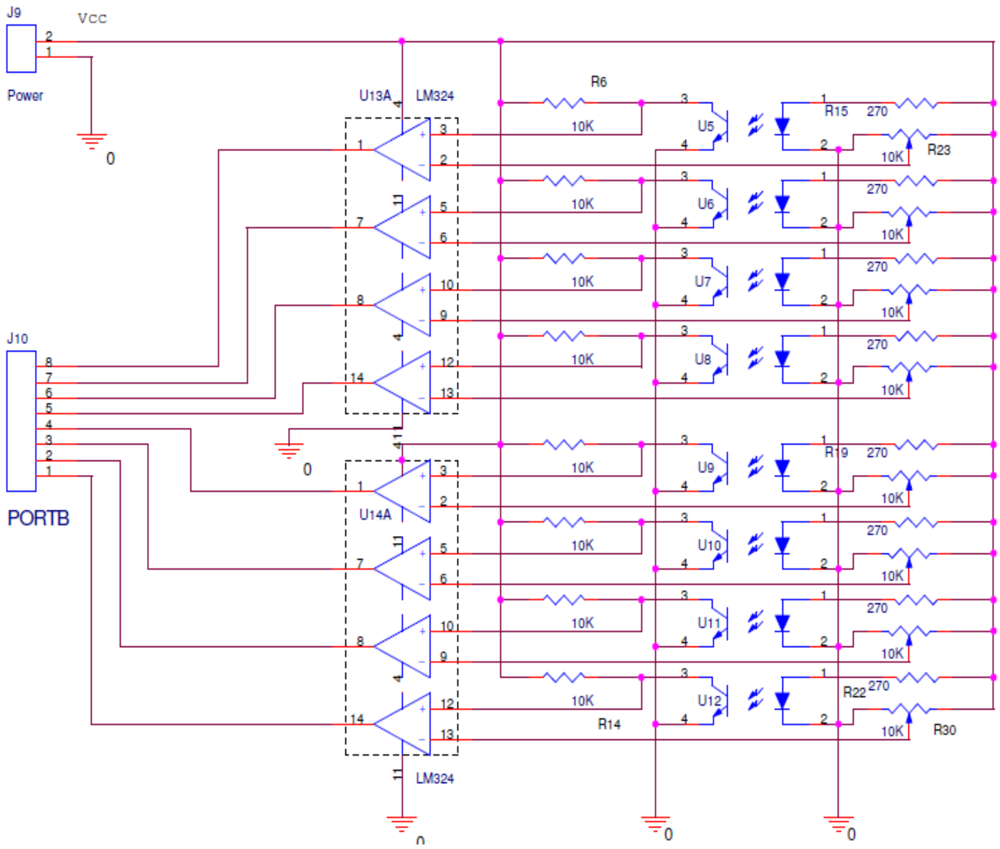

Supplying power to the micro controller, providing the clock pulse, establish the reset circuit and receiving inputs and providing outputs are the main functionalities of this circuit. By closing the sw1 switch in the diagram the micro controller can be reseted. Then the circuit will run again from the beginning.

## Motor Controller Circuit




This circuit controls the rotational speed and the rotational direction of the motor according to the output signal of the micro controller

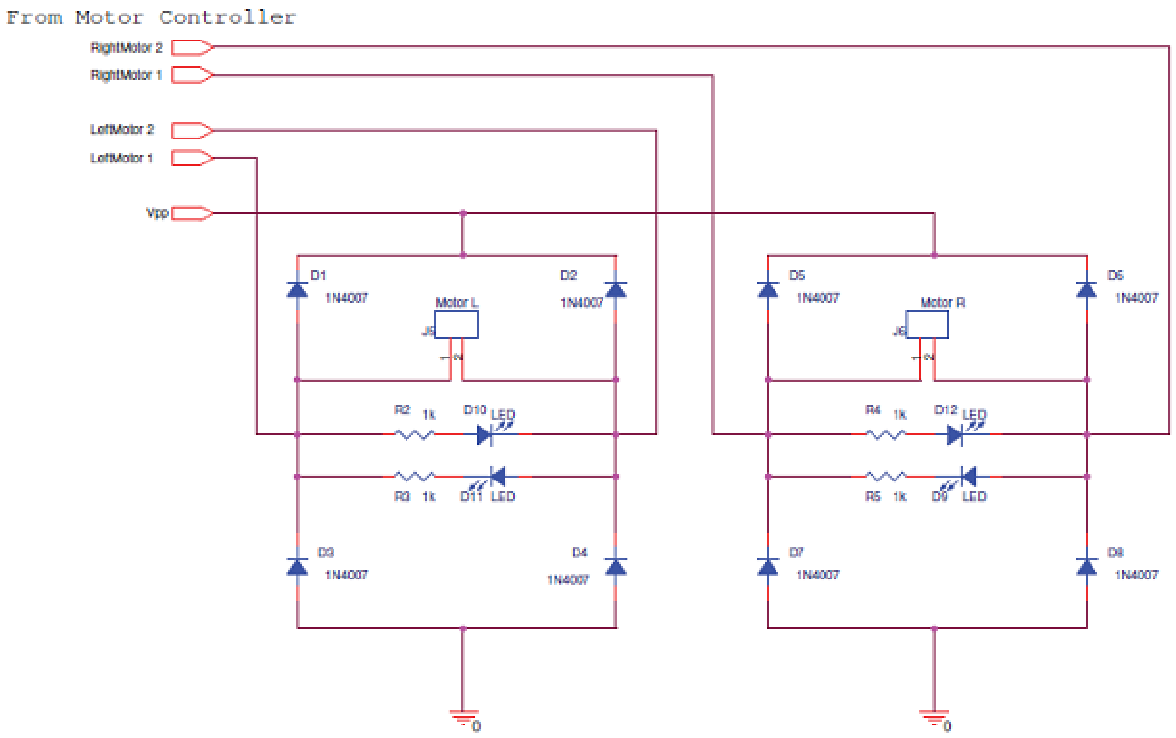
## Infrared Sensor Circuit




This circuit transfer the output signals of the path identifying sensors to the micro controller circuit. In this scenario only 8 sensors are used. But according to the need of sensors the circuit can be extended and built on a veroboard. The gap between the sensors is determined according to the width of the path the robot follows.

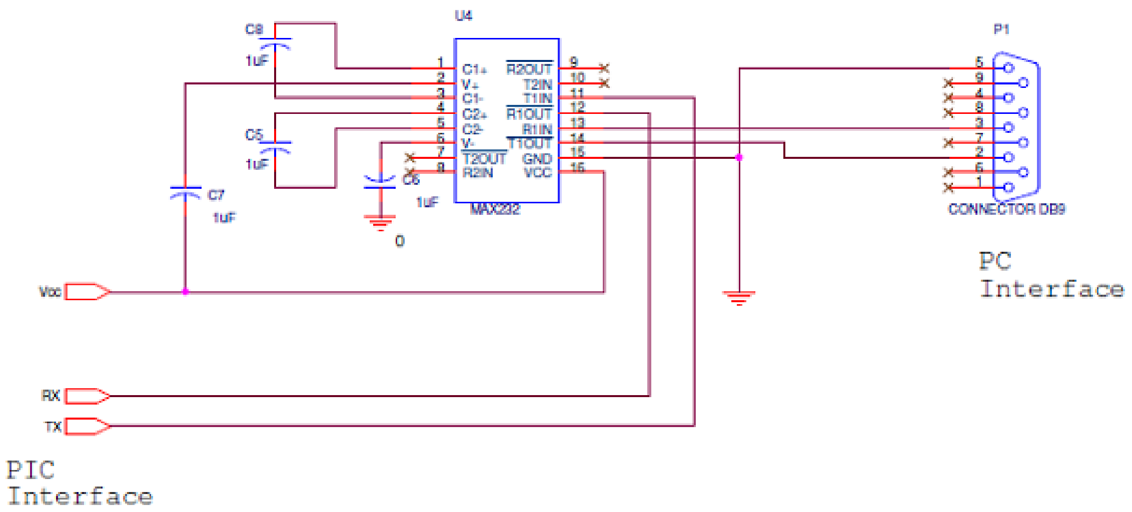
## H-bridge circuit

This circuit supplies the output from L298 circuit to motors. The main purpose of this circuit is to protect the L298 circuit from the back EMF produced due to the high speed switching between on and off states. The two light emitting diodes are there for the convenient observation of the directions of rotation of the motors.



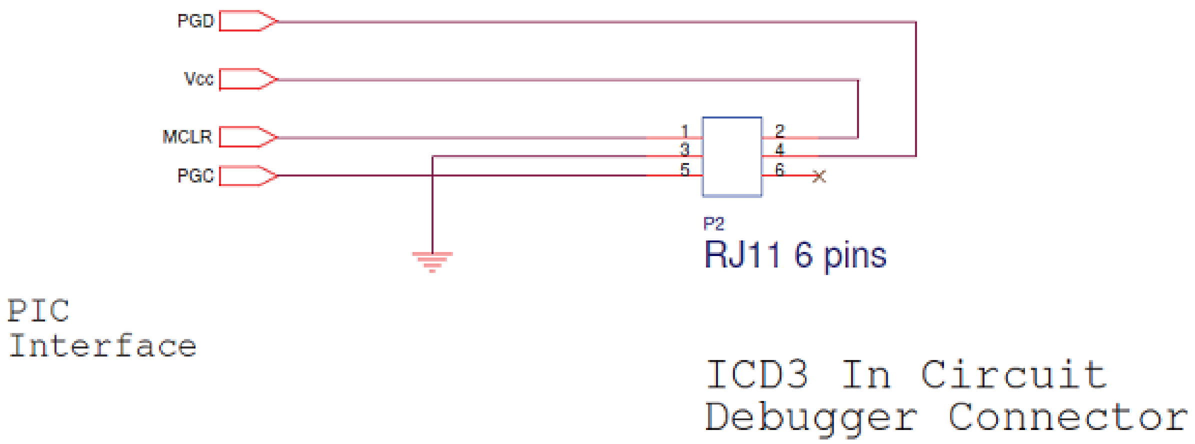

## Serial port circuit

This circuit connects the micro-controller with the serial port of the computer. This can be used to program the micro-controller without disconnecting the micro-controller from the circuit.



### MPLAB ICD connector

This connector is developed for the MPLAB ICD programmer manufactured by Microchip Technology Inc. ICD is a bit expensive device that can be used to program a micro-controller. This connector can be directly connected to an ICD2 or ICD3. But instead of using an RJ11 connector, a programmer like PicKit2 can be used by connecting outputs in the correct manner. If you have such a device and if you have any problems regarding connecting it feel free to contact us via iesl.robogames@gmail.com



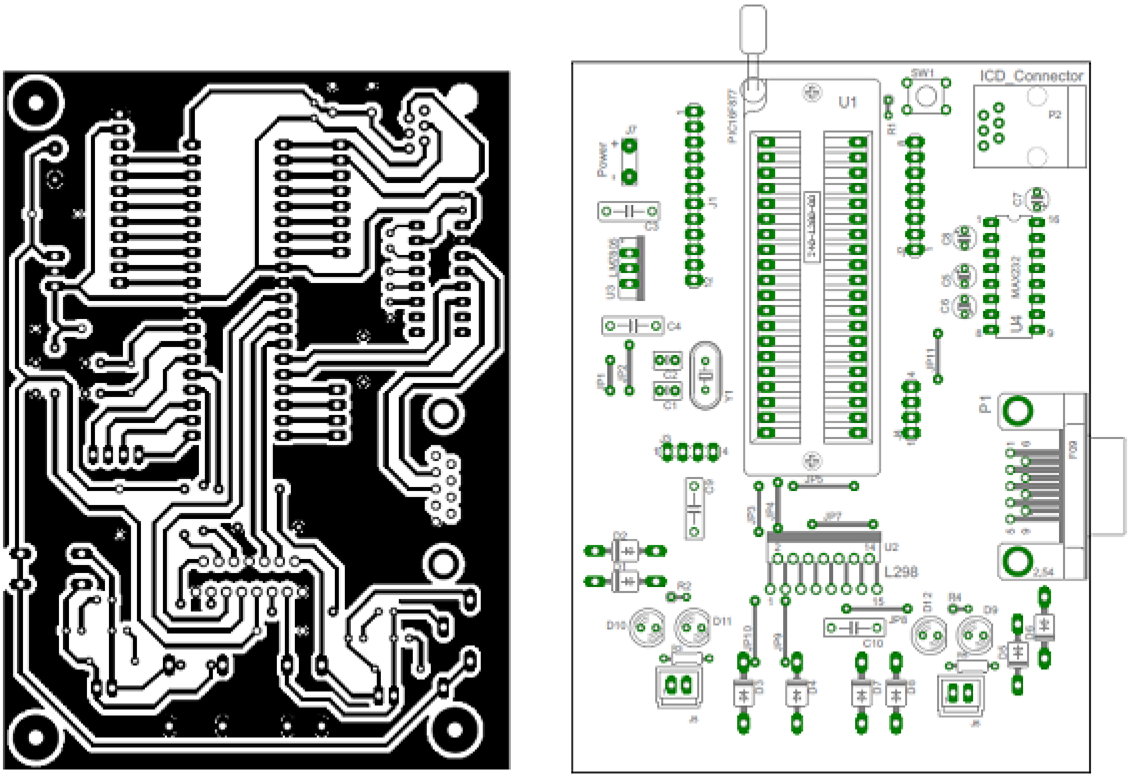
## Required components

| Item | Quantity | Reference | Part |
| --- | --- | --- | --- |
| 1 | 2 | C2,C1 | 22pF |
| 2 | 2 | C3,C4 | .1uF |
| 3 | 4 | C5,C6,C7,C8 | 1uF , 25V |
| 4 | 8 | D1,D2,D3,D4,D5,D6,D7,D8 | 1N4007 |
| 5 | 4 | D9,D10,D11,D12 | LED |
| 6 | 1 | J1 | 12-pin Pin Header |
| 7 | 2 | J10,J2 | 8-pin Pin Header |
| 8 | 2 | J3,J4 | 4-pin Pin header |
| 9 | 1 | J5 | 2 pin Connector – Motor L |
| 10 | 1 | J6 | 2 pin Connector – Motor R |
| 12 | 1 | J9 | 2-pin Pin header |
| 13 | 1 | P1 | CONNECTOR DB9 |
| 14 | 1 | P2 | RJ11 6 pins |
| 15 | 4 | R2,R3,R4,R5 | 1K |
| 16 | 16 | R1,R6,R7,R8,R9,R10,R11,R12, R13,R22,R23,R24,R25,R26, R27,R28,R29 | 10K |
| 17 | 8 | R14,R15,R16,R17,R18,R19, R20,R21 | 270 Ohms |
| 18 | 1 | SW1 | Tach switch |
| 19 | 1 | U1 | PIC16F877A |
| 20 | 1 | U2 | L298 |
| 21 | 1 | U3 | L7805 |
| 22 | 1 | U4 | MAX232CPE |
| 23 | 8 | U5,U6,U7,U8,U9,U10,U11, U12 | IR reflectance sensors |
| 24 | 2 | U13,U14 | LM324 |
| 25 | 1 | Y1 | 10MHz |

## Printed Circuit Board Design

The PCB design is shown in the following diagram. This diagram should not be used for printing and a diagram drawn in correct scale and dimensions for printing is included in the provided CD.

If you find it difficult to print the circuit board contact us via iesl.robogames@gmail.com and we will provide you with a PCB.




Top view of the PCB after connecting the circuit components

## Integrating the circuit

- Make sure that you use an IC Base or a ZIF socket to connect the micro-controller. If you are using a Serial Programmer(JDM Programmer) it would be better to use a ZIF socket.
- Depending on the method that you use to program the micrcontroller, you may skip building the ICD and serial communication parts of the circuit board.
- Please use suitable connectors where J notation is used.
- You should consider the sensor panel as a separate circuit. It should be built in a separate board and connected to the main circuit using two connectors for power and data.

## A few simple example codes

### 1-Blinking an LED

Controlling several outputs according to a given pattern is a simple usage of micro-controllers. Instead of a knight rider circuit or a two bulb robot circuit, you can connect a series of LED bulbs that can be controlled according to a given pattern, to the output pins of the micro-controller.

Let's see how to give commands for a selected output pin. To identify and select pins refer 16F877A datasheet.

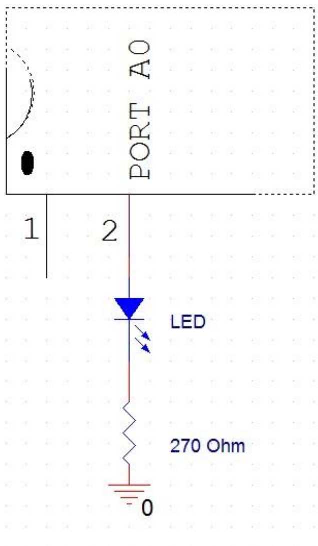
A special characteristic of a micro-controller is that it should be involved in some kind of action as long as power is provided. It cannot be ended or stopped. To achieve this, a while(1){} loop can be used. It will make the code inside the loop run again and again because what it does is checking the condition whether 1 is equal to 1 and if the condition is true running the code inside the loop.

CCS C software helps us to control the pins of a micro-controller easily. You can simply change the logical value of the 0th pin of port A that is the second pin of PIC 16F877A to 1 by including the line output_high(PIN_A0); in your code. This can tested using an LED earthed via a 1 kilo Ohm resistor.

But since the micro-controller takes a negligible period of time to execute one line of code, to observe the changes with time, you must command the micro-controller to wait for some time that is enough for a human to observe the change. We can command the micro-controller to wait 1000 mili seconds by adding the line delay_ms(1000);

output_low(PIN_A0); will bring the logic value of the second pin of the PIC back to logic value 0 and again a command for a time delay is included. The closing curly bracket will lead the micro-controller to check for while(1) condition.

```
void main()
{
  while(1)
  {
   output_high(PIN_A0);
   delay_ms(1000);
   output_low(PIN_A0);
   delay_ms(1000);
  }
}
```



## 2-Reading Input

It is not a difficult task to get inputs to the micro-controller. You can store the input at pin B0 in a variable by input(PIN_B0). But to check the value that was read we will need to get an output.

To do that, in the following code the input value from pin A0 is given out as an output from pin 2.

```
int value;
void main()
{
    while(1)
   {
       value = input(PIN_B0);        //Reading the Input State
       if( value == 1)               //If Input Is high
      {
         output_high(PIN_A0);        //Light up LED
      }
      else
      {
         output_low(PIN_A0);         //Otherwise dont light UP
      }
   }
}
```

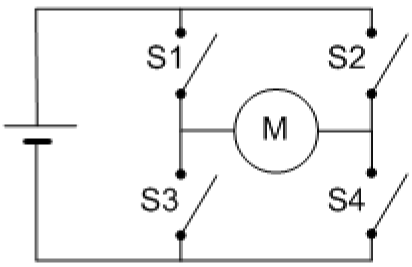
## 3-3-PWM CHECK

### Controlling the angular speed by Pulse Width modulation

The main purpose of this example is to describe how to control the PWM module of the microcontroller by a computer program.

An L298 module is used here because the current output of the microcontroller is not enough to drive a motor. This module uses 3 outputs to control a single motor. One output of these is used to produce the PWM signal whereas the other 2 are used to control the direction.

### The operation of L298




The L298 IC mainly includes 2 H-bridge circuit as shown in the figure. H-bridge is the circuit which takes the shape of the letter ‘H’ consisting 4 switches. Switches S1 and S4 are turned on when we connect one edge to a potential of 5V and the other to 0V. The motor starts rotating in one direction. When we interchange the voltages, S2 and S3 are turned on and the direction of rotation changes. 4 transistors are used inside an L298 instead of these 4 switches. The only thing needed now is to know how to give inputs to get the correct outputs.

E.g. The inputs which we used are shown below.

Left motor:

PWM: RC1/CCP1 (17th pin)
Direction pins: RD6 (29th pin)
RD7 (30th pin)

Right motor:

PWM: RC1/CCP2 (16th pin)
Direction pins: RD5 (28th pin)
RD4 (27th pin)

The PWM module of the microcontroller can be used to give the signal to the PWM pin. The code written for that will be described later. But the inputs should be given to the direction pins. For example, if we want to rotate the motor to a particular direction, the PWM signal should be sent after giving logic 0 to the 29th pin and logic 1 to the 30th pin.

Example code for rotating the motor in a particular direction:

We use the PIC C software for this example. The PIC should be configured before using the PWM unit. Read the article on how to create a new project using ‘PIC C’ for that. The motor can be rotated after the appropriate configurations are done.

First of all, the out puts are given to pins D6 and D7. Then the rotation speed is given by the command “set_pwm_duty(100); A value from 0-1023 can be used instead of 100. The angular velocity is increased as the value increases. To change the direction of rotation, the logic values of the two pins should be interchanged. The complete PIC C project folder is included in the example folder.

```
void main()
{
    output_high(PIN_D6); //Set Direction Pin High
    output_low(PIN_D7); //Other Direction Pin Low
    set_pwm1_duty(100); //Set Rotation Speed (Duty Cycle, 0-1023)
    while(1);
}
```

### 4-Sensor Check

Here output_a(0) and output_e(0) will keep respectively PORT A and PORT E at logic 0. Then, using the method we previously discussed, the states of 8 sensors is observed using 8 LEDs.

Can you identify the pins connected to sensors and the pins connected to bulbs after going through the following code?

```
void main()

{

    output_a(0);

    output_e(0);

    if(INPUT(PIN_B0)) output_high(pin_A0);

    if(INPUT(PIN_B1)) output_high(pin_A1);

    if(INPUT(PIN_B2)) output_high(pin_A2);

    if(INPUT(PIN_B3)) output_high(pin_A3);

    if(INPUT(PIN_B4)) output_high(pin_A4);

    if(INPUT(PIN_B5)) output_high(pin_A5);

    if(INPUT(PIN_B6)) output_high(pin_E0);

    if(INPUT(PIN_B7)) output_high(pin_E1);

}
```

Now what you have to do is to write an algorithm to decide the directions of rotation of the motors according to the signals from the sensor panel and to supply the Pulse Width Modulation signal.

Next steps would be developing a more efficient algorithm and tuning the robot to the environmental conditions and make it move faster.

.