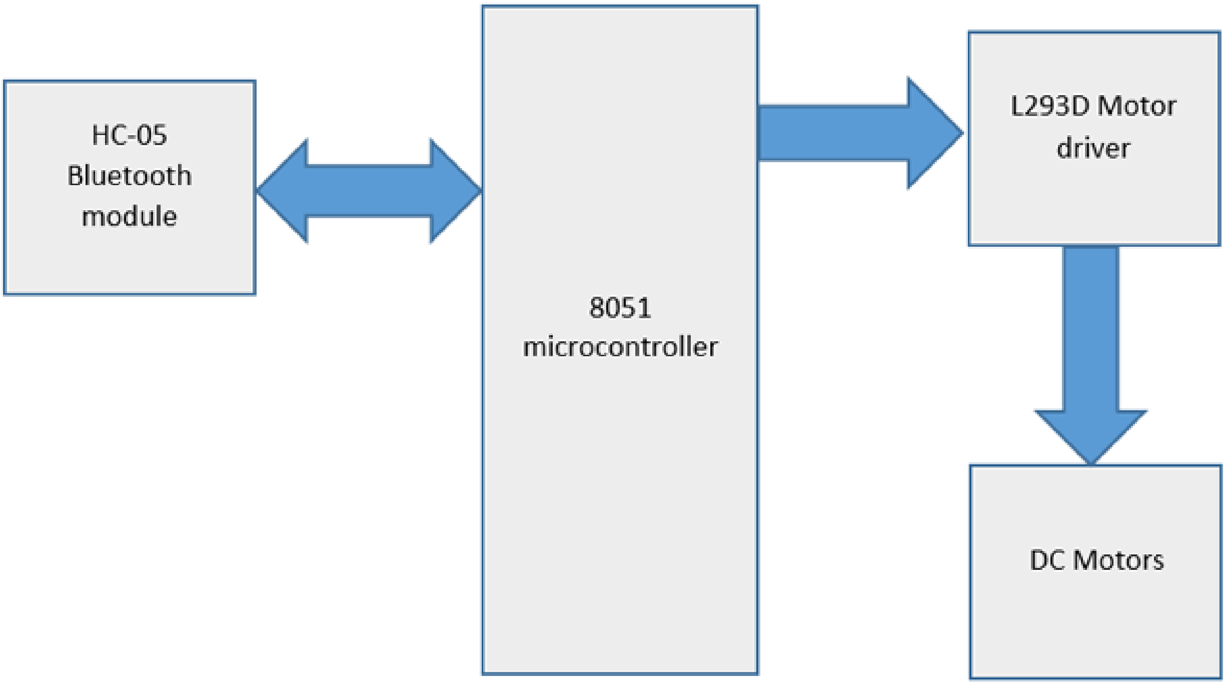
Project Name: Android Controlled Metal Detector Robot

Introduction

The main aim of this Metal Detection by Robot project Report and documentation is to detecting the obstacle (metal), and weapons (gun, knives) this can be done by using Robot. In this project ANTI TANK MINE DETECTOR is automated in the robot. The anti-tank mine detector is used for finding mine when robot moving on the earth. It is the robotic vehicle it can understand the mine and obstacle. Robot is controlled using Android Phone controlled robot using 8051 microcontrollers and Bluetooth module. The robot is designed using DC motors and the direction of DC motors will be controlled by the commands received from the android application. The status of the robot is sent back to the Android app. This project will also help for interfacing of HC-05 Bluetooth module with 8051 microcontrollers. The metal detector consist of oscillator produces the alternating current passes throw the coil and producing alternating magnetic field. For the signal is in condition the signal from the comparator must be match with the microcontroller.



Component Descriptions

- HC-05 Bluetooth Module

  HC-05 is a serial Bluetooth module. It can be configured using AT commands. It can work in three different configurations (Master, Slave, Loop back). In our project we will be using it as a slave. The features of HC-05 module includes,

  - ✦ Typical -80dBm sensitivity.

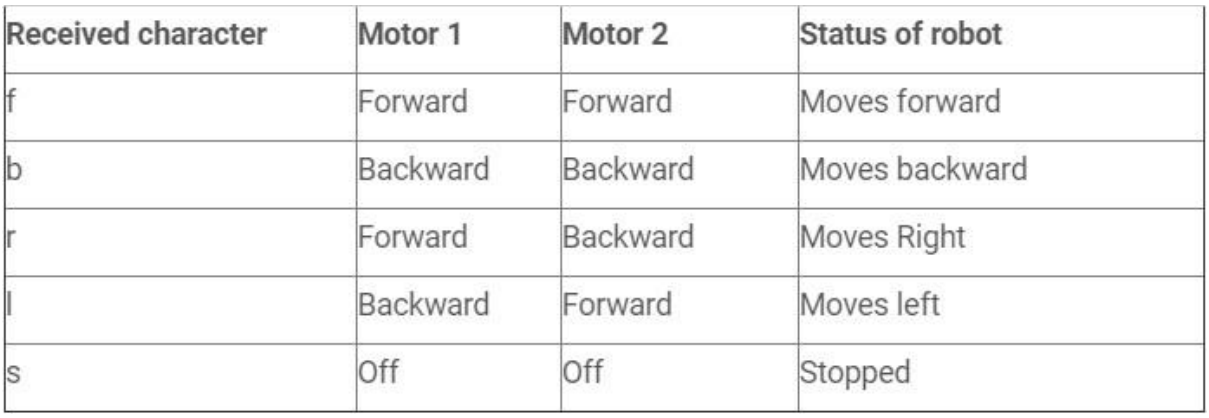
- Default baud rate: 9600bps , 8 data bits , 1 stop bit , no parity.
- Auto-pairing pin code: “1234” default pin code
- It has 6 pins.
- Vcc and Gnd pins are used for powering the HC-05.
- Tx and Rx pins are used for communicating with the microcontroller.
- Enable pin for activating the HC-05 module. when it is low , the module is disabled
- State pin acts status indicator. When it is not paired/connected with any other Bluetooth device, LED flashes continuously. When it is connected/paired with any other Bluetooth device, then the LED flashes with the constant delay of 2 seconds.

- L293D Motor Driver IC

- L293D is a dual H-bridge motor driver IC. This acts as a current amplifier, the output of L293D drives the DC Motors. It contains two inbuilt H-bridge circuits. In common mode of operation , it can drive two dc motors simultaneously in both the directions.

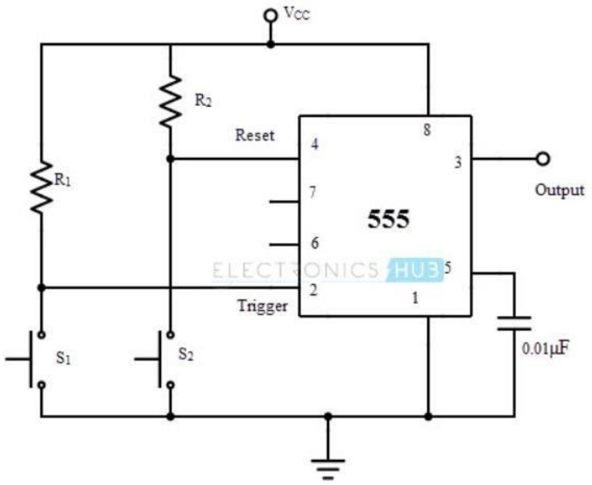
- Working of Android Phone Controlled Robot

- The user of android app sends the data to 8051 microcontroller through HC-05 module. The received data is compared in 8051 microcontroller and the decision is made accordingly. The below table shows the direction of motors and status of robot for different received characters.



Received character	Motor 1	Motor 2	Status of robot
f	Forward	Forward	Moves forward
b	Backward	Backward	Moves backward
r	Forward	Backward	Moves Right
l	Backward	Forward	Moves left
s	Off	Off	Stopped

Working of 555 timer in bi-stable mode for the Metal Detector



The trigger and reset pins (pins 2 and 4 respectively) are connected to the supply through two resistors R1 and R2 so that they are always high. In all the previous cases, the reset pin is not used and in order to avoid any accidental reset, it is simply connected to VCC.

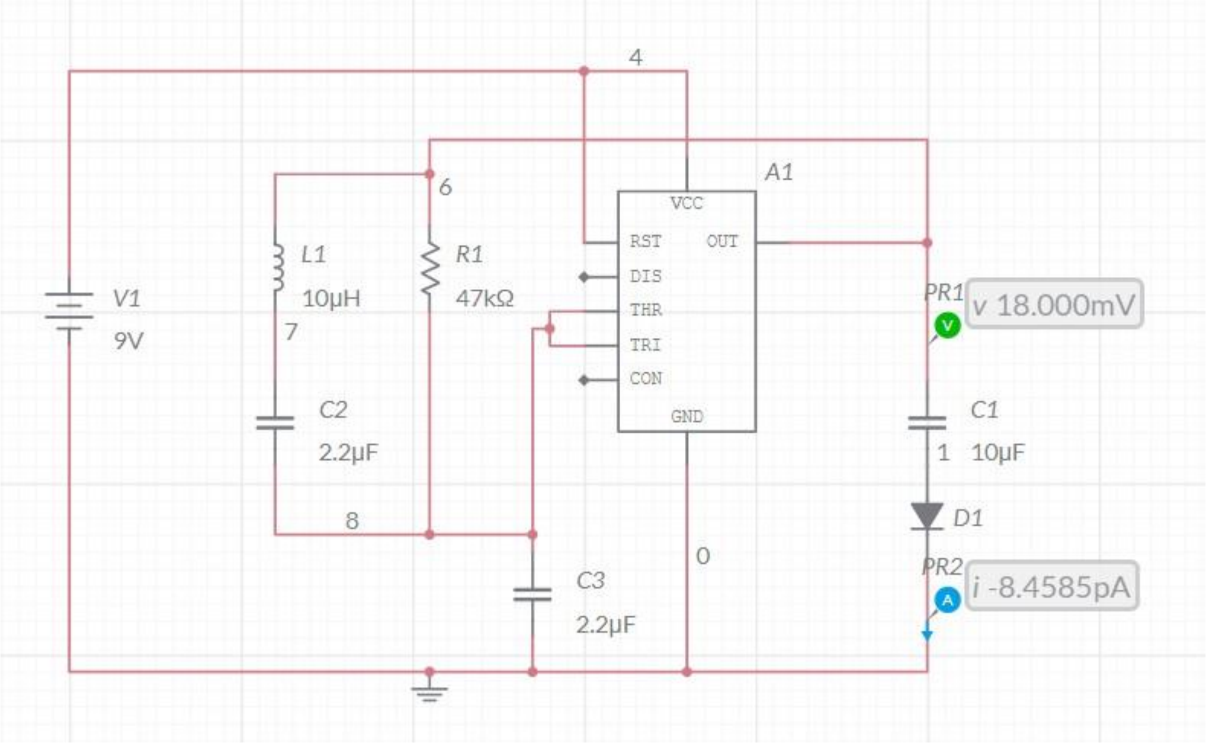
Two switches are connected between these pins and ground in order to make them go low momentarily. The switch at the trigger input will act as S (SET) input for the internal flip-flop. The switch at the reset input will act as reset for the internal flip-flop.

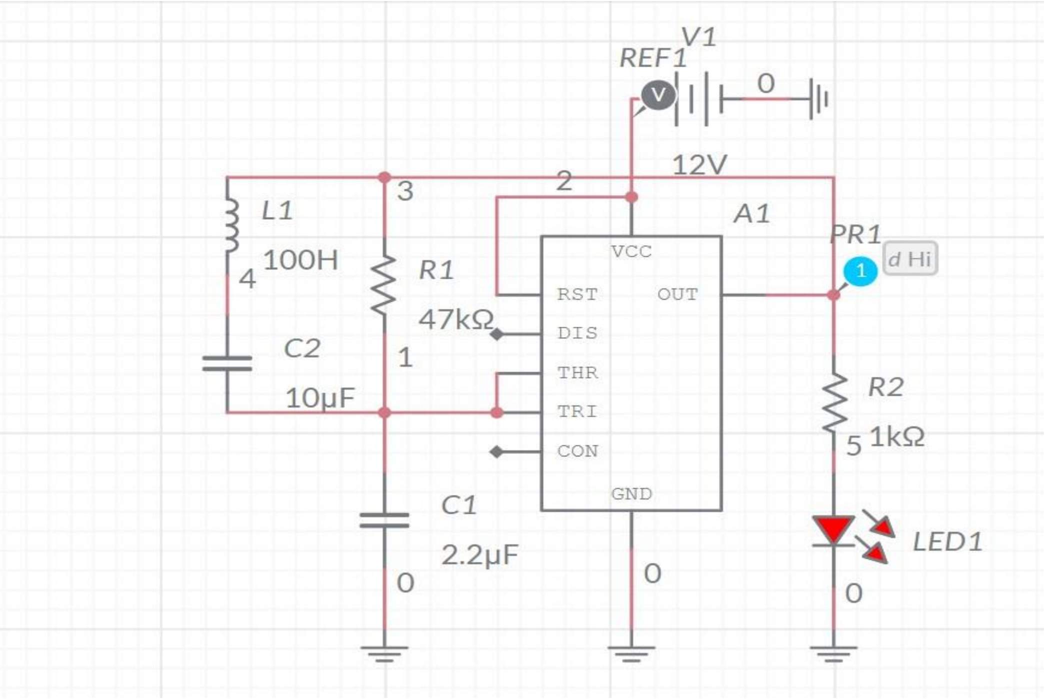
When the switch S1 is pressed, the voltage from VCC will bypass the trigger terminal and is shorted to ground through the resistor R1. Hence, the trigger pulse will momentarily go low and the output of the timer at pin 3 will become HIGH. The output stays HIGH because there is no input from the threshold pin (pin 6 is left open or better if connected to ground) and the output of the internal comparator (comparator 1) will not go high.

When the switch S2 is pressed, the voltage from VCC will bypass the reset terminal and is shorted to ground through the resistor R2. This pin is internally connected to the RESET terminal of the flip-flop. When this signal goes low for a moment, the flip-flop receives the reset signal and RESETs the flipflop.

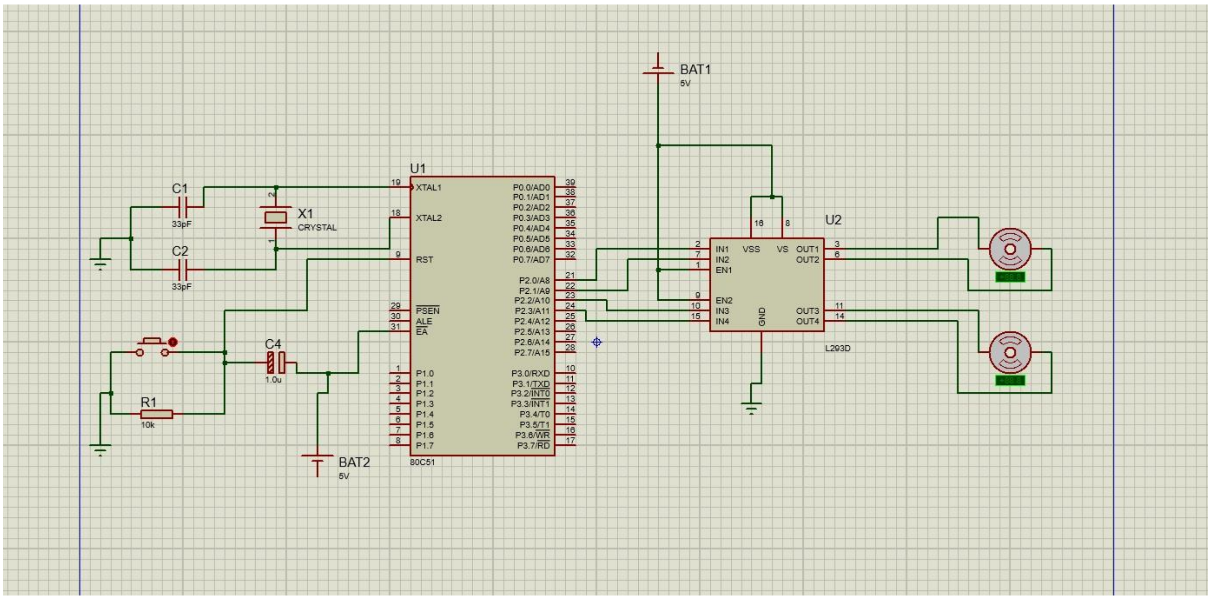
Hence, the output will become LOW and stays there until the trigger is applied.

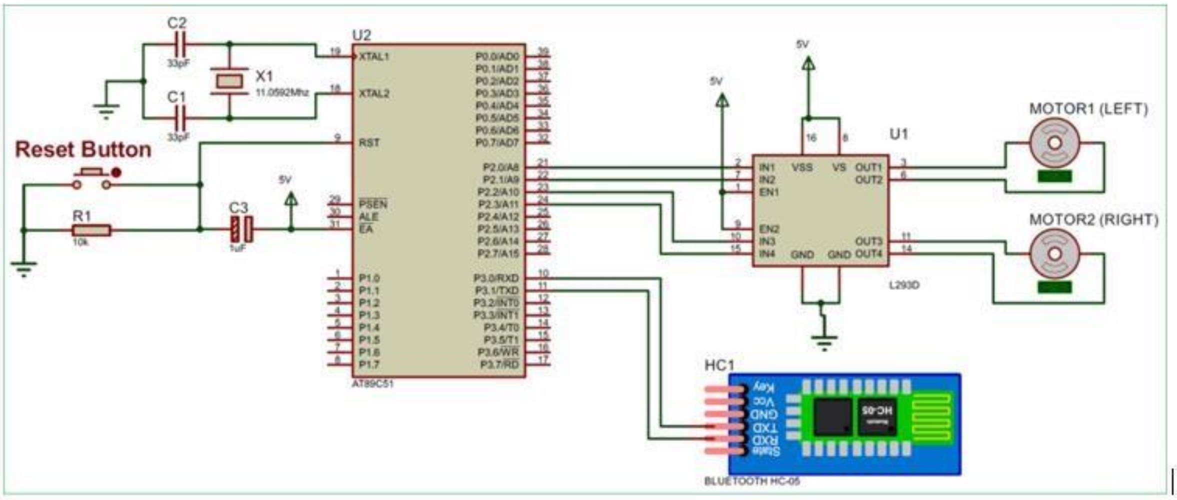
Our implementation of the metal detector circuit





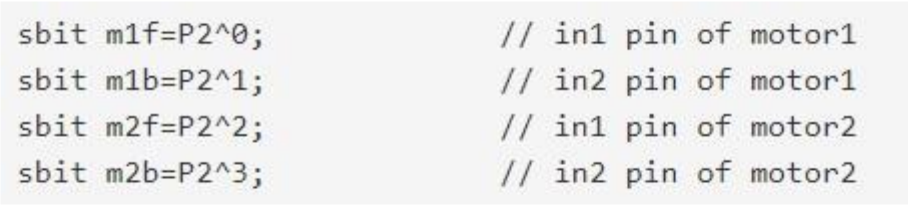
Robot Circuit





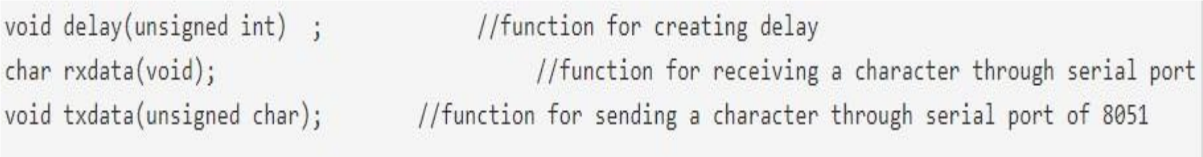
CODE ALGORITHM

For L293D interfacing with 8051 microcontrollers, we have to define pins on which L293D is connected to 8051 microcontroller. In1 pin of motor 1 is connected to P2.0, In2 pin of motor 1 is connected to P2.1, In1 pin of motor 2 is connected to P2.2, In2 pin of motor 2 is connected to P2.3



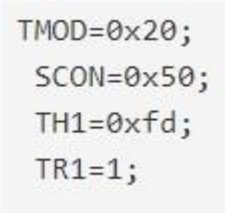
```
sbit m1f=P2^0;              // in1 pin of motor1
sbit m1b=P2^1;              // in2 pin of motor1
sbit m2f=P2^2;              // in1 pin of motor2
sbit m2b=P2^3;              // in2 pin of motor2
```

Next we have to define some functions which are used in the program. Delay function is used to create specified time delay. Txdata function is used to transmit data through serial port . Rxdata function is used to receive data from serial port.



```
void delay(unsigned int)  ;             //function for creating delay
char rxdata(void);                          //function for receiving a character through serial port
void txdata(unsigned char);       //function for sending a character through serial port of 8051
```

In this part of the code we are going to configure 8051 microcontroller for serial communication. TMOD register is loaded with 0x20 for timer 1, mode 2 (auto reload). SCON register is loaded with 0x50 for 8 data bits, 1 stop bit and receive enabled. TH1 register is loaded with 0xfd for baud rate of 9600 bits per second. TR1=1 is used to start the timer.



```
TMOD=0x20;
 SCON=0x50;
 TH1=0xfd;
 TR1=1;
```

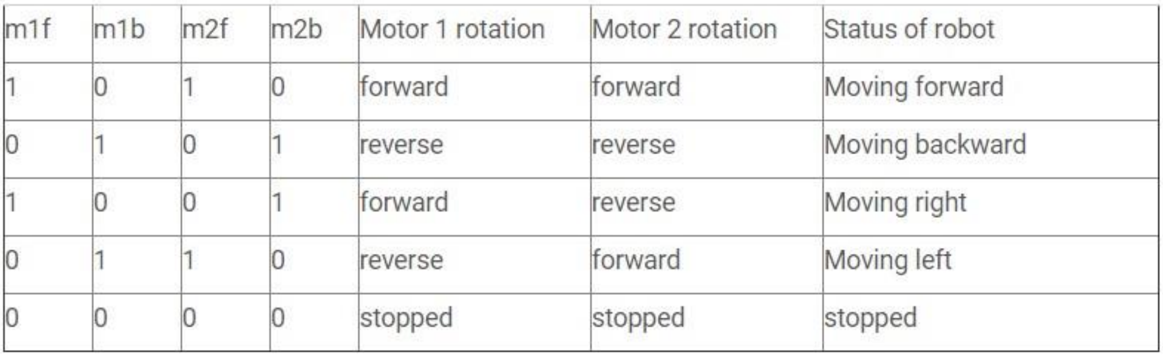
HOW WE ARE RECEIVING DATA

First we poll the RX pin of port 3 till it becomes 1.

This means data is now present in SBUF

Now we just return this data into the variable ‘s’

MAKING THE ROBOT MOVE ON THE BASIS OF RECEIVED DATA

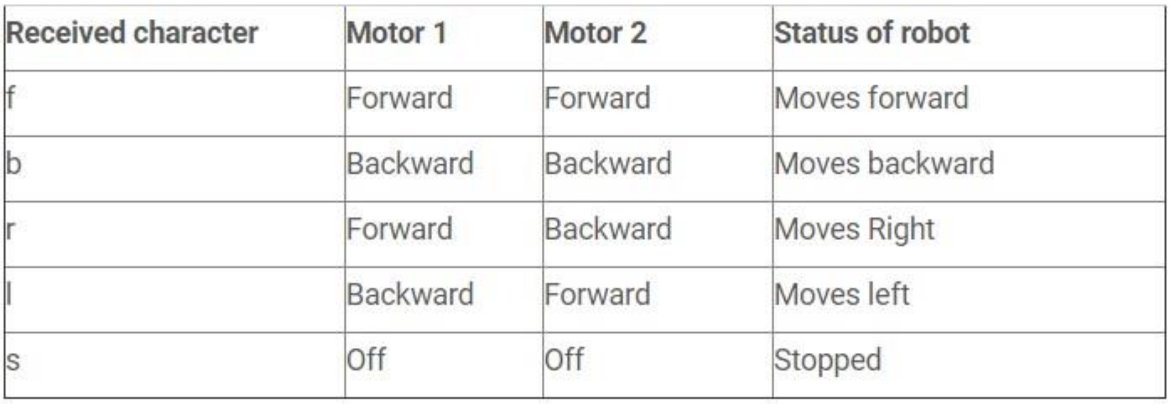


m1f	m1b	m2f	m2b	Motor 1 rotation	Motor 2 rotation	Status of robot
1	0	1	0	forward	forward	Moving forward
0	1	0	1	reverse	reverse	Moving backward
1	0	0	1	forward	reverse	Moving right
0	1	1	0	reverse	forward	Moving left
0	0	0	0	stopped	stopped	stopped

We basically provide a delay after setting or clearing each bit on the basis of the table above.

Example: For moving forward, the IN1 and IN3 pins of the motor driver must be HIGH and the IN2 and IN4 pins must be LOW.

MAKING DECESIONS ON THE BASIS OF ‘S’: THE IF ELSE CHAIN



Received character	Motor 1	Motor 2	Status of robot
f	Forward	Forward	Moves forward
b	Backward	Backward	Moves backward
r	Forward	Backward	Moves Right
l	Backward	Forward	Moves left
s	Off	Off	Stopped

RESULT:

1) Successfully established connection between Bluetooth module and mobile.
2) Proper assembly of chassis and interfacing with the motor driver.
3) Motor successfully working with L239D motor driver.
4) Change in frequency (sound produced) of metal detection when in contact with metal.
5) Working simulation on Proteus ide so code quality is correct.

CODE

```
#include<reg51.h>


unsigned char ch1; unsigned

char s;


sbit m1f=P2^0;            // in1 pin of motor1 sbit

m1b=P2^1;           // in2 pin of motor1 sbit

m2f=P2^2;           // in1 pin of motor2 sbit

m2b=P2^3;           // in2 pin of motor2


void delay(unsigned int)  ;       //function for creating delay char rxdata(void);

//function for receiving a character through serial port of 8051  void txdata(unsigned

char); //function for sending a character through serial port of 8051


void main(void)

 {

unsigned char i; unsigned char msg1[]={"robot is

moving forward"}; unsigned char msg2[]={"robot is

moving backward"}; unsigned char msg3[]={"robot is

moving right"}; unsigned char msg4[]={"robot is

moving left"}; unsigned char msg5[]={"robot is

stopped"};
```

```
TMOD=0x20;   //timer 1 , mode 2 , auto reload
SCON=0x50;    //8bit data , 1 stop bit , REN enabled
TH1=0xfd;     //timer value for 9600 bits per second(bps)
TR1=1;


while(1)             //repeat forever
{
    s=rxdata(); //receive serial data from hc-05 bluetooth module if(s=='f')
//move both the motors in forward direction
    {
        m1f=1;    delay(1);
m1b=0;    delay(1);
m2f=1;    delay(1);
m2b=0;    delay(1);
for(i=0;msg1[i]!='\0';i++)
{
   txdata(msg1[i]);
}


   }


    else if(s=='b')
    {
        m1f=0;     delay(1);
m1b=1;     delay(10);
m2f=0;     delay(10);
m2b=1;     delay(10);
for(i=0;msg2[i]!='\0';i++)
       {
  txdata(msg2[i]);
   }
```

```
     }

     else if(s=='r')
     {
         m1f=1;
delay(1);         m1b=0;
delay(10);         m2f=0;
delay(10);         m2b=1;
delay(10);
for(i=0;msg3[i]!='\0';i++)
{
    txdata(msg3[i]);
}
                }

     else if(s=='l')
     {
         m1f=0;         delay(1);
m1b=1;     delay(1);
m2f=1;     delay(1);
m2b=0;     delay(1);
for(i=0;msg4[i]!='\0';i++)
       {
    txdata(msg4[i]);
   }
    }

     else if(s=='s')
     {
         m1f=0;
delay(1);         m1b=0;
delay(1);         m2f=0;

delay(1);         m2b=0;

delay(1);

for(i=0;msg5[i]!='\0';i++)

       {

  txdata(msg5[i]);

   }

}

      txdata('\n');


                    }

}


char rxdata()

{

  while(RI==0);   //wait till RI becomes HIGH   RI=0;

//make RI low   ch1=SBUF;      //copy received data    return

ch1;     //return the received data to main function.

}


void txdata(unsigned char x)

{

   SBUF=x; //copy data to be transmitted to SBUF

while(TI==0); //wait till TI becomes high

   TI=0; //mae TI low for next transmission

}


void delay(unsigned int z)

{

  unsigned int p ,q;   for(p=0 ; p<z ; p++)

//repeat for 'z' times
```

```
 {
   for(q=0 ; q<1375 ; q++);   //repeat for 1375 times
 }
}
```