A PROJECT REPORT

ON

BLUETOOTH CONTROLLING CAR USING ATMEGA16

Submitted in the partial fulfillment for the award of the

DIPLOMA

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:-

MADAN LAL(1604081029) | MANDEEP SINGH(1604081030)|
PRADEEP RAWAT(1604081036) | RAJIV(1604081045)|
SANDEEP KUMAR(1604081052) | SAWAN KUMAR
(1604081053)

Under the guidance of:-

PROF. KARAMBIR SIR

DEPARTMENT OF ELECTRONICS AND COMMNICATION ENGINNERING

GURU NANAK DEV INSTITUTE OF TECHNOLOGY

GURU NANAK DEV INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the work embodied in the project entitled "BLUETOOTH CONTROLLING CAR" submitted by MADAN LAL(1604081029) | MANDEEP SINGH(1604081030)| PRADEEP RAWAT(1604081036) | RAJIV(1604081045)| SANDEEP KUMAR(1604081052) | SAWAN KUMAR (1604081053) **is the record of their own work and is submitted in the partial fulfillment of their requirement for the ELECTRONICS AND COMMUNICATION ENGINEERING.**

Prof. Karambir sir (Project Guide)

Prof. Mukesh Saxena sir (HOD)

ABSTRACT

"BLUETOOTH CONTROLLING CAR" is a car which is operated through the bluetooth.the operation of car like moving FORWARD,REVERSE,LEFT and RIGHT would be control though the mobile or any device but it should have connectivity of bluetooth.All the operation is done in the BLUETOOTH CONTROLLING CAR by the process in the ATMEGA16 and L293D(MOTOR DRIVVER) and all the MOTOR connected to L293D.

ACKNOWLEDGEMENT

Firstly, we take this opportunity to thank our esteemed institute “GURU NANAK DEV INSTITUTE OF TECHNOLOGY” for making us capable of peparing Project named “*BLUETOOTH CONTROLLING CAR*” . Then we would also like to thank the head of department PROF. MUKESH SAXENA who gave this opportunity to prepare this project. We would also like to thank our teachers whose constant encouragement made it possible for us to take up the challenge of writing the synopsis of the project. Specially thanks to our project guide PROF.KARAMBIR SIR who contribute shaping us in academics as well as in professional career. The design ,implementation and completion of this study would have been impossible without their help an contribution .

We would also like to thank each one of the faculty members for their valuable help and support in encouraging our work on the project.

Finally, we ae deeply indebted to god and our parents,who encouraged us and were there for us in all our good and bad times.

Project Team :

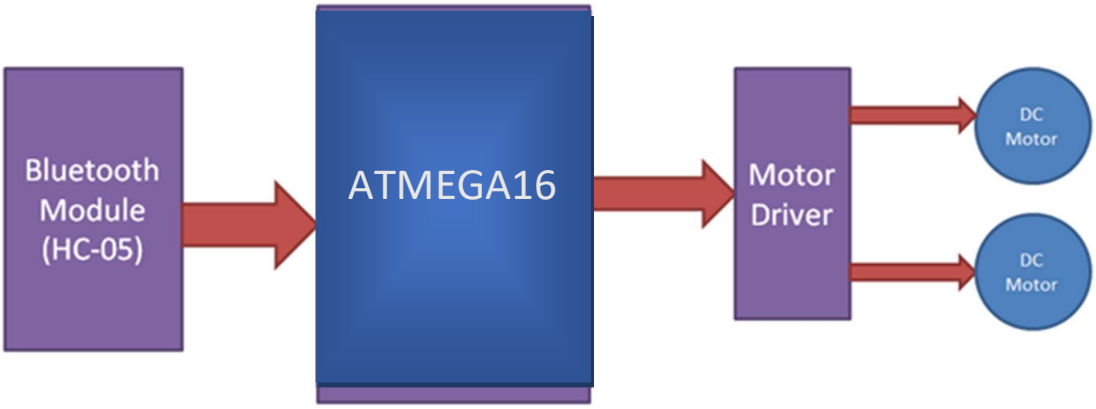
MADAN LAL,MANDEEP SINGH,PRADEEP RAWAT,RAJIV,SANDEEP KUMAR,SAWAN KUMAR

TABLE OF CONTENTS

TOPIC PAGE NUMBER

PREFACE

BLOCK DIAGRAM OF PROJECT



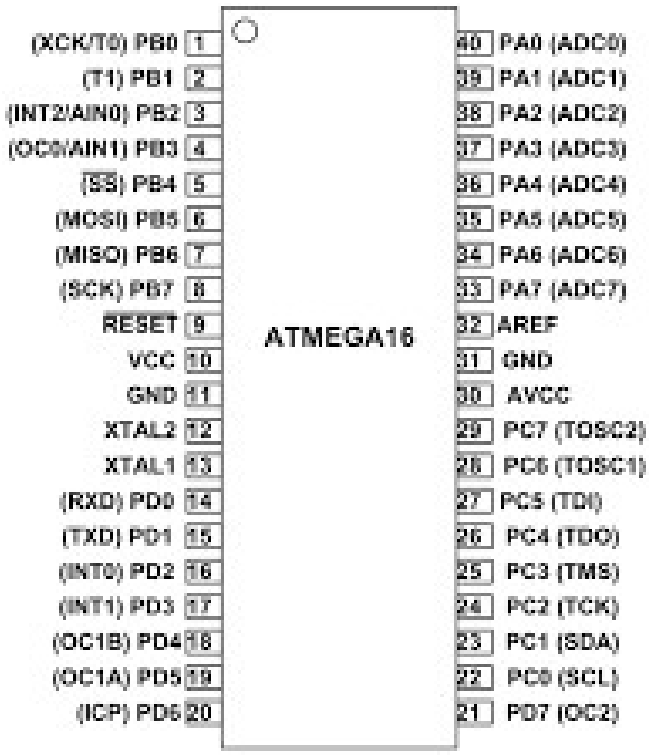
DESCRIPTION

In our project we can control the d.c motors by using the motor driver(L293D) i.e. from ATMEGA16 where the operation takes places for operating the d.c motors, then the d.c motors receives (acts as receiver) the signals, according to the signals being received the direction of the motors is controlled and by using L293D we can give the command

COMPONENTS USED

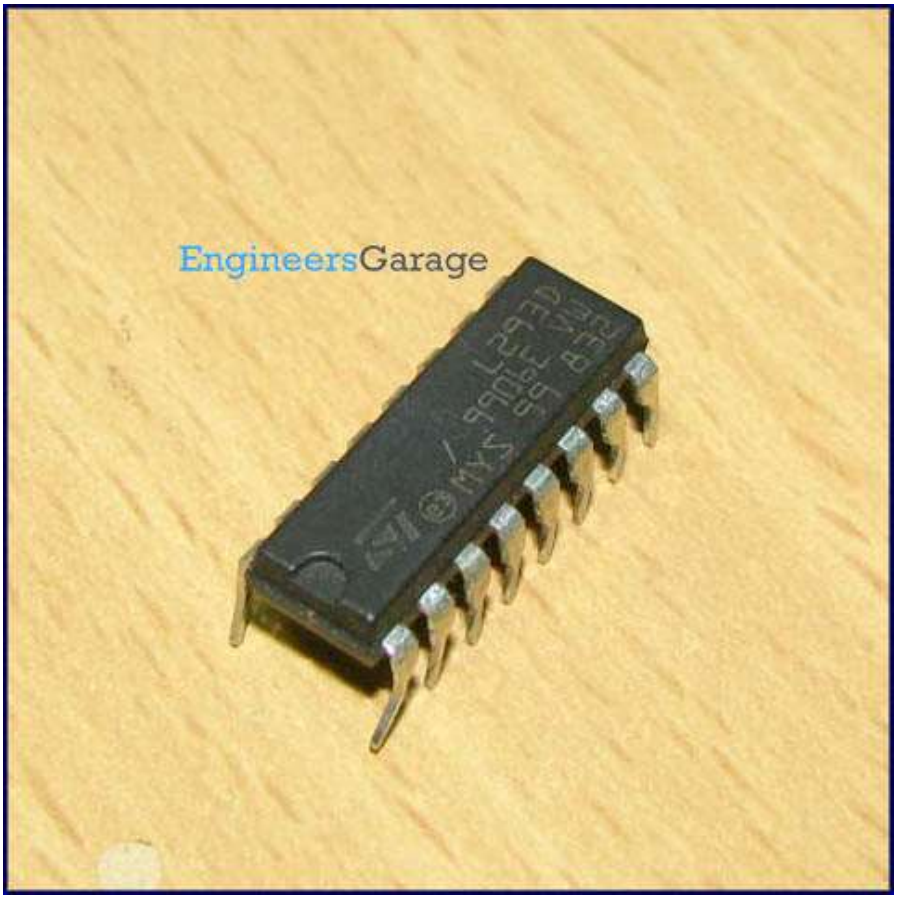
MICROCONTROLLER ATMEGA16

- Microcontroller contains microprocessor, memory (RAM & ROM), I/O interfacing circuit and peripheral devices such as A/D converters, serial I/O, timers etc.
- It has many bit handling instructions.
- Less access times for built-in memory and I/O devices.
- Microcontroller based systems requires less hardware reducing PCB size and increasing reliability.



PIN DIAGRAM OF ATMEGA16 MICROCONTROLLER

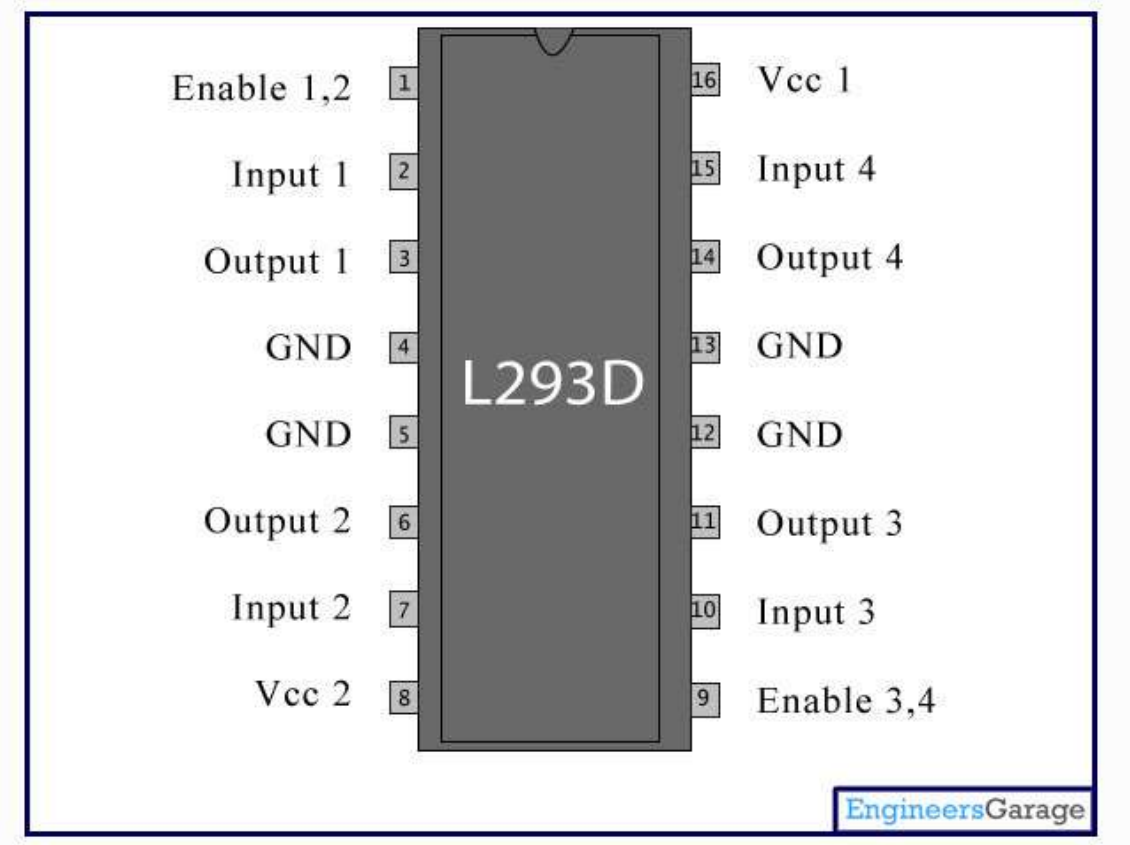
MOTOR DRIVER(L293D)



L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors.

L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.



Pin Description:

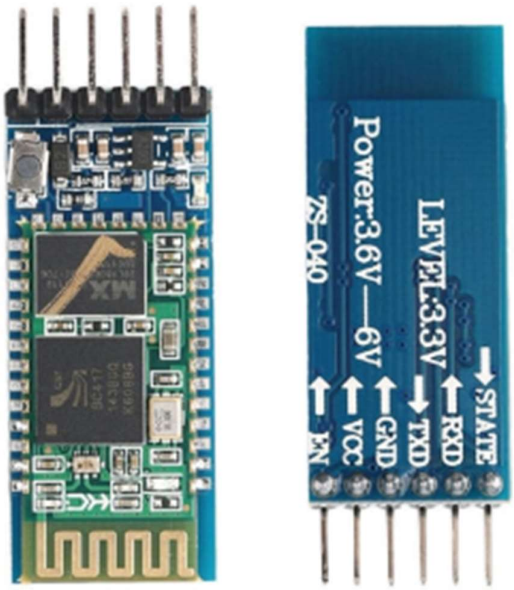
Pin No	Function	Name
1	Enable pin for Motor 1; active high	Enable 1,2
2	Input 1 for Motor 1	Input 1
3	Output 1 for Motor 1	Output 1
4	Ground (0V)	Ground
5	Ground (0V)	Ground
6	Output 2 for Motor 1	Output 2
7	Input 2 for Motor 1	Input 2
8	Supply voltage for Motors; 9-12V (up to 36V)	Vcc $_2$
9	Enable pin for Motor 2; active high	Enable 3,4
10	Input 1 for Motor 1	Input 3
11	Output 1 for Motor 1	Output 3
12	Ground (0V)	Ground
13	Ground (0V)	Ground
14	Output 2 for Motor 1	Output 4
15	Input2 for Motor 1	Input 4
16	Supply voltage; 5V (up to 36V)	Vcc $_1$

DC Motors

DC motors can be applied to movement and locomotion Specifications of most DC motors show high revolutions per minute (rpm) and low torque. Robotics need low rpm and high torque. Gearboxes can be attached to motors to increase their torque while reducing the rpm. The gearbox usually specifies a ratio that describes the rpm in to the rpm out. For instance, a DC motor with an rpm of 8000 is connected to a 1000:1 gearbox. What is the output rpm? 8000 rpm/1000 _ 8 rpm. The torque of the motor is substantially increased. You could estimate that the torque will increase by the same value the rpm decreased.

In reality, no conversion is 100 percent efficient; there always will be efficiency losses. Some DC motors, called *gear head motors,* are built with a gearbox.

BLUETOOTH MODULE(HC-05)



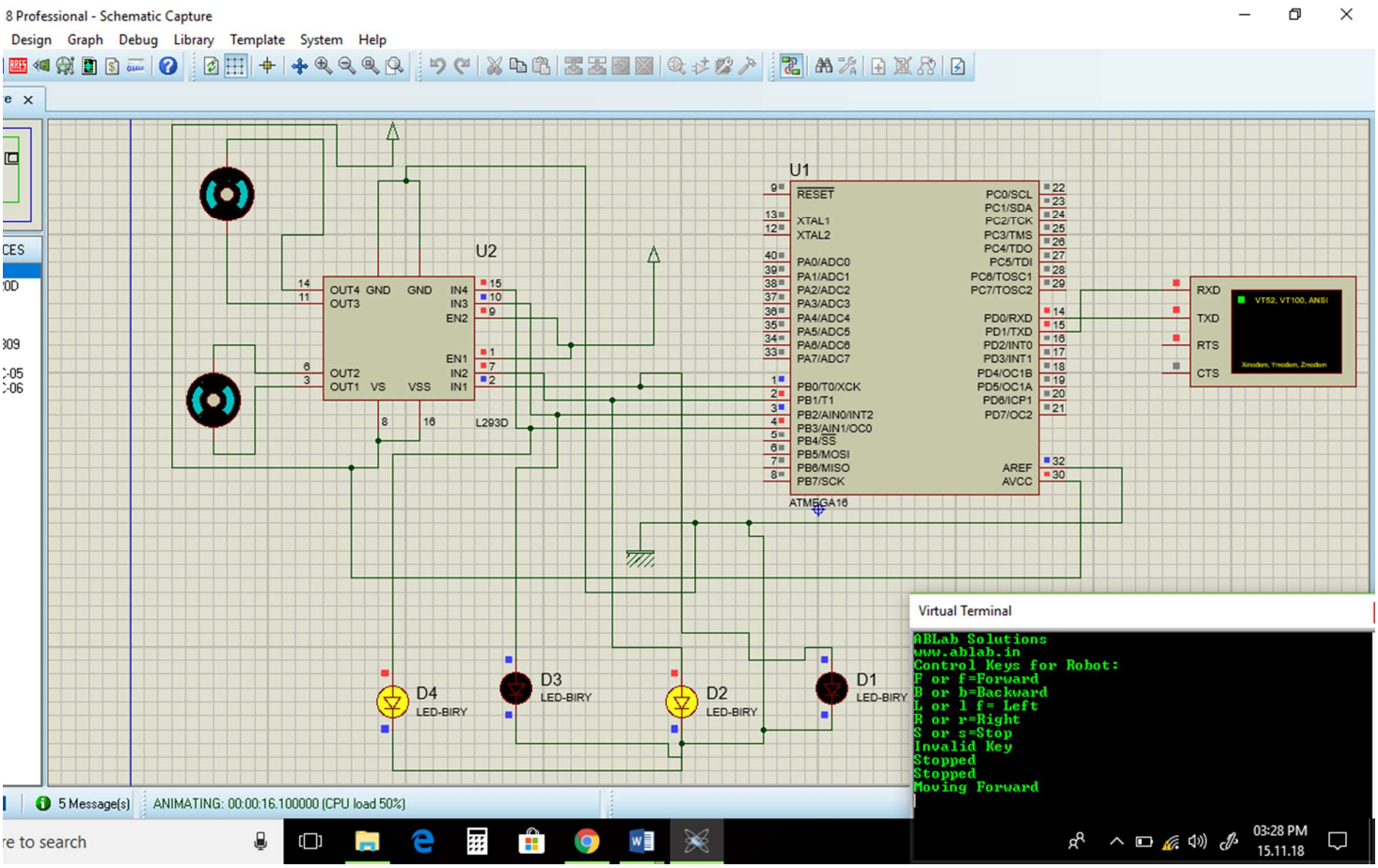
HC-05 Bluetooth Module

HC-05 module is an easy to use **Bluetooth SPP (Serial Port Protocol) module**,designed for transparent wireless serial connection setup.The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication.This serial port bluetooth module is fully qualified **Bluetooth V2.0+EDR (Enhanced Data Rate)**3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses **CSR Bluecore 04**-External single chip Rluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

The Bluetooth module HC-05 is a MASTER/SLAVE module.By default the factory setting is SLAVE.The Role of the module (Master or Slave) can be configured only by AT COMMANDS.The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections.Master module can initiate a

connection to other devices.The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc

CIRCUIT DIAGRAM



CIRCUIT DIAGRAM OF BLUETOOTH CONTROLLING

PROGRAM CODE

```
#include<stdio.h>
 #include<mega16.h>
/*Includes io.h header file where all the Input/Output Registers and
its Bits are defined for all AVR microcontrollers*/

#define    F_CPU    8000000
/*Defines a macro for the delay.h header file. F_CPU is the
microcontroller frequency value for the delay.h header file. Default
value of F_CPU in delay.h header file is 1000000(1MHz)*/

#include<delay.h>
/*Includes delay.h header file which defines two functions,
_delay_ms (millisecond delay) and _delay_us (microsecond delay)*/

/*USART Function Declarations*/
void usart_init();
void usart_data_transmit(unsigned char data );
unsigned char usart_data_receive( void );
void usart_string_transmit(char *string);
```

```
 char receiveddata;
/*HC-05 Bluetooth Function Declarations*/
void hc_05_bluetooth_transmit_byte(char data_byte);
char hc_05_bluetooth_receive_byte(void);
void hc_05_bluetooth_transmit_string(char *transmit_string);


int main()
{
   DDRB=0x0f;


   usart_init();


   hc_05_bluetooth_transmit_string("ABLab Solutions");


   hc_05_bluetooth_transmit_byte(0x0d);
   hc_05_bluetooth_transmit_byte(0x0a);
```

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("www.ablab.in");

*/*Transmits a string to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

*/*Transmits Carriage return to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0a);

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("Control Keys for Robot:");

*/*Transmits a string to Bluetooth of Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

*/*Transmits Carriage return to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0a);

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("F or f=Forward");

*/*Transmits a string to Bluetooth of Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

*/*Transmits Carriage return to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0a);

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("B or b=Backward");

*/*Transmits a string to Bluetooth of Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

*/*Transmits Carriage return to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0a);

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("L or l f= Left");

*/*Transmits a string to Bluetooth of Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

*/*Transmits Carriage return to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0a);

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("R or r=Right");

*/*Transmits a string to Bluetooth of Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

*/*Transmits Carriage return to Bluetooth Module*/*

hc_05_bluetooth_transmit_byte(0x0a);

*/*Transmits New Line to Bluetooth Module for new line*/*

hc_05_bluetooth_transmit_string("S or s=Stop");

*/*Transmits a string to Bluetooth of Module*/*

hc_05_bluetooth_transmit_byte(0x0d);

```
    /*Transmits Carriage return to Bluetooth Module*/

    hc_05_bluetooth_transmit_byte(0x0a);
    /*Transmits New Line to Bluetooth Module for new line*/

    /*Start of infinite loop*/
    while(1)
    {
        receiveddata=hc_05_bluetooth_receive_byte();
        /*Microcontroller will receive a character from Bluetooth
Module*/

        /*Checking the pressed key value to move the robot in different
direction*/
        if(receiveddata == 'F' || receiveddata == 'f')
        {
            PORTB=0x0A;
            /*Robot will move forward direction*/

            hc_05_bluetooth_transmit_string("Moving Forward");
```

```
    /*Transmits a string to Bluetooth of Module*/
}
else if(receiveddata == 'B' || receiveddata == 'b')
{
    PORTB=0x05;
    /*Robot will move backward direction*/

    hc_05_bluetooth_transmit_string("Moving Backward");
    /*Transmits a string to Bluetooth of Module*/
}
else if(receiveddata == 'L' || receiveddata == 'l')
{
    PORTB=0x02;
    /*Robot will move towards left direction*/

    hc_05_bluetooth_transmit_string("Moving Left");
    /*Transmits a string to Bluetooth of Module*/
}
else if(receiveddata == 'R' || receiveddata == 'r')
{
```

```
    PORTB=0x08;
    /*Robot will move towards right direction*/

    hc_05_bluetooth_transmit_string("Moving Right");
    /*Transmits a string to Bluetooth of Module*/
}
else if(receiveddata == 'S' || receiveddata == 's')
{
    PORTB=0x0f;
    /*Robot will stop*/

    hc_05_bluetooth_transmit_string("Stopped");
    /*Transmits a string to Bluetooth of Module*/
}
else
{
    hc_05_bluetooth_transmit_string("Invalid Key");
    /*Transmits a string to Bluetooth of Module*/
}
```

```
        hc_05_bluetooth_transmit_byte(0x0d);
        /*Transmits Carriage return to Bluetooth Module*/

        hc_05_bluetooth_transmit_byte(0x0a);
        /*Transmits New Line to Bluetooth Module for new line*/
    }
}
/*End of program*/



/*USART Function Definitions*/
void usart_init()
{
    UBRRH = 0;
    UBRRL =51;
    UCSRB|= (1<<RXEN)|(1<<TXEN);
    UCSRC |= (1 << URSEL)|(3<<UCSZ0);
}
```

```
void usart_data_transmit(unsigned char data )
{
   while ( !( UCSRA & (1<<UDRE)) )
   ;
   UDR = data;
   delay_ms(1);
}

unsigned char usart_data_receive( void )
{
   while ( !(UCSRA & (1<<RXC)) )
   ;
   return UDR;
}

void usart_string_transmit(char *string)
{
   while(*string)
   {
      usart_data_transmit(*string++);
```

```
    }
}


/*HC-05 Bluetooth Function Definitions*/
void hc_05_bluetooth_transmit_byte(char data_byte)
{
    usart_data_transmit(data_byte);
}

char hc_05_bluetooth_receive_byte(void)
{
    return usart_data_receive();
}

void hc_05_bluetooth_transmit_string(char *transmit_string)
{
    usart_string_transmit(transmit_string);
}
```