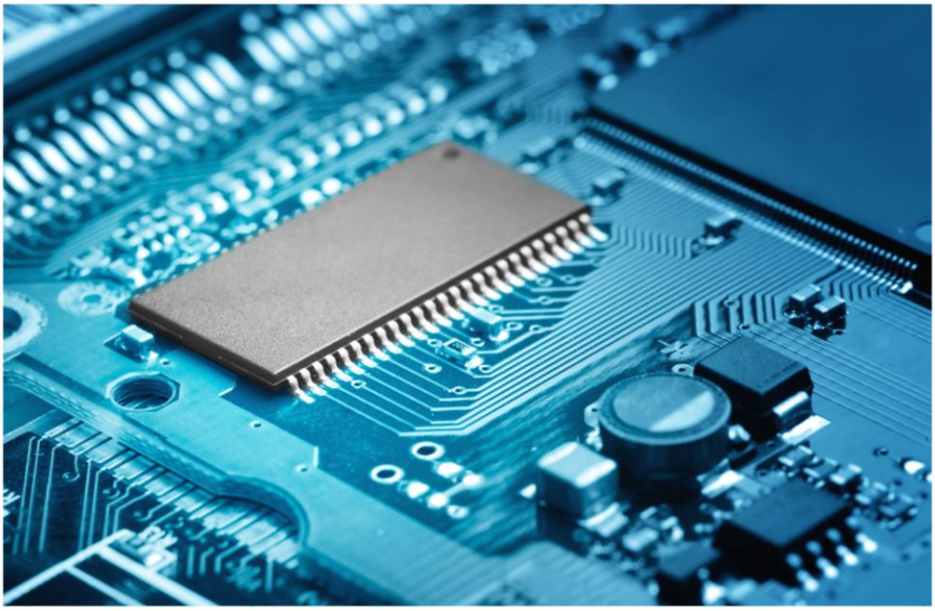


Project Report

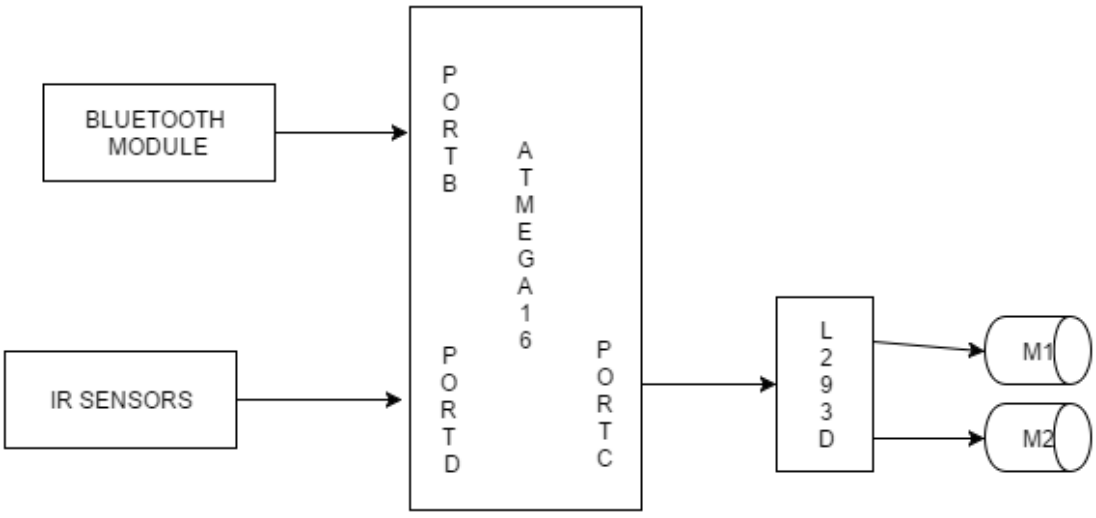
SMART ROBOTIC CAR

Ritik Channa | Embedded Systems |Batch: 8:30-10:30

Objective

To design a smart car that follows line drawn path and switches to manual controls on detecting obstacle or explicitly by the operator

Block Diagram



Working

The project deploys atmega16 as microcontroller, initially the car is set to line follow mode in which car moves around on drawn black line path with the help of two IR sensors set at a distance slight more than the width of line. Car can be switched to manual Bluetooth controlled automatically or manually.

When an obstacle is detected by the third IR sensor in line follow mode the car stops automatically and waits for a while , if the obstacle

is removed it continues on the line marked path else switches to Bluetooth controlled mode automatically

When the operator sends any key character to the car using Bluetooth serial communication, the car switches to the Bluetooth controlled mode. Car now can be controlled by the user using any Bluetooth enabled device

Port C of the microcontroller is connected to the motor driver IC (l293d) which is then further connected to the two motors responsible for the movement of the car

Coding

MAIN.C

```
#include"motor.h"

char ch='l';

void uart_init()

{

   UBRRH=0;

   UBRRL=51;

   UCSRC=0B10000110;

   UCSRB=0B00011000;

   UCSRA=0B00000000;

}

char char_recieve()

{
```

```
    if(UCSRA&128)
    return UDR;
    else
    return 'l';
}
int main()
{
    motor_init();
    uart_init();
    DDRA=0X00;
    PORTA=0XFF;
    DDRB=0X00;
    PORTB=0XFF;
    while(1)
    {
        //line follow mode
        if(ch=='l')
        {
            if((!(PINA&(7<<0)))==0X03)  forward();
            else if((!(PINA&(7<<0)))==0X02)  left();
            else if((!(PINA&(7<<0)))==0X01)  right();
            else
            {
                stop();
```

```
      for(int i=0;i<10000;i++)
      {
         if((!(PINA&(7<<0)))>0X03)
                                    goto z;
      }
      ch='s';
      goto y;
   }
   z:
   ch=char_recieve();
}
else
{
   //bluetooth manual mode
   y:
   while(1)//once manual, stay manual
   {
      ch=char_recieve();
      if(ch=='w') forward();
      else if(ch=='a') left();
      else if(ch=='d') right();
      else if(ch=='s') backward();
      else stop();
   }
```

```
      }
   }
}
```

MOTOR.H

```
#include<avr/io.h>
 void forward()
{
   PORTC&=~(1<<0);
   PORTC|=(1<<1);
   PORTC&=~(1<<2);
   PORTC|=(1<<3);
}
void left()
{
   PORTC&=~(1<<1);
   PORTC|=(1<<0);
   PORTC&=~(1<<2);
   PORTC|=(1<<3);
}
void right()
{
   PORTC&=~(1<<0);
   PORTC|=(1<<1);
   PORTC&=~(1<<3);
```

```
   PORTC|=(1<<2);
}
void backward()
{
        PORTC|=(1<<0);
        PORTC&=~(1<<1);
        PORTC|=(1<<2);
        PORTC&=~(1<<3);
}
void stop()
{
   PORTC&=~(1<<0);
   PORTC&=~(1<<1);
   PORTC&=~(1<<2);
   PORTC&=~(1<<3);

}
void motor_init()
{
   DDRC=0B00001111;
}
```

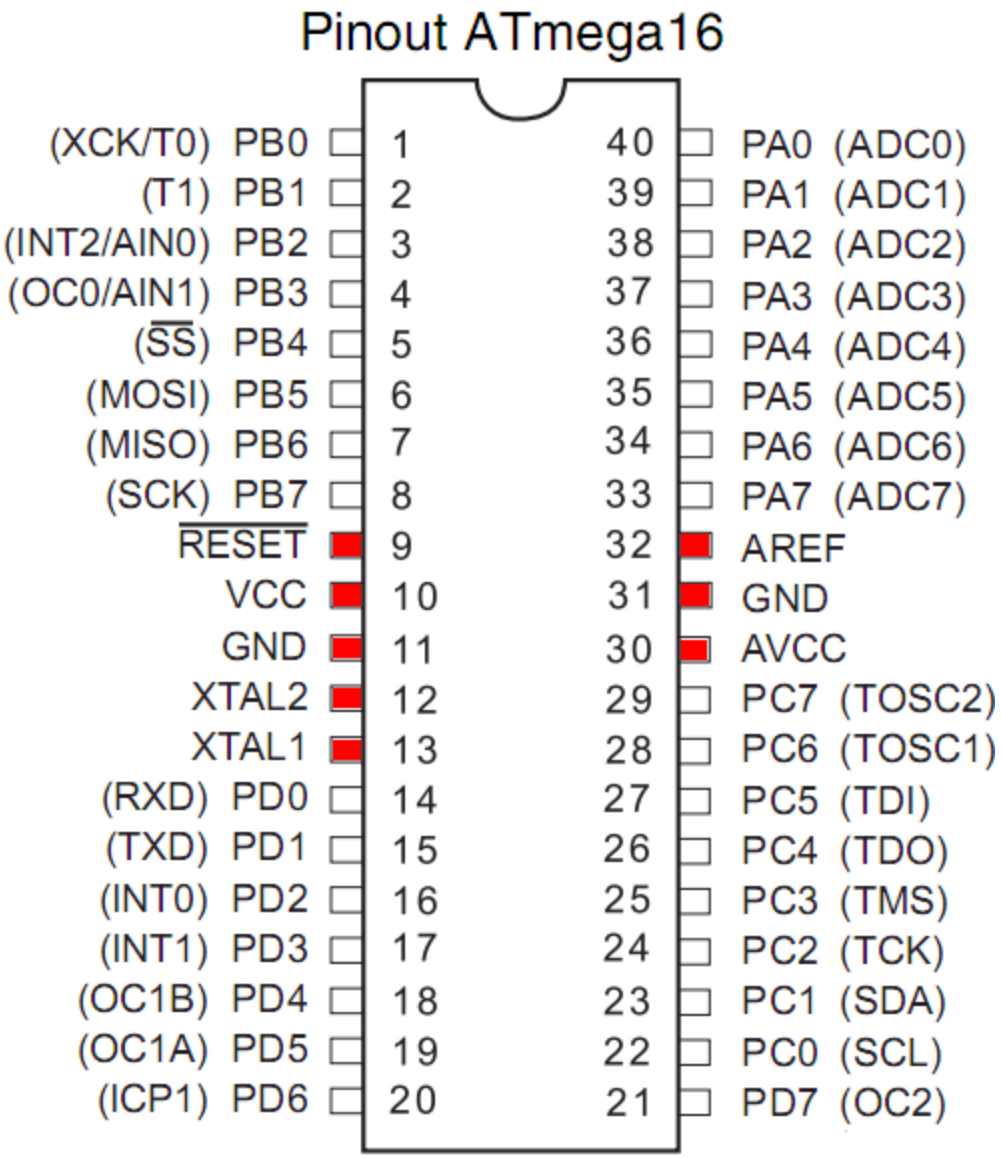
Hardware Requirements

- ## ATMEGA16

  The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves

  Throughputs approaching 1 MIPS per MHz allowing the system designed to optimize power consumption versus processing speed.

Pin Diagram:



Pin Descriptions

- **VCC:** Digital supply voltage.

- **GND**: Ground.

- **Port A (PA7..PA0)**: Port A serves as the analog inputs to the A/D Converter.

  Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

- **Port B (PB7..PB0):** Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

- **Port C (PC7..PC0)**: Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG

interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

- **Port D (PD7..PD0)**: Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

- **RESET**: Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

- **XTAL1:** Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

- **XTAL2**: Output from the inverting Oscillator amplifier.

- **AVCC:** AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.

- **AREF**: AREF is the analog reference pin for the A/D Converter.

- USBASP PROGRAMMER

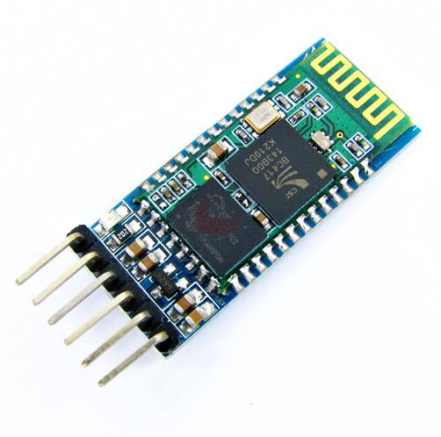
The USBasp is a cheap and simple USB programmer for Atmel AVR's. It basically consists of an ATMega88 or ATMega8, a handful of passive components and as such easily etched and assembled. These AVR programmers are based on USBasp design and connect to your computer's USB port. Not only are they quite compact (70x20mm), but the design is really elegant. The USB interface is achieved by using an atmega8 processor and the rest is done in firmware.

- Works under multiple platforms. Linux, Mac OS X and Windows are tested.
- No special controllers or smd components are needed.
- Programming speed is up to 5kBytes/sec.
- SCK option to support targets with low clock speed (< 1,5MHz).

- Planned: serial interface to target (e.g. for debugging).

BLUETOOTH MODULE

HC-05 is a class-2 Bluetooth module with Serial Port Profile , which can configure as either Master or slave. a Drop-in replacement for wired serial connections, transparent usage. You can use it simply for a serial port replacement to establish connection between MCU, PC to your embedded project and etc.



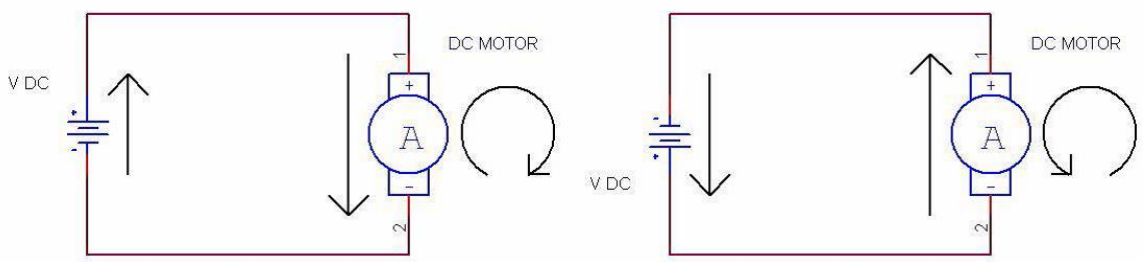
HC-05 Specification:

- Bluetooth protocol: Bluetooth Specification v2.0+EDR
- Frequency: 2.4GHz ISM band
- Modulation: GFSK(Gaussian Frequency Shift Keying)
- Emission power: ≤4dBm, Class 2
- Sensitivity: ≤-84dBm at 0.1% BER
- Speed: Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- Security: Authentication and encryption
- Profiles: Bluetooth serial port

- Power supply: +3.3VDC 50mA
- Working temperature: -20 ~ +75Centigrade
- Dimension: 26.9mm x 13mm x 2.2 mm

• DC MOTOR

A direct current or DC motor is a mechanically commutated electric motor powered from direct current (DC). It converts electrical energy into mechanical energy. It is one of two basic types of motors: the other type is the alternating current or AC motor. Among DC motors, there are shunt-wound, series-wound, compound-wound and permanent magnet motors.

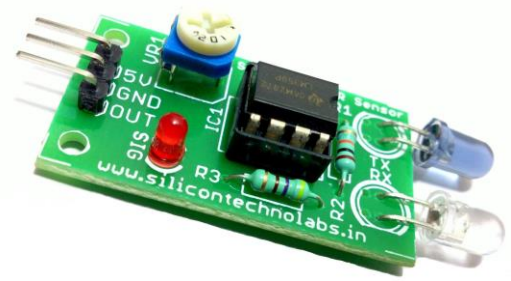


• INFRARED SENSOR

An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An Infrared (IR) sensor is used to detect obstacles in front of the robot or to differentiate between colors depending on the configuration of the sensor.

An IR sensor consists of an emitter, detector and associated circuitry. The circuit required to make an IR sensor consists of two parts; the emitter circuit and the receiver circuit.

The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, its resistance and correspondingly, its output voltage, change in proportion to the magnitude of the IR light received. This is the underlying principle of working of the IR sensor



- MOTOR DRIVER IC

A motor driver IC is an integrated circuit chip which is usually used to control motors in autonomous robots. Microprocessors operate at low voltages and require a small amount of current to operate while the motors require a relatively higher voltages and current. Thus current cannot be supplied to the motors from the microprocessor. This is the primary need for the motor driver IC.

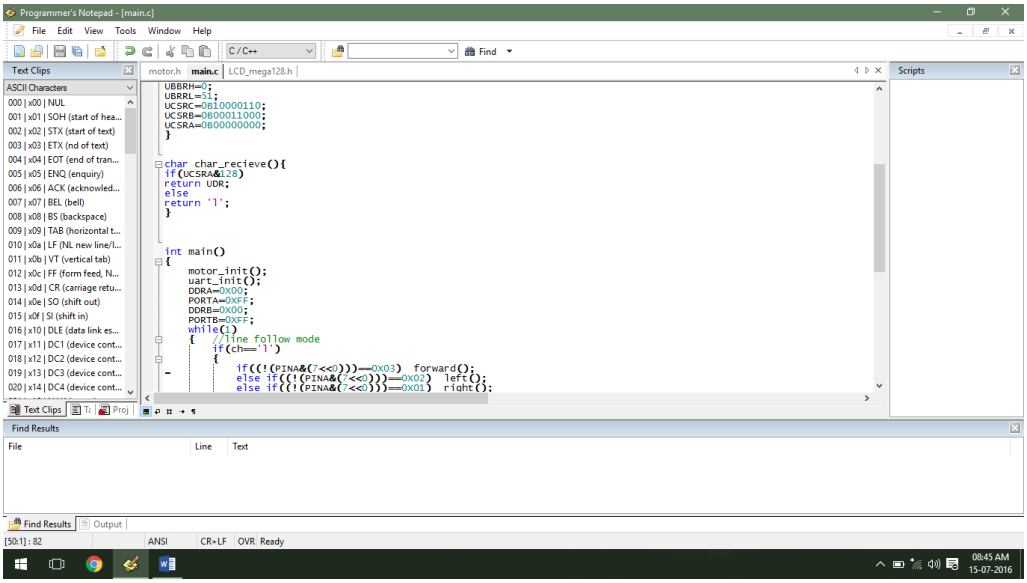
The L293D IC receives signals from the microprocessor and transmits the relative signal to the motors. It has two voltage pins, one of which is used to draw current for the working of the L293D and the other is used to apply voltage to the motors. The L293D switches it output signal according to the input received from the microprocessor.

Software Requirements

- ## WINAVR

  WinAVR (pronounced "whenever") is a suite of executable, open source software development tools for the Atmel AVR series of RISC microprocessors hosted on the Windows platform. It includes the GNU GCC compiler for C and C .

  WinAVR contains all the tools for developing on the AVR. This includes avr-gcc (compiler), avrdude (programmer), avr-gdb (debugger), and more! WinAVR is used all over the world from hobbyists sitting in their damp basements, to schools, to commercial projects.



Distribution also includes the standard operating system for UNIX tools such as Find, make, grep , awk , sed , etc., and based on the Scintilla editor for programming . Part of the package AVR-GCC cross-compiler supports not only the input languages C and C + +, but Objective-C, and provides a complete development environment for AVR32. WinAVR has no master

source code AVR equipment settings and interface with different devices, but the code generated by the compiler master CVAVR, it is possible to compile in WinAVR.

- ## BLUETERM

  Blueterm is a VT-100 terminal emulator for communicating with any serial device using a Bluetooth serial adapter. The RFCOMM/SPP protocol emulates serial communications over Bluetooth.

  Blueterm is basically a terminal emulator program that communicates over Bluetooth. It consists of several activities with Blueterm being the most important. It discovers, pairs, and connects with a remote Bluetooth device that supports SPP/RfComm.

  It helps to control a robotic car with mobile. As we type a character on Blueterm the car responds according to the function set to the character.

Future Enhancements

More IR sensors can be installed to the car to avoid collision in any direction and Artificial intelligence can be added to the car to make decisions to avoid obstacles.

This system can be used as transportation system in industry level where tracks are still being used as this will be more economical viable and safe