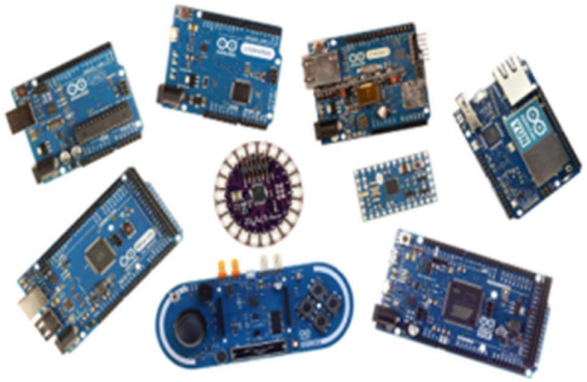
ARDUINO BOOT-CAMP WITH PROTEUS

INTRODUCTION:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

ARDUINO:

Arduino is a name of a company which provides different types of Arduino boards along with Arduino Software (IDE).



Features of Different Types of Arduino Boards:

Arduino Board	Processor	Memory	Digital I/O	Analogue I/O
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KB SRAM, 512KB flash	54	12 input, 2 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	20	12 input, 0 output

ARDUINO UNO:

The Uno is a huge option for our initial projects. This Arduino board depends on an ATmega328P based microcontroller. As compared with other types of Arduino boards, it is very simple to use. It consists of 14-digital I/O pins, where 6-pins can be used as PWM (pulse width modulation outputs), 6-analog inputs, a reset button, a power jack, a USB connection, an In-Circuit Serial Programming header (ICSP), etc. It includes everything required to hold up the microcontroller; simply attach it to a PC with the help of a USB cable and give the supply to get started with an AC-to-DC adapter or battery.

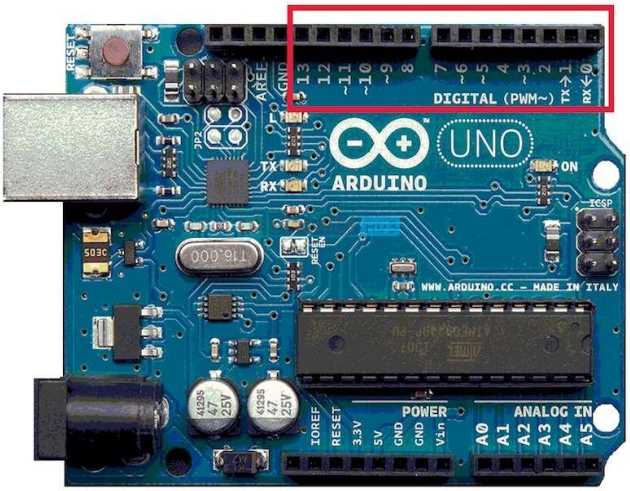
Arduino Uno is the most frequently used board and it is the standard form apart from all the existing Arduino Boards. This board is very useful for beginners.

Arduino Uno Specification:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Length: 68.6 mm
- Width: 58.4 mm
- Weight: 25 g

Powering up the Arduino Uno:

The Arduino Uno board can be powered via a USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector. The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

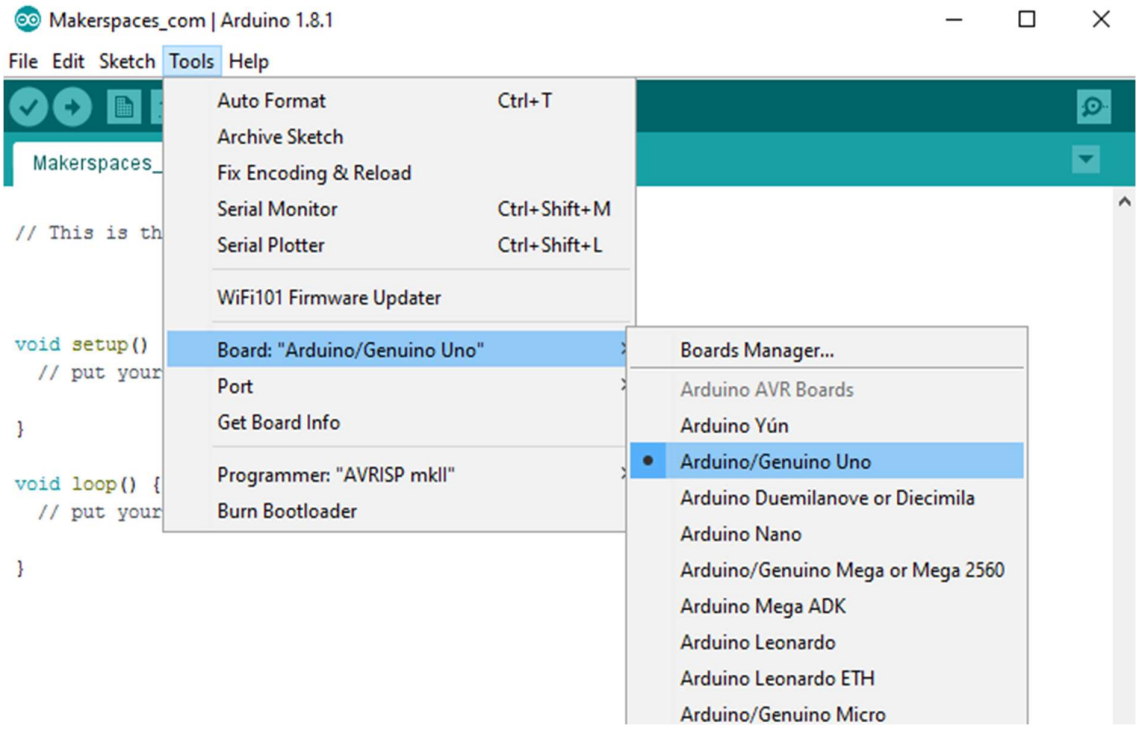


GETTING STARTED WITH ARDUINO:

Before you can start working with Arduino, you need to make sure you have the IDE software installed on your computer. This program allows you to write, view and upload the code to your Arduino Uno board. You can download the IDE for free on Arduino's website. Once the IDE is installed, you will need to connect your Arduino to your computer. To do this, plug one end of the USB cable to the Arduino Uno and then the other end of the USB to your computer's USB port

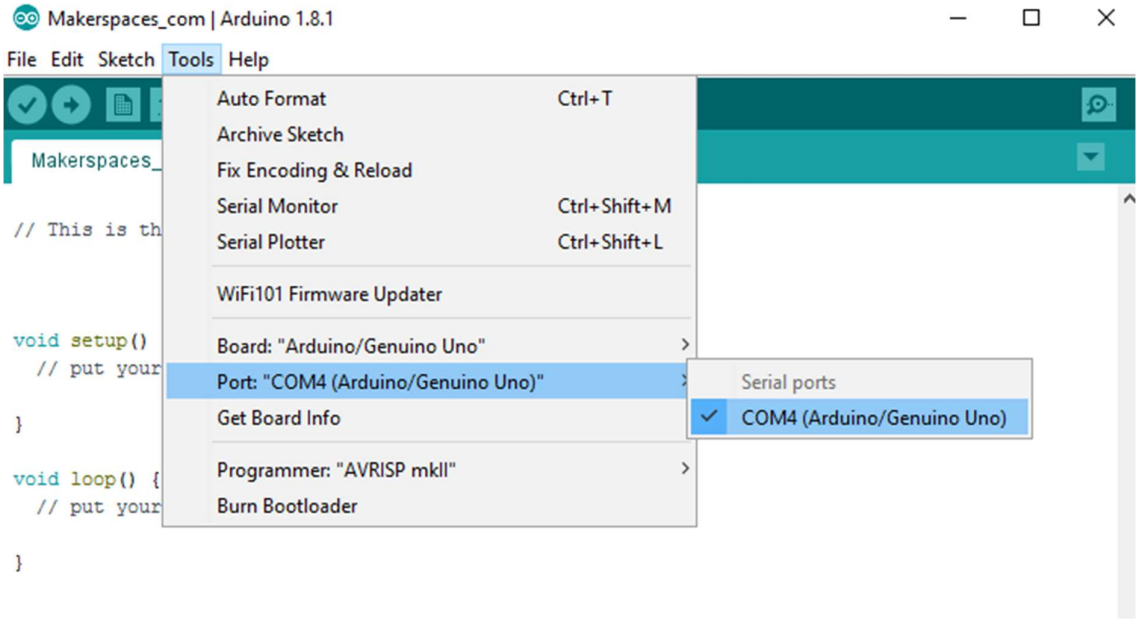
SELECT THE BOARD:

Once the board is plugged in, you will need to open the IDE and click on **Tools** > **Board** > **Arduino Uno** to select the board.



SELECT SERIAL PORT:

Next, you have to tell the Arduino which port you are using on your computer. To select the port, go to **Tools > Port** and then select the port that says **Arduino.**



BASIC SYNTAX:

```
void setup()
{
  // put your setup code here, to run once:

}

void loop()
{
  // put your main code here, to run repeatedly:

}
```

- ⇨ `void setup()` executes only once in the code; in void setup() we define the pinMode(13,OUTPUT), serial communication(Serial.begin(9600)), lcdObject.begin(16,2) ect.
- ⇨ `void loop()` executes infinite number of times, we commonly write the main logical code in this loop.

I/O FUNCTIONS:

The pins on the Arduino board can be configured as either inputs or outputs. We will explain the functioning of the pins in those modes. It is important to note that a majority of Arduino analog pins, may be configured, and used, in exactly the same manner as digital pins.

pinMode( ) Function:

The pinMode() function is used to configure a specific pin to behave either as an input or an output.

pinMode() Function Syntax

```
void setup()
{
pinMode(pin,mode);

}
```

- pin – the number of the pin whose mode you wish to set.
- `mode` – INPUT, OUTPUT, or INPUT_PULLUP.

digitalWrite() Function

The **digitalWrite()** function is used to write a HIGH or a LOW value to a digital pin.

digitalWrite() Function Syntax

```
void loop()
{
 digitalWrite(pin,value);

}
```

- pin − the number of the pin whose mode you wish to set.
- value − HIGH, or LOW.

analogRead( ) function

Arduino is able to detect analog voltage applied to one of its analog pins(A0,A5).

analogRead() function Syntax

```
void loop()
{
x=analogRead(pin);

}
```

- pin − the number of the analog input pin to read from (A0,A5)
- x- is variable of type int or float to store the value of voltage

TIME:

delay () function

The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function. However, the problem is, the delay() function is not a good way to make your program wait, because it is known as a "blocking" function.

delay() function Syntax

```
delay(ms);
```

where, **ms** is the time in milliseconds to pause (unsigned long).

delayMicroseconds () function

The **delayMicroseconds()** function accepts a single integer (or number) argument. This number represents the time and is measured in microseconds. There are a thousand microseconds in a millisecond, and a million microseconds in a second.

Currently, the largest value that can produce an accurate delay is 16383. This may change in future Arduino releases. For delays longer than a few thousand microseconds, you should use the delay() function instead.

delayMicroseconds() function Syntax

```
delayMicroseconds(us);
```

where, **us** is the number of microseconds to pause (unsigned int)

1.BLINKING OF LED

LEDs are small, powerful lights that are used in many different applications. To start, we will work on blinking an LED, the Hello World of microcontrollers.

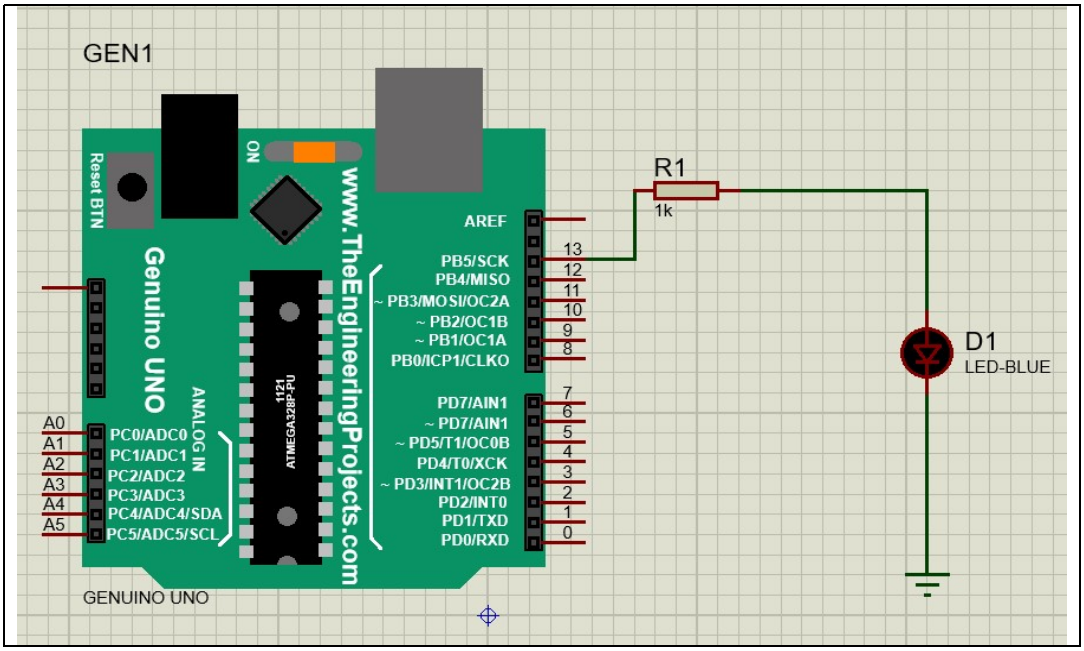
CODE :

```
void setup()
{
pinMode(13,OUTPUT);


}

void loop() {
 digitalWrite(13,HIGH);
delay(100);
digitalWrite(13,LOW);
delay(100);

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- MINRES 1K
- LED BLUE

CIRCUIT:



2.SERIAL COMMUNICATION

Serial Communication is the most important topic in Arduino uno. The arduino can able to send and receive data serially from a devices called PC, Bluetooth ect.

Here we use the PC(laptop) to send and receive the data from the Arduino.

This can be done by using the Serial monitor in the Arduino(IDE) in practical applications of Arduino. But , for the simulation we are using the VIRTUAL TERMINAL in the proteus for serial communication.

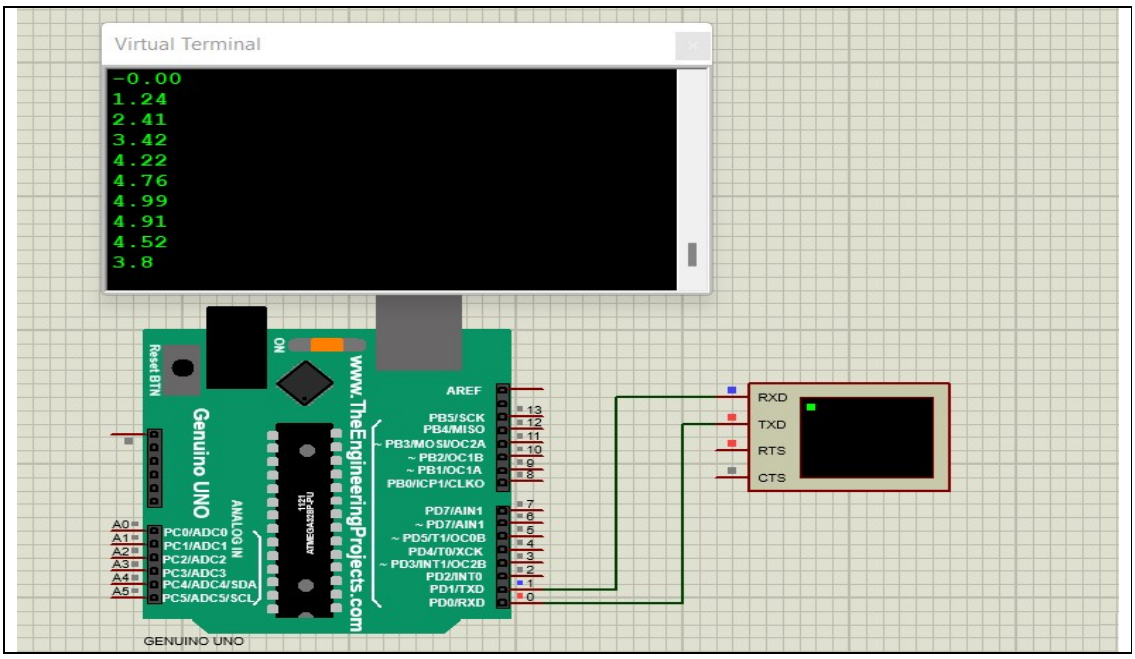
CODE:

```
float t=0,x;

void setup()
{
 Serial.begin(9600);
}

void loop()
{
  x=5*sin((2*PI*4*t));
  Serial.println(x);
  t=t+0.01;
}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- VIRTUAL TERMINAL

CIRCUIT:



3.DIGITAL READ

Arduino reads the digital input provided at the digital input pins

Here we explain with the help of switch connected to 4V supply

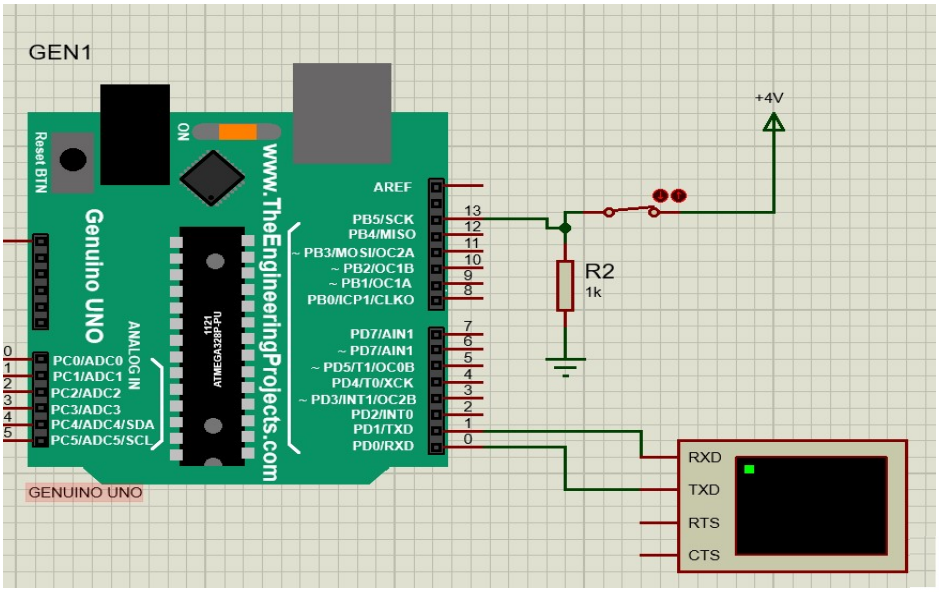
CODE:

```
int x;

void setup()
{
  pinMode(13,INPUT);
  Serial.begin(9600);
}

void loop()
{

x=digitalRead(13);
Serial.println(x);
delay(300);

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- MINRES 1K
- SWITCH
- POWER
- VIRTUAL TERMINAL

CIRCUIT:



4.ANALOG READ:

Same as digital Read( ) ,analogRead( ) can able to read analog volage at the analog pins.

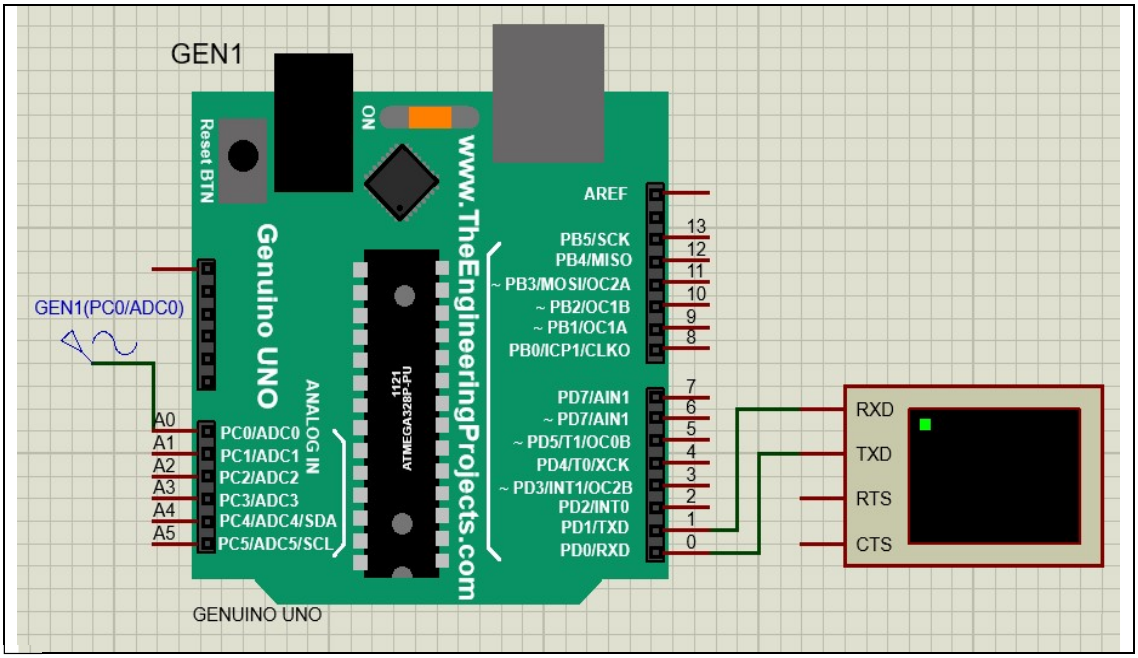
CODE:

```
float x;

void setup()
{
pinMode(A0,INPUT);
Serial.begin(9600);

}

void loop()
{
x=analogRead(A0);
Serial.println(x);
delay(100);
}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- VIRTUAL TERMINAL
- SINE

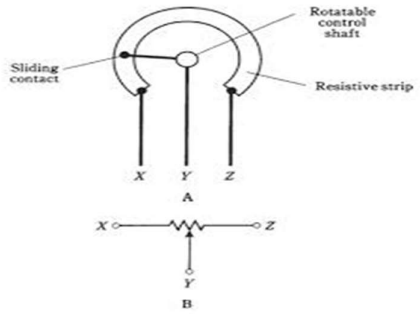
CIRCUIT:



5.POTENTIOMETER

This potentiometer gives gives the variable voltage ,which drops across the lower resister of potentiometer.

Simply it is based on analogeRead( ) function.



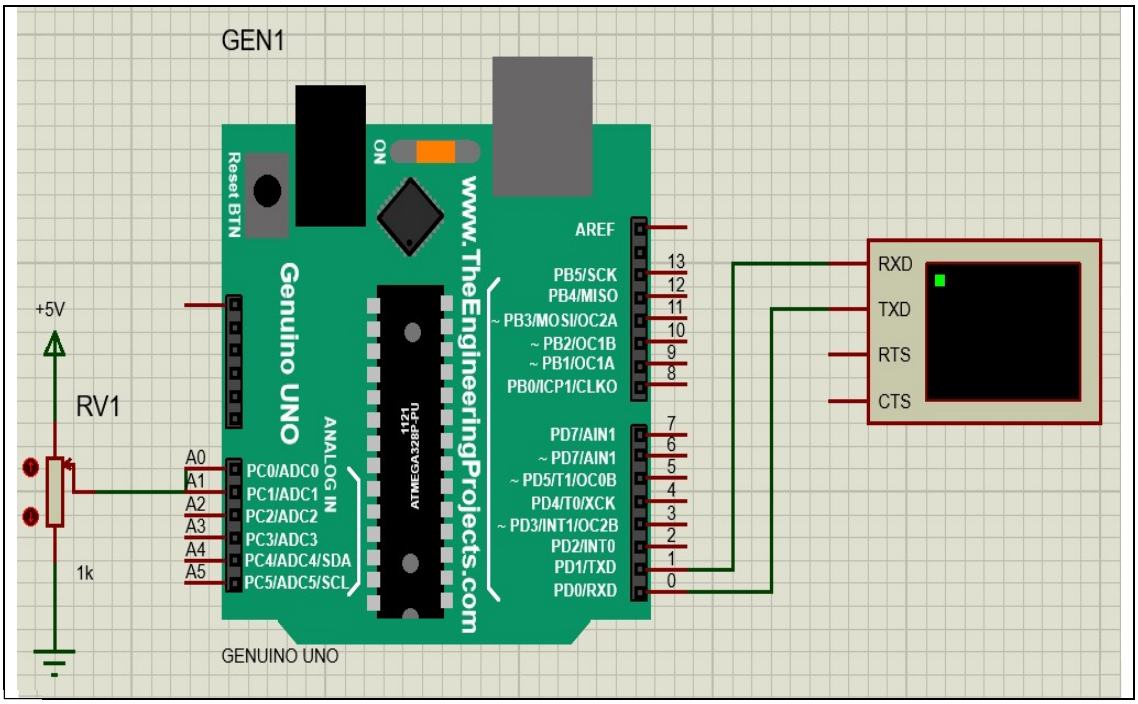
CODE:

```
int x;

void setup() {
  pinMode(A0,INPUT);
  Serial.begin(9600);
}

void loop()
{
 x= analogRead(A0);
 Serial.println(x);
 delay(100);

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- POT
- VIRTUAL SERIAL MONITOR

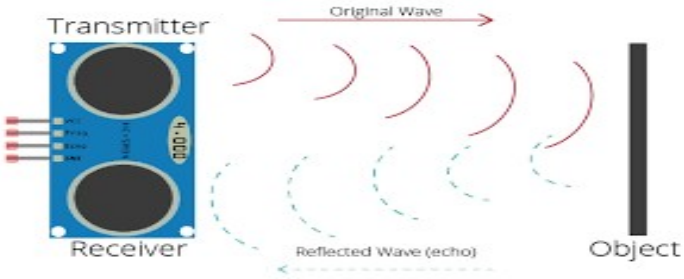
CIRCUIT:


GEN1
Reset BTN
ON
www.TheEngineeringProjects.com
Genuino UNO
ANALOG IN
ATMEGA328P-PU
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
13
12
11
10
9
8
7
6
5
4
3
2
1
0
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
+5V
RV1
1k
GENUINO UNO
RXD
TXD
RTS
CTS

6.ULTRASONIC SENSOR

The ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.



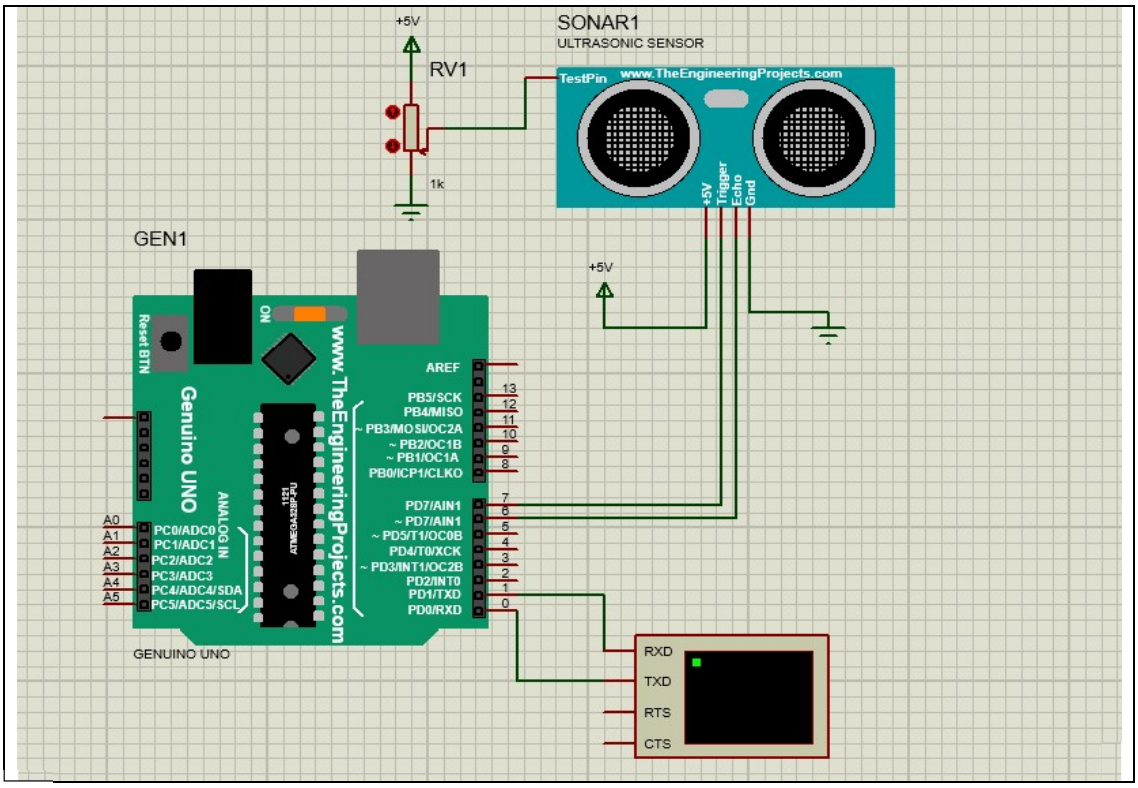
CODE:

```
long dur;

void setup() {
  Serial.begin(9600);
  pinMode(7,OUTPUT);
  pinMode(6,INPUT);


}

void loop()
{
digitalWrite(7,LOW);       //pulse to generate UltraSonicSound waves
delayMicroseconds(2);
digitalWrite(7,HIGH);
delayMicroseconds(10);
digitalWrite(7,LOW);

dur=pulseIn(6,HIGH);
Serial.println(dur*0.034/2);


}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- ULTRASONIC SENSOR
- POT(potentiometer)
- VERTUAL TERMINAL

CIRCUIT:



7.SERVO MOTOR

A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft. If the coded signal changes, the angular position of the shaft changes. In practice, servos are used in radio-controlled airplanes to position control surfaces like the elevators and rudders. They are also used in radio-controlled cars, puppets, and of course, robots.

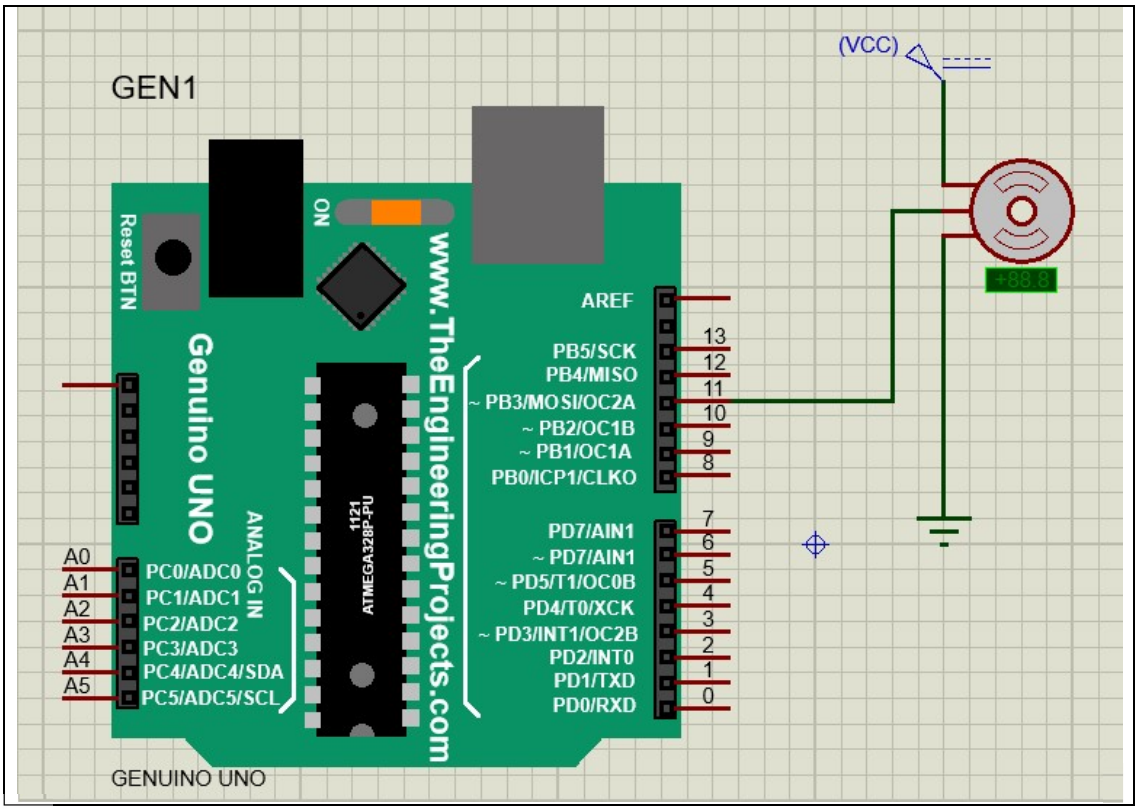
CODE:

```
#include<Servo.h>

Servo s1;

void setup()
{
   s1.attach(11);

}

void loop()
{
    s1.write(0);
    delay(1000);

    s1.write(60);
    delay(1000);

    s1.write(130);
    delay(1000);
}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- SERVO MOTOR
- POWER

CIRCUIT:


GEN1
(VCC)
Reset BTN
ON
www.TheEngineeringProjects.com
Genuino UNO
ANALOG IN
ATMEGA328P-PU
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
GENUINO UNO

8.DC MOTOR

By connecting an L293 bridge IC to an Arduino, you can control a DC motor.

A direct current, or DC, motor is the most common type of motor. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.

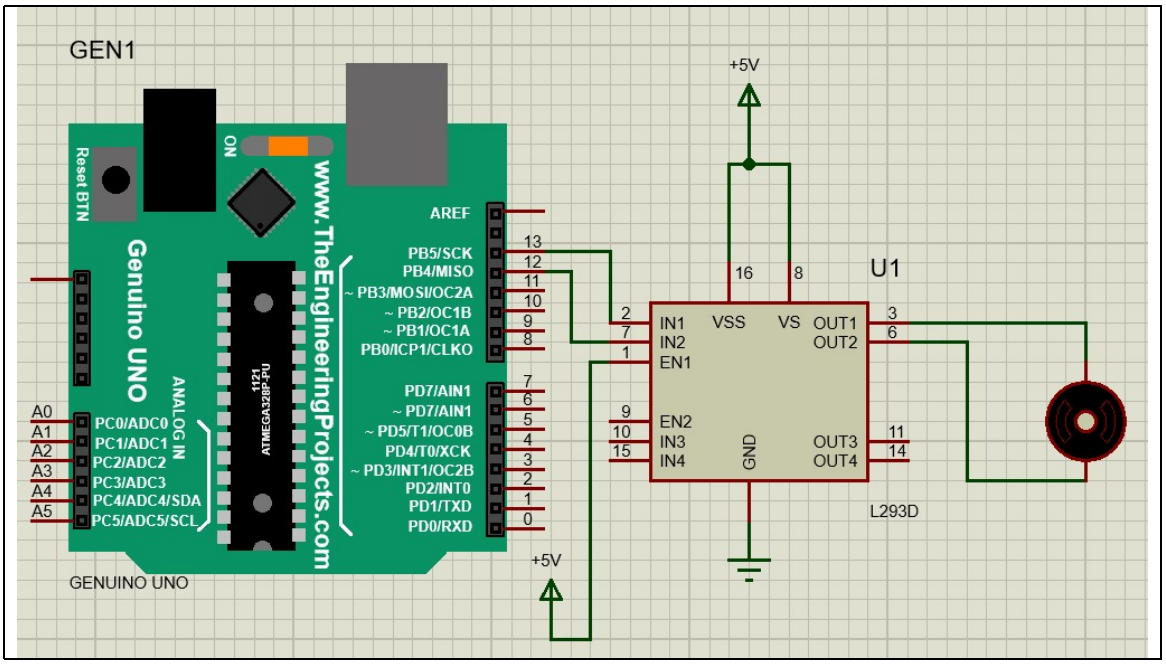
COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- L293
- POWER

CODE:

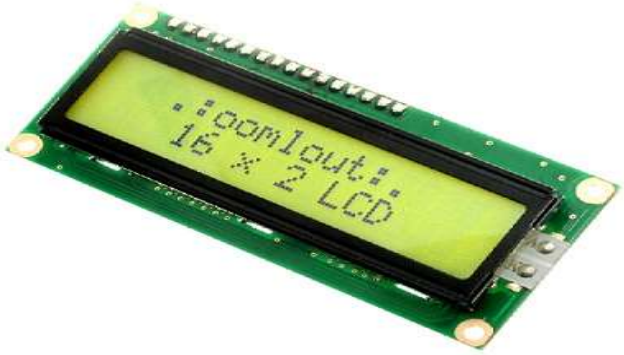
```
void setup()
{
 pinMode(13,OUTPUT);
 pinMode(12,OUTPUT);

}


void loop()
{

 digitalWrite(13,HIGH);
 digitalWrite(12,LOW);
 delay(5000);


 digitalWrite(13,LOW);
 digitalWrite(12,HIGH);
 delay(5000);

}
```

CIRCUIT:

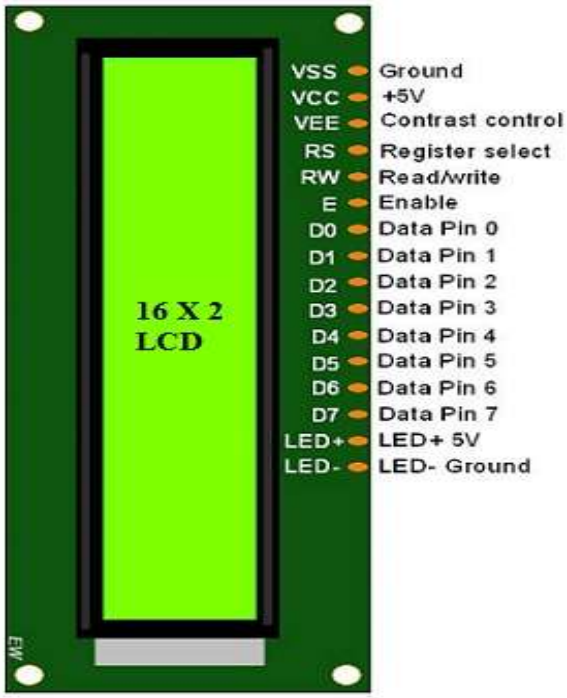

GEN1
Reset BTN
ON
www.TheEngineeringProjects.com
Genuino UNO
ANALOG IN
ATMEGA328P-PU
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
GENUINO UNO
+5V
+5V
U1
IN1
IN2
EN1
EN2
IN3
IN4
VSS
VS
GND
OUT1
OUT2
OUT3
OUT4
L293D

9.LCD DISPLAY

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.



- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, used to connect a microcontroller unit pin and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or writes operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation, and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in 8-wire mode, 8-pins are connected to microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V
- Pin 16 (-ve pin of the LED): This pin is connected to GND.



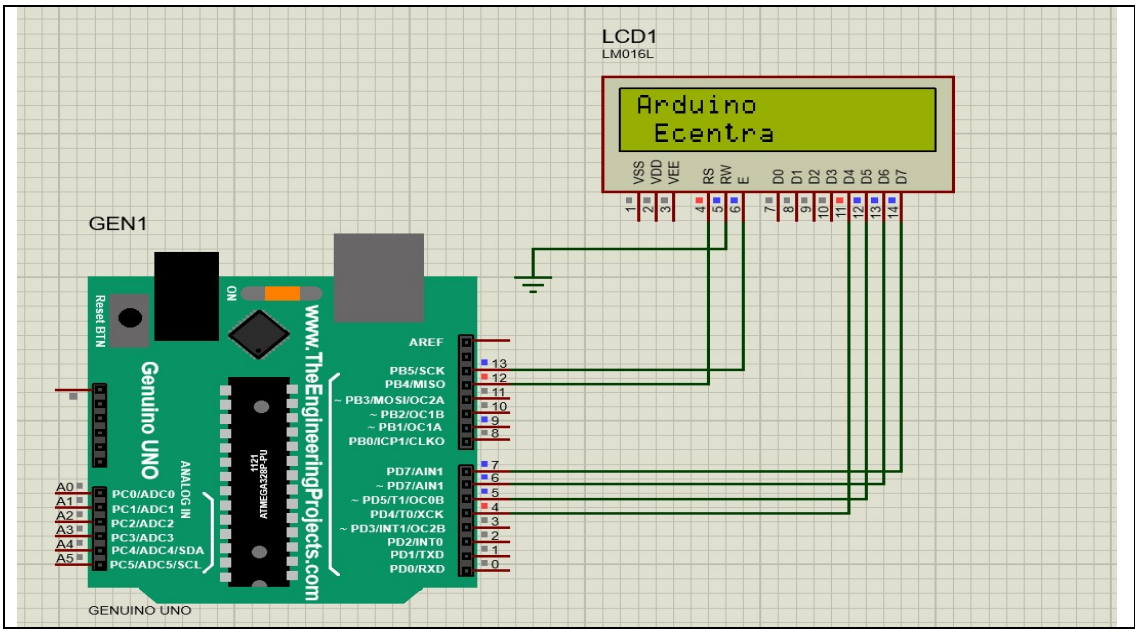
CODE:

```
#include<LiquidCrystal.h>

LiquidCrystal P1(12,13,4,5,6,7);

void setup() {
 P1.begin(16,2);
 P1.print("Arduino");

 }

void loop() {

P1.setCursor(1,1);
P1.print("Ecentra");
delay(1000);
P1.clear();
delay(1000);

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- LCD DISPLAY

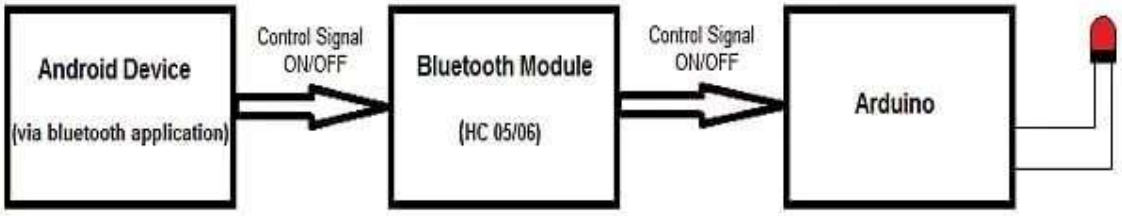
CIRCUIT:


LCD1
LM016L
Arduino
Ecentra
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
GEN1
Reset BTN
ON
www.TheEngineeringProjects.com
Genuino UNO
ANALOG IN
ATMEGA328P-PU
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
GENUINO UNO

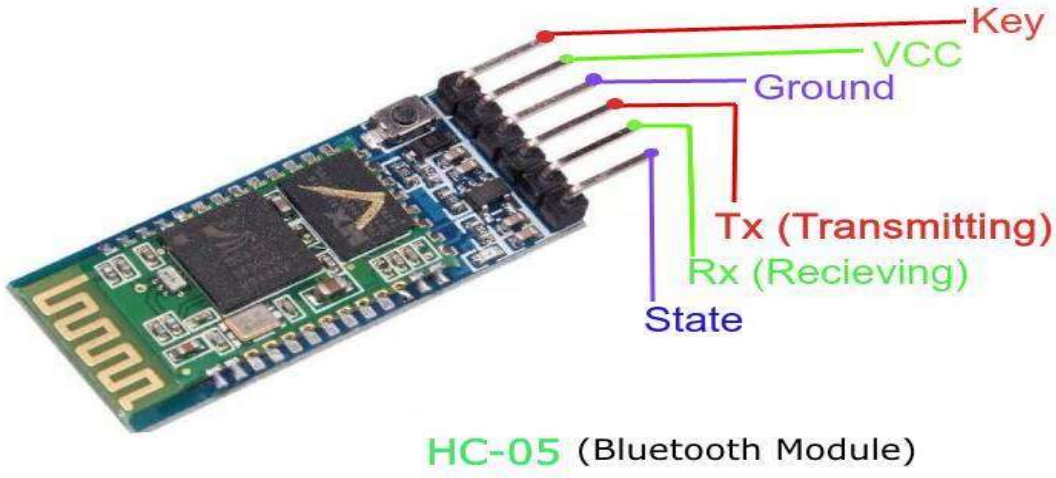
10.BLUETOOTH LED

Interface an Android smartphone with an Arduino via Bluetooth to control an LED from your phone.

There are three main parts to this project. An Android smartphone, a Bluetooth transceiver, and an Arduino.



HC 05/06 works on serial communication. The Android app is designed to send serial data to the Arduino Bluetooth module when a button is pressed on the app. The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module (connected to RX pin of Arduino). The code uploaded to the Arduino checks the received data and compares it. If the received data is 1, the LED turns ON. The LED turns OFF when the received data is 0. You can open the serial monitor and watch the received data while connecting.



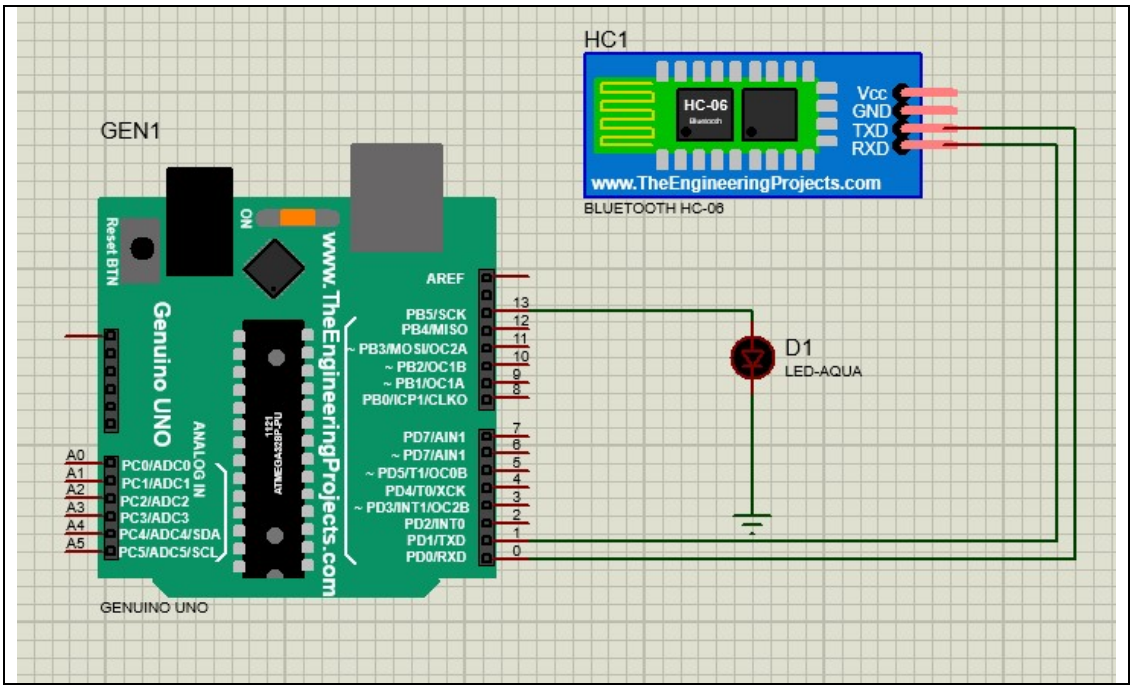
CODE:

```
String x;

void setup()
{
  Serial.begin(9600);
  pinMode(13,OUTPUT);
}

void loop()
{
  while(Serial.available())
     {  x=Serial.readString(); }


  if(x=="on" or x=="ON")
     {  digitalWrite(13,HIGH); }

  else if(x=="off" or x=="OFF")
     {  digitalWrite(13,LOW);  }

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- BLUETOOTH
- LED

CIRCUIT:



11.BLUETOOTH DISPLAY

It is a combination of both LCD display and Bluetooth module, the messages which are send through the android app will be displayed in LCD display.

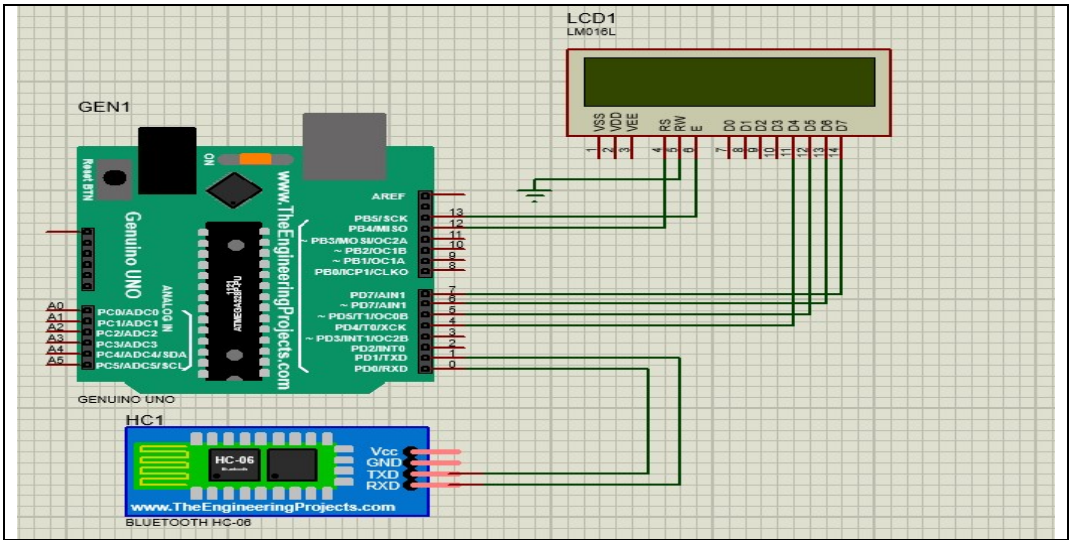
CODE:

```
#include<LiquidCrystal.h>

LiquidCrystal l1(12,13,4,5,6,7);
String x;

void setup()
{
  l1.begin(16,2);
  Serial.begin(9600);
  l1.print("ARDUINO");
}

void loop()
{
   while(Serial.available())
      {x=Serial.readString();}

      l1.clear();
      l1.print(x);

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- BLUETOOTH
- LCD DISPLAY

CIRCUIT:



12.HOMEAUTOMATION

Home automation is simply a Bluetooth led project ,but relays are used extra to control the high voltage home appliance.

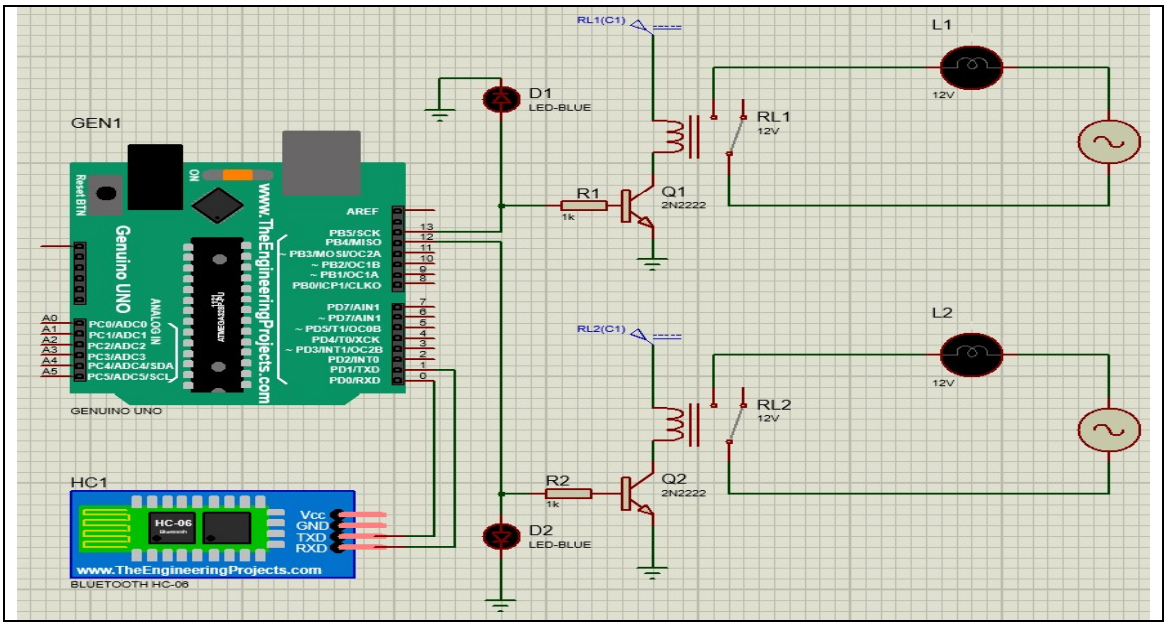
CODE:

```
char x;
void setup()
{
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  Serial.begin(9600);
}

void loop()
{

  while(Serial.available())
       { x=Serial.read();}


     if(x=='A')
        {digitalWrite(13,HIGH);}

     else if(x=='a')
        {digitalWrite(13,LOW);}

     if(x=='B')
        {digitalWrite(12,HIGH);}

     else if(x=='b')
        {digitalWrite(12,LOW);}

}
```

COMPONENTS REQUIRED:

- ARDUINO UNO
- BLUETOOTH
- RELAY
- LAMP
- TRANSISTOR
- MINRES 1K(resistance)
- ALTERNATOR
- POWER

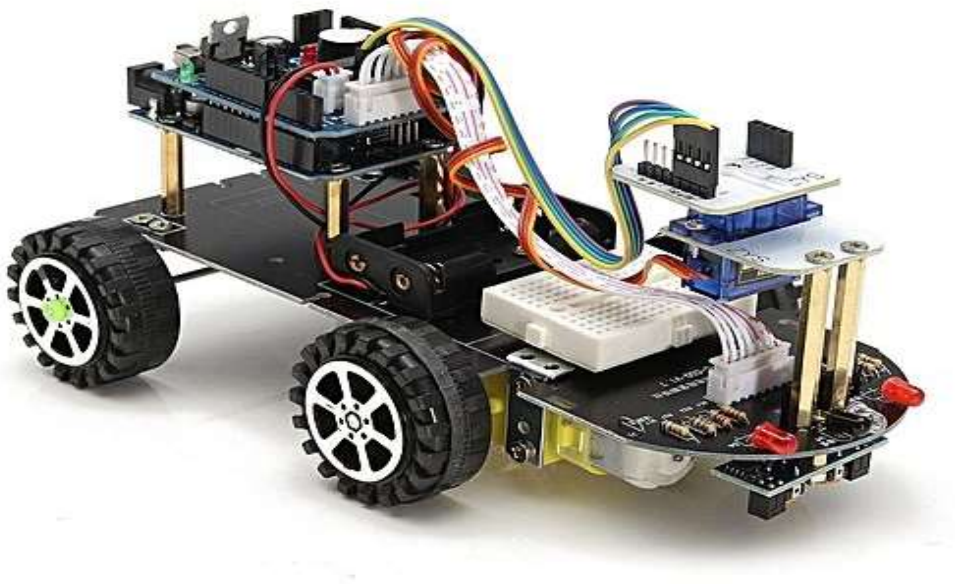
CIRCUIT:


RL1(C1)
L1
12V
D1
LED-BLUE
GEN1
RL1
12V
R1
1k
Q1
2N2222
Genuino UNO
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
GENUINO UNO
RL2(C1)
L2
12V
RL2
12V
HC1
HC-06
Vcc
GND
TXD
RXD
www.TheEngineeringProjects.com
BLUETOOTH HC-06
R2
1k
Q2
2N2222
D2
LED-BLUE

13.BLUETOOTH ROBOT

Bluetooth Robot is a device which is operated by the commands given by smart phone through Bluetooth.

Here we use the motor driver to control the DC motors, because the Arduino can not give high voltages and currents to control the DCmotors.



CODE:

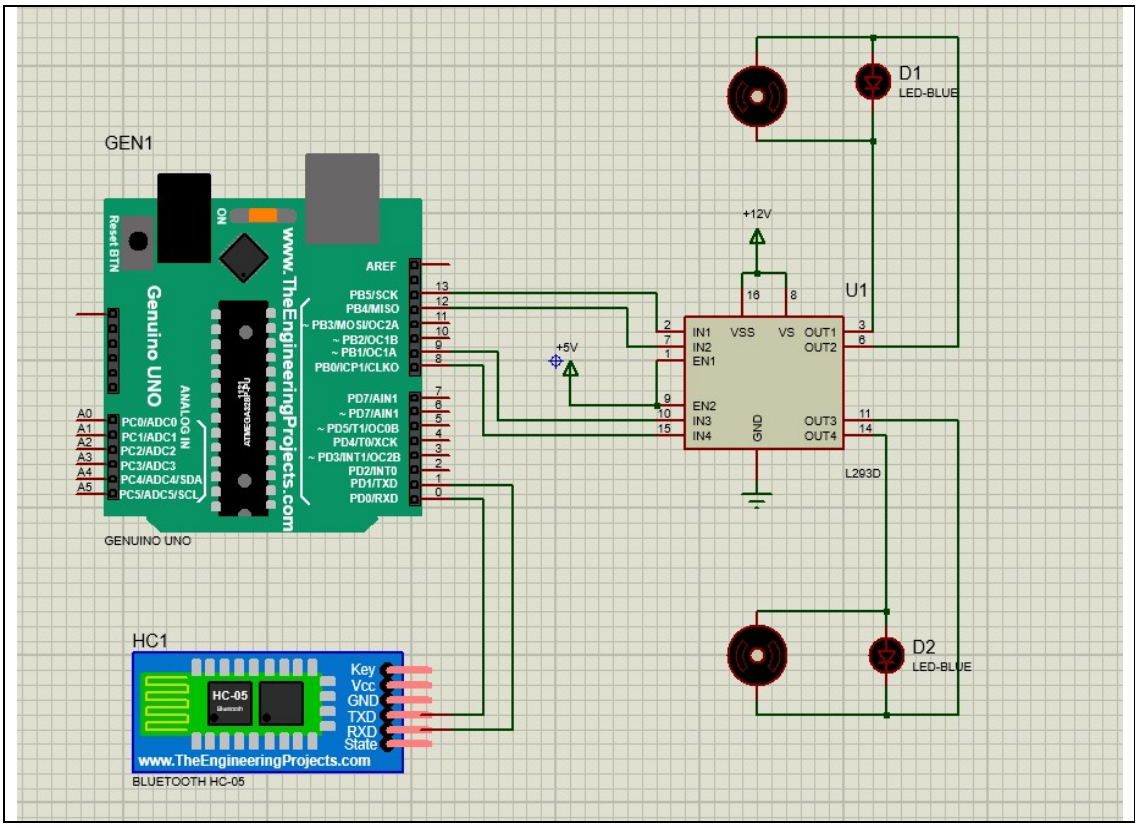
```
void fun(int p,int q)
  {
     digitalWrite(12,p);
     digitalWrite(13,!p);

     digitalWrite(8,q);
     digitalWrite(9,!q);
  }


void Stop()
  {

     digitalWrite(13,0);
     digitalWrite(12,0);
     digitalWrite(9,0);
     digitalWrite(8,0);
  }
```

```
char x;
void setup()
{
  Serial.begin(9600);
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
}

void loop()
{
  while(Serial.available())
       { x=Serial.read();}



  if(x=='Z')
   {
     Stop();
   }

  else if(x=='R')
   {
     fun(1,0);
     delay(1000);
     Stop();
   }

  else if(x=='L')
  {
     fun(0,1);
     delay(1000);
     Stop();
  }

  else if(x=='F')
  {
     fun(1,1);
  }

  else if(x=='B')
  {
    fun(0,0);
  }

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- BLUETOOTH
- L293D
- DC MOTOR
- LED
- POWER

CIRCUIT:


GEN1
Reset BTN
ON
Genuino UNO
ANALOG IN
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
GENUINO UNO
+5V
+12V
U1
IN1
IN2
EN1
EN2
IN3
IN4
VSS
VS
OUT1
OUT2
OUT3
OUT4
GND
L293D
D1
LED-BLUE
D2
LED-BLUE
HC1
Key
Vcc
GND
TXD
RXD
State
HC-05
www.TheEngineeringProjects.com
BLUETOOTH HC-05

16.OBSTACLE AVOIDER

CODE:

```
void fun(int p,int q)
  {
    digitalWrite(12,p);   //This is for motor 1
    digitalWrite(13,!p);

    digitalWrite(8,q);   //This is for motor 2
    digitalWrite(9,!q);
  }

void Stop()
  {

     digitalWrite(13,0);   //To stop the running of motors
     digitalWrite(12,0);
     digitalWrite(9,0);
     digitalWrite(8,0);
  }


float dur,dis;
void setup()
{
  Serial.begin(9600);
  pinMode(13,OUTPUT); //This 4 pins are connected two motor driver
  pinMode(12,OUTPUT);
  pinMode(9,OUTPUT);
  pinMode(8,OUTPUT);
  pinMode(7,OUTPUT); //This 7,8 pins are connected to Ultrasonic
  pinMode(6,INPUT);
}

void loop()
{
   digitalWrite(7,LOW);
   delayMicroseconds(2);
   digitalWrite(7,HIGH);
   delayMicroseconds(10);
   digitalWrite(7,LOW);

  dur=pulseIn(6,HIGH);
  dis=dur*0.034/2;
  Serial.println(dis);

        if(dis<=7)
        {
              fun(0,1);
              delay(300);
              Stop();
        }

        else            //if there is no obstacle near to the sensor
        {
              fun(1,1);
        }

}
```

COMPONENTS REQUIRED:

- GENUINO UNO (Arduino uno)
- L293D
- DC MOTOR
- ULTRASONIC SENSOR
- POWER

CIRCUIT: