Eng.mohammed Albhaisi

Lab#3 : arduino to proteus simulation

for simulate Arduino program that you wrote you have to have these programs :

1-Arduino C

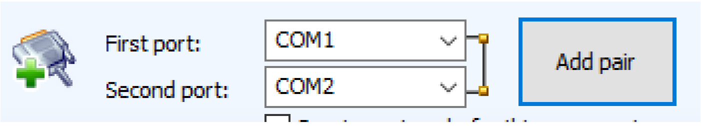
2-proteus

3- Virtual Serial Port Driver

4-Arduino library to proteus

You must do following Steps if you want to simulate your program:

1-We must make virtual port using Virtual Serial Port Driver

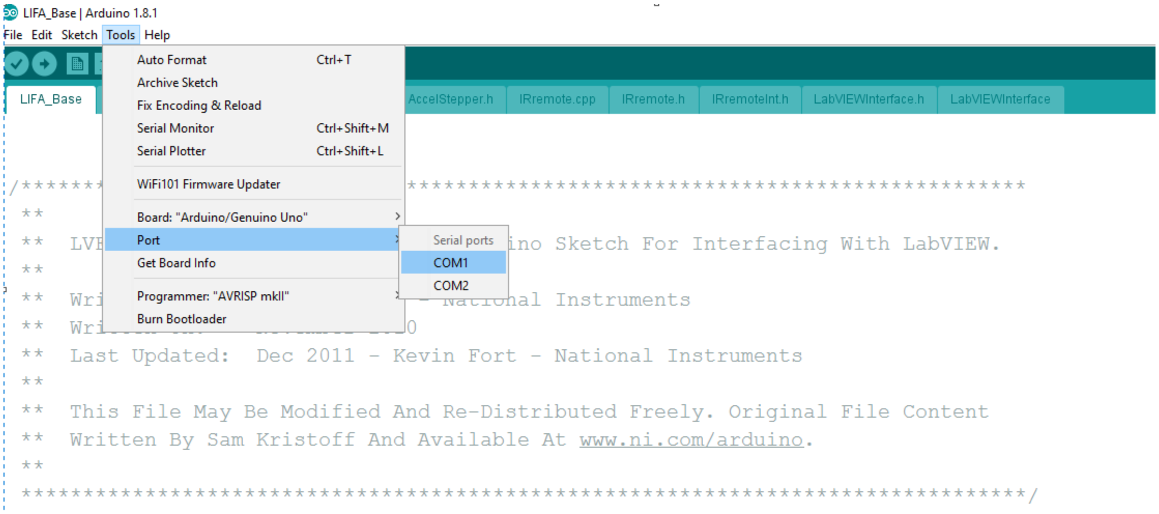


First port: COM1

Second port: COM2

Here we make virtual port between COM1 and COM2

2- choose port and Arduino type from tools



Board: "Arduino/Genuino Uno"

Port: COM1

Programmer: "AVRISP mkII"

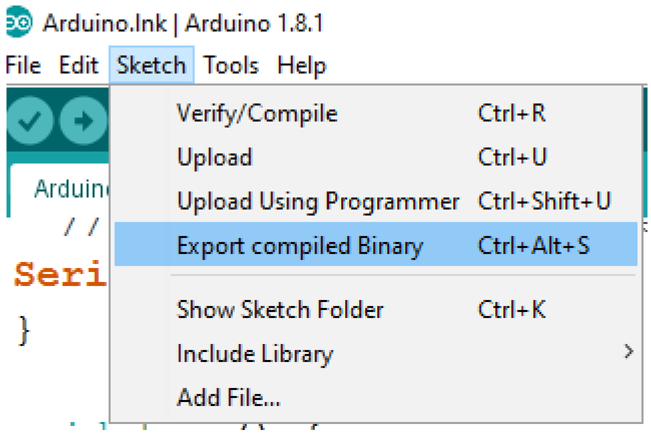
```
/*******                ***********************************************
 **
 **  LVI                ino Sketch For Interfacing With LabVIEW.
 **
 **  Wri                - National Instruments
 **  Wri                0
 **  Last Updated:  Dec 2011 - Kevin Fort - National Instruments
 **
 **  This File May Be Modified And Re-Distributed Freely. Original File Content
 **  Written By Sam Kristoff And Available At www.ni.com/arduino.
 **
 *********************************************************************************/
```

Hints : Here we connected arduino to COM 1

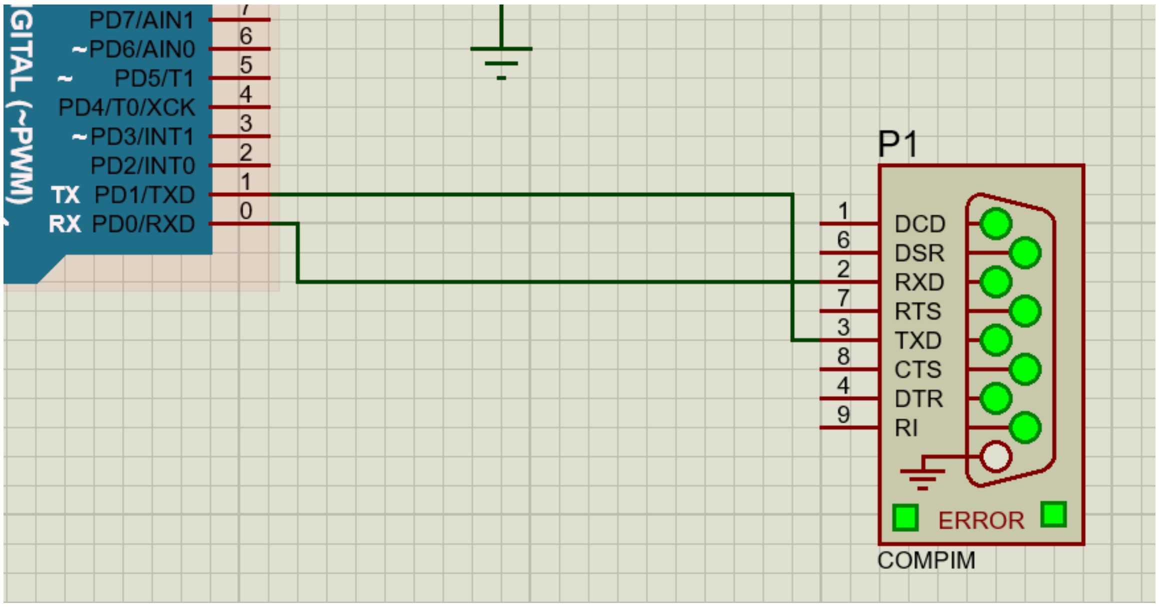
3-write your code , then write code to `initialize` serial communication (you may need it sometimes)

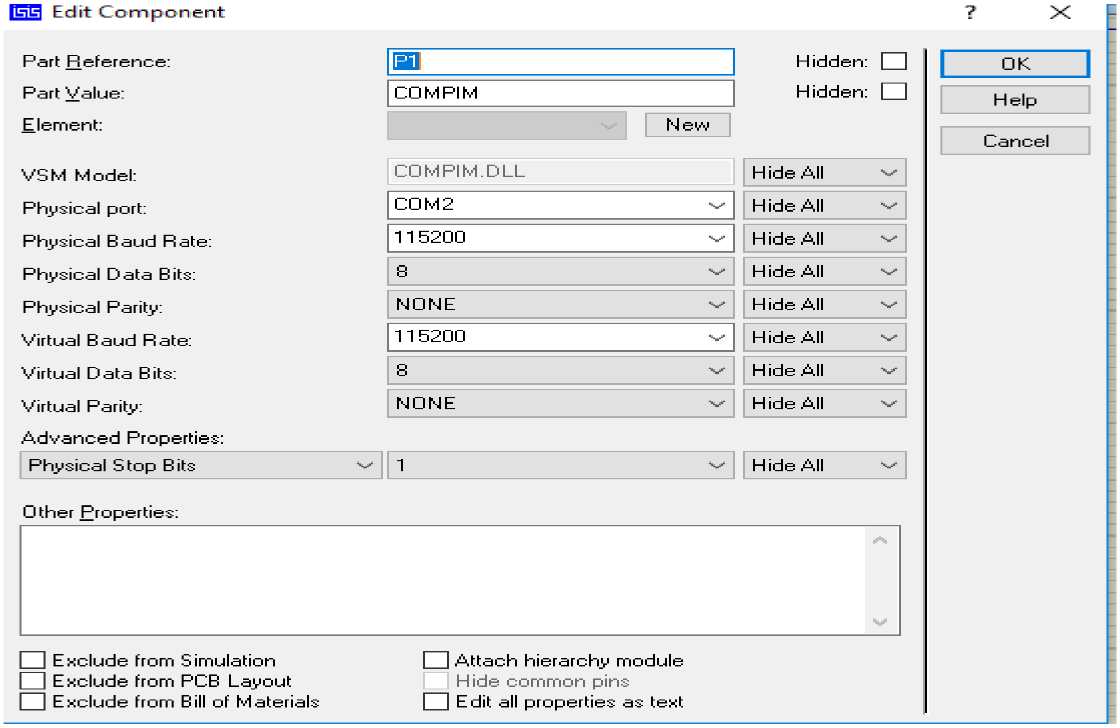
```
void setup()
{
//initialize serial communications at a 9600 baud rate
Serial.begin(9600);
}
```

4-after you finish your code , then choose from Sketch > Export compiled Binary to get hex file .


Arduino.Ink | Arduino 1.8.1
File
Edit
Sketch
Tools
Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Arduin
//
Seri
}

5- choose Arduino uno from protues ,then connect COMPIM(serial port) to arduino Rx from Arduino to Rx in COMPIM and vice versa ,then choose port COM2 in COMPIM ,and change baud rate to 9600 (or try 115200)





Edit Component

Part Reference: P1 Hidden: □

Part Value: COMPIM Hidden: □

Element: New

VSM Model: COMPIM.DLL Hide All

Physical port: COM2 Hide All

Physical Baud Rate: 115200 Hide All

Physical Data Bits: 8 Hide All

Physical Parity: NONE Hide All

Virtual Baud Rate: 115200 Hide All

Virtual Data Bits: 8 Hide All

Virtual Parity: NONE Hide All

Advanced Properties:

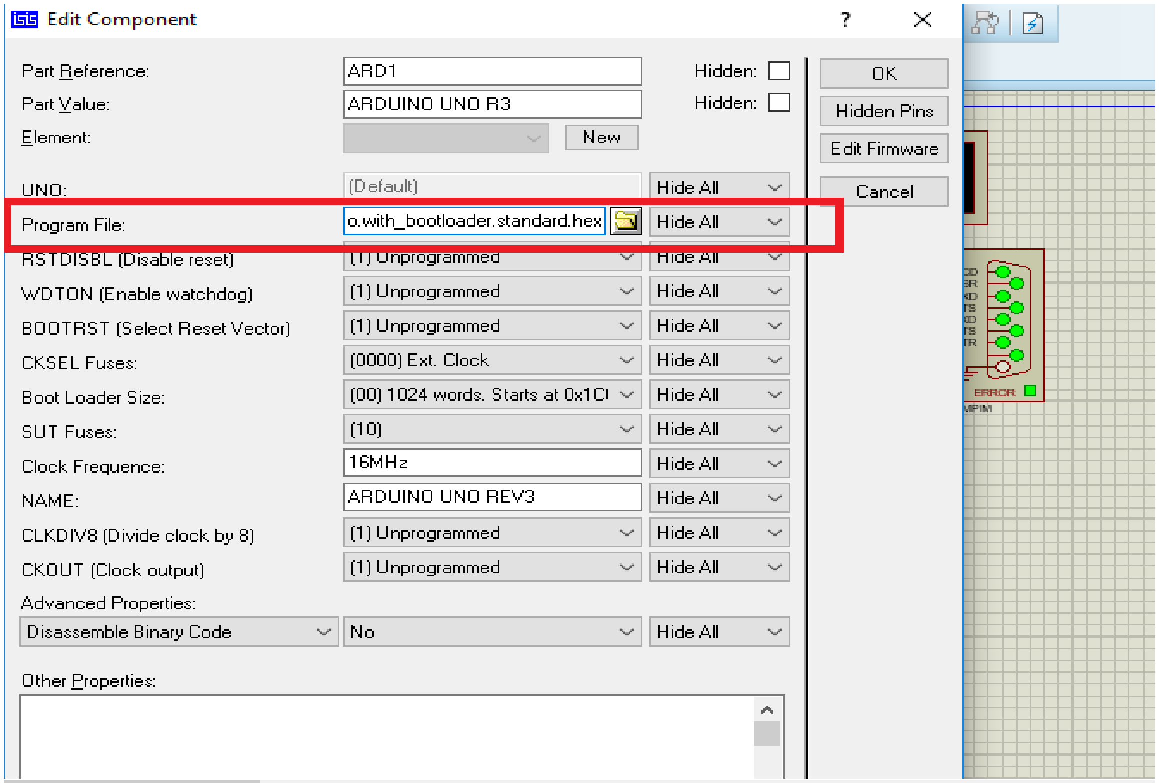
Physical Stop Bits 1 Hide All

Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Exclude from Bill of Materials
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text

Hints : here we choose COM 2 because we connect arduino to COM 1

6-upload hex file to arduino that you export it in step number four .



Edit Component

Part Reference: ARD1 Hidden: □

Part Value: ARDUINO UNO R3 Hidden: □

Element: New

UNO: (Default) Hide All

Program File: o.with_bootloader.standard.hex Hide All

RSTDISBL (Disable reset) (1) Unprogrammed Hide All

WDTON (Enable watchdog) (1) Unprogrammed Hide All

BOOTRST (Select Reset Vector) (1) Unprogrammed Hide All

CKSEL Fuses: (0000) Ext. Clock Hide All

Boot Loader Size: (00) 1024 words. Starts at 0x1C(Hide All

SUT Fuses: (10) Hide All

Clock Frequence: 16MHz Hide All

NAME: ARDUINO UNO REV3 Hide All

CLKDIV8 (Divide clock by 8) (1) Unprogrammed Hide All

CKOUT (Clock output) (1) Unprogrammed Hide All

Advanced Properties:

Disassemble Binary Code No Hide All

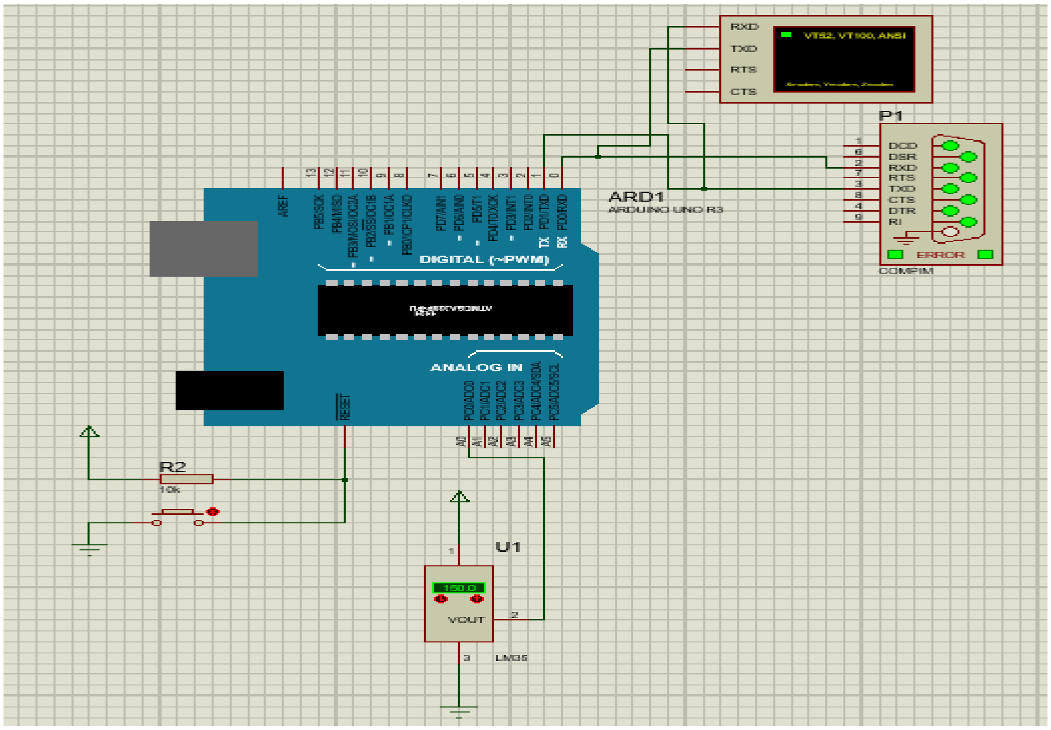
Other Properties:

Example # 1 : lm 35 sensor

First code Arduino C :

```
const int sensorPin=A0;
int reading;
float voltage;
float temperatureC;
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
reading=analogRead(sensorPin);
voltage=reading*5.0/1024;
Serial.print(voltage);
Serial.println("volts");
temperatureC=(voltage -0.5)*100;
Serial.println("Temperature is:'");
Serial.print(temperatureC);
Serial.println("degrees C");
delay(1000);
}
```

Second connect your circuit at proteus program as following :



Example # 2 : PWM

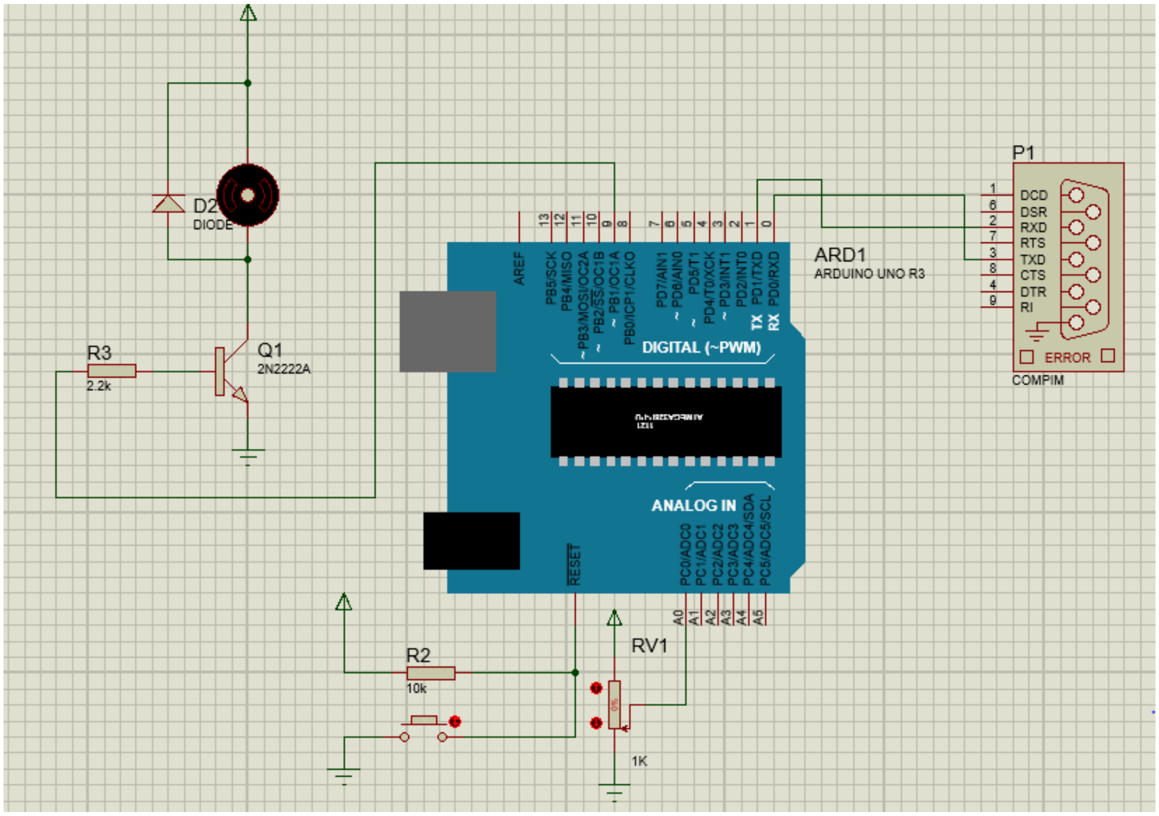
Arduino C :

```
int pwmPin=9;

void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(pwmPin,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
int potValue=analogRead(A0);
int newpotValue=map(potValue,0,1023,0,255);
analogWrite(pwmPin,newpotValue);
}
```

Second proteus :



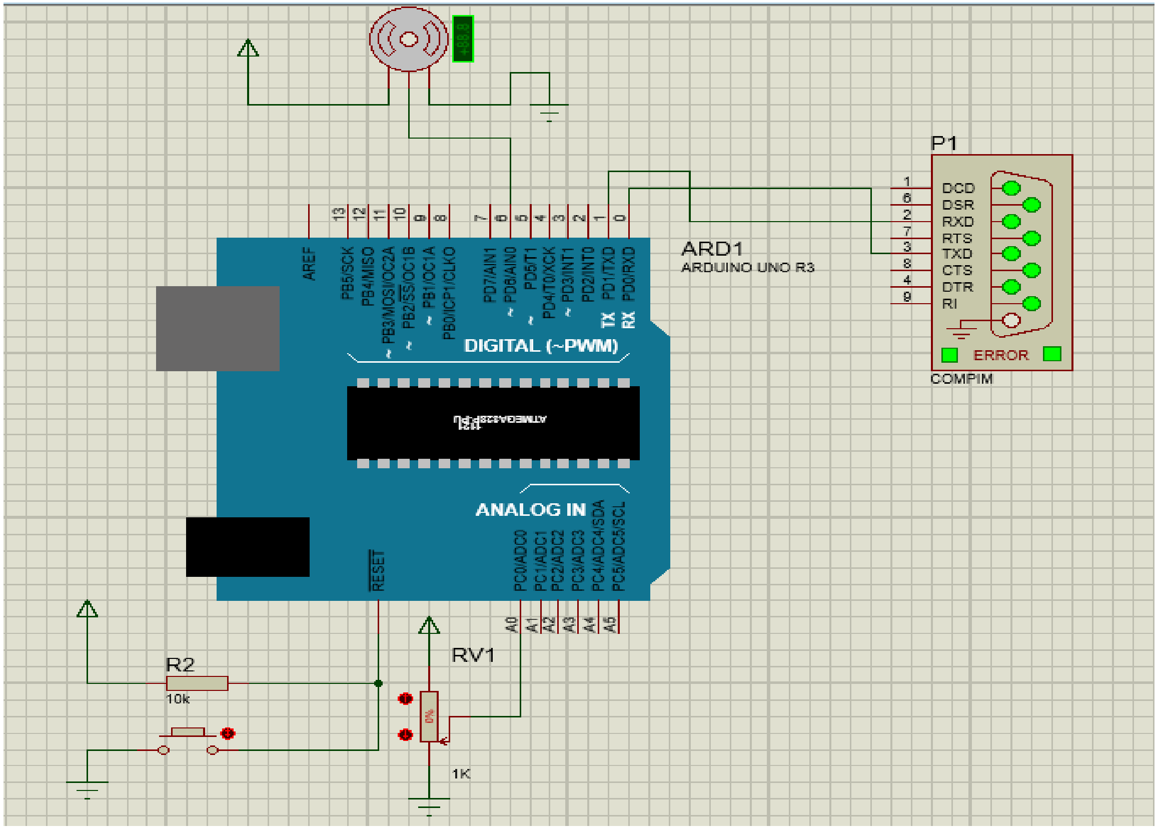
Example # 3 : Servo motor

Arduino C :

```
#include <Servo.h>
Servo myServo ;
void setup() {
  // put your setup code here, to run once:
myServo.attach(6);
}

void loop() {
  // put your main code here, to run repeatedly:
int potValue=analogRead(A0);
int angleValue =map(potValue,0,1023,0,180);
myServo.write(angleValue);
delay(10);
}
```

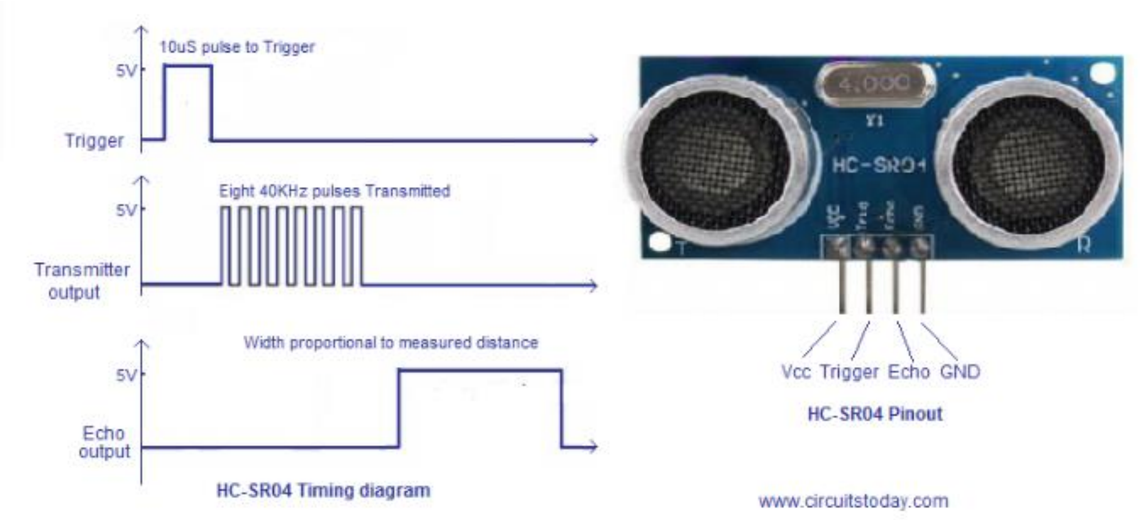
Second proteus :

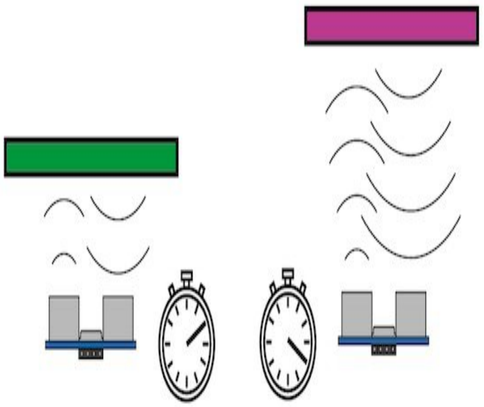


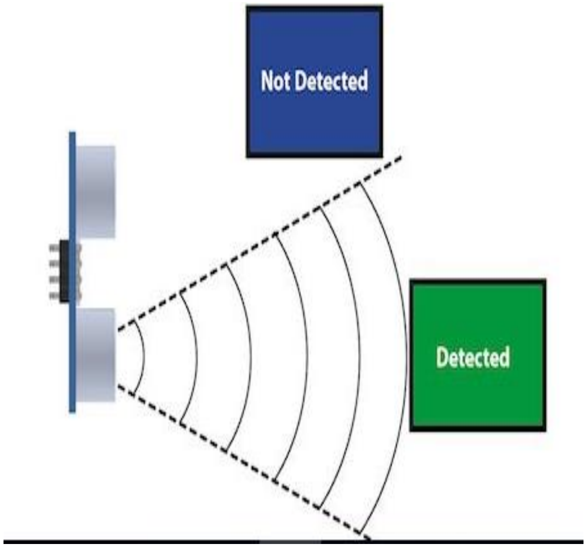
Example # 4 : Ultra sonic

HC-SR04 ultrasonic ranging module.

HC-SR04 is the ultrasonic ranging module used here. HC-SR04 consists of an ultrasonic transmitter, receiver and necessary electronics for making it a stand alone system. The operating principle is very simple. It sends 8 pulses of 40KHz sound waves, and picks up the reflected wave. The time lag between the transmission and reception is measured and the distance is calculated from the equation D=TS/2. Where D is the distance, T is the time lag and S is the speed of sound. The output of the HC-SR04 will be a pulse whose width is proportional to the distance. From the datasheet, width of the output signal will be 58uS for a distance of 1cm. What we need to do is to just send a 10uS wide trigger signal to the trigger pin of the module and wait for the output pulse at the echo pin of the module. Timing diagram and pin out of the ultrasonic module is given below.



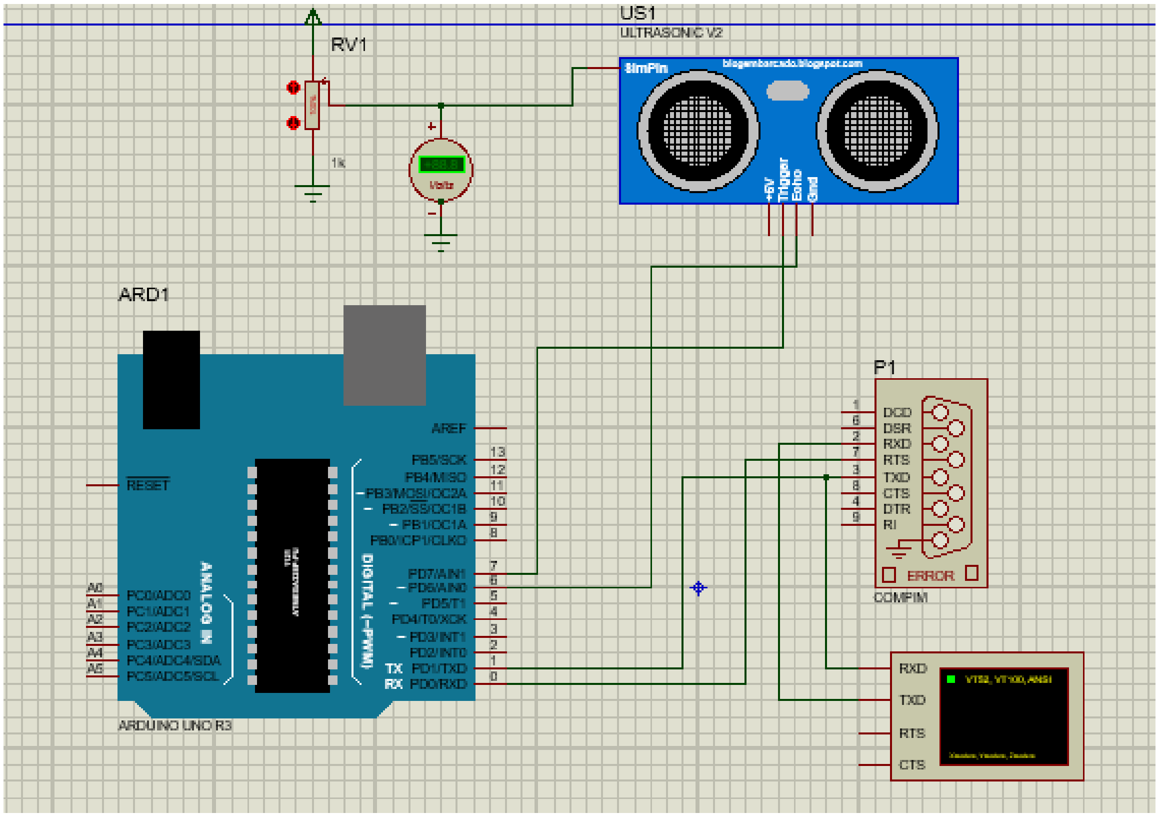




Arduino C :

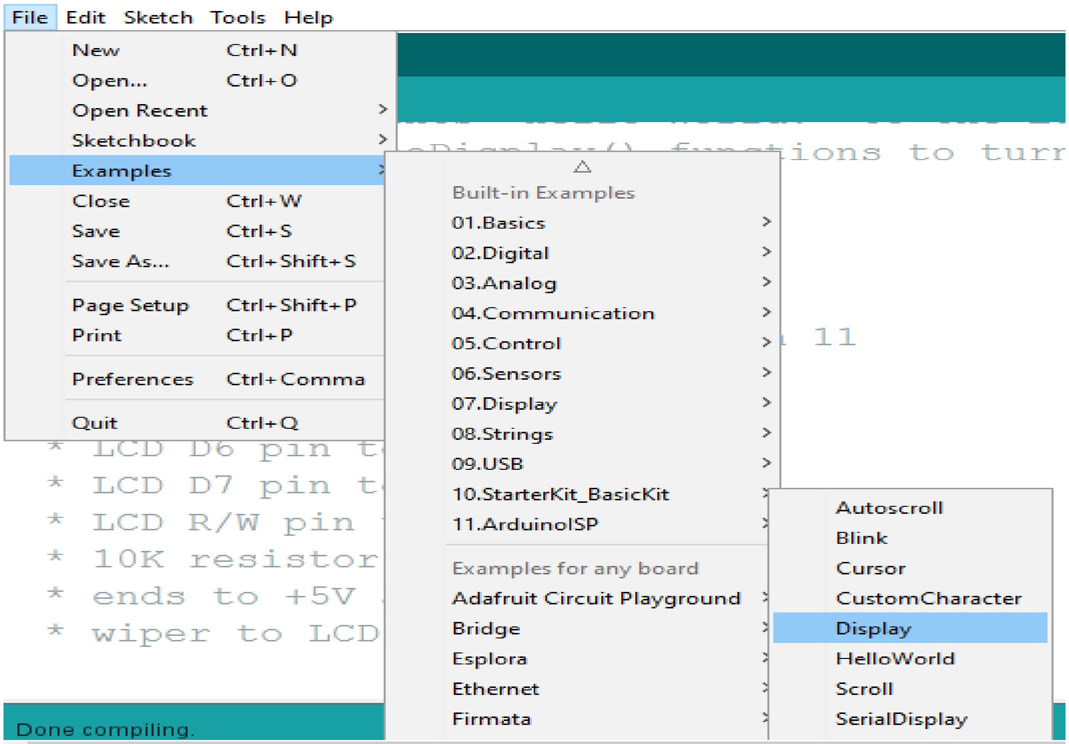
```
const int pingPin = 7; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 6; // Echo Pin of Ultrasonic Sensor
void setup()
{
Serial.begin(9600); // Starting Serial Terminal
}
void loop()
{
long duration, inches, cm;
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(10);
digitalWrite(pingPin, LOW);
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
}
long microsecondsToInches(long microseconds)
{
return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
return microseconds / 29 / 2;
}
```

Second proteus :



Example # 5 :LCD to arduino

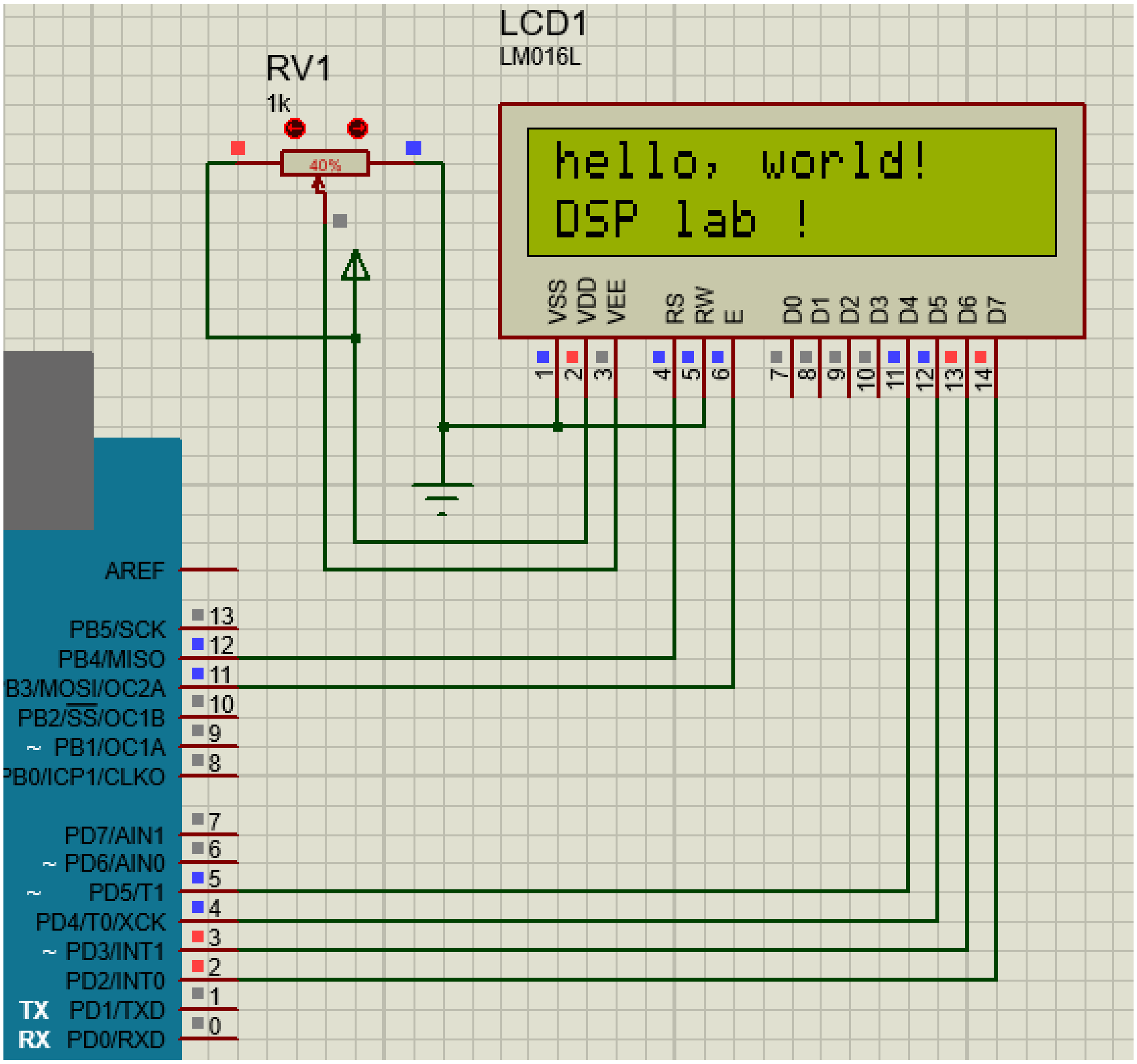
To get more information go to example as shown :



Arduino C :

```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
  lcd.setCursor(0,1) ;
  lcd.print("DSP lab ! ");
}
void loop() {
  // Turn off the display:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
}
```

Second proteus :



```
hello, world!
DSP lab !
```

Exercise :

1- Do all example in your report (arduino C and proteus)
2- Simulate code that make Traffic Light (red , yellow, green)