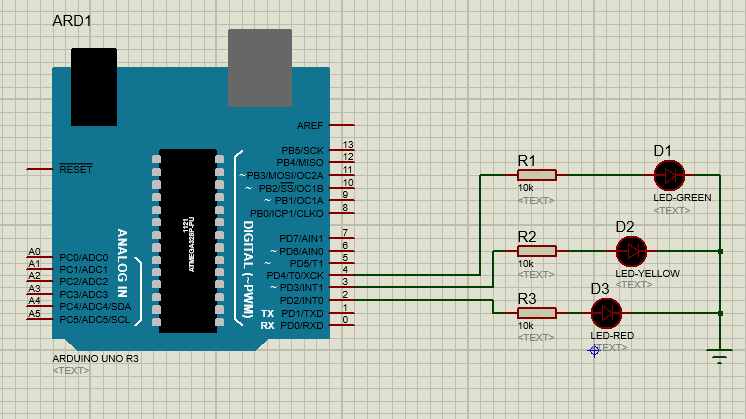
**Embedded system Hardware training**

**LED Blinking**



**Arduino Instruction**

#define GreenLed 4

#define YellowLed 3

#define RedLed 2

void setup() {

// put your setup code here, to run once:

pinMode(GreenLed, OUTPUT);

pinMode(YellowLed, OUTPUT);

pinMode(RedLed, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(GreenLed, HIGH);

digitalWrite(YellowLed, LOW);

digitalWrite(RedLed, LOW);

delay(500);

digitalWrite(GreenLed, LOW);

digitalWrite(YellowLed, LOW);

digitalWrite(RedLed, HIGH);

delay(500);

digitalWrite(GreenLed, LOW);

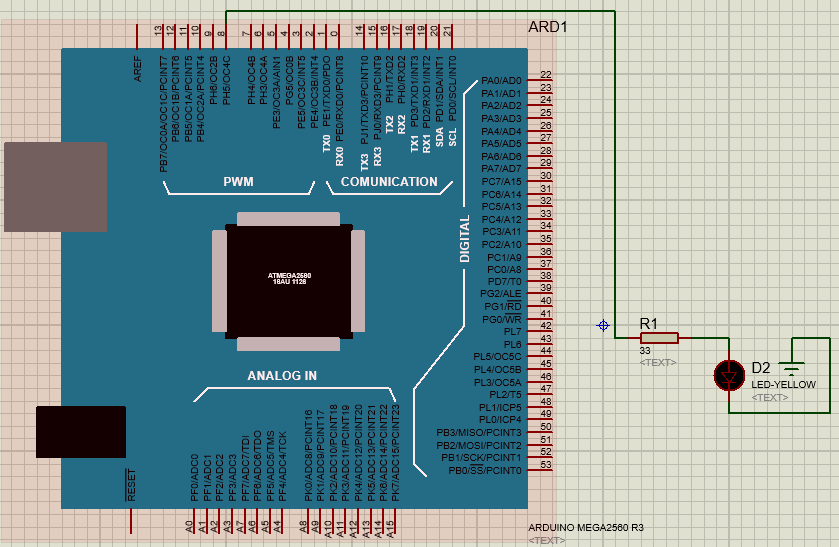
digitalWrite(YellowLed, HIGH);

digitalWrite(RedLed, LOW);

delay(300);

}

**LED fade**



**Arduino Instruction**

int led = 8; // the PWM pin the LED is attached to

int brightness = 0; // how bright the LED is

int fadeAmount = 5; // how many points to fade the LED by

void setup() {

// put your setup code here, to run once:

pinMode(led, OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

// set the brightness of pin 9:

analogWrite(led, brightness);

// change the brightness for next time through the loop:

brightness = brightness + fadeAmount;

// reverse the direction of the fading at the ends of the fade:

if (brightness == 0 || brightness == 255)

{

fadeAmount = -fadeAmount ;

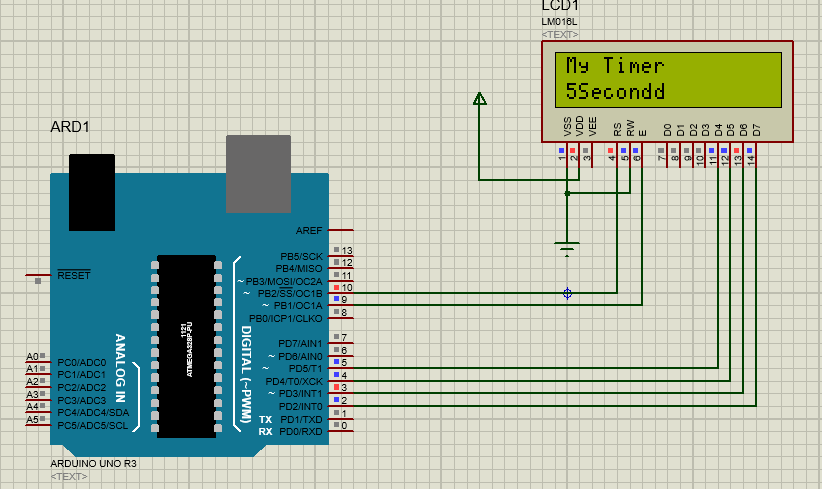
}

// wait for 30 milliseconds to see the dimming effect

delay(300);

}

**Arduino Timer With LCD**



**Arduino Instruction**

#include <LiquidCrystal.h>

LiquidCrystal LCD(10,9,5,4,3,2);

int mycounter=0;

void setup() {

// put your setup code here, to run once:

LCD.begin(16,2);

LCD.setCursor(0,0);

LCD.print("My Timer");

}

void loop() {

// put your main code here, to run repeatedly:

for(mycounter=1; mycounter<=10; mycounter=mycounter+1){

LCD.setCursor(0,1);

LCD.print(mycounter);

LCD.print("Second");

delay(100);

}

for(mycounter=10; mycounter>=1; mycounter=mycounter-1){

LCD.setCursor(0,1);

LCD.print(mycounter);

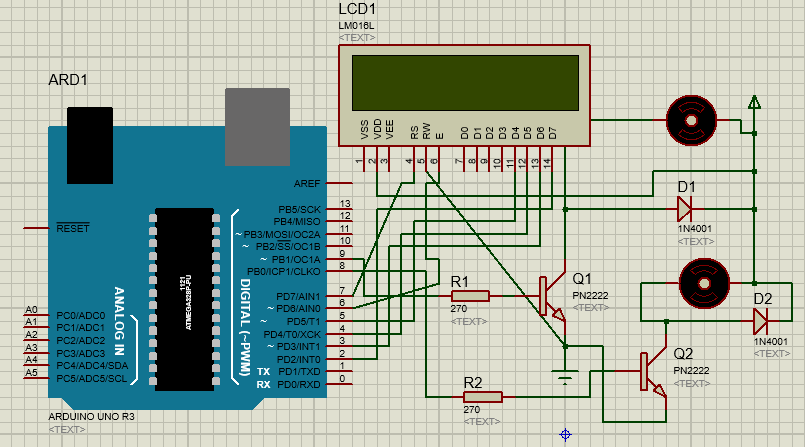
LCD.print("Second");

delay(100);

}

}

**Arduino with DC Motor**



**Arduino instruction**

#include <LiquidCrystal.h>

LiquidCrystal LCD(7,6,5,4,3,2);

#define motorpin 9

#define motorp 8

void setup() {

// put your setup code here, to run once:

pinMode(motorpin,OUTPUT);

pinMode(motorp,OUTPUT);

LCD.begin(16,2);

LCD.setCursor(0,0);

LCD.print("Motor Detection");

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(motorpin,LOW);

delay(100);

digitalWrite(motorpin,HIGH);

digitalWrite(motorp,LOW);

LCD.setCursor(0,1);

LCD.print("MOTOR1 Start");

delay(800);

digitalWrite(motorpin,LOW);

LCD.setCursor(0,1);

LCD.print("M1\_Stop");

delay(100);

digitalWrite(motorp,HIGH);

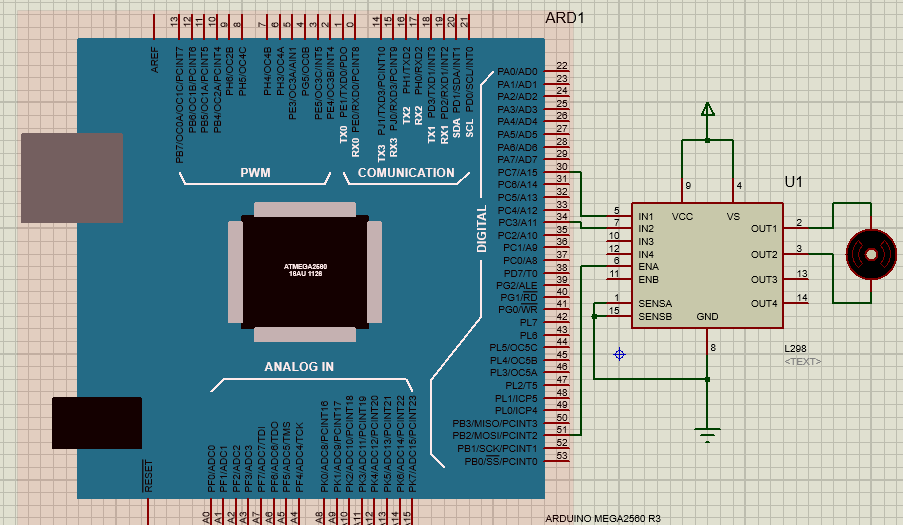
LCD.setCursor(0,1);

LCD.print("MOTOR2 Start");

delay(800);

}

**DC Motor Direction Control**



**Arduino instruction**

#define in1 30

#define in2 34

#define ena 51

void setup() {

// put your setup code here, to run once:

pinMode(in1,OUTPUT);

pinMode(in2,OUTPUT);

pinMode(ena,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

//For Clock wise motion , in\_1 = High , in\_2 = Low

digitalWrite(in1,HIGH) ;

digitalWrite(in2,LOW) ;

analogWrite(ena,255) ;

//Clockwise for 3 secs

delay(3000) ;

//For brake

digitalWrite(in1,HIGH) ;

digitalWrite(in2,HIGH) ;

delay(1000) ;

//For Anti Clock-wise motion - IN\_1 = LOW , IN\_2 = HIGH

digitalWrite(in1,LOW) ;

digitalWrite(in2,HIGH) ;

delay(3000) ;

//For brake

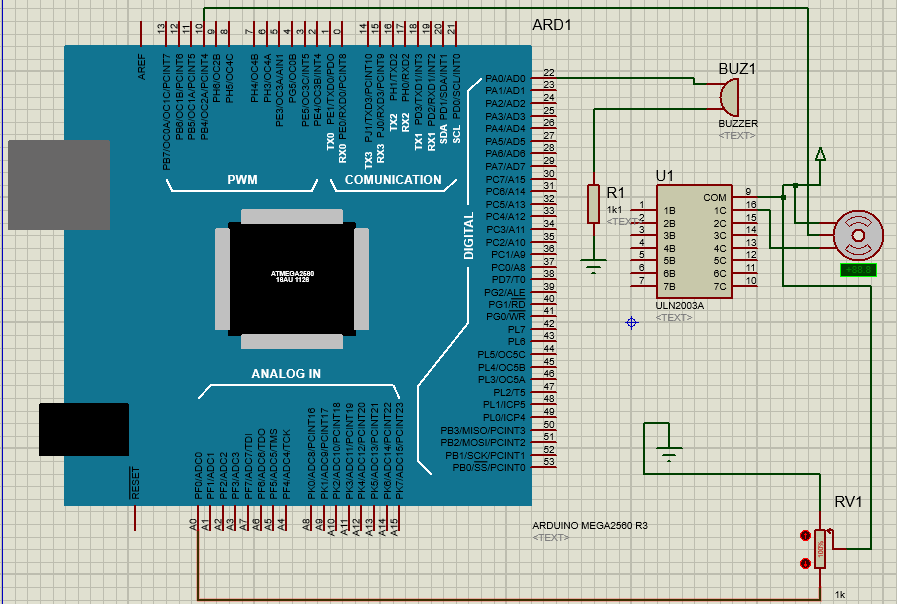
digitalWrite(in1,HIGH) ;

digitalWrite(in2,HIGH) ;

delay(1000) ;

}

**Servo Motor Direction Control**



**Arduino instruction**

#include <Servo.h>

#define b 22

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer

int val=0; // variable to read the value from the analog pin

int servodelay=25;

void setup() {

// put your setup code here, to run once:

myservo.attach(10);

pinMode(b,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

for(val=15; val<=170; val=val+1){

myservo.write(val); // sets the servo position according to the scaled value

delay(servodelay);

}

for(val=180; val>=15; val=val-1){

myservo.write(val); // sets the servo position according to the scaled value

delay(servodelay);

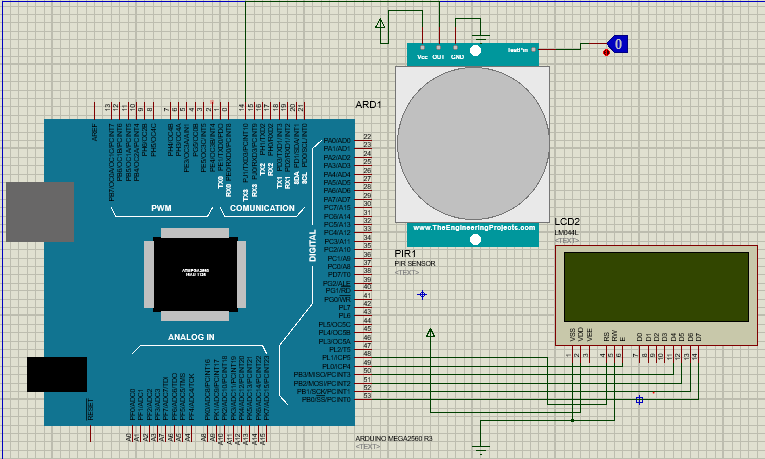
}

digitalWrite(b,HIGH);

delay(15);

}

**Motion detection with PIR**



**Arduino instruction**

#include <LiquidCrystal.h>

LiquidCrystal LCD(48,49,50,51,52,53);

void setup() {

// put your setup code here, to run once:

LCD.begin(20,4);

}

void loop() {

// put your main code here, to run repeatedly:

int pirpin=digitalRead(14);

LCD.setCursor(0,0);

LCD.print("Study PIR");

if(pirpin==LOW){

LCD.setCursor(0,1);

LCD.print("HELLO");

delay(2);

LCD.clear();

}

else if(pirpin==HIGH){

LCD.setCursor(0,1);

LCD.print(" WLCOME");

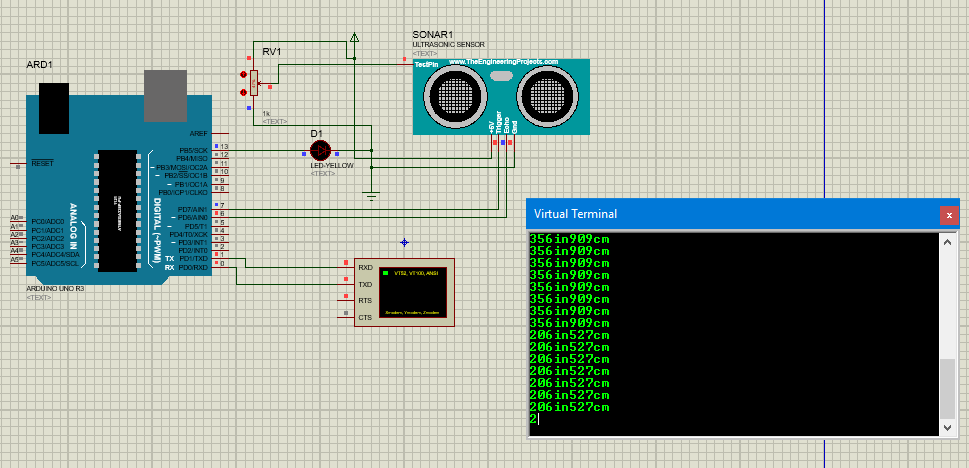
delay(2);

LCD.clear();

}

}

**Distance measurement using Ultrasonic sensor**



**Arduino instruction**

#define ledpin 13

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

pinMode(echoPin, INPUT);

pinMode(pingPin, OUTPUT);

pinMode(ledpin, OUTPUT);

}

void loop()

{

long duration, inches, cm;

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(10);

digitalWrite(pingPin, LOW);

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;

}

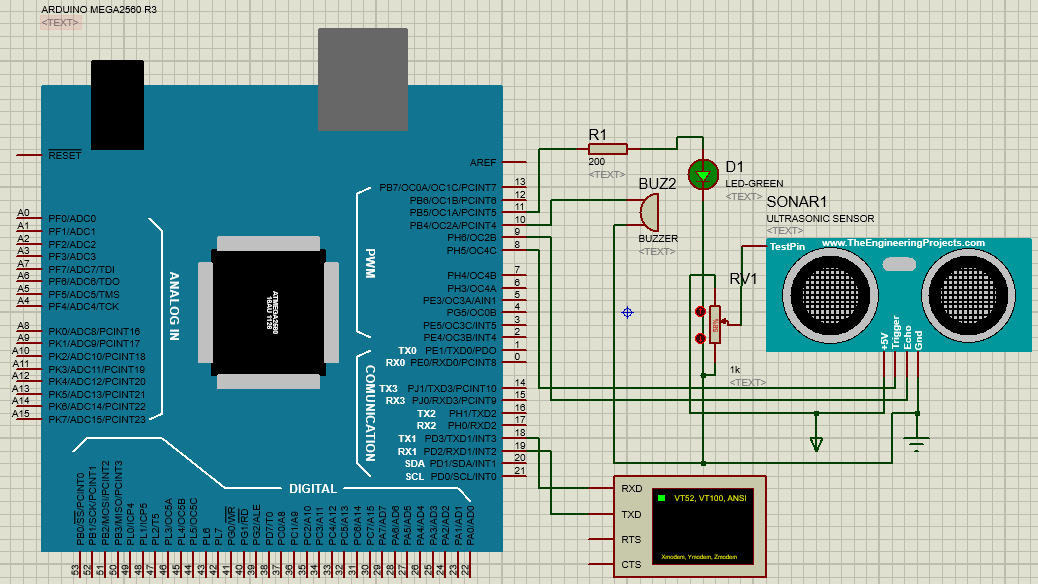
long microsecondsToCentimeters(long microseconds)

{

return microseconds / 29 / 2;

}

**Alarm system using Ultrasonic Sensor and Buzzer**



**Arduino Instruction**

#define trigpin 8

#define echopin 9

#define buzzer 10

#define ledpin 11

long duration;

int distance;

int alarm;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(8,OUTPUT);

pinMode(9,INPUT);

pinMode(10,OUTPUT);

pinMode(11,OUTPUT);

}

void loop() {

// put your main code here, to run repeatedly:

digitalWrite(trigpin,LOW);

delayMicroseconds(2);

digitalWrite(trigpin,HIGH);

delayMicroseconds(20);

digitalWrite(trigpin,LOW);

duration=pulseIn(echopin,HIGH);

distance=duration\*0.034/2;

alarm=distance;

if(alarm<=5){

digitalWrite(buzzer,HIGH);

digitalWrite(ledpin,HIGH);

}

else{

digitalWrite(buzzer,LOW);

digitalWrite(ledpin,LOW);

}

Serial.print("Distance of object: ");

Serial.println(distance);

}

**Control Servo Motor with Joystick**

#include <Servo.h>

Servo joyservo;

int joyX= 0;

int joyY=1;

int joystick\_value;

void setup() {

// put your setup code here, to run once:

joyservo.attach(14);

}

void loop() {

// put your main code here, to run repeatedly:

joystick\_value=analogRead(joyX);

joystick\_value=map(joystick\_value, 0, 1023, 0, 180);

joyservo.write(joystick\_value);

}