Indian Institute of Information Technology, Una H.P.

PRACTIUM II

Light Sensing Alarm System

Electronics LAB Report File

By – Prakhar Jaiswal (21229)

Submitted To –

Dr. Gurpreet Kaur,

Dr. Naman Garg,

Dr. Ankur

Aim: - Introduction to Arduino and making a new project by interfacing Arduino with different components discussed below.

Table of Contents: -

ARDUINO BOARD

Arduino is an open-source electronic prototyping platform based on flexible easy to use hardware and software. Arduino boards are able to read inputs- light on a sensor, a finger on a bottom, or a twitter message and turn it out into an output activating a motor, turning on an LED, publishing something online. Further, 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 board Explanation: -

1. The Arduino Uno can be powered by USB cable or directly supplying 9-12v from the barrel jack. The circuitry operates at 5v dc which in case input more than that is regulated with the help of 7805 voltage regulator. The 7805-voltage regulator is used regulate the voltage supplied to the Arduino board and manage it through processor and other elements.

2. **Voltage pins:** There are 3v and 5v output voltage pins and two ground. Most of the Arduino components operate at 5v or 3.3v and so can be powered with these pins. There are several ground ports which can be used to give ground to your circuit and components. There is a VIN pin which can be used to power the Arduino UNO from an external source.

3. **Crystal Oscillator:** There are certain case when the processor has to deal with time-signal issues, in order to balance it the crystal oscillator is used. The crystal oscillator is the only way the Arduino is able to calculate the time. There is a number printed on the top of the crystal. The number indicates the frequency of the crystal, in most of them the frequency is 16 MHZ or 16,000,000 hertz.

4. **Digital Input/output pins:** Arduino UNO board has 14 digital input/ output pins, out of which contains 6 PWM (pulse width modulation).

5. **Analog Input/ Output pins**: The Arduino UNO board has 6 Analog input and output pins from A0 to A5. These pins can read Analog signals like temperature, proximity, humidity and converts into digital values.

6. **Power indicator LED**: These LED indicates that the board is powered up correctly or not.

7. **Reset Button:** Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded in Arduino board.

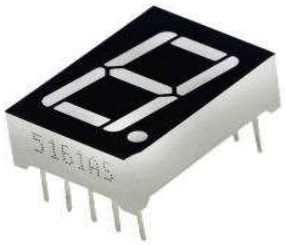
8. **TX(Transmitter) and RX(Receiver) Pins:** These pins are used for serial communication and the corresponding LED glowing indicated if the data is being sent by TX and if the data is being received by RX.

Arduino code:

1. Setup: This is called only when the Arduino is powered on or reset. loop: The loop function runs continuously till the device is powered off.
2. Pin Mode: It is used to target a pin for input or output.
3. Digital Write (13, LOW): It makes the output voltage on pin 13 and 0 V.
4. Libraries: Libraries are a collection of code that makes it easy for user to connect to a sensor, display, module, etc.

Seven Segment

Seven segments are used to display only numbers in the Arduino. The seven-segment display has seven LEDs arranged in the shape of number eight. They are easy to use and cost effective.



Seven segment displays are of two types:

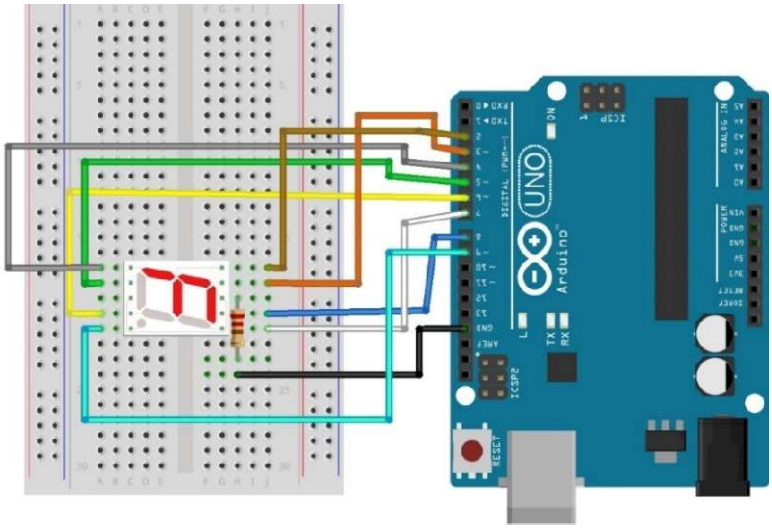
- Common anode.
- Common cathode.

The Internal structure of both types is nearly the same. The difference is the polarity of the LEDs and common terminal.

1. **Common Cathode:** - In a common cathode seven-segment display all seven LEDs plus a dot LED have the cathodes connected to pins 3 and pin 8. To use this display, we need to connect GROUND to pin 3 and pin 8 and connect +5V to the other pins to make the individual segments light up. We can check this easily by multi meter.
2. **Common Anode: -** The common anode display is the exact opposite. In a common-anode display, the positive terminal of all the eight LEDs is connected together and then connected to pin 3 and pin 8. To turn on an individual segment, you ground one of the pins. The following diagram shows the internal structure of the common-anode seven-segment display.

Seven Segment with Arduino

Circuit Diagram: -



Arduino Code

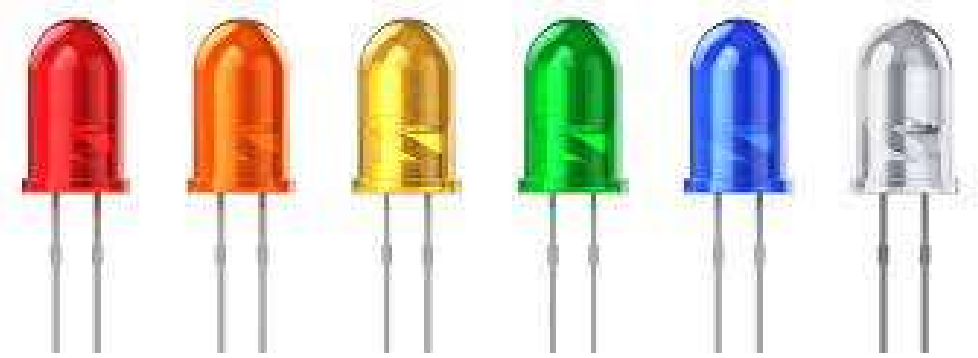
```
#define A 8
#define B 7
#define C 6
#define D 5
#define E 4
#define F 3
#define G 2
#define DP 9 // decimal
#define common_cathode 0
#define common_anode 1
bool segMode = common_cathode; // set this to your segment type, my segment is common_ca
thode
int seg[] {A,B,C,D,E,F,G,DP}; // segment pins
byte chars = 35; // max value in the array "Chars"
byte Chars[35][9] {
            {'0',1,1,1,1,1,1,0,0},//0
            {'1',0,1,1,0,0,0,0,0},//1
            {'2',1,1,0,1,1,0,1,0},//2
            {'3',1,1,1,1,0,0,1,0},//3
            {'4',0,1,1,0,0,1,1,0},//4
            {'5',1,0,1,1,0,1,1,0},//5
            {'6',1,0,1,1,1,1,1,0},//6
            {'7',1,1,1,0,0,0,0,0},//7
            {'8',1,1,1,1,1,1,1,0},//8
            {'9',1,1,1,1,0,1,1,0},//9
            {'a',1,1,1,0,1,1,1,0},//A/10
            {'b',0,0,1,1,1,1,1,0},//b/11
            {'c',1,0,0,1,1,1,0,0},//C/12
            {'d',0,1,1,1,1,0,1,0},//d/13
            {'e',1,0,0,1,1,1,1,0},//E/14
            {'f',1,0,0,0,1,1,1,0},//F/15
            {'g',1,0,1,1,1,1,0,0},//G/16
            {'h',0,1,1,0,1,1,1,0},//H/17
            {'i',0,0,0,0,1,1,0,0},//I/18
            {'j',0,1,1,1,1,0,0,0},//J/19
            {'l',0,0,0,1,1,1,0,0},//L/20
            {'n',0,0,1,0,1,0,1,0},//n/21
            {'o',0,0,1,1,1,0,1,0},//o/22
            {'p',1,1,0,0,1,1,1,0},//P/23
            {'q',1,1,1,0,0,1,1,0},//q/24
            {'r',0,0,0,0,1,0,1,0},//r/25
            {'s',1,0,1,1,0,1,1,0},//S/26
            {'t',0,0,0,1,1,1,1,0},//t/27
            {'u',0,1,1,1,1,1,0,0},//U/28
            {'y',0,1,1,1,0,1,1,0},//y/29
            {'-',0,0,0,0,0,0,1,0},//-/30
            {'.',0,0,0,0,0,0,0,1},//./31
            {']',1,1,1,1,0,0,0,0},//]/32
            {'[',1,0,0,1,1,1,0,0},//[/33
            {'_',0,0,0,1,0,0,0,0},//_/34
            };
void setup() {
```

```
void setup() {
// set segment pins as OUTPUT
pinMode(seg[0],OUTPUT);
pinMode(seg[1],OUTPUT);
pinMode(seg[2],OUTPUT);
pinMode(seg[3],OUTPUT);
pinMode(seg[4],OUTPUT);
pinMode(seg[5],OUTPUT);
pinMode(seg[6],OUTPUT);
pinMode(seg[7],OUTPUT);
}
void setState(bool mode) //sets the hole segment state to "mode"
{ for(int i = 0;i<=6;i++)
{
digitalWrite(seg[i],mode);
}
}
void Print(char Char) // print any character on the segment ( Note : you can't use capit
al characters )
{
int charNum = -1;// set search resault to -1
setState(segMode);//turn off the segment
for(int i = 0; i < chars ;i++){//search for the enterd character
if(Char == Chars[i][0]){//if the character found
charNum = i;//set the resault number into charNum ( because this function prints the cha
racter using it's number in the array )
}
}
if(charNum == -1 )// if the character not found
{
for(int i = 0;i <= 6;i++)
{
digitalWrite(seg[i],HIGH);
delay(100);
digitalWrite(seg[i],LOW);
}
for(int i = 0;i <= 2;i++)
{
delay(100);
setState(HIGH);
delay(100);
setState(LOW);
}
}else // else if the character found print it
{
for(int i = 0;i<8;i++)
{digitalWrite(seg[i],Chars[charNum][i+1]);
}
}
}
```

```
void Print(int num) // print any number on the segment
{
 setState(segMode);//turn off the segment
if(num > chars || num < 0 )// if the number is not declared
{
 for(int i = 0;i <= 6;i++)
 {
 digitalWrite(seg[i],HIGH);
 delay(100);
 digitalWrite(seg[i],LOW);
 }
 for(int i = 0;i <= 2;i++)
{
 delay(100);
 setState(HIGH);
 delay(100);
 setState(LOW);
}
}else // else if the number declared, print it
{
 if(segMode == 0){ //for segment mode
for(int i = 0;i<8;i++)
   {digitalWrite(seg[i],Chars[num][i+1]);
   }
}
else{
 for(int i = 0;i<8;i++)
   {digitalWrite(seg[i],!Chars[num][i+1]);
   }
}
}
}
void loop() {
  for(int i = 0;i < chars;i++) //print
 {
   Print(i);
  delay(1000);
 }
 //Print(number or character); // print any number or character on the segment ( Note :
you can't use capital characters )
 //setState(state); //sets the hole segment state to "mode"
}
```

Light Emitting Diode (LED)

A Light Emitting Diode (LED) is a semiconductor device, which can emit light when an electric current passes through it. To do this, holes from p-type semiconductors recombine with electrons from n-type semiconductors to produce light

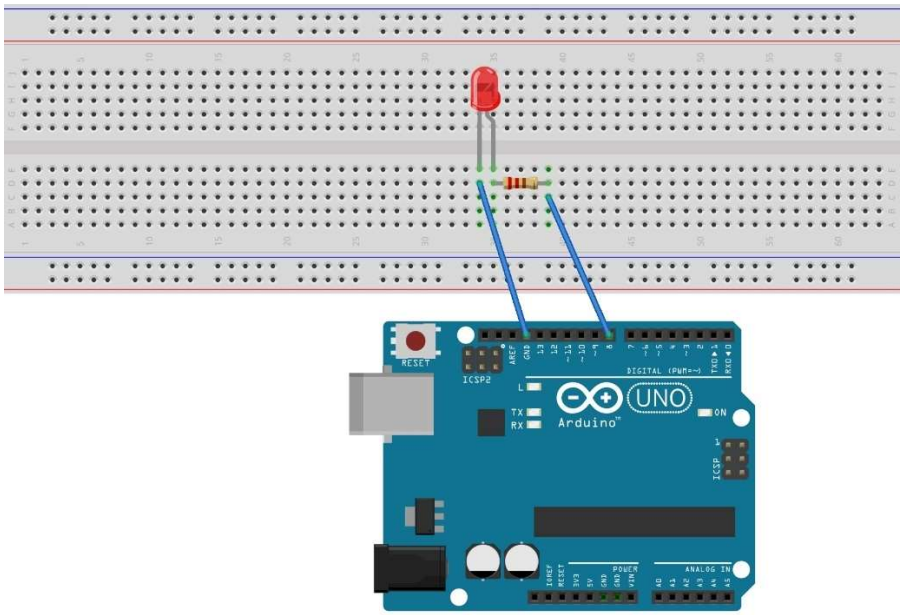


Light-emitting diodes are heavily doped p-n junctions. Based on the semiconductor material used and the amount of doping, an LED will emit a colored light at a particular spectral wavelength when forward biased.

LED with Arduino: -

The aim of this project is to blink the LED in a smarter way using Arduino. The blinking simply means the ON and OFF of the LED, it's simpler as that of switching the power supply ON and OFF.

Circuit Diagram: -



Arduino Code: -

```
int ledpin = 13;                          //set pin no of ardinuo to which le
d is connected
void setup() {
  pinMode(ledpin, OUTPUT);                // initialize digital pin LED_BUILT
IN as an output
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledpin, LOW );    // turn the LED off by making the voltag
e LOW
  delay(1000);
  digitalWrite(ledpin, HIGH);    // turn the LED ON by making the voltage
HIGH
  delay(1000);
  }
```

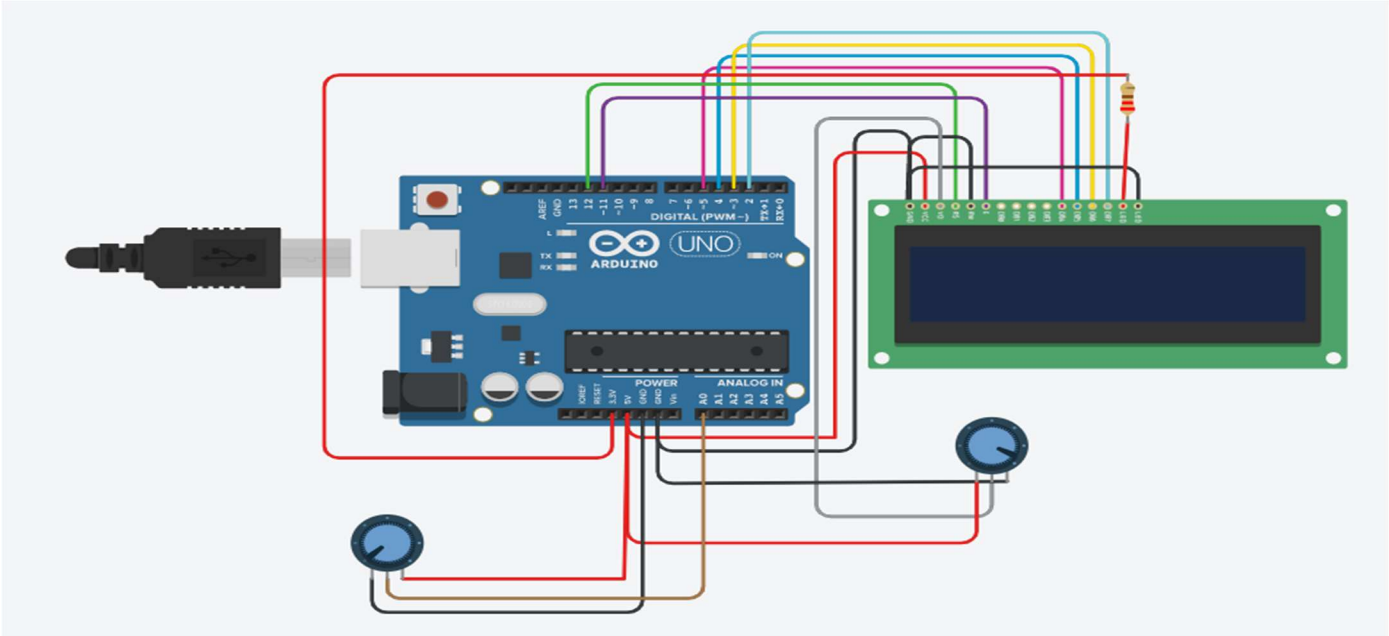
LCD display

An LCD character display is a unique type of display that can only output individual ASCII characters with fixed size. Using these individual characters then we can form a text.

The number of the rectangular areas define the size of the LCD. The most popular LCD is the 16×2 LCD, which has two rows with 16 rectangular areas or characters. Of course, there are other sizes like 16×1, 16×4, 20×4 and so on, but they all work on the same principle. Also, these LCDs can have different background and text colour.

LCD with Arduino: -

Circuit Diagram: -



Arduino Code: -.

```
 lcd.begin(16, 2);

}

void loop() {


  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Hello Hackster");
  lcd.setCursor(0, 1);
  lcd.print("Value : ");
  lcd.setCursor(10, 1);
  lcd.print(analogRead(A0));
  Serial.println(analogRead(A0));
delay(500);
}
```

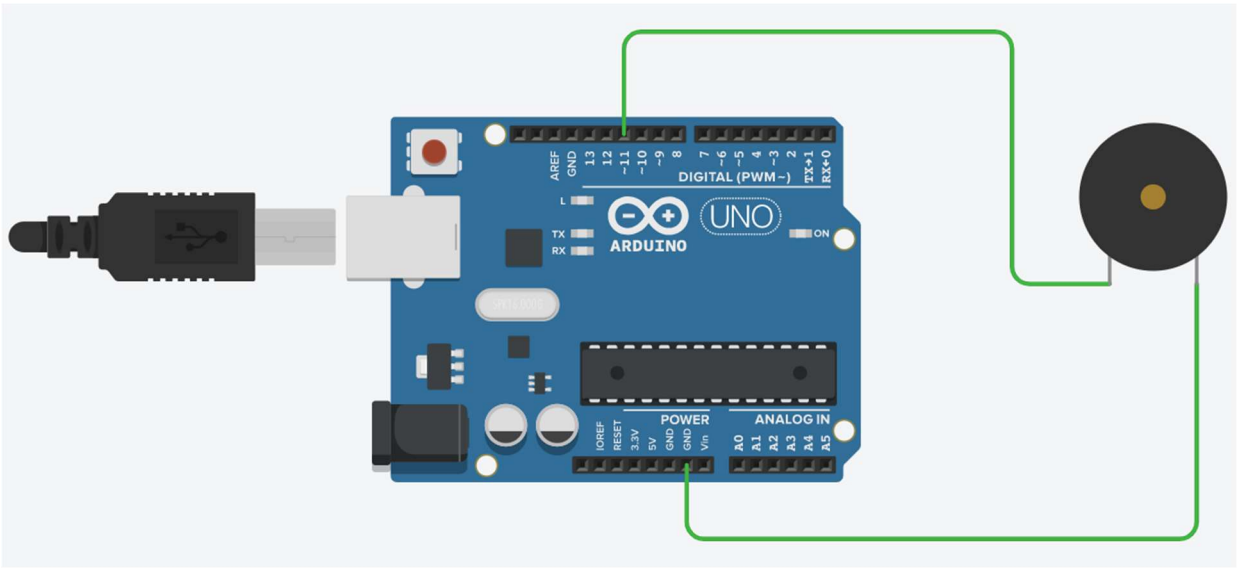
Buzzer

Buzzer is a kind of voice device that converts audio model into sound signal. It is mainly used to prompt or alarm. According to different design and application, it can produce music sound, flute sound, buzzer, alarm sound, electric bell and other different sounds.

Simply change the frequency of the voltage sent to the piezo and it will start generating sounds by changing shape very quickly.

Buzzer with Arduino: -

Circuit Diagram: -



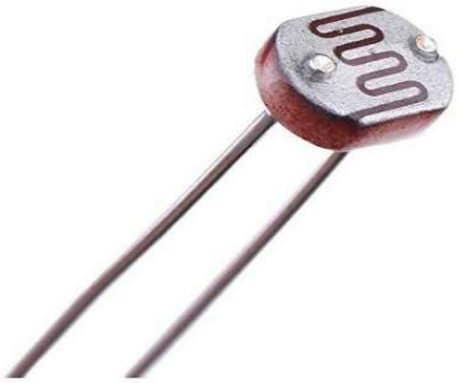
Arduino Code: -

```
void setup() {
pinMode(11,OUTPUT);
}

void loop() {
tone(11,200);
delay(500);
noTone(11);
 delay(500);
}
```

LDR

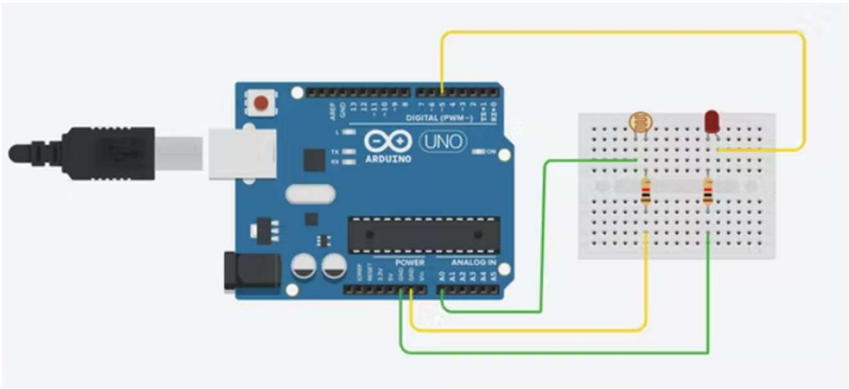
LDR (Light Dependent Resistor) as the name states is a special type of resistor that works on the photoconductivity principle means that resistance changes according to the intensity of light. Its resistance decreases with an increase in the intensity of light. It is often used as a light sensor, light meter, Automatic Street Lights, and in areas where we need to have light sensitivity. It is also called a Light Sensor.



It works on the principle of photoconductivity whenever the light falls on its photoconductive material, it absorbs its energy and the electrons of that photoconductive material that is in the valence band get excited and go to the conduction band and thus increasing the conductivity as per the increased in light intensity.

LDR with Arduino: -

Circuit Diagram: -



Arduino Code: -

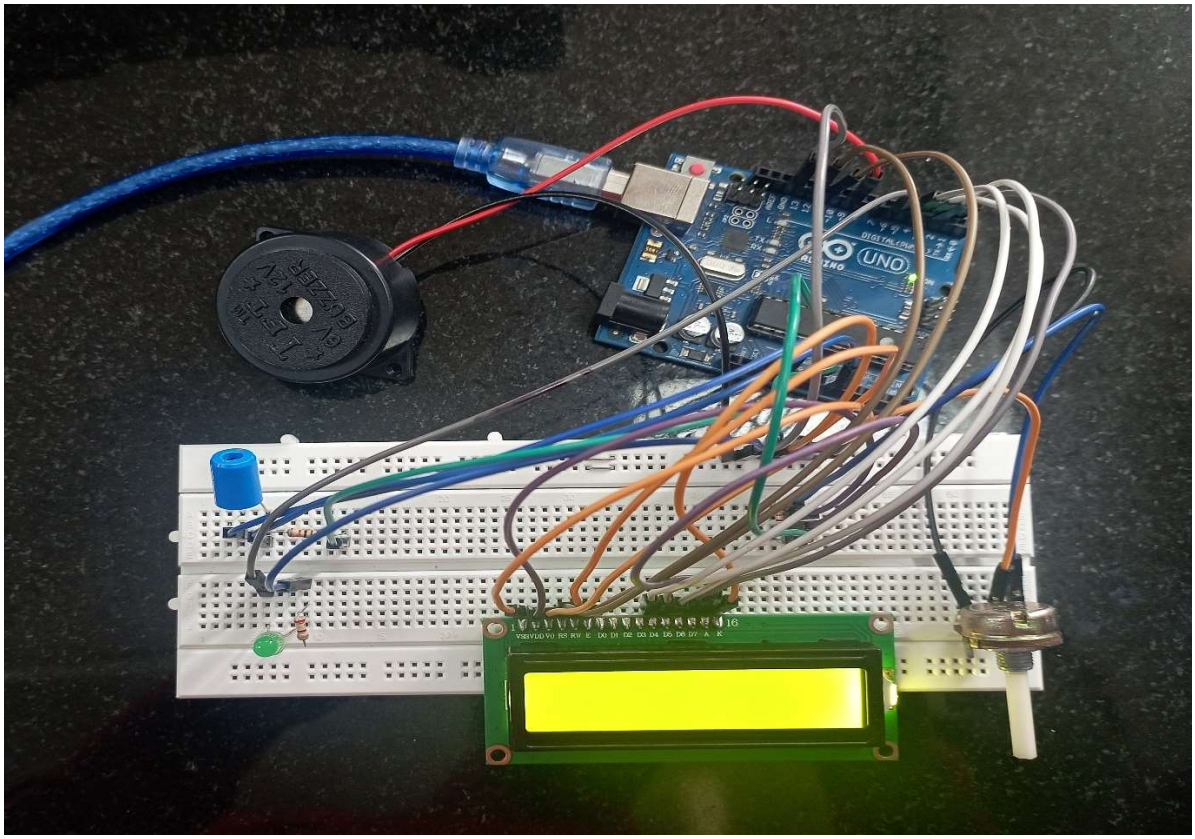
```
const int ledPin = 5; // digital pin 5
const int ldrPin = A0; // analog pin 0
void setup() { // The setup() function will only run once, after each po
werup or reset of the Arduino board.
Serial.begin(9600);
pinMode(ledPin, OUTPUT); // Here LED is determined as an output or an ind
icator.
pinMode(ldrPin, INPUT); // Here LDR sensor is determined as input.
}
void loop() { // Void loop is ran again and again and contains main code
.
int ldrStatus = analogRead(ldrPin);
if (ldrStatus <= 200) {digitalWrite(ledPin, HIGH); // If LDR senses darkness led
pin high that means led will glow.
Serial.print("Darkness over here,turn on the LED :");
Serial.println(ldrStatus);
}else {
digitalWrite(ledPin, LOW); // If LDR senses light led pin low that means
led will stop glowing.
Serial.print("There is sufficient light , turn off the LED : ");
Serial.println(ldrStatus);
}
}
```

Project Name: - Light Sensing Alarm System

Components: -

- LDR
- Buzzer
- LCD Display
- LED
- Arduino Board
- Jumper Wires
- Potentiometer
- Resistors

Circuit Diagram: -



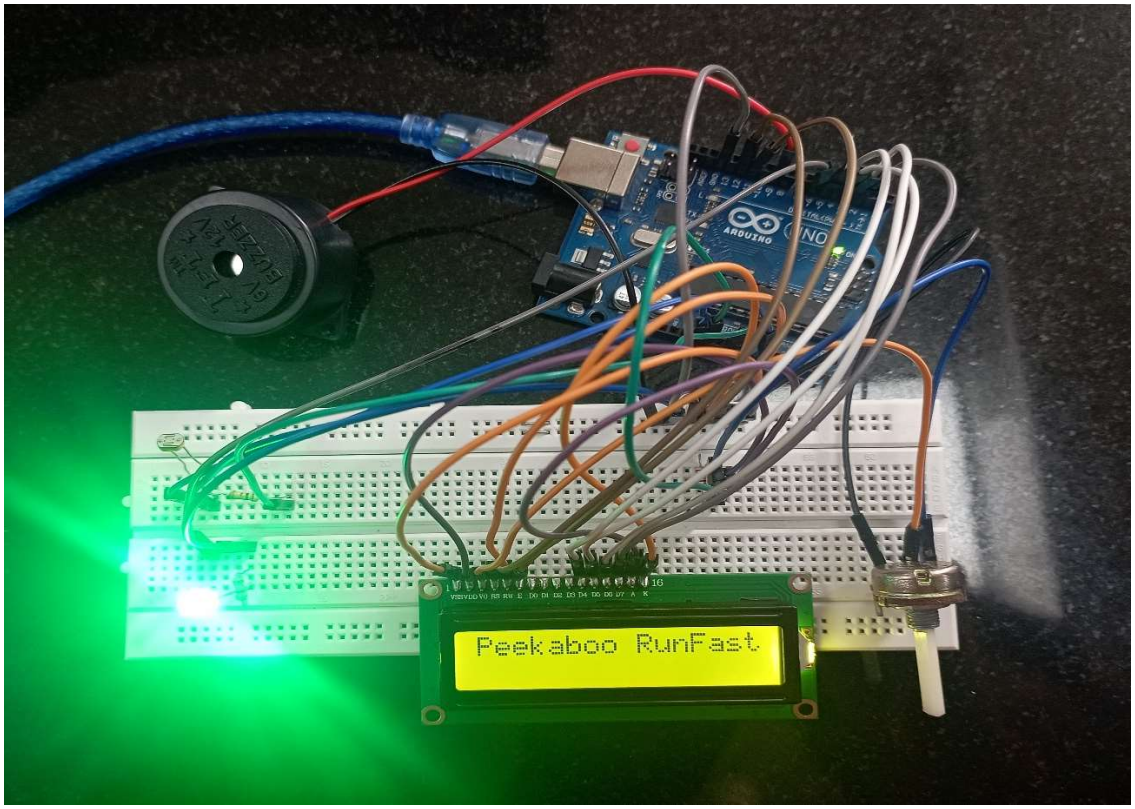
Arduino Code: -

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
int ldr = A0; //Set A0(Analog Input) for LDR.
int value = 0;
const int buzzer = 9; //buzzer to arduino pin 9
int ledpin = 6; //LED to pin 6

void setup() {
  Serial.begin(9600);
  pinMode(6, OUTPUT); // Set LED - pin 6 as an output
  lcd.begin(16, 2);
  pinMode(buzzer, OUTPUT); // Set buzzer - pin 9 as an output
}

void loop() {
  value = analogRead(ldr); //Reads the Value of LDR(light).
  //Serial.println("LDR value is :");//Prints the value of LDR to Serial Monitor.
  //Serial.println(value);
  if (value < 300)
  {
    digitalWrite(6, LOW);
    lcd.clear();
    noTone(buzzer);
  }
  else
  {
    digitalWrite(6, HIGH);
    tone(buzzer, 500);
    lcd.setCursor(0, 0);
    lcd.print("Peekaboo ");
    lcd.print("RunFast");
    //lcd.print(millis() / 1000);
    //delay(100);
    //noTone(buzzer);
    //delay(100);
  }
}
```

Observations: -



When there is no light is sensed by the LDR then there is no output delivered by LED, Buzzer and LCD Display. So, when there is some part of light is sensed by the LDR then LED starts to glow, LED displays a message and Buzzer starts to beep indicating that light has been sensed by the LDR.

Applications: -

- Home Security System.
- Dark Room Sensor.

Conclusion: -

Through this project we have created a Light sensing alarm system that can be used as Security alarm system for a Home or even for deaf people.