Home Security Alarm System

✓ **The problem to be solved by this project**

The need for home security alarm systems nowadays is a serious demand. As the number of crimes are increasing every day, there has to be something that will keep us safe. We are all aware of the high-end security systems present in the market but they are not easily available to everyone. I therefore intend to provide a solution by constructing a cost-efficient electronic system that has the capability of sensing the motion of the intruders and setting off the alarm. The basic idea behind this project is that all the bodies generate some heat energy in the form of infrared which is invisible to human eyes. But it can be detected by electronic motion sensor. The project involves the use of Arduino, motion sensor, buzzer, LCD

display and a simple program. The sensor detects any motion in its permissible range and triggers the alarm. It will also send the signal to Arduino which processes the signal and set off the alarm along with detection message on display. With this system we can easily set up a security alarm in our home for unwanted intruders

Introduction

I have designed an interesting and cheap home security alarm. This Gadget helps you to protect your

house from thieves. In this project we are going to use an Arduino Uno R3 Board, P.I.R Sensor module,

LCD and some other components. This Project can either powered with 9V Battery or with U.S.B of

your computer.

This is a basic motion-sensing alarm that detects when someone enters the area. When an intruder is

detected, it activates a siren. Our body generates heat energy in the form of infrared which is invisible

to human eyes. But it can be detected by electronic sensor. This type of sensor is made up of crystalline

material that is Pyroelectric. In this project, we are using P.I.R. Motion Sensor Module as an infrared

sensor that generates electric charge when exposed in heat and sends a signal to Arduino. According to

level of the infrared in front of sensor, Arduino displays the status on L.C.D and start buzzing speaker

and glows the L.E.D. A simple program is running on Arduino which checks sensor if anything is

moved or new object has been detected.

COMPONENTS REQUIRED

❖   Arduino Uno
❖   P.I.R Sensor Module
❖   L.C.D(16 X 2)
❖   9V Battery
❖   9V Battery Clip
❖   L.E.D
❖   Piezo Buzzer
❖   Breadboard
❖   Jumper Wires
❖   An USB Cable
❖   A Computer

WORKING

This system is a basic motion activated alarm. It is built around an Arduino Microcontroller. It is

connected to a **PIR** motion sensor, a buzzer, a resistor, and a pair of external terminals. The whole

system is battery powered so that it is easily portable. Once you have the code, you can connect all the

external parts. The easiest way to do this is with a breadboard. This will let you make temporary

connections to test everything out.

<u>Step 1:</u> Connecting the P.I.R sensor to Arduino:

1. Connect Vcc pin of **P.I.R** sensor to positive terminal of Arduino (5V).

2. Connect Gnd pin of **P.I.R** sensor to any ground pin of Arduino.

3. Connect out pin of **P.I.R** sensor to Pin no. -7 of Arduino.

Step 2: Connecting L.E.D and Piezo Buzzer to Arduino Connecting L.E.D

Connect Positive terminal (Longer Lead) Of L.E.D To Arduino Pin no. 13.

Connect Negative terminal (Shorter Lead) Of L.E.D To Any Ground Pin.

Connecting Piezo Buzzer

Connect Positive terminal (Red Wire) Of Buzzer to Arduino Pin no. 10.

Connect Negative terminal (Black Wire) Of Buzzer to Any Ground Pin.

Step 3: Connecting L.C.D to Arduino:

To wire your LCD screen to your Arduino, connect the following pins:

LCD RS pin to digital pin 12

LCD Enable pin to digital pin 11

LCD D4 pin to digital pin 5

LCD D5 pin to digital pin 4

LCD D6 pin to digital pin 3

LCD D7 pin to digital pin 2

Additionally, wire a 10K pot to +5V and GND, with its wiper (output) to LCD screens VO pin (pin3).

Step 4: Programming Arduino:

1. Download Arduino IDE 1.0.6 from https://www.arduino.cc/en/main/software.

2. Connect Your Arduino to your computer using USB Cable.

3. Open Arduino IDE, choose your correct board from Tools--Boards

4.Choose Your Correct Port from Tools--Serial Port

6. Copy the following sketch which appears in your Web Browser to your Arduino Sketch Page.

7. Click on Upload Icon or go to File—Upload

Step 5: Drill Holes in the Housing:

Next we need to drill a few holes in the housing so that we can mount all the parts. Start by using a ¼"

hole in one end of the housing. This will be where we mount the buzzer. Then use a ¾" hole saw to

drill a hole in the other side of the housing. This will be where we mount the motion sensor

Step 6: Glue the Motion Sensor and the Buzzer in Place

Apply a small amount of hot glue around the motion sensor where it lines up with the hole in the

housing. Then press the motion sensor into the hole. Apply more hot glue around the outside and hold

it in place until the glue cools. Then apply a small amount of hot glue to the face of the buzzer. Align

the hole in the buzzer with the hole in the housing and press it in place. Hold the buzzer in this position

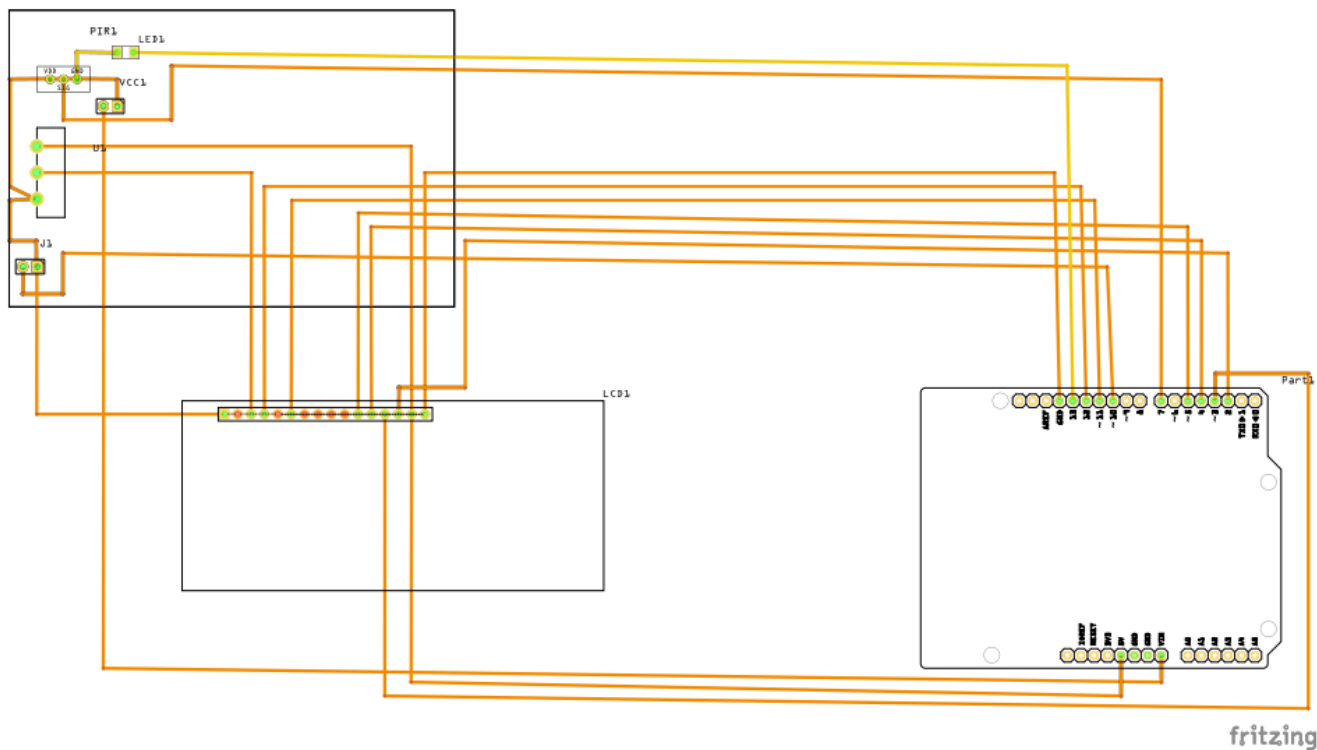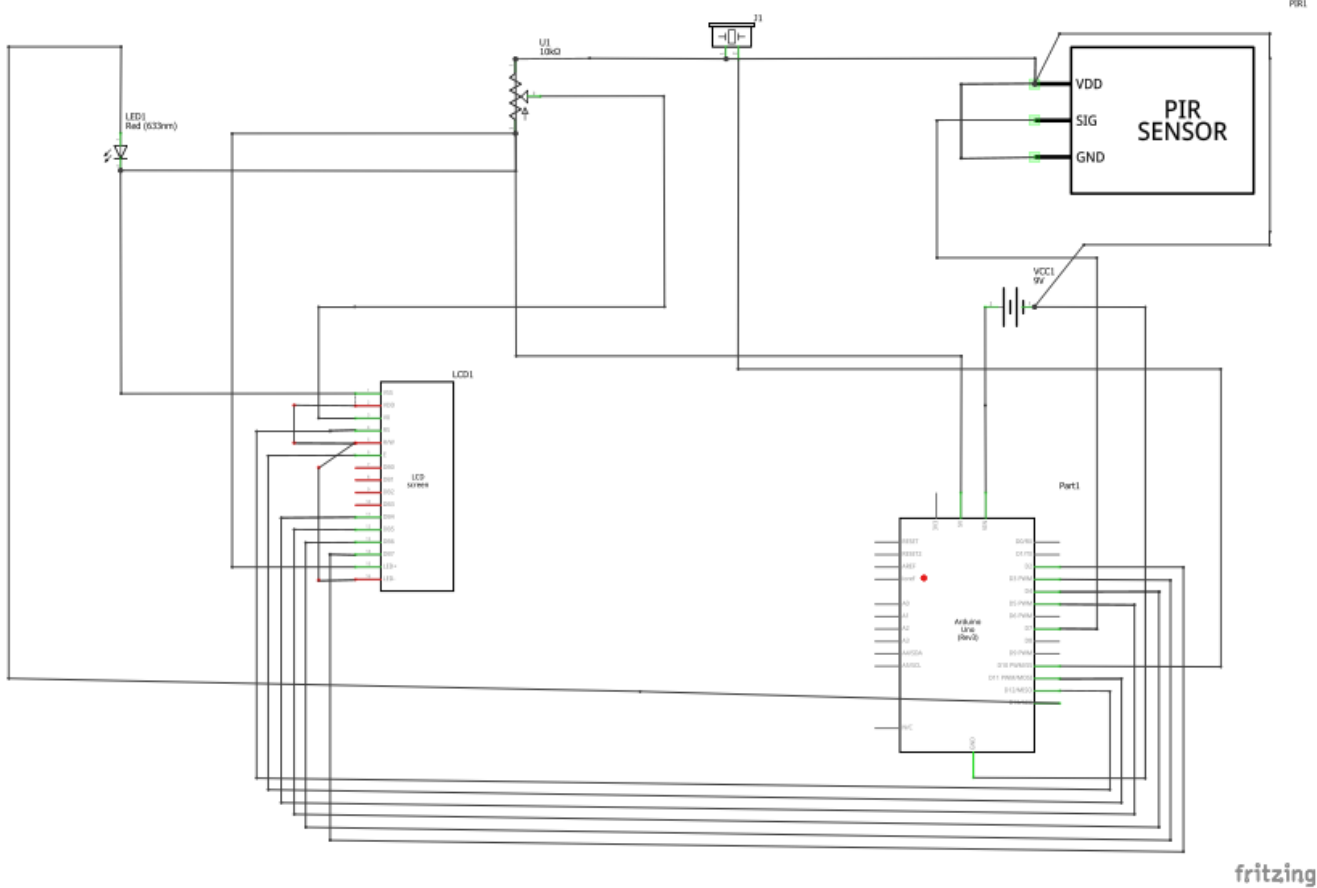until the glue dries.

APLLICATIONS

This type of motion sensing alarm system can be easily employable for security purposes at banks,

various offices and even for sensitive establishments such as for military. We can easily set up this

system for household purposes

HOME SECURITY SYSTEM CIRCUIT
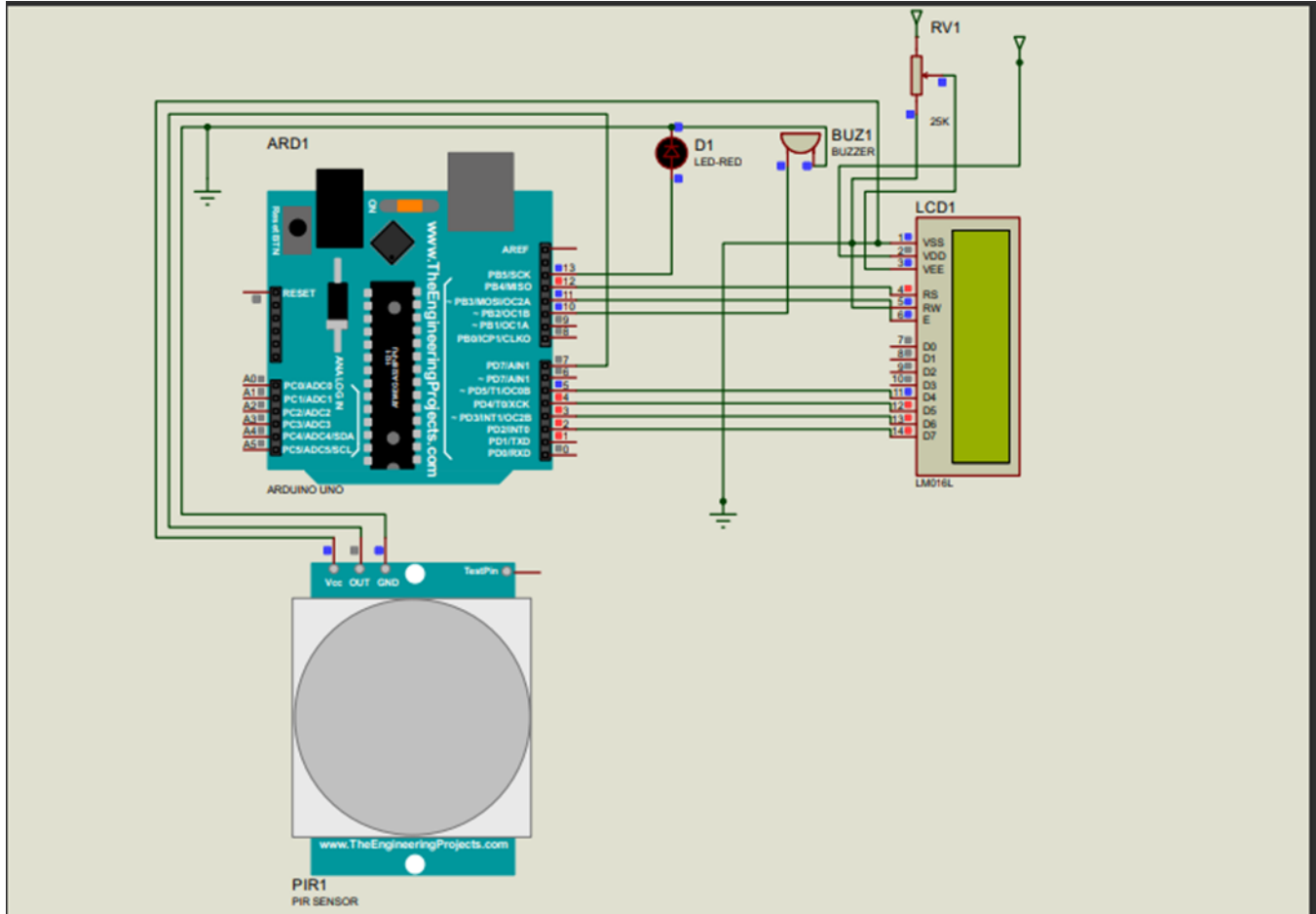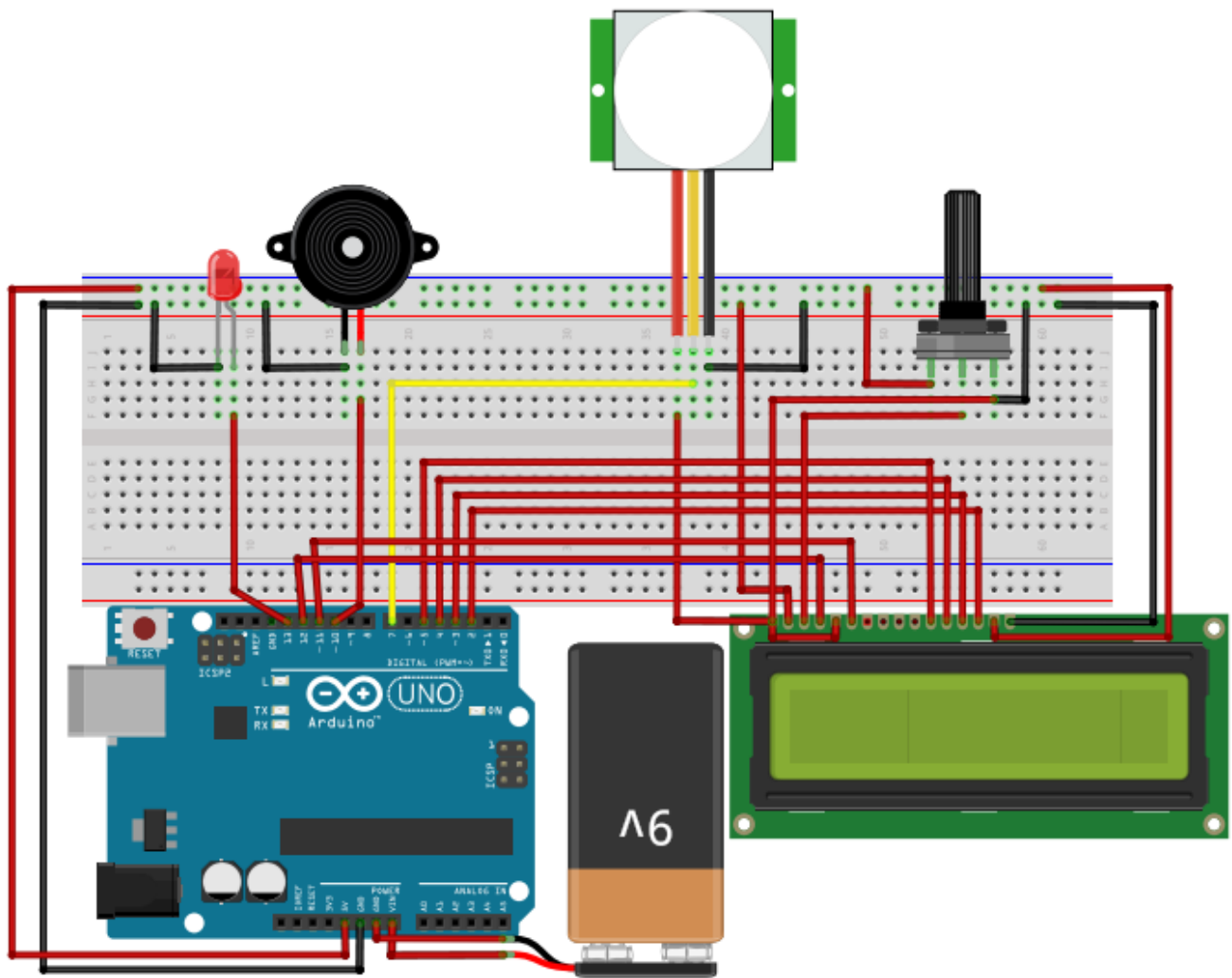
Arduino source code

```
#include <LiquidCrystal.h>
int ledPin = 13;
int inputPin = 7;
int pirState = LOW;
int val = 0;
int pinSpeaker = 10;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() { pinMode(ledPin, OUTPUT);
 pinMode(pinSpeaker, OUTPUT);
 Serial.begin(9600);
 lcd.begin(16, 2);
 lcd.setCursor(2, 0);
 lcd.print("P.I.R Motion");
 lcd.setCursor(5, 1);
 lcd.print("Sensor");
 delay(4000);
 lcd.clear();
 lcd.setCursor(2, 0);
 lcd.print("Developed By");
```

```
 lcd.setCursor(2, 1);
 lcd.print("IRUMVA JEAN PIERRE");
delay(5000);
lcd.clear();
lcd.setCursor(0, 0);
 lcd.print("Processing Data.");
 delay(3000);
 lcd.clear();
 lcd.setCursor(3, 0);
 lcd.print("Waiting For");
 lcd.setCursor(3, 1);
 lcd.print("Motion....");
}
void loop(){
 val = digitalRead(inputPin);
 if (val == HIGH)
 {
 digitalWrite(ledPin, HIGH);
 playTone(300, 300);
 delay(150);
```

```
if (pirState == LOW)
{
Serial.println("Motion detected!");
lcd.clear() ;
lcd.setCursor(0, 0);
lcd.print("Motion Detected!");
pirState = HIGH;
}
}
else
{
digitalWrite(ledPin, LOW);
playTone(0, 0);
delay(300);
if (pirState == HIGH)
{
Serial.println("Motion ended!");
lcd.clear() ;
lcd.setCursor(3, 0);
lcd.print("Waiting For");
lcd.setCursor(3, 1);
```

```
lcd.print("Motion....");

pirState = LOW;

}

}

}

void playTone(long duration, int freq)

{

duration *= 1000;

int period = (1.0 / freq) * 100000;

long elapsed_time = 0;

while (elapsed_time < duration)

{

digitalWrite(pinSpeaker,HIGH);

delayMicroseconds(period / 2);

digitalWrite(pinSpeaker, LOW);

delayMicroseconds(period / 2);

elapsed_time += (period);

}

}
```