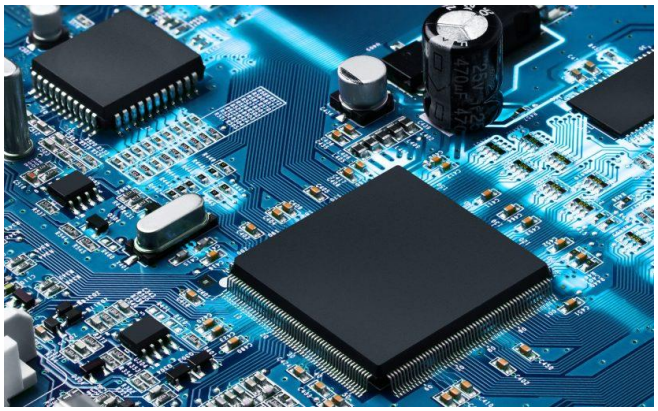
JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Subject: EMBEDDED SYSTEMS



TOPIC: HOME AUTOMATION SYSTEM USING ARDUINO UNO

By:

1. Vishal S Chincholi (01JST17CS178)

2. Chandan Kumar M (01JST18CS402)

3. Dhanush B L (01JST18CS404)

4. Meraj Ahmed (01JST17CS086)

Under the Guidance of

Madhusudan R

Associate Professor

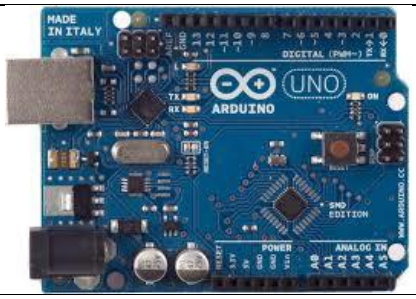
1. INTRODUCTION

Home automation system makes the operations of various home appliances more convenient and saves energy. With the energy saving concept, home automation or building automation makes life very simple nowadays. It involves automatic controlling of all electrical or electronic devices in homes or even remotely through wireless communication. Centralized control of lighting equipments, , security systems, kitchen appliances and all other equipments used in home systems is possible with this system.

This system is mainly implemented by sensors and controlling devices as shown in the figure. The sensors detects light, motion, temperature and other sensing elements, and then send that data to the main controlling devices.

Controllers may be personal computers/laptops, touch pads, smart phones, etc., attached to the controlling devices like programmable-logic controllers that receive the information from the sensors. The programmable controller allows to connect various sensors and actuators through various input and output modules whether they are analog or digital.

2. COMPONENTS USED

SLNO	Component Name	Component
1	Arduino UNO	
2	Ultrasonic Sensor	
3.	Servo Motor	
4.	Electric Bulb	

5.	Buzzer	
6.	PIR Motion Sensor	
7.	Light Dependent Resistor	
8.	Realy	
9.	Fire Sensor	

3. MODULES

Module 1: PARKING GATE

Whenever a car comes near gate, ultrasonic sensor detects the car. Once the car is detected, servo motor is activated which opens the gate.

Components used:

1. Ultrasonic sensor
2. Servo motor

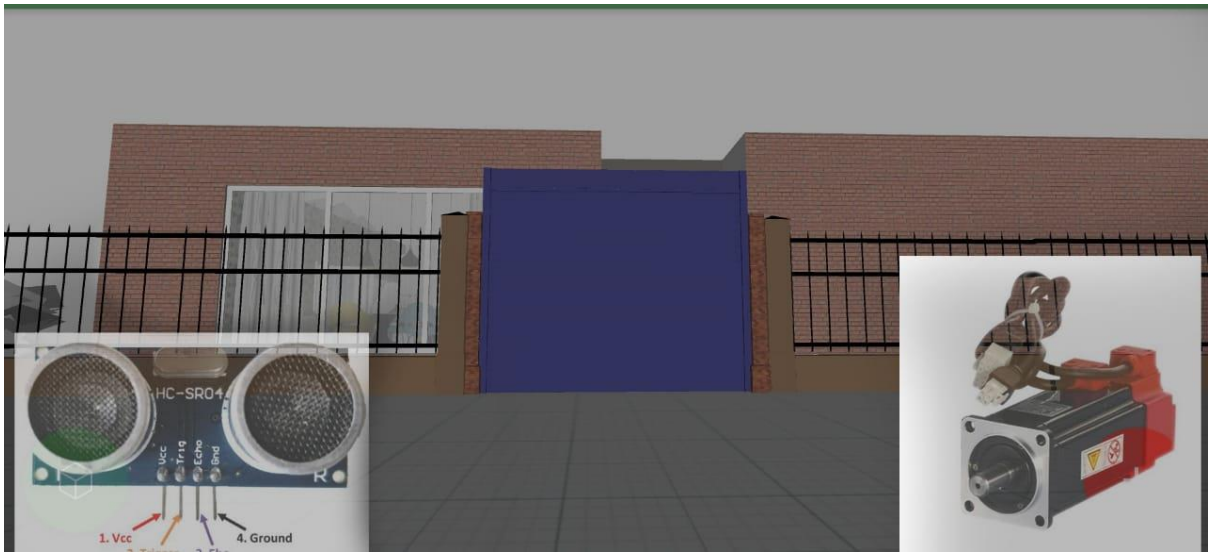


Figure: Parking gate

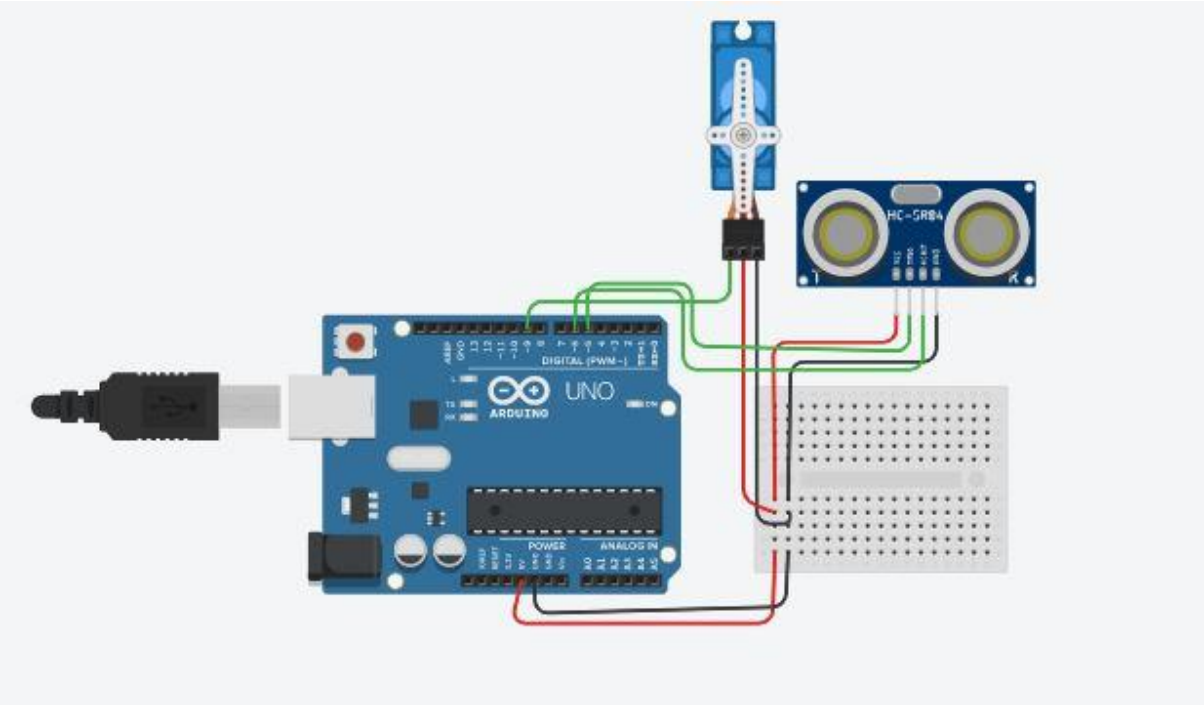


Figure: Circuit for Parking Gate

Code:

```
#include <Servo.h>
Servo servo1;
long duration;
int distance;
const int trigPin = 5;
const int echoPin = 6;

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
Serial.begin(9600);
  servo1.attach(9);
}

void loop() {

                    digitalWrite(trigPin, LOW);
```

```
                delayMicroseconds(2);
                digitalWrite(trigPin, HIGH);
                delayMicroseconds(10);
                digitalWrite(trigPin, LOW);

                duration = pulseIn(echoPin, HIGH);

                distance= duration*0.034/2;

                Serial.print("Distance: ");
                Serial.println(distance);

  if(distance<=50){
    servo1.write(90);
   delay(5000);}

  else if(distance>=50){
    delay(5000);
    servo1.write(0);}
}
```

Module 2: HALL

PIR motion sensor detects the object in motion. Whenever a motion is detected, it activates the relay to turn on the light.

Components used:

1. PIR sensor
2. LDR
3. Electric bulb

Figure: Hall

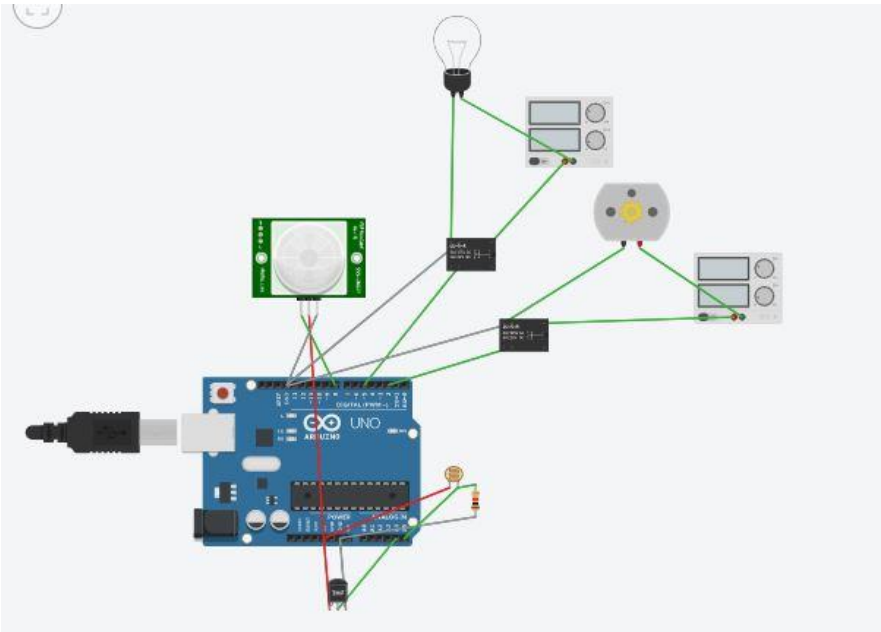


Figure: Circuit for Hall

Code:

```
int x,y,t ;
void setup ()
{
  pinMode( 8, INPUT);
  pinMode( 2, OUTPUT);
  pinMode( 5, OUTPUT);
  pinMode( A5, INPUT);
  pinMode( A4, INPUT);
  Serial.begin(9600);
}
void loop ()
{
  x= digitalRead(8);
  y= analogRead(A5);
  t= analogRead(A4);
  t= (t*5)/1.024;
```

```
t= t/10;
Serial.print("Temperature:  ");
Serial.println(t);
Serial.print("Human detected (1:yes, 0:no):  ");
Serial.println(x);
Serial.print("milivolt through LDR:  ");
Serial.println(y);

if (x<1)
{
  digitalWrite(5,0);
  digitalWrite(2,0);

}
else
{
  if ((y<550) && (t>32))
  {
    {digitalWrite(5,1);
     digitalWrite(2,1);}
    delay(4000);

  }

  else
  {
                  if ((y>550) && (t>32))
                 {
        {digitalWrite(5,0);
         digitalWrite(2,1);}
        delay(4000);

                 }

                   else
                   {
                          if ((y<550) && (t<32))
                          {
              {digitalWrite(5,1);
```

```
            digitalWrite(2,0);}
           delay(4000);

                        }

                        else
                        {
              {digitalWrite(5,0);
               digitalWrite(2,0);}
              delay(1000);
                        }
      }
}}}
```

Module 3: KITCHEN

The fire sensor continuously monitors the temperture. Whenever the temperature goes beyond the threshold value (room temperature by default) it activates the buzzer to indicate the emergency to the home owner.

Components used:

1. Fire sensor
2. Buzzer

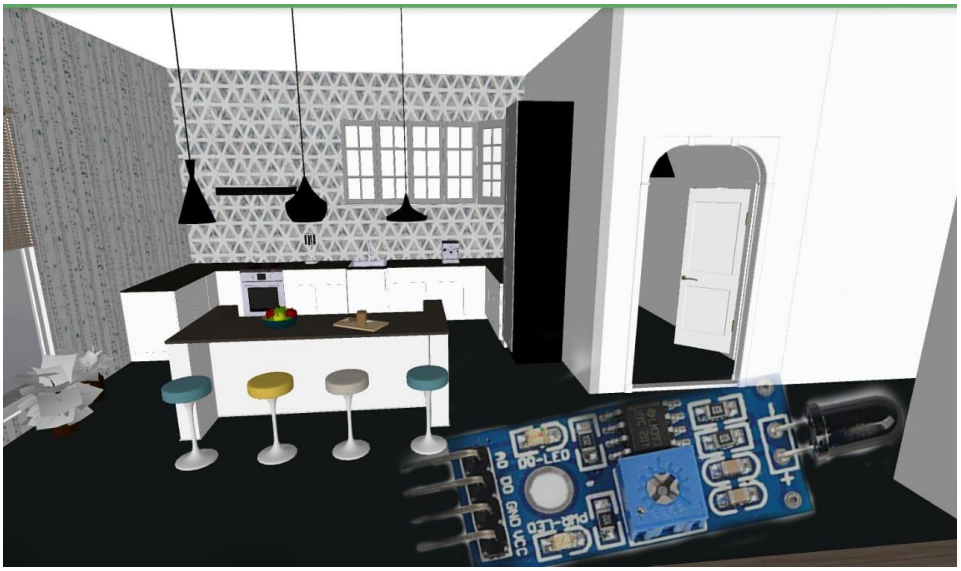


Figure: Kitchen

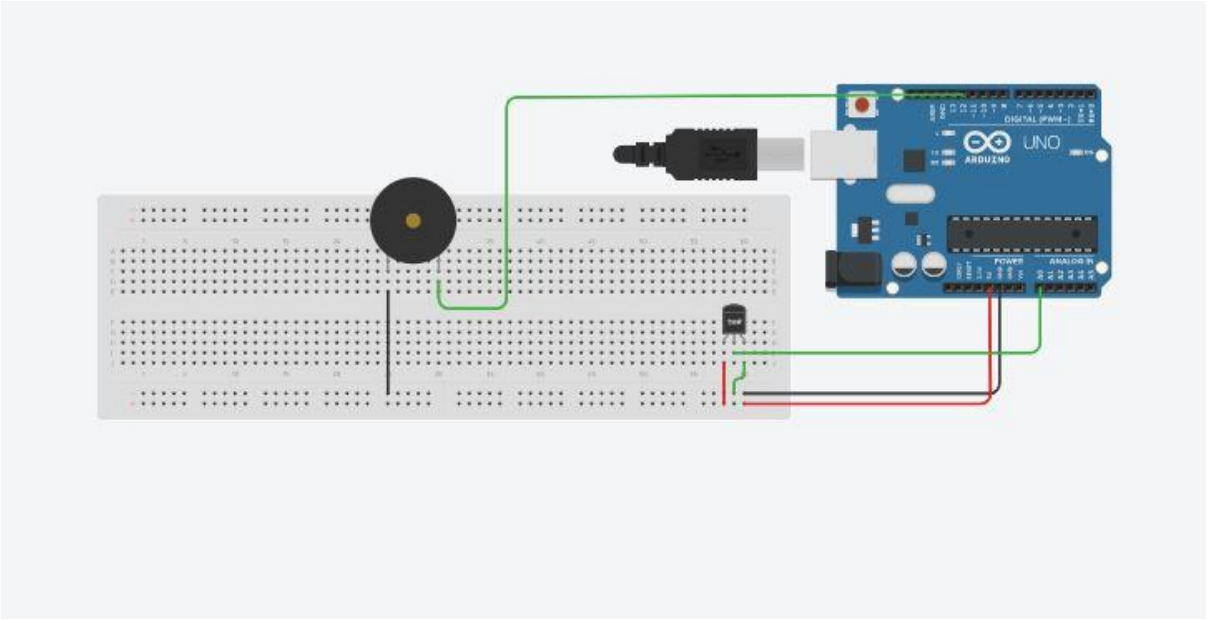


Figure: Circuit for Kitchen

Code:

```
const int temperaturePin = 0;
int buzzer = 12;

void setup()
{

pinMode(buzzer, OUTPUT);
}

void loop()
{
float voltage, degreesC;
voltage = getVoltage(temperaturePin);
degreesC = (voltage - 0.5) * 100.0;
```

```
  if(degreesC>37)
  {
  digitalWrite(buzzer, LOW);
  delay(500);  //delay half a second
  tone(12, 10000, 100);
  }
}

float getVoltage(int pin)
{

  return (analogRead(pin) * 0.004882814);
}
```

Module 4: BEDROOM

During night, when everyone inside the room decide to sleep, this system is turned on. It involves two home safety measures. One is, when a burglar enters the room, the motion is detected by PIR motion detector sensor and the buzzer is activated. Two is, when the burglar approaches the safety vault, it is detected by the ultrasonic sensor and buzzer is triggered.

Components used:

1. PIR sensor
2. Ultrasonic Sensor
3. Buzzer

Figure: Bedroom

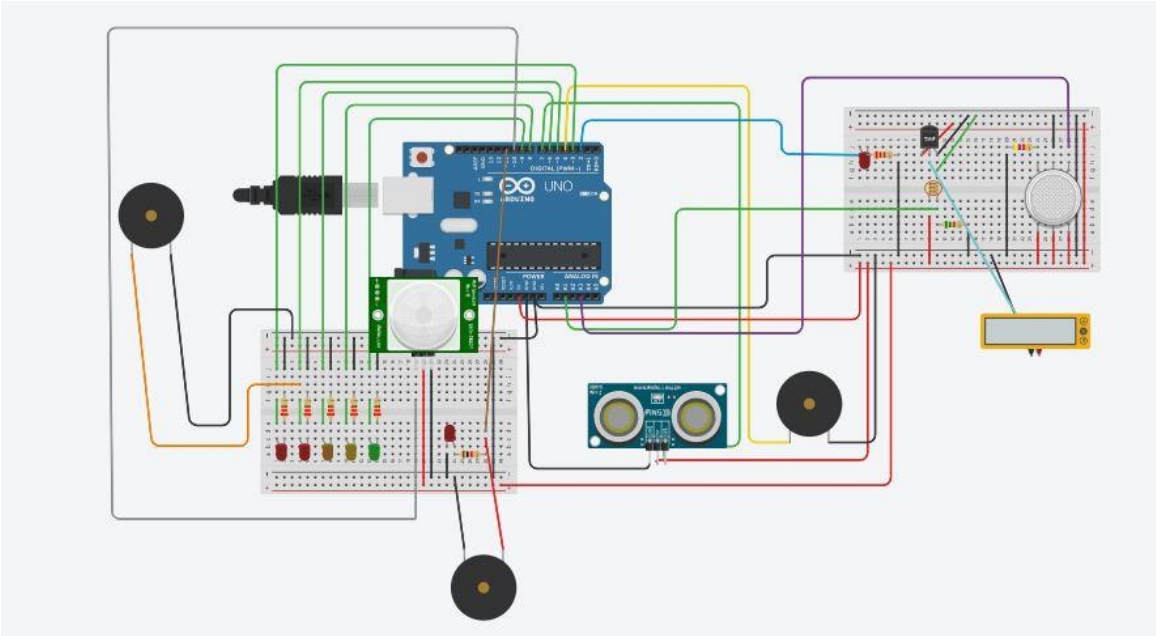


Figure: Circuit for Bedroom

Code:

```
const int PingPin=7;
const int buzzPin=4;
const int ledPin = 2;
const int ldrPin = A1
#define PIN_LED_1   9
#define PIN_LED_2   8
#define PIN_LED_3   6
#define PIN_LED_4   5
#define PIN_LED_5   3
#define PIN_GAS     A3
const int delayBetweenReads = 5000;
const int sensorPin = A5;
void setup()
{
  //initialize serial communication
```

```
  Serial.begin(9600);
  pinMode(buzzPin,OUTPUT);
  pinMode(ledPin, OUTPUT);  //initialize the LED pin as an output
  pinMode(ldrPin, INPUT);   //initialize the LDR pin as an input
  pinMode(PIN_LED_1, OUTPUT);
  pinMode(PIN_LED_2, OUTPUT);
  pinMode(PIN_LED_3, OUTPUT);
  pinMode(PIN_LED_4, OUTPUT);
  pinMode(PIN_LED_5, OUTPUT);
  pinMode(10 , INPUT ); // signal of pir sensor
  pinMode (11 , OUTPUT ); // output for motion detection
}
void loop()
{
  //establish variables for duration of Ping
  // give a short low pulse beforehand to ensure a clean high pulse
  long duration,cm;
  pinMode(PingPin,OUTPUT);
  digitalWrite(PingPin,LOW);
  delayMicroseconds(2);
  digitalWrite(PingPin,HIGH);
  delayMicroseconds(5);
  digitalWrite(PingPin,LOW);
  pinMode(PingPin,INPUT);
  duration = pulseIn(PingPin,HIGH);
  cm=microsecondsToCentimeters(duration);
  Serial.print("Distance: ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  if(cm<100)
  {
    digitalWrite(buzzPin,HIGH);
  }
  else
  {
    digitalWrite(buzzPin,LOW);
  }
```

```
  int ldrStatus = analogRead(ldrPin);   //read the status of the LDR value

  //check if the LDR status is <= 300
  //if it is, the LED is HIGH

   if (ldrStatus <=300) {
    digitalWrite(ledPin, HIGH);              //turn LED on
   }

  else {

    digitalWrite(ledPin, LOW);          //turn LED off

  }

  int value = map(analogRead(PIN_GAS), 300, 750, 0, 100);
  digitalWrite(PIN_LED_1, HIGH);
  digitalWrite(PIN_LED_2, value >= 20 ? HIGH : LOW);
  digitalWrite(PIN_LED_3, value >= 40 ? HIGH : LOW);
  digitalWrite(PIN_LED_4, value >= 60 ? HIGH : LOW);
  digitalWrite(PIN_LED_5, value >= 80 ? HIGH : LOW);

if (digitalRead(10) == HIGH) // check if PIR is triggered
  {
                digitalWrite(11,HIGH);
                delay(100);

                digitalWrite(11,LOW) ;
                delay(100);
  }
  delay(100);
  }

long microsecondsToCentimeters(long microseconds)
{
  return microseconds/29/2;
}
```

4. REFERENCES

1. Tinkercad

2. YouTube