SMART HOME AUTOMATION

Using Embedded System

By, Kaviselvan T

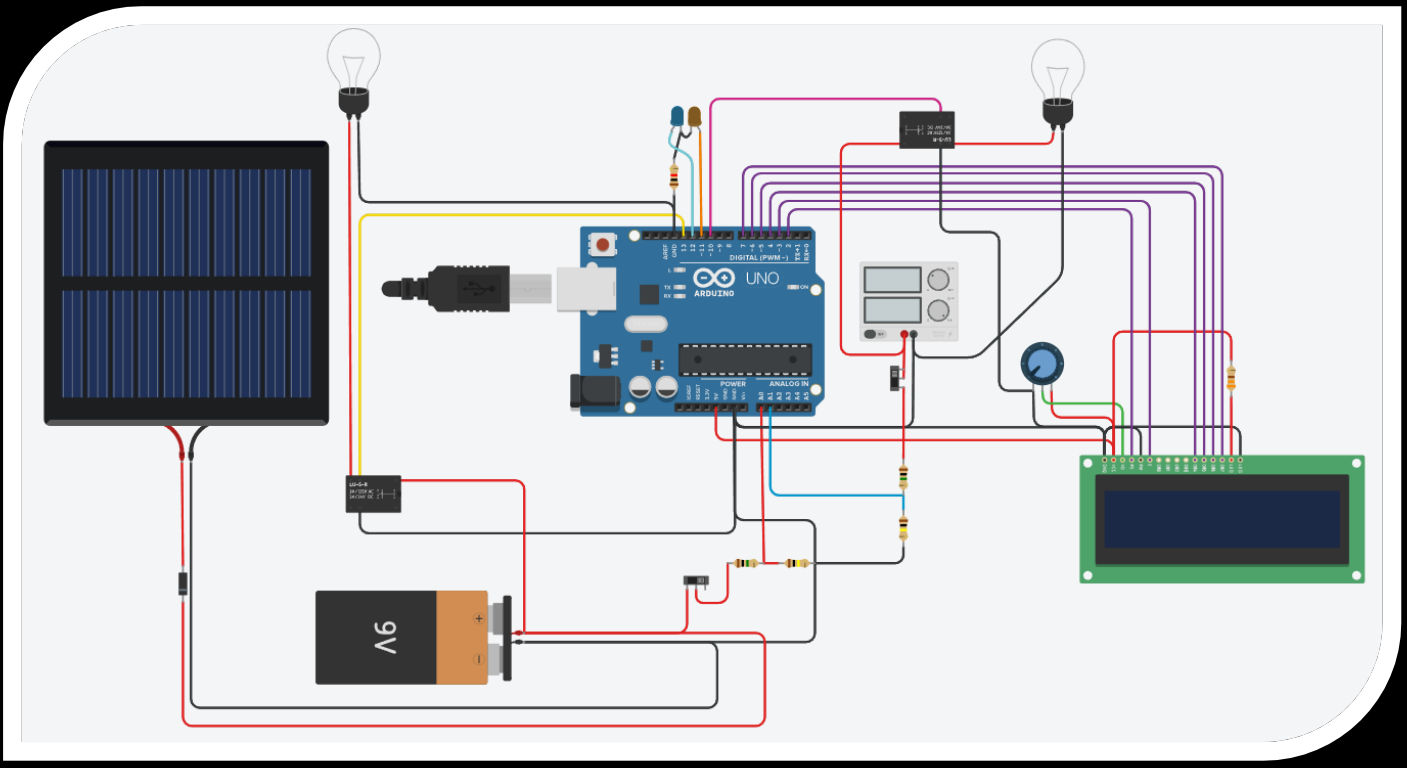
BE.ECE

Smart Home

Welcome to my Smart Home Automation project, an innovative endeavour aimed at transforming traditional living spaces into intelligent, responsive, and energy efficient environments. In this project, I have harnessed the power of embedded systems to develop a comprehensive smart home solution that integrates a variety of sensors, microcontrollers, and actuators. This system is designed to provide enhanced convenience, safety, and efficiency through automated control of home appliances and monitoring systems.

The core of this project revolves around using Arduino boards to manage and control various aspects of home automation, including power supply management, automatic lighting, climate control, security systems, and more. By leveraging solar energy and advanced sensing technologies, this smart home system not only ensures uninterrupted power but also optimizes energy usage.

Power Supply for Home Automation

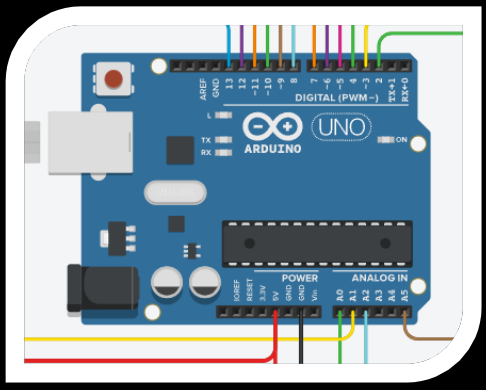


In the smart home system, the Arduino board controls the overall power supply. The primary power source is connected to an external power supply. In the event of a power outage, solar-charged batteries are utilized to provide power to the home. Solar power is essential for charging the batteries, and they can also be charged using the primary power source from the home. The status of the battery power level and the transition between power sources is displayed and notified on a 2x16 display.

Kaviselvan T

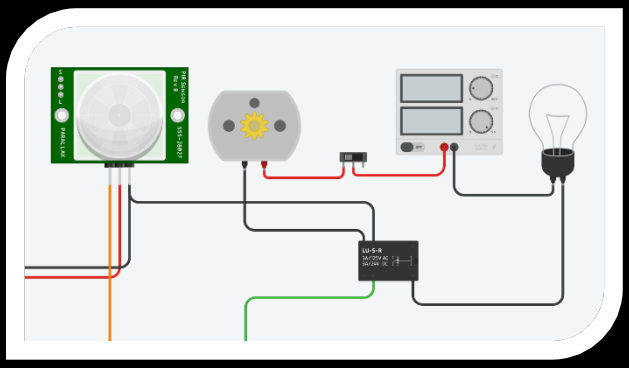
Arduino Boards

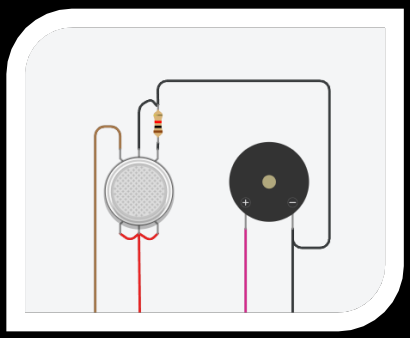
Arduino Uno boards used in this setup, each handling different aspects of the home automation system. This board is used to interconnect various embedded devices for sensing and actuation. This board is only for prototype purpose only. For this project we used three board to compensate all the devices.



Automatic AC & Lighting

The PIR sensor detects human motion and activates the air cooler and lighting in the room. It saves energy by automatically turning off the AC motor and lights when no one is present. It uses an external power supply.





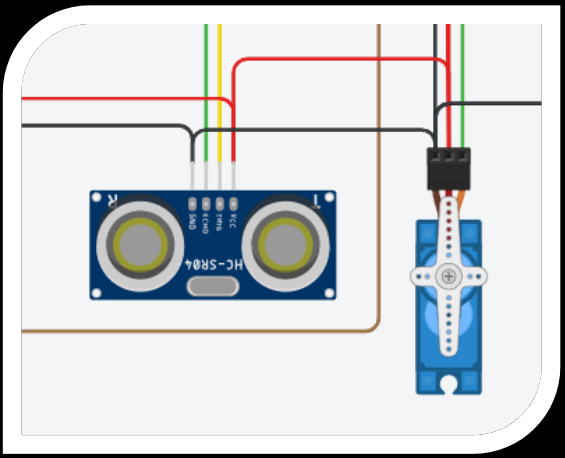
Smoke Detector and Alarm System

Gas sensors detect harmful gases and smoke in the home, and then send a signal through the alarm system to notify the residents.

Kaviselvan T

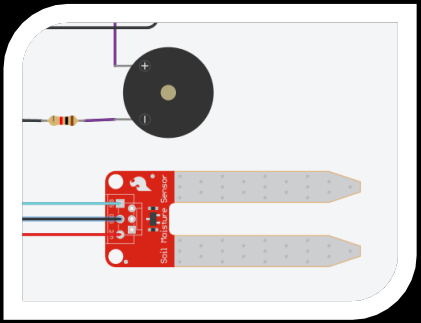
Automatic Door Control

The ultrasonic distance sensor and servo in the door are used to detect a person in front of the door and open it using the servo connected to the door hinges. After a certain amount of time, it will automatically close.



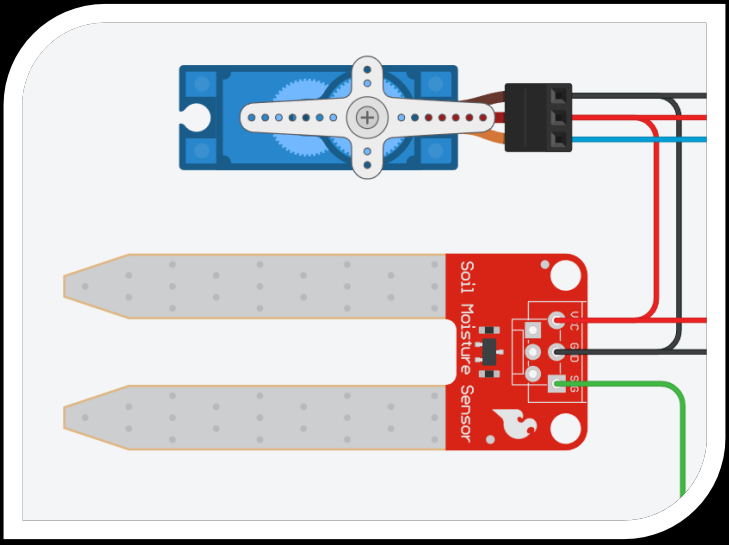
Rain Detection

The rain sensor is placed in the environment to continuously monitor the surroundings. When rainwater droplets pass through the sensor, it closes the circuit to send a notification to the controller, which triggers the buzzer.

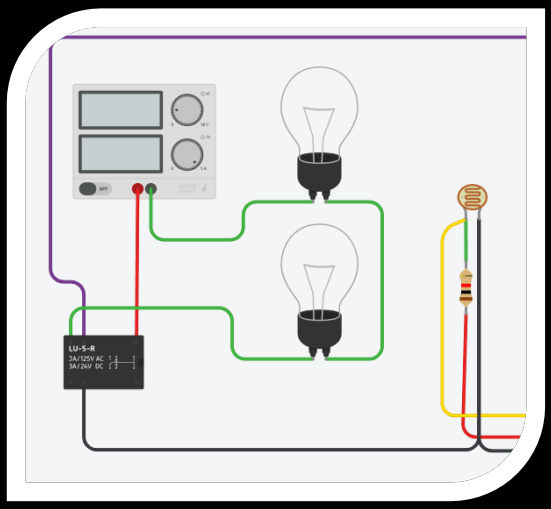


Moisture Monitor for Plants

Soil moisture sensor is connected to the plant's field. The sensor continuously monitors the soil moisture scale. When the water moisture level decreases to a certain amount, the controller opens the water tap using a servo motor. the water moisture amount is reached, it turns off the tap.

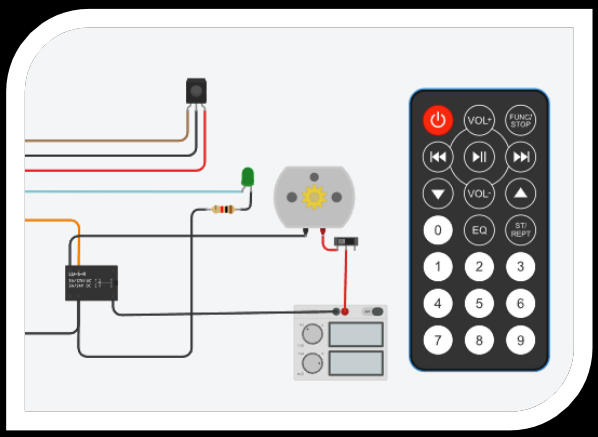


Kaviselvan T



Automatic Lighting for Night

The photoresistor sensor is placed here to observe the daylight and send the information to the controller. When the sensor detects a change in daylight conditions, the home lighting is turned on automatically.



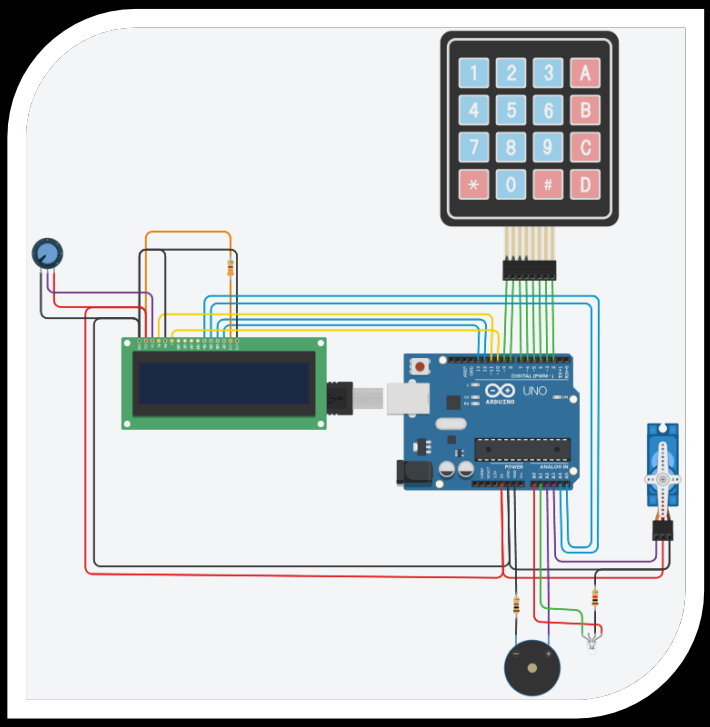
Manual Controls for Motors

IR sensor and a remote are used to control the overall motor actions for different devices such as fans, AC, and water pumps. The entire system is set up for manual control. External power can be assigned to heavy-duty motors.

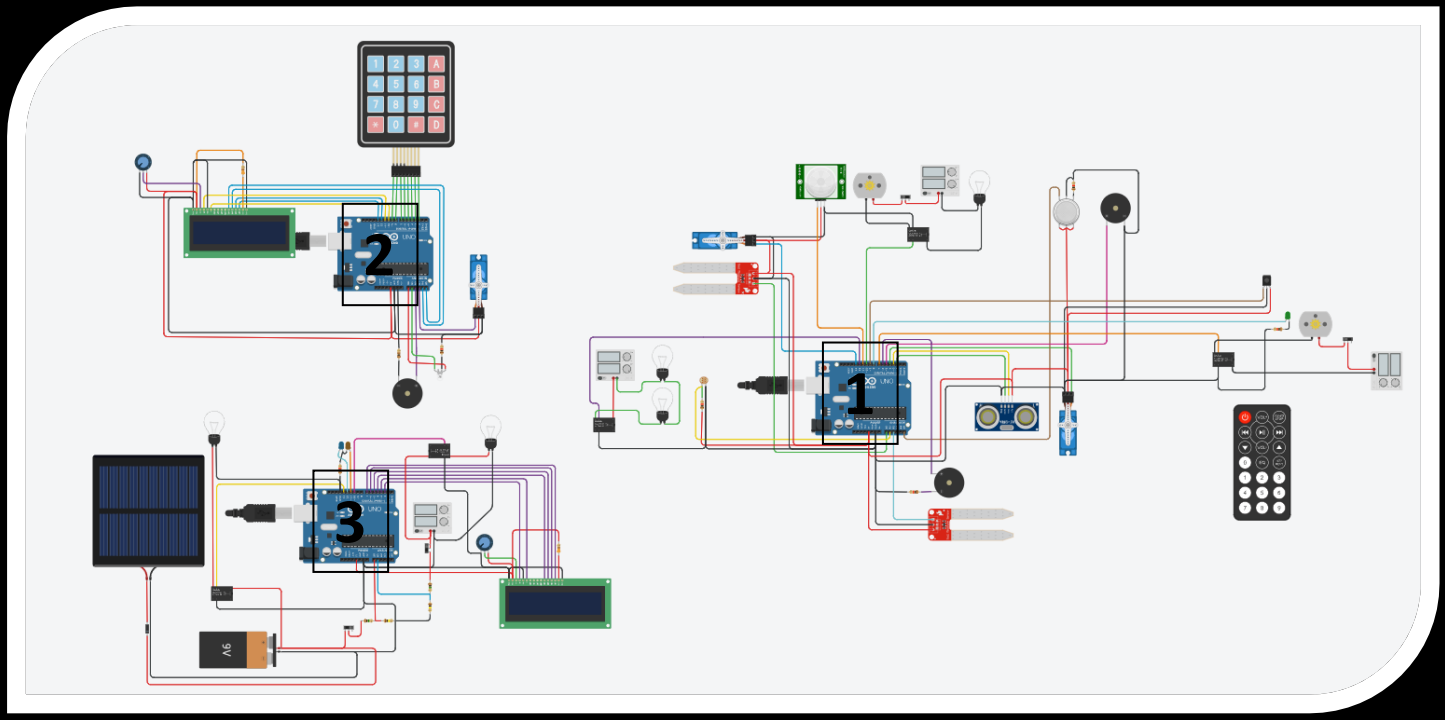
Kaviselvan T

Smart Lock

When the numeric code is entered on the keyboard, it unlocks the passkey. The controller then operates the door servo to open when the passkey is entered correctly. If the wrong passkey is entered, the controller activates the buzzer.



Kaviselvan T



Program: Arduino UNO 1

```
#include <Servo.h>
#include <IRremote.hpp>
//------------Door----------------//
Servo door_servo;

const int trigPin = 3;
const int echoPin = 2;
long duration;
int distance;

//----tap water for plant---------//
Servo tap;
int plantmoisture;

//---------Rain Alert-------------//
const int rainbuzzer = 6;
int rainsensor;

//---------FAN & LIGHT-------------//
int PIR_motion = 11;
int fan_light_switch = 10;
int pir_1;

//---------Gas Detection-------------//
int gas_buzzer = 5;
int gas_sensor = A5;
int gas_limit = 100;

//---------LDR lightning-------------//
```

```
int LDR = A1;
int ldr_light_switch = 12;

//---------IR remote control---------//
#define IR_RECEIVE_PIN 9
int LEDx_PIN = 8; //LED pin
int fan_switch_ = 7;//fan switch pin


void setup()
{
  Serial.begin(9600);

  //------------Door----------------//
  door_servo.attach(4); //servo pin 4
  pinMode(trigPin,OUTPUT);
  pinMode(echoPin,INPUT);

  //----tap water for plant---------//
  tap.attach(13); //servo-tap water

  //---------Rain Alert-------------//
  pinMode(rainbuzzer,OUTPUT); //signal to buzzer

  //---------FAN & LIGHT-------------//
  pinMode(PIR_motion,INPUT);
  pinMode(fan_light_switch,OUTPUT);//fan and light

  //---------Gas Detection-------------//
  pinMode(gas_buzzer,OUTPUT);//buzzer

  //---------LDR lightning-------------//
  pinMode(LDR,INPUT);
  pinMode(ldr_light_switch,OUTPUT);//light

  //---------IR remote control---------//
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK);//IR signals
  pinMode(LEDx_PIN, OUTPUT); //LED pin
  pinMode(fan_switch_,OUTPUT); //fan pin

}


void loop()
{
  ////////////////////////////////////
  //------------Door----------------//
  digitalWrite(trigPin,LOW);//reset trig pin
  delayMicroseconds(2);

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

```
duration = pulseIn(echoPin,HIGH);
distance = (duration*0.034)/2; //constant=speed of sound 0.034

//door servo
if (distance<=150)
  {
  door_servo.write(90);
  Serial.print("|| Door Opened!  || Distance:");
  Serial.println(distance);
  Serial.print("\n");
}
else {
  door_servo.write(0);
  Serial.print("|| Door closed!  || Distance:");
  Serial.println(distance);
  Serial.print("\n");
}
////////////////////////////////////

//----tap water for plant---------//
plantmoisture = analogRead(A0); //values 0-1023
if (plantmoisture<=200)
{
  tap.write(90);
  Serial.print("|| Plant Dried!!  || ");
  Serial.println(plantmoisture);
  Serial.print("\n");
}
else {
  tap.write(0);
}
////////////////////////////////////

//---------Rain Alert-------------//
rainsensor = analogRead(A2); //values 0-1023
if (rainsensor>=100)
{
  tone(rainbuzzer,100);
  Serial.print("|| Raining!!  || ");
  Serial.println(rainsensor);
  Serial.print("\n");
}
else {
  noTone(rainbuzzer);
}
////////////////////////////////////

//---------FAN & LIGHT-------------//
pir_1 = digitalRead(PIR_motion);
if (pir_1 == HIGH)
{
  digitalWrite(fan_light_switch,HIGH);
```

```
  }
  else {
    digitalWrite(fan_light_switch,LOW);
  }

  //---------Gas Detection-------------//
  int gas_val = analogRead(gas_sensor); //read sensor value

  if (gas_val > gas_limit)
    {
    tone(gas_buzzer, 600);
    Serial.print("|| Gas Sensor Value = ");
    Serial.print(gas_val);
    Serial.print("\n");
    }
  else {
    noTone(gas_buzzer);
  }
  ////////////////////////////////////

  //---------LDR lightning----------//
  int ldr_val = analogRead(LDR); //read sensor value

  if (ldr_val >= 500)
  {
    digitalWrite(ldr_light_switch,HIGH);
  }
  else {
    digitalWrite(ldr_light_switch,LOW);
  }
  Serial.print("|| LDR Sensor Value = ");
  Serial.print(ldr_val);

  Serial.print("\n ");
  ////////////////////////////////////

  //---------IR remote control---------//
  if (IrReceiver.decode()) {
    Serial.println(IrReceiver.decodedIRData.decodedRawData, HEX); // Print raw data
    IrReceiver.printIRResultShort(&Serial); // Print complete received data in one line
    IrReceiver.printIRSendUsage(&Serial);   // Print the statement required to send this
data

    switch (IrReceiver.decodedIRData.command) {
      case 0x10: // press 1 to turn ON light
        digitalWrite(LEDx_PIN, HIGH);
        break;
      case 0xc: // press 0 to turn OFF light
        digitalWrite(LEDx_PIN, LOW);
        break;
      case 0x5: // press play to turn ON FAN
        digitalWrite(fan_switch_, HIGH);
        break;
```

```
      case 0x0: // press power key to turn OFF FAN
        digitalWrite(fan_switch_, LOW);
        break;
    }
    IrReceiver.resume(); // Prepare to receive next IR signal
  }
  ///////////////////////////////////


  delay(10);
}
```

Program: Arduino UNO 2

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>

//----------------------keypad----------------------------//
const byte ROWS = 4;//rows in the keypad
const byte COLS = 4;//columns in the keypad

char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};//key layout

byte rowPins[ROWS] = {9, 8, 7, 6};//pins connected to rows
byte colPins[COLS] = {5, 4, 3, 2};//pins connected to columns

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

//-------------------------LCD-----------------------------//
LiquidCrystal lcd(11, 10, A4, A5, 12, 13);//(rs, enable, d4, d5, d6, d7)

//-------------------------Password-----------------------------//
#define Password_Length 5
char Data[Password_Length];
char Passkey[Password_Length] = "1234";
byte data_count = 0;//passkey_count = 0;
char customKey;

//-------------door action---------//
int G_doorled = A1;//green led
int R_doorled = A0;//red led
int door_buzzer1 = A2;//buzzer
Servo door_ser; //door servo
```

```
void setup(){
  lcd.begin(16, 2);// set up the LCD's number of columns and rows:
  lcd.clear();// Clears the LCD screen
  pinMode(G_doorled, OUTPUT);
  pinMode(G_doorled, OUTPUT);
  pinMode(door_buzzer1,OUTPUT);//buzzer
  door_ser.attach(A3);//door servo
}

void loop(){
  lcd.setCursor(0,0);
  lcd.print("Enter Password:");

  customKey = customKeypad.getKey();
  if (customKey){
    Data[data_count] = customKey;//array Data[...]
    lcd.setCursor(data_count,1);//display data_count'th column & 1'th row
    lcd.print(Data[data_count]);
    data_count++;
    }

  if(data_count == Password_Length-1){
    lcd.clear();

    if(!strcmp(Data, Passkey)){ //Data & Master comparing
      lcd.print("Correct");
      digitalWrite(G_doorled, HIGH);
      door_ser.write(90);//door open
      delay(10000);
      digitalWrite(G_doorled, LOW);
      door_ser.write(0);//door close
      }
    else{
      lcd.print("Incorrect");
      digitalWrite(R_doorled, HIGH);

      tone(door_buzzer1,600);//buzzer if wrong
      delay(5000);
      }

    lcd.clear();
    clearData();//clear data from data_count(from void clearData())
  }
}

void clearData(){
  while(data_count !=0){
    Data[data_count--] = 0;
  }
  return;
}
```

Program: Arduino UNO 3

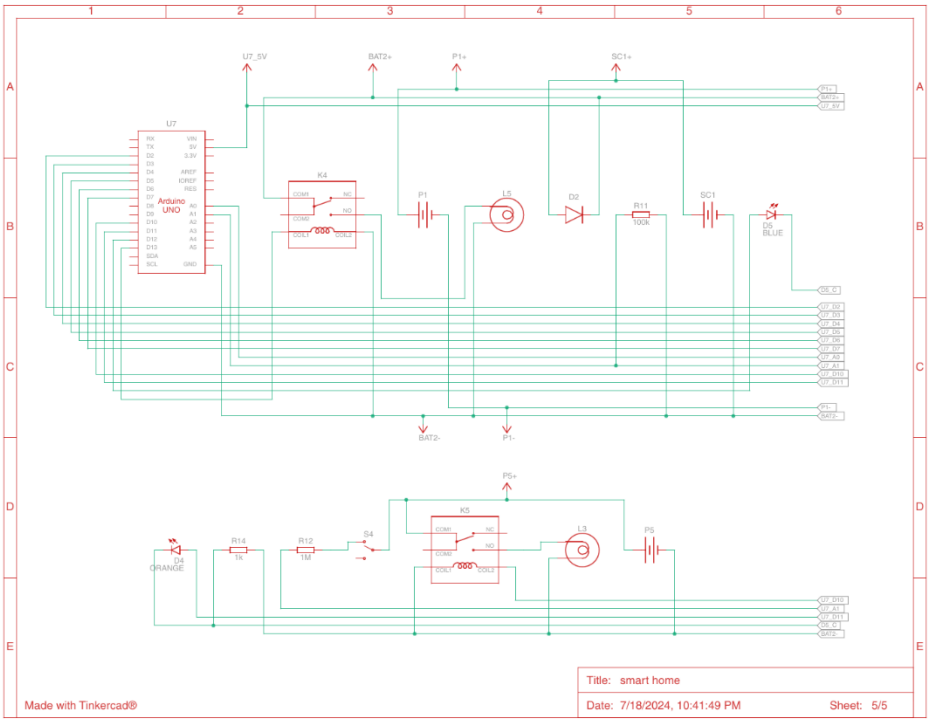
```
#include <LiquidCrystal.h>

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);//(rs, enable, d4, d5, d6, d7)

int led_pin_Bon = 12;//battery
int led_pin_PSon = 11;//power supply

int power_switch_Bt = 13;//battery load's switch
int power_switch_Ps = 10;//power supply load's switch

void setup(){
  lcd.begin(16, 2);//LCD's number of columns and rows
  lcd.clear();//Clears LCD screen

  pinMode(led_pin_Bon,OUTPUT);//power from battery
  pinMode(led_pin_PSon,OUTPUT);//power from power supply

  pinMode(power_switch_Bt,OUTPUT);//battery to load switch
  pinMode(power_switch_Ps,OUTPUT);//power supply to load switch

  //initially
  digitalWrite(led_pin_PSon, LOW);//orange led off
  digitalWrite(power_switch_Ps,LOW);//power supply OFF
  digitalWrite(led_pin_Bon, LOW);//blue led off
  digitalWrite(power_switch_Bt,LOW);//battery OFF
}

void loop()
{
  switchPower();
}

 void switchPower()
{
  //POWER SUPPLY
  int sensorValue_B = analogRead(A0); //read the A0 pin value
  float voltage_B = sensorValue_B * (5.00 / 1023.00) * 11; //convert the value to a true
voltage.(for 10v)
  lcd.setCursor(0,0);
  lcd.print("BT.voltage=");
  lcd.print(int round(voltage_B)); //print the battery voltage to LCD
  lcd.print(" V");

  //BATTERY
  int sensorValue_PS = analogRead(A1); //read the A0 pin value
  float voltage_PS = sensorValue_PS * (5.00 / 1023.00) * 11; //convert the value to a true
voltage.(for 9v)
  lcd.setCursor(0,1);
  lcd.print("PS.voltage=");
  lcd.print(int round(voltage_PS)); //print the power supply voltage to LCD
```

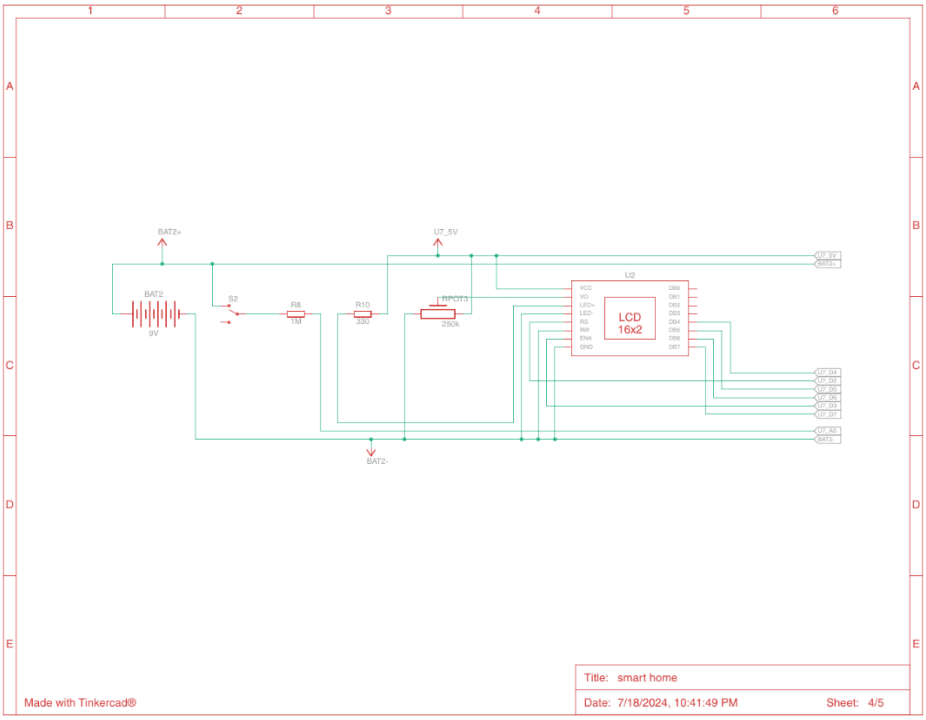
```
  lcd.print(" V");


  //switch between battery and power supply
   //both are LOW-------------
   if ((voltage_B <= 5) && (voltage_PS <= 5))
   {
     digitalWrite(led_pin_PSon, LOW);//orange led off
     digitalWrite(power_switch_Ps,LOW);//power supply OFF
     digitalWrite(led_pin_Bon, LOW);//blue led off
     digitalWrite(power_switch_Bt,LOW);//battery OFF

     lcd.clear();
     lcd.print("NO POWER");//indicate no power
   }
   else if (voltage_PS <= 5 || voltage_B <= 5)
   {
     //battery power------------
     if (voltage_PS <= 5)
     {
       digitalWrite(led_pin_Bon, HIGH);//blue led
       digitalWrite(power_switch_Bt,HIGH);//battery to load
     }
     else
     {
       digitalWrite(led_pin_Bon, LOW);//blue led off
       digitalWrite(power_switch_Bt,LOW);//battery OFF
     }
     //power supply------------
     if (voltage_B <= 5)
     {
      digitalWrite(led_pin_PSon, HIGH);//orange led
      digitalWrite(power_switch_Ps,HIGH);//power supply to load
     }
     else
     {
       digitalWrite(led_pin_PSon, LOW); //orange led off
       digitalWrite(power_switch_Ps,LOW);//power supply OFF
     }
   }

   //both are HIGH------------
   if ((voltage_B > 5) && (voltage_PS > 5))
   {//connect to power supply
     digitalWrite(led_pin_PSon, HIGH);//orange led
     digitalWrite(power_switch_Ps,HIGH);//power supply to load
     digitalWrite(led_pin_Bon, LOW);//blue led off
     digitalWrite(power_switch_Bt,LOW);//battery OFF
   }

   delay(100);
}
```

Components:

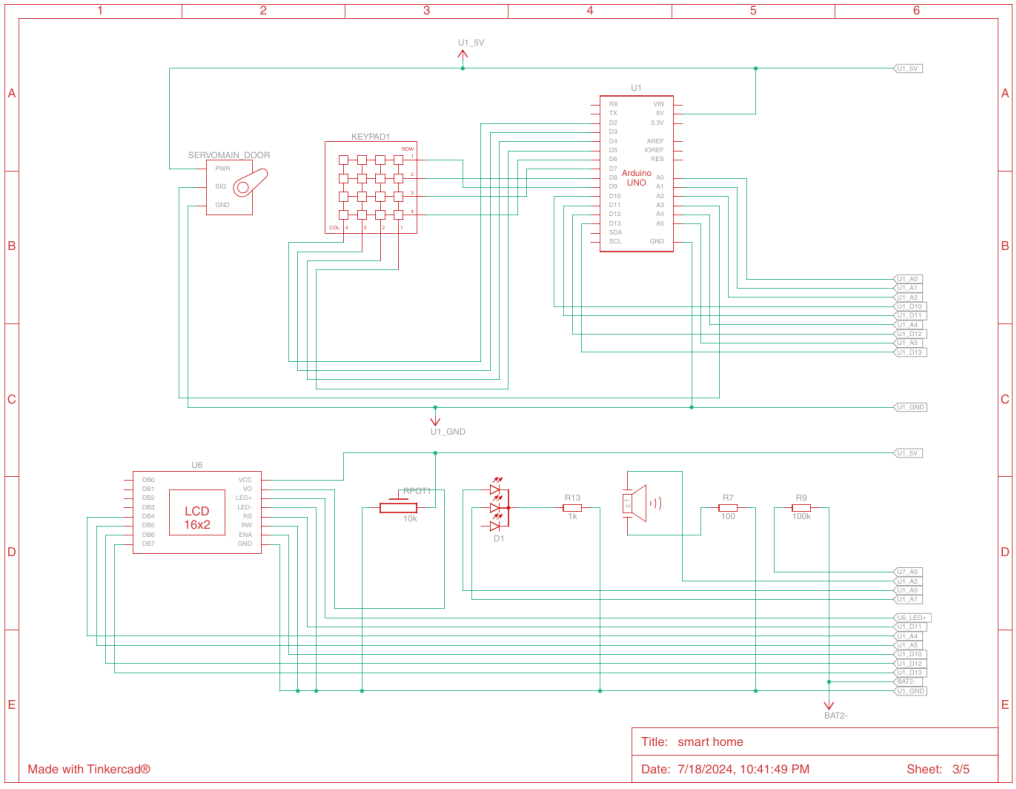
Name	Quantity	Component
SERVOdoor, SERVOwater tap, SERVOmain door	3	Positional Micro Servo
SEN1, SEN2	2	Soil Moisture Sensor
R1	1	Photoresistor
R2, R4, R5, R3, R13, R14	6	1 kΩ Resistor
L1, L2, L5, L4, L3	5	Light bulb
GAS1	1	Gas Sensor
PIEZO1, PIEZO2, PIEZO3	3	Piezo
SC1	1	5 V, 100 mA Solar Cell
P1, P2, P3	3	5 , 5 Power Supply
U2, U6	2	LCD 16 x 2
BAT2	1	9V Battery
K1, K2, K3, K4, K5	5	Relay SPDT
M1, M2	2	DC Motor
PIR1	1	PIR Sensor
KEYPAD1	1	Keypad 4x4
U4	1	IR sensor
S1, S2, S3, S4	4	Slideswitch
U5, U1, U7	3	Arduino Uno R3
R6, R10	2	330 Ω Resistor
Rpot1	1	10 kΩ Potentiometer
R7	1	100 Ω Resistor
Rpot3	1	250 kΩ Potentiometer
R8, R12	2	1 MΩ Resistor
R9, R11	2	100 kΩ Resistor
P5	1	9 , 5 Power Supply
D2	1	Diode
D3	1	Green LED
DIST1	1	Ultrasonic Distance Sensor (4-pin)
D1	1	LED RGB
D4	1	Orange LED
D5	1	Blue LED

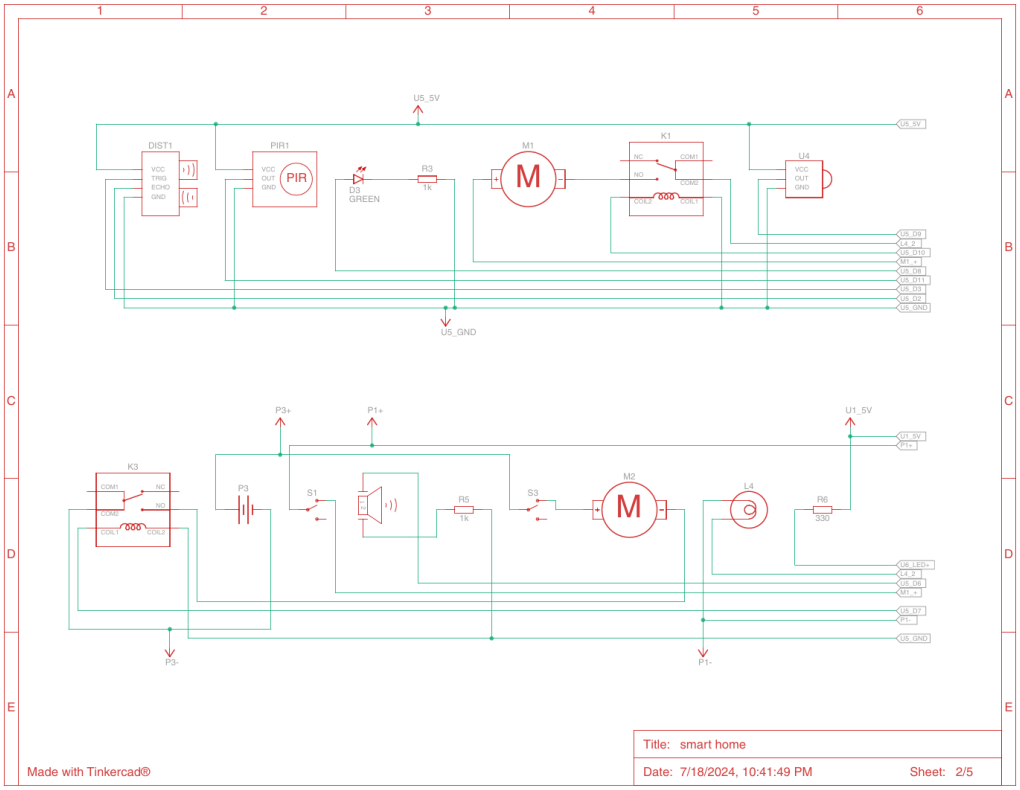
Schematic:



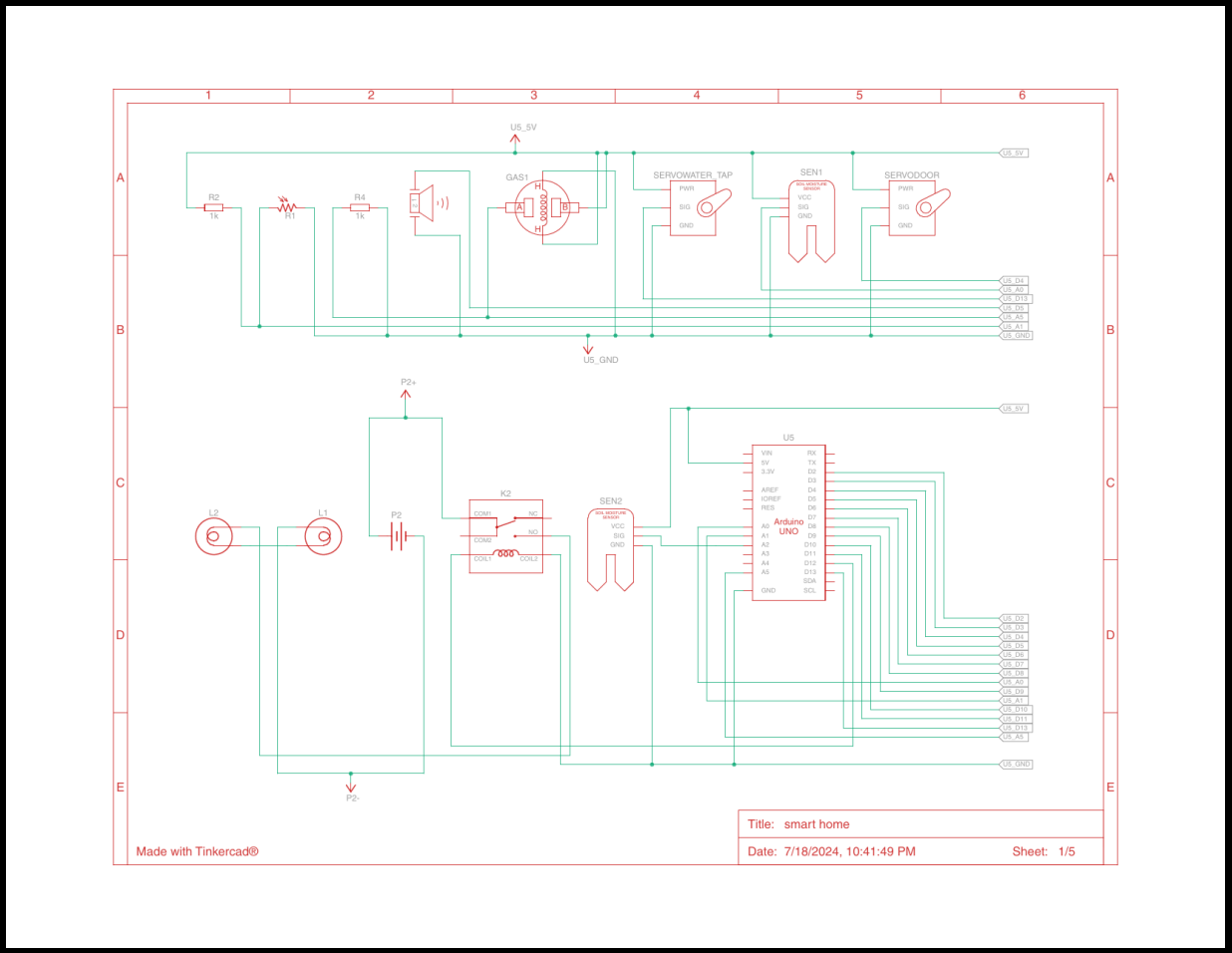


Kaviselvan T





Kaviselvan T



Conclusion

This Smart Home Automation project demonstrates the potential of embedded systems to enhance the convenience, safety, and energy efficiency of modern homes. By integrating various sensors and automated controls, the system offers a sophisticated solution to everyday challenges in home management.

Kaviselvan T