



INTERN CALL

Smart Street Lights

Abstract:

This project aims to create a smart street light system using Arduino Uno and LDR sensor. The system will automatically turn off the streetlights during daytime and turn them on at night, providing efficient and automated lighting control.

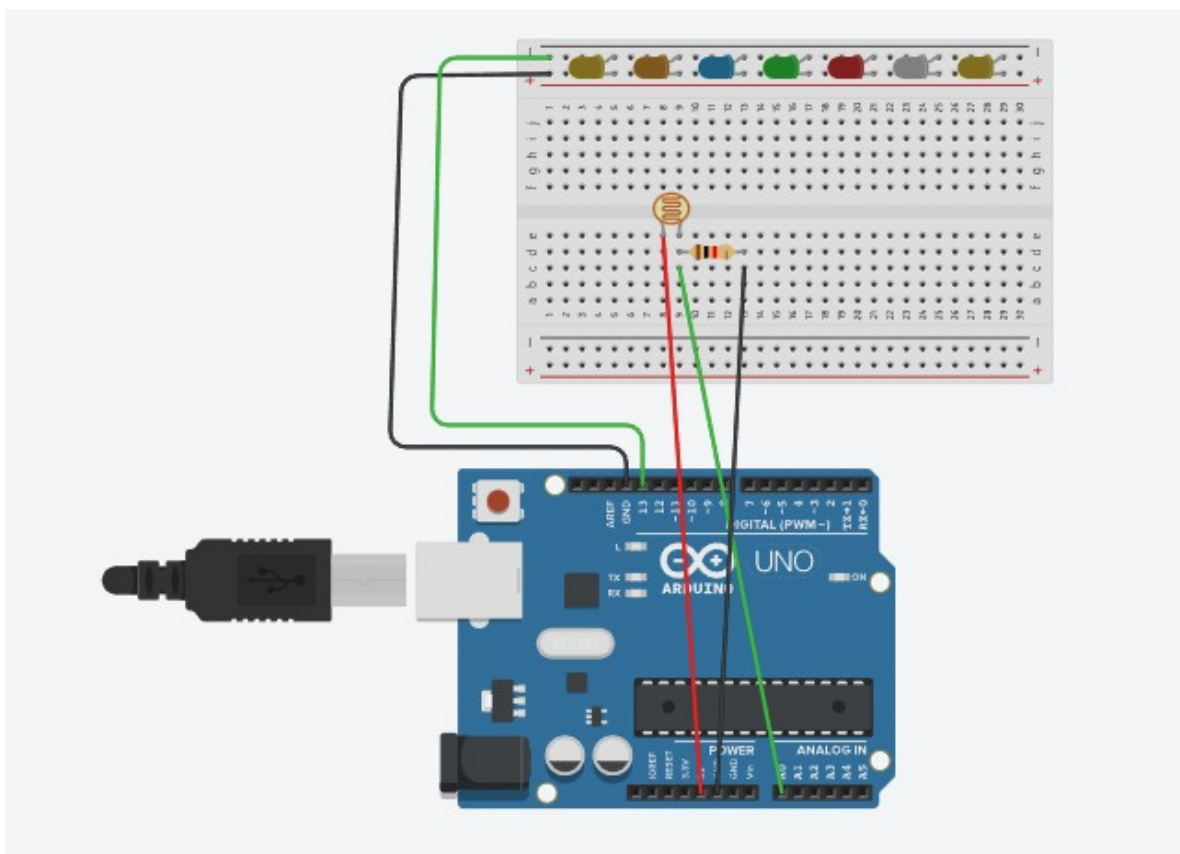
Components Required:

1. Arduino Uno
2. LDR (Light Dependent Resistor) sensor
3. Resistor
4. Jumper wires
5. Breadboard
6. LEDs (for simulation purposes in Tinkercad)
7. USB cable for uploading code to Arduino
8. Tinkercad account (for simulation)



INTERN CALL

Circuit Diagram:





INTERN CALL

Code :

```
1 // Constants
2 const int sensor = A0; // Photoresistor at Arduino analog pin A0
3 const int ledPin = 13; // LED pin at Arduino pin 13
4
5 // Variables
6 int value; // Store value from photoresistor (0-1023)
7
8 void setup() {
9   pinMode(ledPin, OUTPUT); // Set ledPin - 13 pin as an output
10  pinMode(sensor, INPUT); // Set pResistor - A0 pin as an input (optional)
11 }
12
13 void loop() {
14   value = analogRead(sensor);
15
16   // You can adjust the threshold value based on your LDR characteristics
17   if (value > 500) {
18     digitalWrite(ledPin, LOW); // Turn LED off
19   } else {
20     digitalWrite(ledPin, HIGH); // Turn LED on
21   }
22
23   delay(500); // Small delay
24 }
25
```



INTERN CALL

Output:

The street lights will be controlled automatically based on the ambient light levels detected by the LDR sensor. During daytime (when the sun is up), the street lights will be turned off, conserving energy. As the sun sets, the LDR sensor will detect low light levels and trigger the street lights to turn on, ensuring safe and well-lit streets during the night.



Weather Monitoring System

Abstract:

This project involves building a Weather Monitoring System using an Arduino Uno and an LM35 temperature sensor. The system categorizes temperature into three levels: low, medium, and high, and indicates them using LEDs. This enables users to easily interpret temperature conditions.

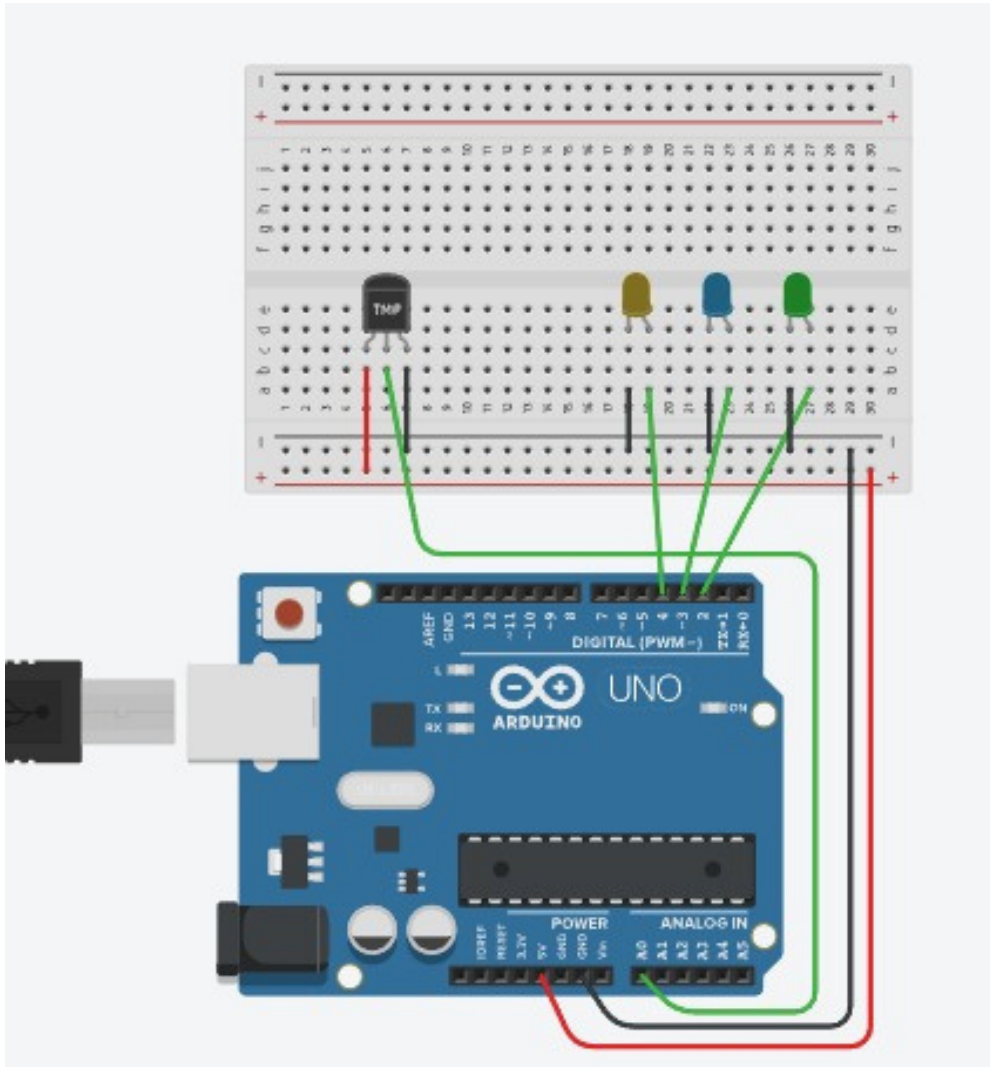
Components Required:

1. Arduino Uno
2. LM35 temperature sensor
3. Jumper wires
4. Breadboard
5. LEDs (for indicating temperature levels)
6. USB cable for uploading code to Arduino
7. Tinkercad account (for simulation)



INTERN CALL

Circuit Diagram:



Code:



INTERN CALL

```
1 void setup() {
2   pinMode(A0, INPUT);
3   pinMode(2, OUTPUT);
4   pinMode(3, OUTPUT);
5   pinMode(4, OUTPUT);
6   Serial.begin(9600);
7 }
8
9 void loop() {
10   int temperature = analogRead(A0);
11
12   Serial.println(temperature);
13
14   if (temperature >= 100) {
15     digitalWrite(2, HIGH);
16   }
17   else{
18     digitalWrite(2, LOW);
19   }
20
21   if (temperature >= 200) {
22     digitalWrite(3, HIGH);
23   }
24   else{
25     digitalWrite(3, LOW);
26   }
27
28   if (temperature >= 300) {
29     digitalWrite(4, HIGH);
30   }
31   else{
32     digitalWrite(4, LOW);
33   }
34
35   delay(1000);
36 }
37
```



Output:

The Weather Monitoring System will provide visual feedback using LEDs to indicate the current temperature level. Based on the temperature readings from the LM35 sensor, if the temperature is considered cool, a low temperature LED will light up. If the temperature is within the normal range, a medium temperature LED will illuminate. Lastly, if the temperature is high, the high temperature LED will be activated. This system offers a simple and intuitive way to monitor temperature conditions.

Smart Watering Alert



INTERN CALL

Abstract:

The Smart Watering Alert system utilizes an Arduino Uno along with a moisture sensor to detect soil moisture levels. When the soil is dry, the system triggers an alert through both sound (buzzer) and light (LED) to notify users that the plants require watering. Conversely, when the soil is wet, no alert is given, allowing for regular watering routines to proceed undisturbed.

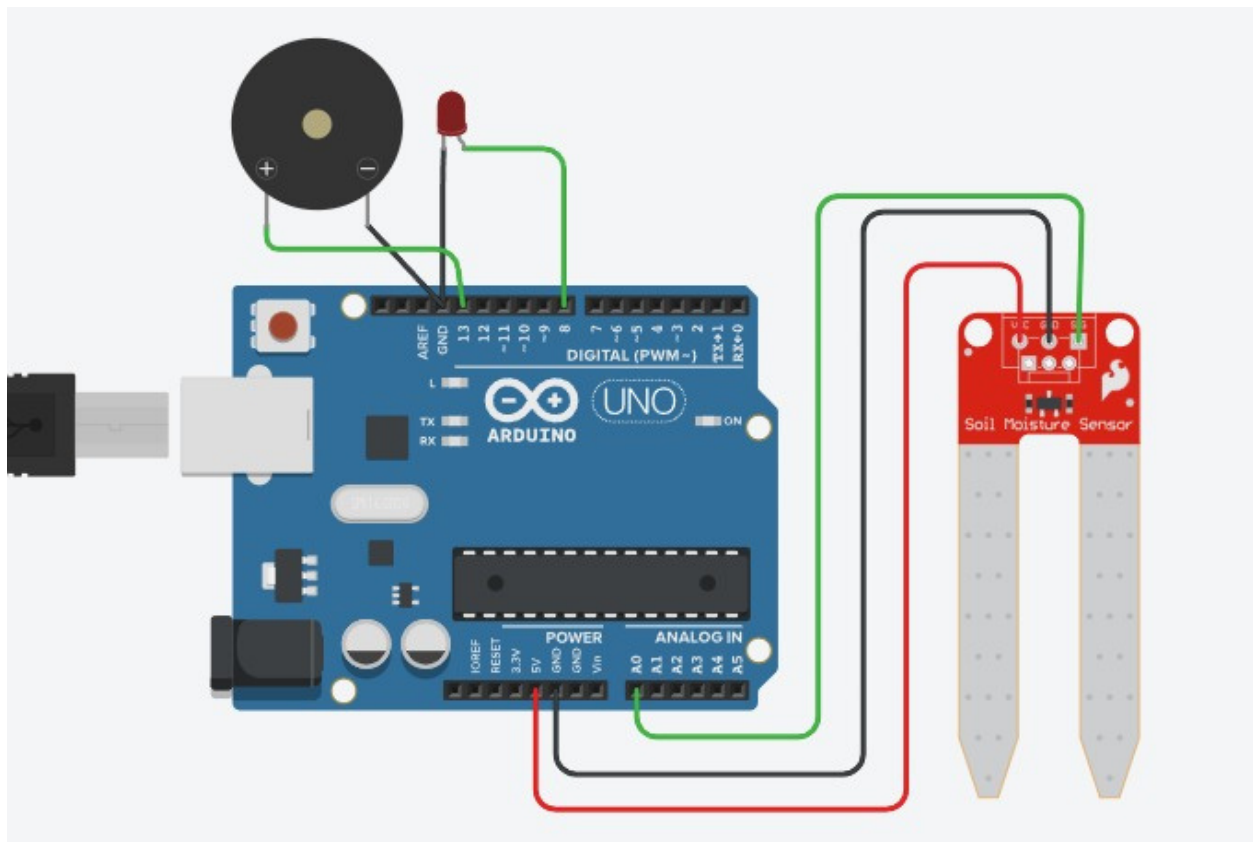
Components Required:

1. Arduino Uno
2. Moisture sensor
3. Jumper wires
4. Breadboard
5. Buzzer (piezo)
6. LEDs
7. USB cable for uploading code to Arduino
8. Tinkercad account (for simulation)



INTERN CALL

Circuit Diagram:





INTERN CALL

Code 1: Without knowing the status of the soil in monitor.

```
1  const int moistureSensorPin = A0; // Analog pin connected to the moisture sensor
2  const int ledPin = 8; // Built-in LED pin
3
4  void setup() {
5      pinMode(moistureSensorPin, INPUT);
6      pinMode(ledPin, OUTPUT);
7  }
8
9  void loop() {
10     // Read moisture level from the sensor
11     int moistureLevel = analogRead(moistureSensorPin);
12
13     // Check if soil is dry (you may need to adjust this threshold)
14     if (moistureLevel < 500) {
15         digitalWrite(ledPin, HIGH); // Turn on LED
16     } else {
17         digitalWrite(ledPin, LOW); // Turn off LED
18     }
19
20     delay(1000); // Delay for stability
21 }
```



INTERN CALL

Code 2: By knowing the soil condition in serial monitor.

```
1  const int moistureSensorPin = A0; // Analog pin connected to the moisture sensor
2  const int ledPin = 8; // Built-in LED pin
3  void setup() {
4      pinMode(moistureSensorPin, INPUT);
5      pinMode(ledPin, OUTPUT);
6      Serial.begin(9600);
7  }
8  void loop() {
9      // Read moisture level from the sensor
10     int moistureLevel = analogRead(moistureSensorPin);
11     // Check if soil is dry (you may need to adjust this threshold)
12     if (moistureLevel < 500) {
13         digitalWrite(ledPin, HIGH); // Turn on LED
14         Serial.println("Dry");
15     } else {
16         digitalWrite(ledPin, LOW); // Turn off LED
17         Serial.println("Wet");
18     }
19     delay(1000); // Delay for stability
20 }
```



Output:

When the soil moisture level is detected as dry by the moisture sensor, the Smart Watering Alert system activates both the buzzer and LED to alert users. This combined audio and visual alert serves as a prompt for watering the plants. In contrast, when the soil moisture level is sufficient (wet), no alert is given, allowing for normal watering routines to continue uninterrupted.