Automatic Light Controller using LDR and Arduino UNO Controller

AIM:

To design and simulate an automatic light controller using an LDR and Arduino UNO controller in Proteus.

Software Required:

- Arduino IDE
- Proteus

PROCEDURE:

Arduino IDE

Step 1: Open the Arduino IDE
Step 2: Go to "File" and select "New File" option
Step 3: Type the program
Step 4: Go to "File" and select "Save" option to save the program
Step 5: Go to "Sketch" and select "Verify/Compile" options
Step 6: If no error, a Hex file will be generated in the temporary folder

Proteus

Step 7: Open the Proteus software
Step 8: Go to "File", select "New Design", and click "OK" button
Step 9: Select component mode and click "Pick Devices" from the library
Step 10: Type the component name in the keyword to select the components and click "OK" button
Step 11: Design the circuit as per the diagram
Step 12: Double click the Arduino controller and upload the hex file generated by Arduino IDE
Step 13: Click "Start" button and check the output

THEORY:

A light sensor and street light control system using Arduino is designed to measure the intensity of light. It automatically controls street lights based on the light intensity detected. In this experiment, we utilize the Arduino UNO R3 for implementation. A Light-Dependent Resistor (LDR) is used for light detection. To provide isolation between the Arduino and the 220V AC street light, a relay is incorporated.

Light Dependent Resistor (LDR)

A Light Dependent Resistor (LDR) is used to detect changes in light intensity or as a light sensor. An LDR is essentially a variable resistor that changes its resistance with the intensity of light. When the light intensity is high, the LDR has low resistance, and when the light intensity decreases, the LDR offers high resistance. Therefore, there is an inverse relationship between the intensity of light and the resistance of the LDR. To measure the resistance of an LDR and calculate the intensity of light, we can use the Arduino UNO R3 board, which has six analog-to-digital converter channels. However, these channels can only measure voltage and cannot measure resistance directly. To overcome this, we use signal conditioning by connecting a 10K ohm resistor in series with the LDR and a 5-volt source. This circuit converts the resistance into voltage form, which can then be measured using the analog-to-digital converter of the Arduino. The measured voltage across the LDR can be converted back into resistance using the voltage division formula.

Relay

In this experiment, a relay is used to provide isolation between low-voltage circuitry and high-voltage circuitry. Arduino is also used to provide a control signal to the relay whenever the intensity of light falls below a certain level. The control signal is generated from pin 13 of Arduino, which is used as an output pin.

Light Sensor with Arduino

The circuit diagram shown in the image below represents a light sensor and street light control system using Arduino. Let’s break down the components and their functions:

- **Arduino UNO R3:** The Arduino UNO R3 board serves as the main microcontroller in this project. It receives input from the light sensor and controls the street lights based on the detected light intensity.
- **Light Dependent Resistor (LDR):** The LDR is a sensor that detects changes in the intensity of light. It is connected in series with a 10K ohm resistor and a 5-volt power supply. The resistance of the LDR varies with the intensity of light falling upon it.
- **Analog-to-Digital Converter (ADC):** The Arduino UNO R3 board has six ADC channels that can measure voltage. In this project, the measured voltage across the LDR is converted into a corresponding resistance value using a voltage divider circuit. This resistance value is then converted into a digital value by the ADC for further processing.
- **Relay:** The relay is used to provide isolation between the low-voltage circuitry (Arduino) and the high-voltage circuitry (street lights). It acts as a switch that is

controlled by the Arduino. When the light intensity falls below a certain level, the Arduino sends a control signal to the relay, which then switches on or off the street lights accordingly.

Overall, this circuit diagram showcases how the Arduino, LDR, ADC, relay, and transistor are interconnected to create a light sensor and street light control system. The Arduino reads the resistance value of the LDR, converts it into a digital value, and based on that, controls the street lights using the relay.

LDR Features:

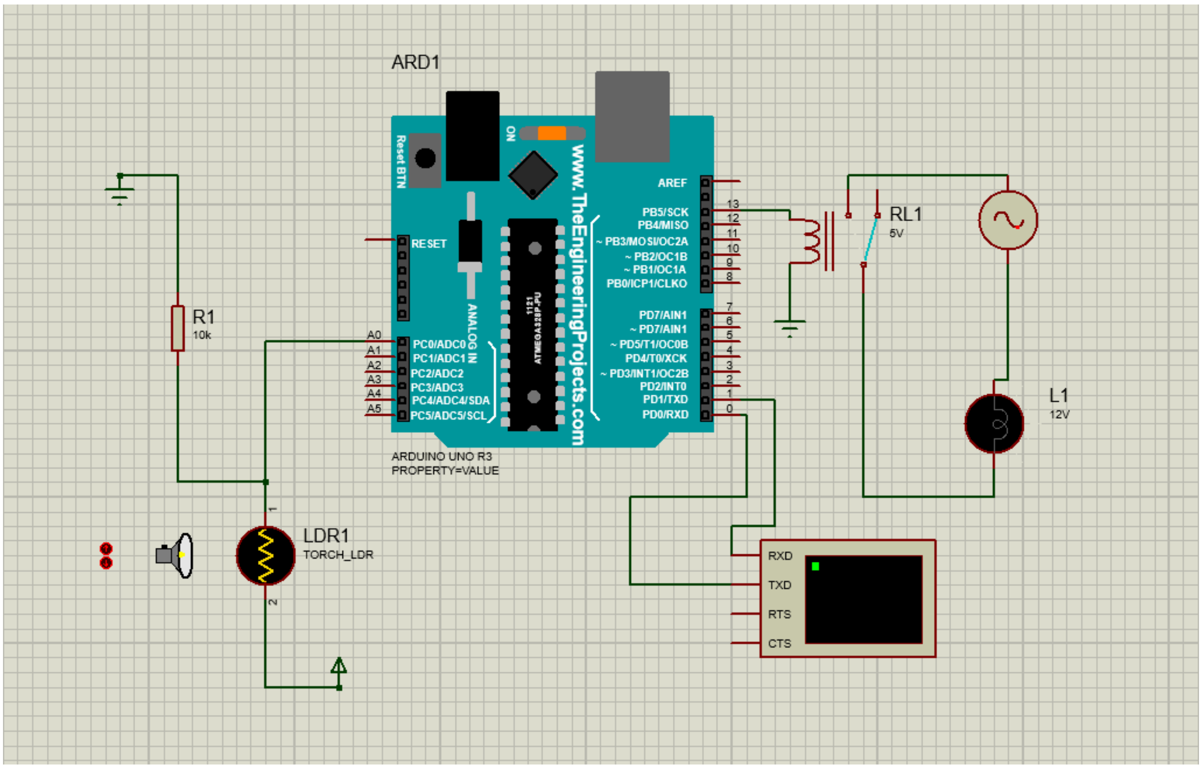
Features of LDR are as follows:

1. High reliability.
2. Light weight.
3. Wide spectral response.
4. Wide ambient temperature range.

PROGRAM:

```
// Define LDR pin

const int ldrPin = A0; // LDR connected to Analog pin A0

// Define Relay pin

const int relayPin = 13; // Relay connected to Digital pin 13

// Define threshold for light intensity

// This value can be adjusted based on ambient light conditions

const int lightThreshold = 500; // Adjust as needed (0-1023)

void setup() {

  // Initialize Serial communication for debugging

  Serial.begin(9600);

  // Set relay pin as an output

  pinMode(relayPin, OUTPUT);

  // Initially turn off the light
  digitalWrite(relayPin, LOW);
}
void loop() {
  // Read the analog value from the LDR
  int ldrValue = analogRead(ldrPin);
  // Print LDR value to Serial Monitor for debugging
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);
  // Check if light intensity is below the threshold (it's dark)
  if (ldrValue < lightThreshold) {
    // Turn on the street light
    digitalWrite(relayPin, HIGH);
    Serial.println("Street light ON");
  } else {
    // Turn off the street light
    digitalWrite(relayPin, LOW);
    Serial.println("Street light OFF");
  }
  // Small delay to avoid rapid switching and stabilize readings
  delay(1000); // Delay for 1 second
}
```

OUTPUT:



RESULT:

Thus, the automatic light controller was designed and simulated using an LDR and Arduino UNO controller.