**Slip 7: To write a program for LDR to vary the light intensity of LED using Arduino.(in proteus)**

Below is a simple Arduino program to control the brightness of an LED based on the light  intensity detected by a Light Dependent Resistor (LDR). The LDR changes its resistance based  on the light falling on it, allowing us to read light intensity values.

**Components Needed:**

- Arduino (e.g., Arduino Uno)

- LDR (Light Dependent Resistor)

- LED

- Resistor

-Virtual Terminal

**Circuit Connections:**

1. LDR Setup:

- Connect one terminal of the LDR to the 5V pin on the Arduino.

- Connect the other terminal of the LDR to one terminal of a 10k-ohm resistor. Connect the other terminal of the resistor to the ground (GND).

- Connect the point between the LDR and the resistor to an analog pin (e.g., A0) on the Arduino.  This

point gives a voltage that varies with light intensity.

2. LED Setup:

- Connect the anode (long leg) of the LED to a digital pin (e.g., D9) on the Arduino through a  220-ohm

resistor.

- Connect the cathode (short leg) of the LED to the ground (GND).

3. Virtual terminal setup: Add a virtual terminal to the schematic capture and connect the RX of  virtual terminal to the TXD of Arduino component and TX of virtual terminal to RXD of  Arduino component.

1

**Arduino code:**

**Step 1: Write this code in main.py**

// Pin definitions

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

const int ledPin = 9; // LED connected to digital pin 9

void setup() {

 pinMode(ledPin, OUTPUT); // Set LED pin as output

 Serial.begin(9600); // Initialize serial communication for debugging }

void loop() {

 int ldrValue = analogRead(ldrPin); // Read the LDR value (0-1023)

 // Map LDR value to PWM range (0-255)

 int ledBrightness = map(ldrValue, 0, 1023, 255, 0); // Invert for brightness control

 // Set the LED brightness

 analogWrite(ledPin, ledBrightness);

 // Print the LDR value for debugging

 Serial.print("LDR Value: ");

 Serial.print(ldrValue);

 Serial.print(" | LED Brightness: ");

 Serial.println(ledBrightness);

 delay(100); // Small delay for stability

}

2 

**Step 2:** Copy the same code in Arduino ide

**Step 3:** Choose the correct board (Arduino Uno)

**Step 4:** Save the sketch and compile it.

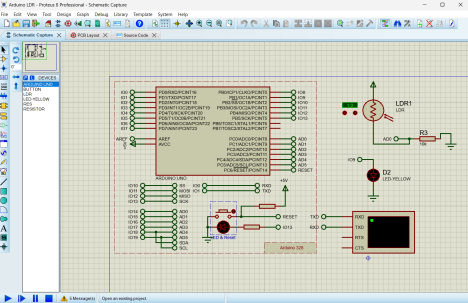
**Step 5:** Go to Sketch > Export Compiled Binary. This will save the .hex file in  your Arduino project folder.

**STEP 6:** Go to proteus and double click on the Arduino component.

**STEP 7:** Find the field for the program file or hex file, and browse to select the  .hex file you exported from the Arduino IDE.

**STEP 9:** Run the simulation.

**Connections:**

****3