LAUNCH THE PROJECT - 10

Light Meter

(using LDR)

On AYNOP® UNO Launchpad Kit

Author: AYNOP Enterprises | Doc Version: 1.0 | Date: 18-9-2025

Table of Contents

1. Overview

This project demonstrates how to use a **photoresistor (LDR)** to measure light levels with the Arduino® UNO R4 Minima. The light intensity is displayed on the **Serial Monitor** as both raw ADC values and a percentage.

You will learn to:

- Connect an LDR in a voltage divider circuit.
- Read analog input values using analogRead()**.**
- Map raw readings to a light percentage.
- Control an LED based on brightness thresholds.

2. Components Required

- Arduino® UNO R4 Minima board
- USB Type-C data cable
- 1 × Photoresistor (LDR)
- 1 × Resistor (10 kΩ recommended, for voltage divider)
- Breadboard
- Jumper wires

3. Software Required

- Arduino IDE (v2.3.6 or later recommended)
- No additional libraries required.

Note:

We assume the **Arduino UNO R4 Minima board package** is already installed on your machine, as explained in the 00_Getting_Started/00_GettingStarted_Arduino_R4_Minima guide. If it is not installed, please refer to that document and complete the installation before proceeding.

4. Hardware Setup

This section explains how to connect the components for the **LIGHT Meter** project. It includes a **Wiring Diagram** and a **Circuit Schematic**.

4.1 Wiring Diagram

The **photoresistor (LDR)** is connected as part of a **voltage divider circuit**. This allows the Arduino to measure changing light levels as varying voltages.

- Connection order**: VCC → LDR → Resistor → GND**.
- The junction between the **LDR and resistor** is connected to **analog pin A0**.
- The onboard LED (pin D13) is already on the board and needs no extra wiring**.**

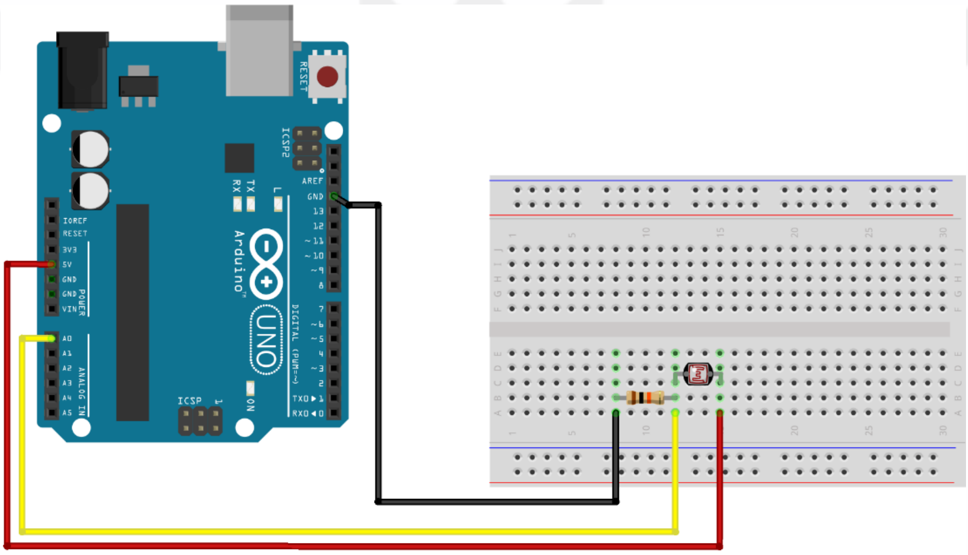


Figure 4.1 – Wiring diagram for LDR Light Meter project

💡 **Tip:** Always disconnect the USB cable before making or changing hardware connections.

4.2 Circuit Schematic

- **VCC → LDR → A0 → Resistor → GND**
- **Onboard LED → pin 13 → GND**

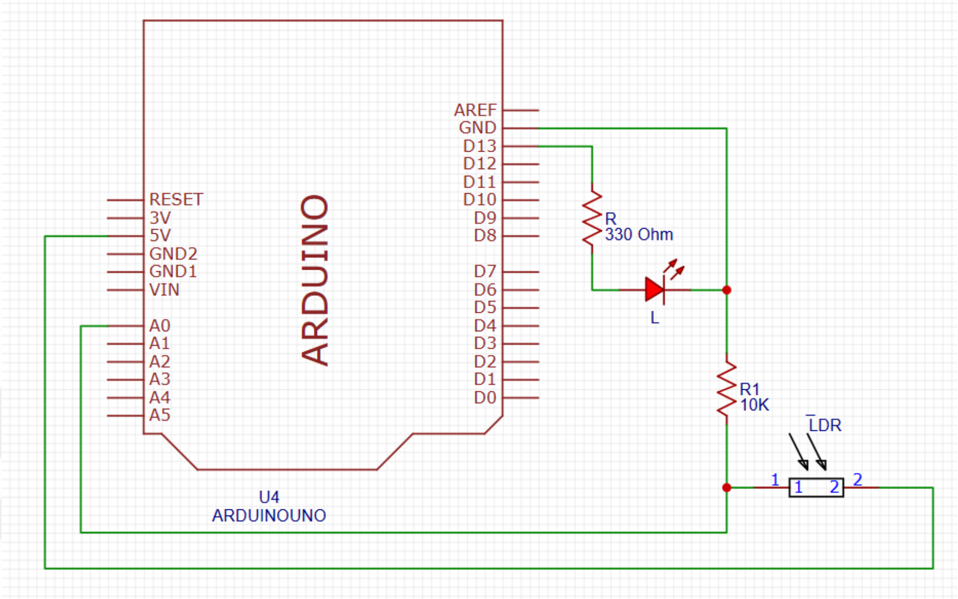


Figure 4.2 – Circuit schematic for LDR Light Meter project

5. Principle – How It Works

A **photoresistor (LDR)** changes its resistance depending on light intensity:

- **Bright light → Low resistance.**
- **Darkness → High resistance.**

In this project, the LDR is combined with a fixed resistor in a **voltage divider**:

- One end of the LDR is connected to **5V (VCC)**.
- The other end goes to **analog pin A0**, then continues through a **10 kΩ resistor** to **GND.**
- This way, the voltage at A0 changes with light intensity

The Arduino measures this voltage using its **10-bit ADC** (*analogRead*() gives values between **0 and 1023**)

- High light → higher voltage at A0 → higher reading
- Darkness → lower voltage at A0 → lower reading

The code then:

- Uses a threshold (**darkThreshold**) to decide if it's dark enough to turn ON the onboard LED.
- Maps the raw reading into a **percentage (0–100%)** for easier understanding.

6. Procedure – Steps to Run

1. **Build the Circuit**
   a. Assemble the connections for light meter project as shown in the **Wiring Diagram (Figure 4.1)**.

2. **Connect the Board**
   a. Use a USB Type-C data cable to connect your UNO R4 Minima to your computer.

3. **Open the Project Code**
   a. Simply **double-click** the file *10_Light_Meter.ino* in the project folder, and it will open directly in the Arduino IDE (if installed).

4. **Confirm Board Selection**
   a. The IDE usually auto-detects the UNO R4 Minima if the package is installed.
   b. If not installed, refer to the *00_Getting_Started/00_GettingStarted_Arduino_R4_Minima* document to install the necessary board package.
   c. Verify that *Arduino UNO R4 Minima* is selected in the IDE's board selector (top toolbar).

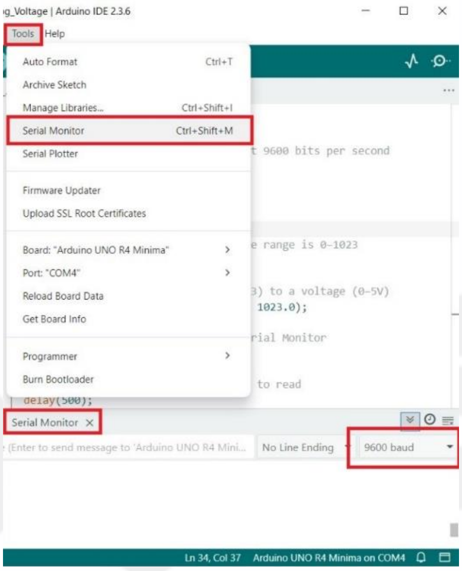
5. **Upload the Code**
   a. Click the **Upload** button (arrow icon) in the top-left corner of the IDE.
   b. Wait until the console displays "**Done uploading**."

6. **Observe the Behaviour**
   a. Move your hand or turn off all-light source near the LDR.
   b. The onboard LED turns ON in dark conditions (below threshold)
   c. Observe the real time raw ADC value and percentage of light **on** Serial Monitor.

6.1 How to Use the Serial Monitor:

The Serial Monitor is a **tool in the Arduino IDE** that lets you see text/data sent by the microcontroller (MCU) over the USB connection.

- In Arduino IDE, go to **Tools → Serial Monitor**, or press **Ctrl + Shift + M**.
- A new window will open showing real-time data from your UNO R4 Minima.
- At the bottom-right of the Serial Monitor window, set the baud rate to **9600**. This must match the *Serial.begin*(9600) in the code.


g_Voltage | Arduino IDE 2.3.6
Tools
Help
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "Arduino UNO R4 Minima"
Port: "COM4"
Reload Board Data
Get Board Info
Programmer
Burn Bootloader
Serial Monitor
(Enter to send message to 'Arduino UNO R4 Mini...
No Line Ending
9600 baud
Ln 34, Col 37 Arduino UNO R4 Minima on COM4

```
t 9600 bits per second
e range is 0-1023
3) to a voltage (0-5V)
 1023.0);
rial Monitor
 to read
delay(500);
```

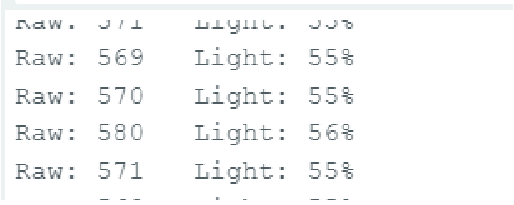
Figure 6.1 – Arduino IDE Serial Monitor set to 9600 baud

💡 **Tip:** If the Serial Monitor shows nothing, check that:

- The correct **COM port** is selected.
- The baud rate is set to **9600**

7. Expected Output

- When light is bright → high raw value, LED OFF.
- When it's dark → low raw value, LED ON.
- The Serial Monitor will show values like:



```
Raw: 569   Light: 55%
Raw: 570   Light: 55%
Raw: 580   Light: 56%
Raw: 571   Light: 55%
```

8. Code

The source code for this project is included in the downloaded folder:

📁 *uno-launchpad-kit/01_Basic_Projects/10_Light_Meter/10_Light_Meter.ino*

👉 **Tip:**

- To open the project, simply **double-click the .ino file**. If the Arduino IDE is installed, it will launch automatically and load the code.
- If you **haven't installed the Arduino IDE yet**, please refer to:
  📁 uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima
  to **download and install it.**

8.1 Function References

- setup() – runs once when the board is powered on or reset.
- loop() – runs continuously after setup() finishes.
- analogRead(pin) – reads analog input (0–1023).
- map(value, fromLow, fromHigh, toLow, toHigh) – re-maps values to a different range.
- digitalWrite(pin, value) – writes HIGH/LOW to a pin.
- pinMode(pin, mode) – configures pin as INPUT/OUTPUT.
- Serial.begin(baudrate) – starts serial communication.
- Serial.print() / Serial.println() – prints to Serial Monitor.
- delay(ms) – pauses execution.

📚 **For more details and advanced usage, visit:**
🔗 Arduino Language Reference — The official guide for all Arduino functions.

9. Troubleshooting Tips

- **No output on Serial Monitor?**
  - Ensure baud rate is set to **9600**.
  - Ensure Serial Monitor is open after upload.

- **LED behaviour inverted?**
  - Swap LDR and resistor positions in the voltage divider.
  - Adjust the **darkThreshold** value in the code

- **LED never turns ON or OFF?**
  - Test with a flashlight or cover the LDR.
  - Try lowering or raising **darkThreshold**.

- **Upload error in Arduino IDE?**
  - Verify that the correct board (**Arduino UNO R4 Minima**) is selected in the IDE.
  - Check that the correct **COM port** is chosen.

- **Board not detected via USB?**
  - Ensure you are using a **data-capable USB Type-C cable** (some cables only provide charging).
  - Try reconnecting the cable or using a different USB port.

- **Board not listed in Arduino IDE?**
  - If you don't see **Arduino UNO R4 Minima** in the board selector, the **board package is not installed**.
  - To fix this, follow the installation steps in:
    📂 *uno-launchpad-kit/00_Getting_Started/00_GettingStarted_Arduino_R4_Minima*

💡 **Tip:** If nothing works, press the **RESET** button on the UNO R4 Minima and try uploading the code again.

10. License

This content (source code and documentation) is licensed under **Creative Commons BY-NC-SA 4.0**.
© 2025 AYNOP. You may use, modify, and share for personal and educational purposes only.
Commercial use or redistribution without prior written permission is strictly prohibited.
Refer to the **LICENSE** file for complete details.

11. Support & Feedback

We value your feedback and are happy to assist with any questions, troubleshooting, or suggestions you may have.

✉ Email: *support@aynop.com*

When sending an email, please include your kit name (AYNOP® UNO Launchpad Kit) and, if applicable, the project name in the subject line. This will help our team respond faster and more accurately. We aim to respond to all queries within 2–3 business days. Your feedback helps us improve our products and create even better learning experiences for all Arduino beginners.