

## Task 1: Button Press – Digital Input & Output

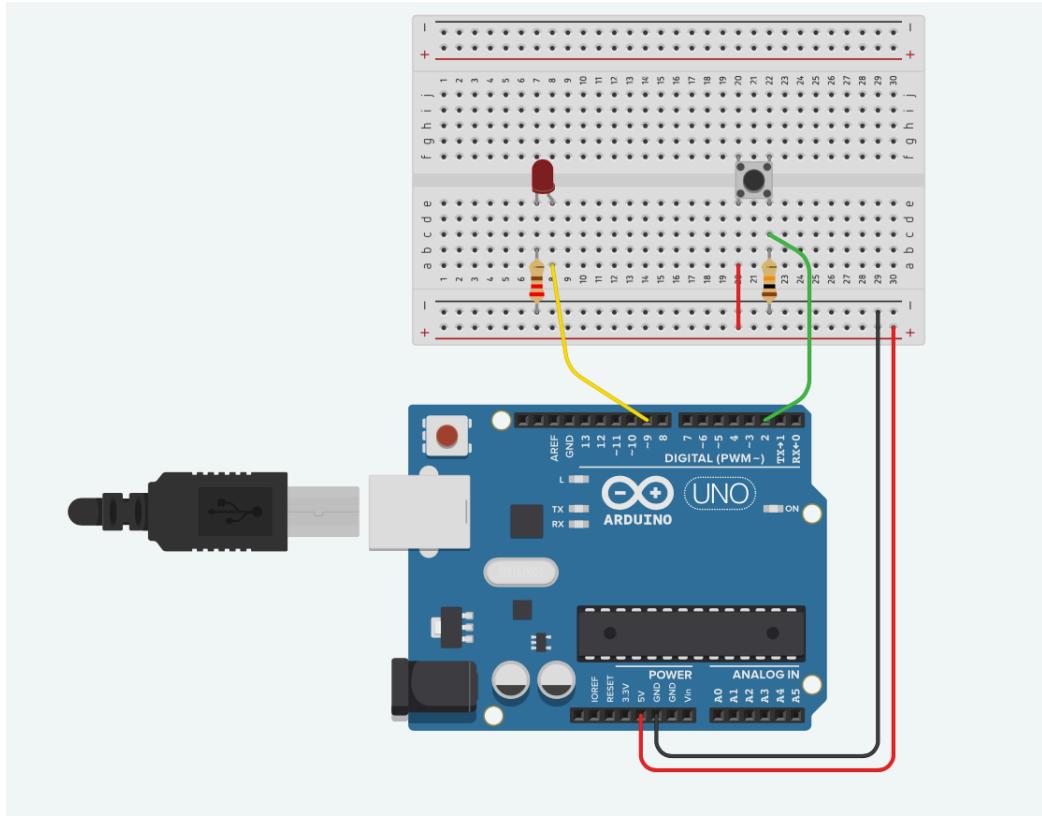
### Objective

Use a **button** to control an **LED** (digital only). Step 2, add a potentiometer to dim the LED when the button is held.

### Components

- 1x LED
- 1x Button
- 1x Potentiometer (for step 2)
- 2x Resistors (220Ω, 10kΩ)
- Breadboard and jumper wires

### Circuit diagram



### Task 01 - Starter Code

```
// Code to use a button to switch on an LED

int ledPin = 9;      // define the pin for the LED output
int buttonPin = 2; // define the pin for the button input

void setup(){
    // set the pinMode for each of the pins we'll need
    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop(){
    // create a variable to hold the state of the button
    // read the state of the button at the begining of every loop
    int buttonState = digitalRead(buttonPin);

    // check if the button is pressed = when the button is pressed
    // the pin should be receiving a 5V / a HIGH signal
    if (buttonState == HIGH){
        // if the button is pressed switch on the LED
        digitalWrite(ledPin, HIGH);
    }else {
        // if the button is not pressed the LED should be off
        digitalWrite(ledPin, LOW);
    }
    // wait for 20 milliseconds before starting the new loop
    delay(20);
}
```

### Expansion Challenge aka. Step 2

Add a potentiometer and change the code so that:

- When the button is held, the LED brightness depends on the potentiometer value (hint: use `analogWrite()` to read the potentiometer).

## Task 2: Light-Controlled LED – Analog Input + Digital Threshold

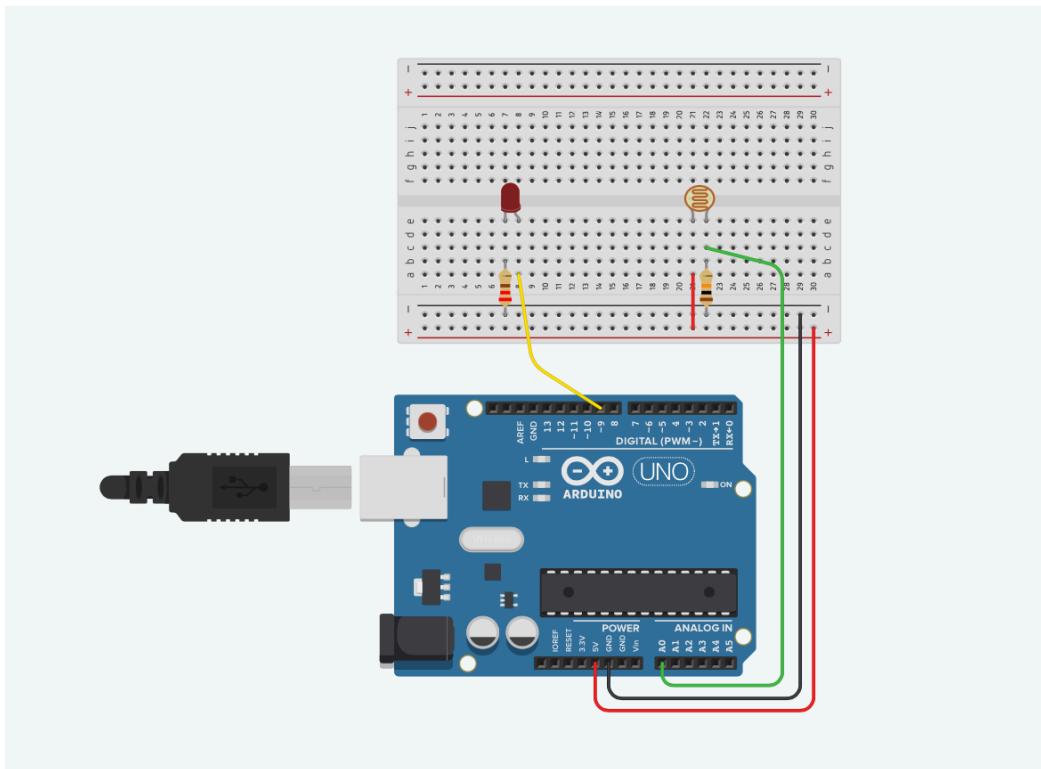
### Objective

Use an **LDR (light sensor)** to detect ambient light and turn on an LED when it gets dark (analog input + digital output). Then fade the LED based on brightness.

### Components

- 1x LDR (Photoresistor)
- 1x LED
- 1x 10kΩ Resistor
- 1x 220Ω Resistor
- Breadboard and jumper wires

### Circuit diagram



## Task 2 - Starter Code

```
// Code to use a light sensor to switch on an LED

int ledPin = 9;      // define the pin for the LED output
int ldrPin = A0; // define the analog pin for the button input
int threshold = 500;

void setup(){
    // set the pinMode for each of the pins we'll need
    pinMode(ledPin, OUTPUT);
}

void loop(){

    // create a variable to hold the level of light
    // read the lightlevel as an analog value expect values from 0-
    1023 (10bit integer)
    int lightlevel = analogRead(ldrPin);

    // check if the lightlevel is below the threshold
    // the analog pin shloud receive a value somewhere between 0-1023
    if (lightlevel < threshold){
        // if the lightlevel is below the threshold switch on the LED
        digitalWrite(ledPin, HIGH);
    }else {
        // if it's "bright enough" switch off the LED
        digitalWrite(ledPin, LOW);
    }
    // wait for 20 milliseconds before starting the new loop
    delay(20);
}
```

## Expansion Challenge

Replace `digitalWrite()` with `analogWrite()` so the LED fades in/out based on light level.

## Task 3: Toggle Modes – Digital Button + Analog Output

### Objective

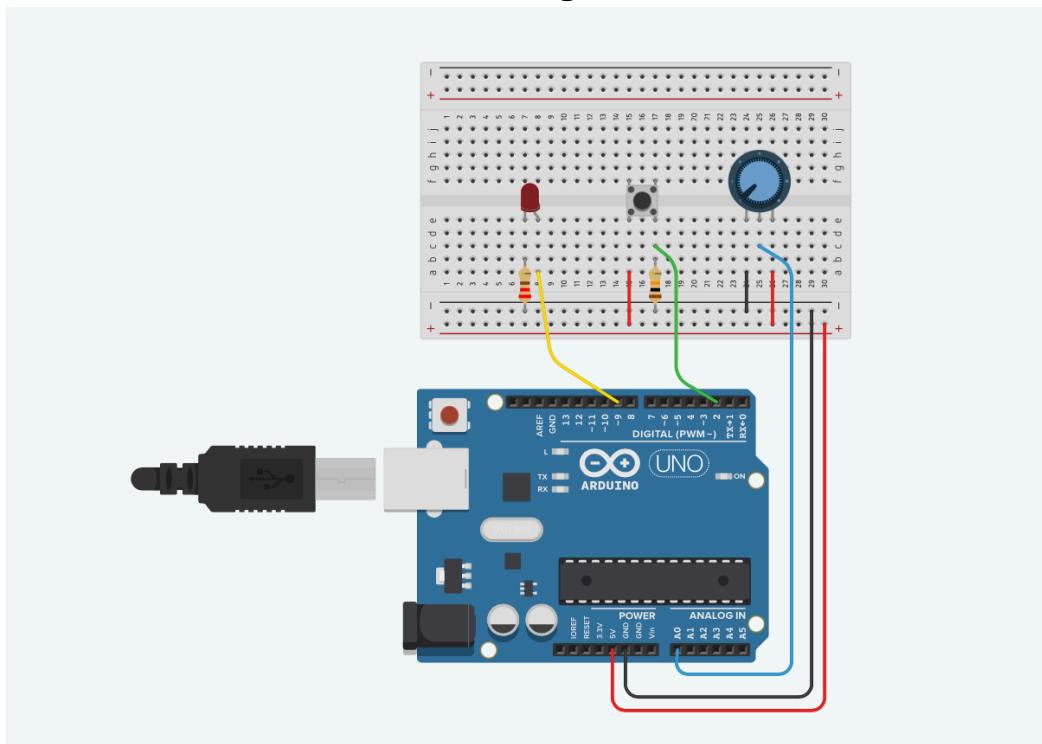
Use a **button** to toggle between two modes:

1. LED is fully ON.
2. LED brightness controlled by a **potentiometer**.

### Components

- 1x LED
- 1x Button
- 1x Potentiometer
- 2x Resistors (220Ω, 10kΩ)
- Breadboard and jumper wires

### Circuit diagram



### Task 3 - Starter Code

```
// Code to use a button to switch between
// analog and digital mode for LED control

int ledPin = 9;      // define the pin for the LED output
int buttonPin = 2; // define the pin for the button input
int potPin = A0;    // define the pin for the potentiometer input
bool mode = false;
bool lastButton = true;

void setup(){
    // set the pinMode for each of the pins we'll need
    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop(){
    // create a variable to hold the state of the button
    int currentButton = digitalRead(buttonPin);

    if (lastButton == true && currentButton == false){
        mode = !mode;
        delay(40); // debounce the button interaction
    }
    // save the state of the button for the next check
    lastButton = currentButton;

    // check of mode is true or false
    // if it is true get the value from the potentiometer
    if (mode == true){
        int ledBrightness = map(analogRead(potPin), 0, 1023, 0, 255);
        analogWrite(ledPin, ledBrightness);
    } else {
        // if the variable mode is false just switch the LED on
        digitalWrite(ledPin, HIGH);
    }
}
```

### Expansion Challenge

Add a second button to reset to “default mode”.