Understanding the Circuit and Code

**Circuit Diagram:**

The provided circuit diagram shows a simple setup with a button connected to pin 2 of an Arduino. A 10k resistor is used as a pull-up resistor, ensuring that the input pin is pulled high when the button is not pressed.

**Code Explanation:**

The code reads the state of the button and prints it to the serial monitor. Here's a breakdown:

1. **setup() function:**
   - Initializes serial communication at a baud rate of 9600 bits per second.
   - Sets pin 2 as an input pin to read the button state.
2. **loop() function:**
   - **Reads the button state:** The digitalRead(2) function reads the value on pin 2. When the button is not pressed, the input pin is pulled high (HIGH state). When the button is pressed, the input pin is connected to ground (LOW state).
   - **Prints the button state:** The Serial.println(buttonState) function prints the read value (1 for HIGH, 0 for LOW) to the serial monitor.
   - **Delay:** A short delay is added to prevent overwhelming the serial monitor with data.

**Why Ground and Input Can Be on the Same Line:**

In this circuit, the button and the resistor create a voltage divider. When the button is not pressed, the resistor pulls the input pin high. When the button is pressed, it effectively shorts the input pin to ground, creating a low voltage.

**Key Points:**

- The 10k resistor acts as a pull-up resistor, ensuring that the input pin is high when the button is not pressed.
- The digitalRead() function is used to read the state of the button.
- The Serial.println() function is used to print the button state to the serial monitor.
- The delay() function prevents excessive serial output.

By understanding these concepts, you can effectively read the state of a button using an Arduino and process the input data.

Video: Push Switch : Working of a Tactile switch

Video: Short and sweet: pull up/pull down resistors explained