How to Make an Analog Button Using Copper Conductive Tape

Materials Needed:

- **Copper conductive tape** (available at electronics or craft stores)
- **Non-conductive base** (such as cardboard, plastic, or wood)}
- **Wires** (for circuit connections)
- **Microcontroller** (Arduino, Raspberry Pi, or your preferred option)
- **Resistors** (optional, to adjust sensitivity, i used a 220 Ohms)
- **Multimeter** (to test continuity)

Steps:

1. Create the button base

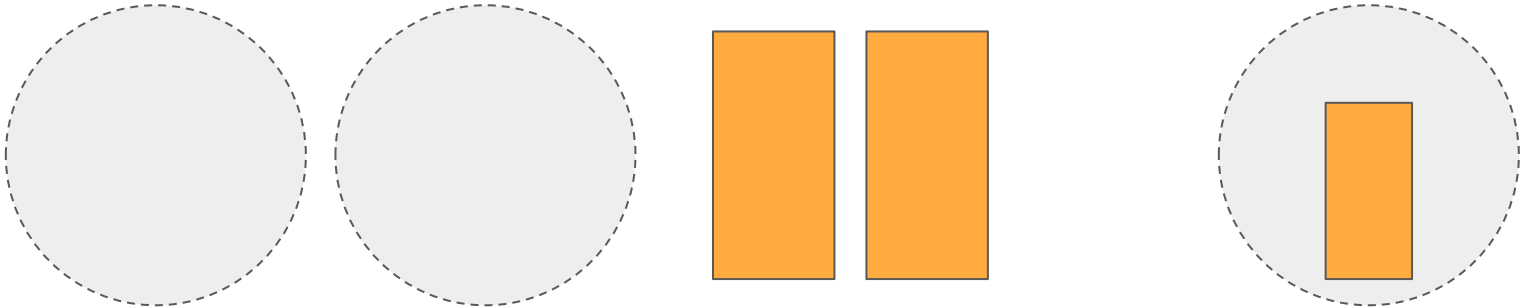
Cut a piece of cardboard or plastic, which will be the base for your button. This material should be non-conductive since it will serve as the support for your circuit.

2. Cut the copper conductive tape

Cut two strips of copper conductive tape, approximately 2-3 cm long (depending on the button size you want).

Place one of the strips on the base (this will be one of the electrodes).

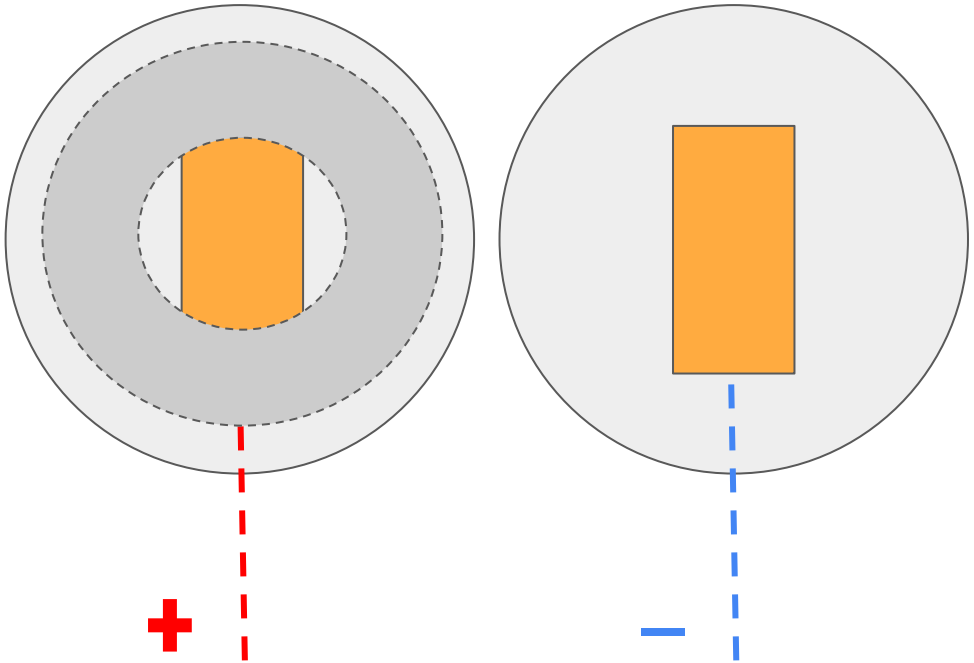
Make sure it's well-adhered without any wrinkles.



Steps:

3. Prepare the contact material

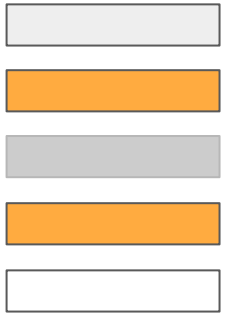
Cut a small piece of conductive foam or any flexible material that you can press and that returns to its original shape. This material will act as the "pusher" that connects the two copper contacts when pressed.



4. Place the second strip of copper tape

Place the second strip of conductive copper tape on top of the contact material, making sure there’s a small air gap between the two copper strips (the one on the base and the one over the contact material).

This will ensure the strips don't touch directly when the button isn't pressed.



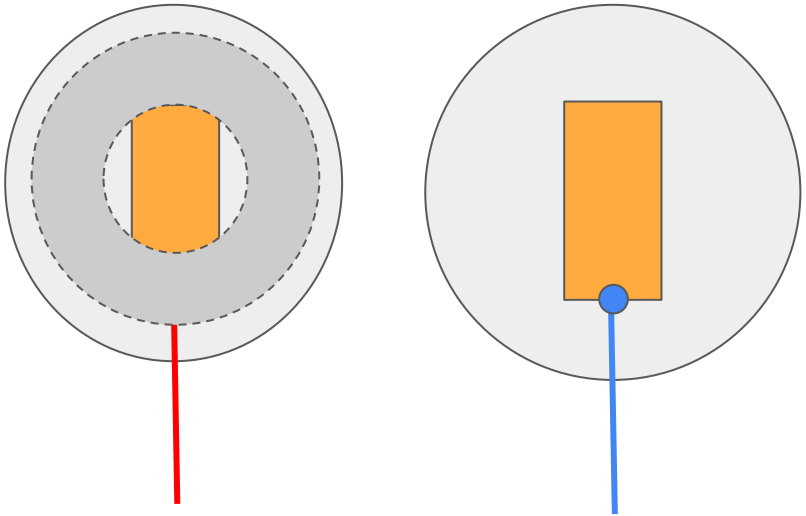
1.Fabric
2. Copper tape
3.Foam
4.Copper tape
5.Fabric

Button Sandwich

6. Connect the wires

Attach a wire to each copper tape strip. You can stick the wire ends directly to the copper tape or solder them for a better connection.

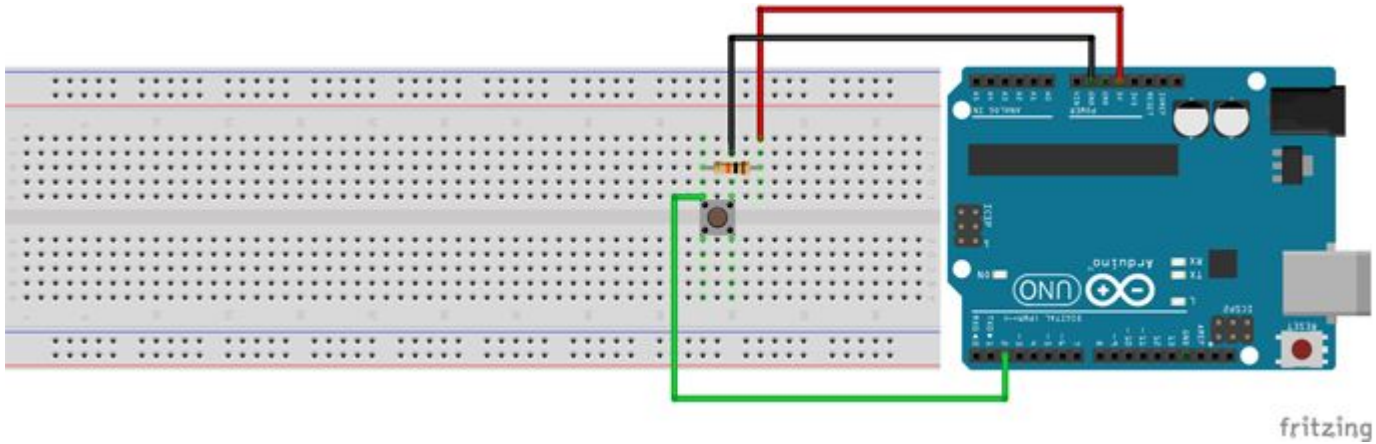
The other end of the wires will connect to the microcontroller inputs (like an Arduino).



7. Programming the button and connections

Connect the microcontroller to your computer.

Set up the inputs in your code (I've attached the code, which simply reads the 3 pin).



8. Code

```
const int pulsadorPin = 3;
int valorPulsador = 0;

void setup() {
    // Activamos los pines de entrada y salida
    pinMode(pulsadorPin, INPUT);
}

void loop() {
    // Leemos el valor del pin
    valorPulsador = digitalRead(pulsadorPin);
  // Encendemos o apagamos el pin del LED según convenga
  if (valorPulsador == HIGH) {
Serial.print ("Encendido");
Serial.println (pulsadorPin);
  }
  else {
Serial.print ("Apagado");
Serial.println (pulsadorPin);
  }
}
```

8. Testing and adjustments

Connect the circuit and open the serial monitor (in the Arduino IDE, for example) to see how the value changes when you press the button.

If needed, adjust the resistor or button pressure for more accurate readings.

If the button isn’t responding correctly, check the connections and use a multimeter to ensure the copper tape is making contact when pressed.

Project Expansion:

Variable sensitivity: You can add more resistors to adjust the button’s sensitivity.

Multiple buttons: Use several strips of copper tape in parallel to create more buttons within the same circuit.