

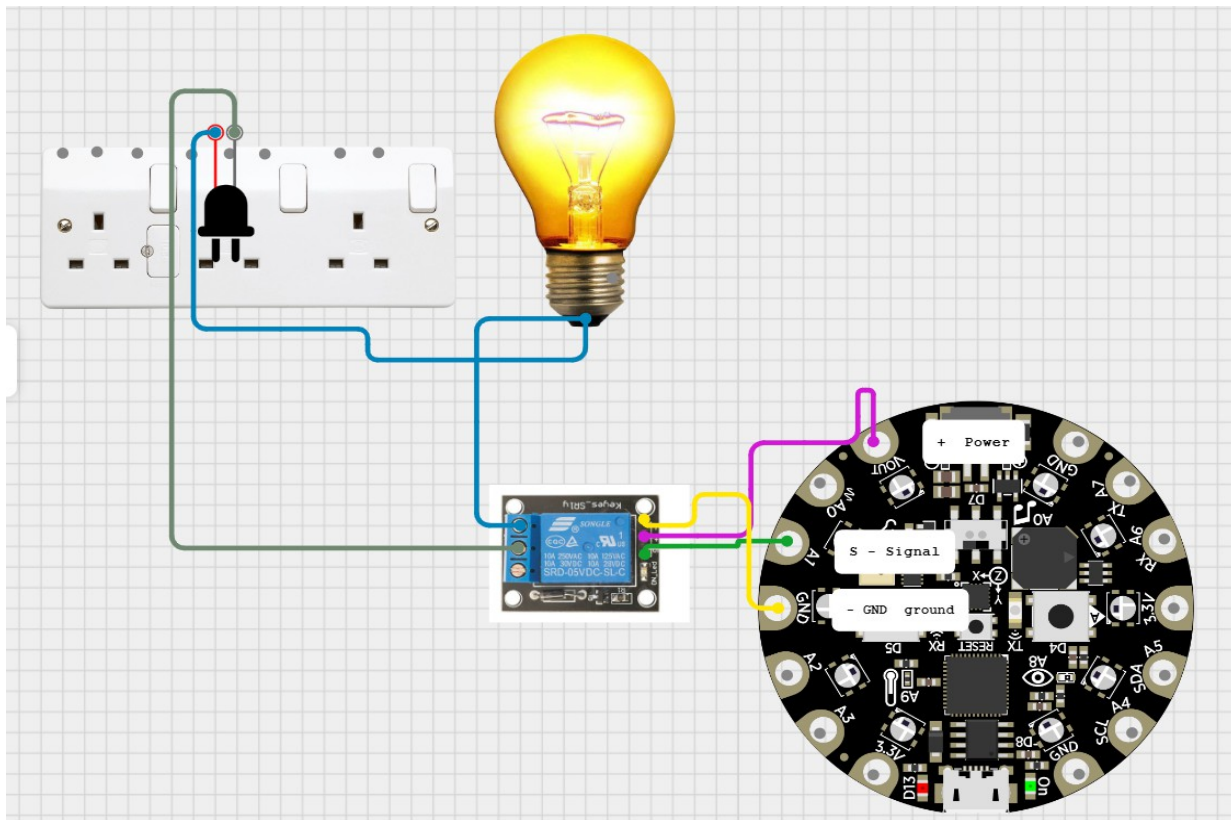
Using relay to set up a circuit

Step 1: Connect the CPX

- Plug in the CPX board to your computer using a USB cable.
- Make sure the CPX board appears as a drive named **CIRCUITPY** when connected. If not, CircuitPython may need to be installed (refer to previous lesson).
- Open the **Mu Editor**.

About Relay

Step 1: Circuit setup



- Relay modules usually have three pins: + (VCC), - (GND), and Signal (IN).
- Connect the Signal (IN) pin of the relay to CPX A1

- Connect the - (GND) pin of the relay to CPX GND
- Connect the + (VCC) pin of the relay to CPX 3.3V (or 5V if required by the relay)
- Connect your external light and battery through the relay using the Normally Open (NO) and Common (COM) terminals
- Connect GND of relay to CPX GND
- Connect VCC of relay to 3.3V (or external 5V if needed)

5V Relay Terminals and Pins



- Connect your external light and battery through the relay (normally open)

Clcik

1. What `relay.value` means in code

- `relay.value = True` → The CPX sends **HIGH (3.3V)** to the relay module → **Energizes** the relay coil.
 - `relay.value = False` → The CPX sends **LOW (0V)** → **De-energizes** the relay coil.
-

2. Relay behavior

a) Normally Open (NO) relay contact

- When `relay.value = False` (**de-energized**) → Circuit is **open** → Device **OFF**.
- When `relay.value = True` (**energized**) → Circuit **closes** → Device **ON**.

b) Normally Closed (NC) relay contact

- When `relay.value = False` (**de-energized**) → Circuit is **closed** → Device **ON**.
 - When `relay.value = True` (**energized**) → Circuit **opens** → Device **OFF**.
-

Summary

- For **NO** contacts: `True = ON`, `False = OFF`
- For **NC** contacts: `True = OFF`, `False = ON`

So, the `relay.value` in code is always about energizing the relay coil, not directly ON/OFF of the device.
The **contact type** (NO or NC) decides how your device behaves.