# How to use a Rotary Encoder Button

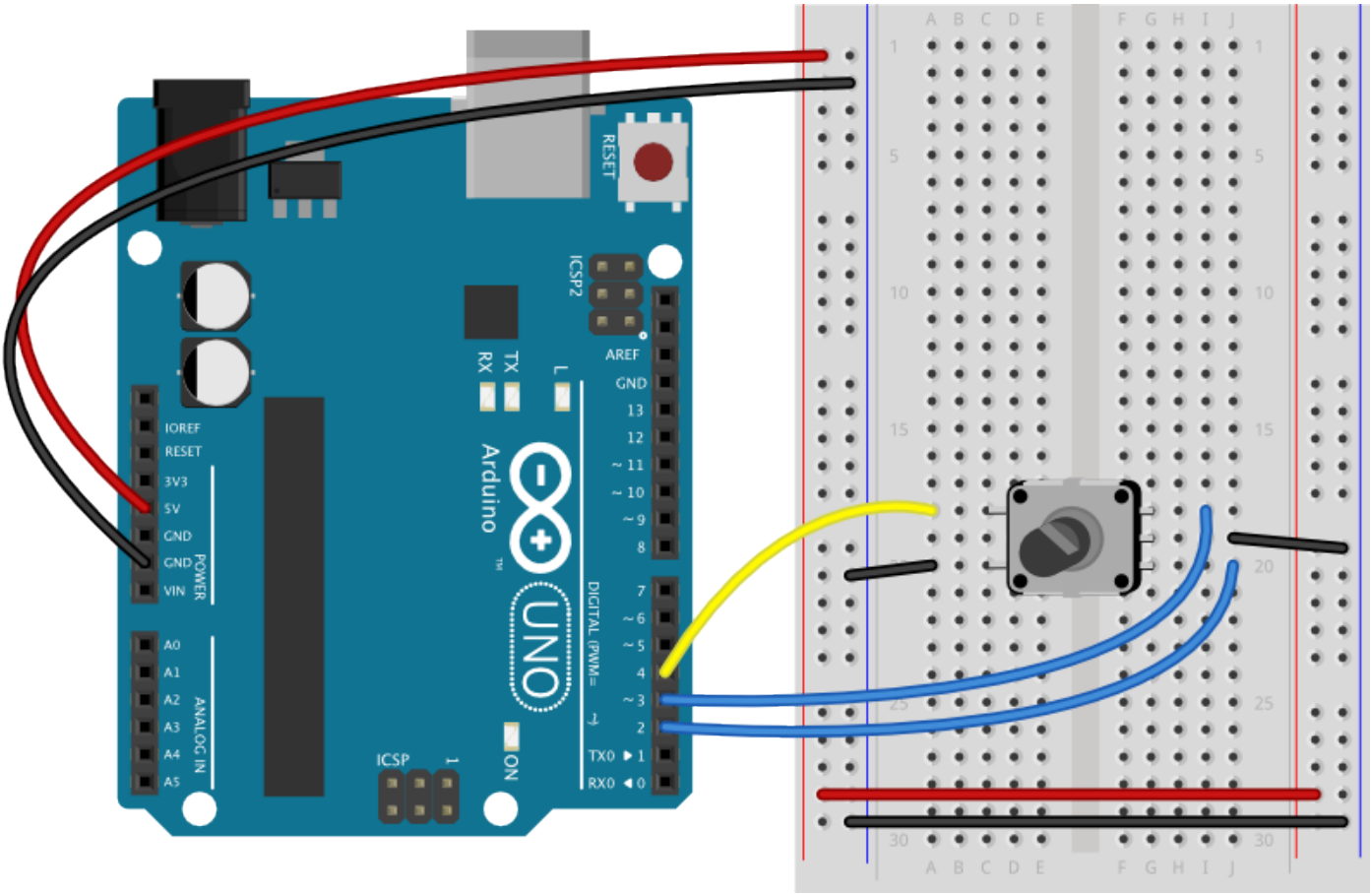
## What is a Rotary Encoder Button?

A rotary encoder button is an input device that combines: a **Rotary encoder** — to detect rotational direction (clockwise/counterclockwise) and steps and a **Push button** — to detect when the knob is pressed down. This component is smaller in size and significantly cheaper than a regular rotary encoder.

This rotary encoder is useful as a rotation sensor or selector and looks similar to potentiometers. These rotary encoders rotate all the way around continuously and are divided up into 24 'segments'. Each segment has a clicky feeling to it, and each movement clockwise or counter-clockwise causes the two switches to open and close.

## Wiring

1. **Top 2 Pins:**
   - one pin to D4
   - one pin to GND
2. **Bottom 3 Pins:**
   - Right Pin to D3
   - middle pin to GND
   - Left Pin to D2




# Library

**EncoderStepCounter** library will be used. We have a tutorial on **how to install a library** here.

# Getting started

```
#include <EncoderStepCounter.h>

const int pin1 = 2;
const int pin2 = 3;

// Create encoder instance:
EncoderStepCounter encoder(pin1, pin2);
```

```
// encoder previous position:
int oldPosition = 0;

const int buttonPin = 4;    // pushbutton pin
int lastButtonState = LOW;  // last button state
int debounceDelay = 5;      // debounce time for the button in ms



void setup() {
  Serial.begin(9600);
  // Initialize encoder
  encoder.begin();
  // Initialize interrupts
  attachInterrupt(digitalPinToInterrupt(pin1), interrupt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(pin2), interrupt, CHANGE);
  // set the button pin as an input_pullup:
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop() {
  // if you're not using interrupts, you need this in the loop:
  encoder.tick();

  // read encoder position:
  int position = encoder.getPosition();
  // read the pushbutton:
  int buttonState = digitalRead(buttonPin);
  //  // if the button has changed:
  if (buttonState != lastButtonState) {
    // debounce the button:
    delay(debounceDelay);
    // if button is pressed:
    if (buttonState == LOW) {
      Serial.print("you pressed on position: ");
      Serial.println(position);
    }
  }
  // save current button state for next time through the loop:
  lastButtonState = buttonState;

  // reset the encoder after 24 steps:
```

```
  if (position % 24 == 0) {
    encoder.reset();
    position = encoder.getPosition();
  }
  // if there's been a change, print it:
  if (position != oldPosition) {
    Serial.println(position);
    oldPosition = position;
  }
}

// Call tick on every change interrupt
void interrupt() {
  encoder.tick();
}
```

---

Revision #2
Created 26 June 2025 13:51:10 by Joanne Leung
Updated 26 June 2025 13:59:33 by Joanne Leung