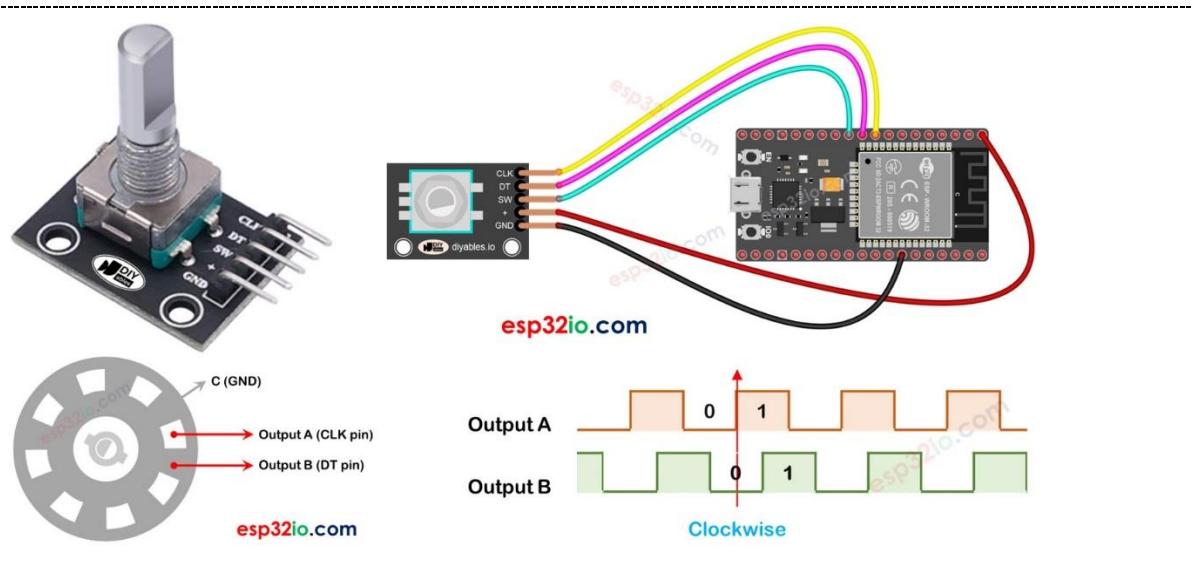


Getting Start ESP32: ESP32 GPIO + ESP32 Interface

Mission 10/12 – ESP32 + Rotary Encoder

1. Read <https://esp32io.com/tutorials/esp32-rotary-encoder>
2. Read <https://www.allnewstep.com/b/318>

Rotary Encoder เช่นเชอร์ตตรวจจับการหมุน สำหรับรับข้อมูลโดยการบิดหมุนที่ตัว Rotary จะได้เลขค่าตามที่เรากำหนดไว้ในการเพิ่มหรือลดค่า ของการหมุนแต่ละล้อ ก็ใช้สำหรับกำหนดค่าขึ้นลง เช่นการปรับโวลุ่มเสียงเพลง



A rotary encoder module has 4 pins:

- ◆ **CLK pin (Output A):** is the main pulse that tells us how much rotation has occurred. Whenever you turn the knob by one detent (click) in either direction, the CLK pin outputs a signal that completes a full cycle (**LOW → HIGH → LOW**).
- ◆ **DT pin (Output B):** acts like the CLK pin but outputs a signal lags behind CLK signal by 90 degrees. It helps us figure out the direction of rotation (clockwise or anticlockwise).
- ◆ **SW pin:** comes from the encoder's button. It's normally open. When we add a pull-up resistor to this pin, the SW pin will be **HIGH** when the knob isn't pushed and **LOW** when it's pushed.
- ◆ **VCC pin (+):** needs to be connected to **VCC** (between 3.3 and 5 volts)
- ◆ **GND pin:** needs to be connected to **GND (0V)**



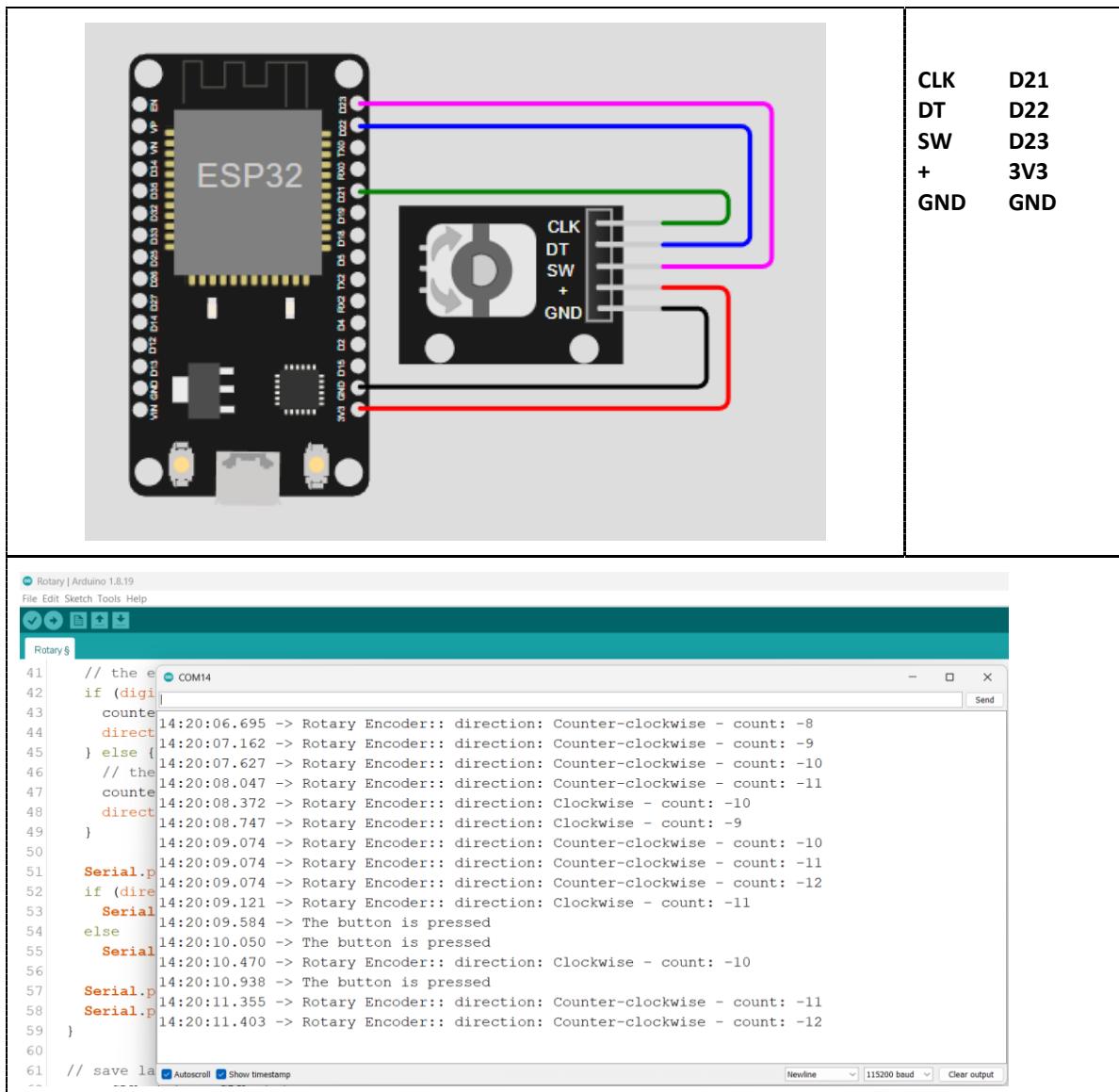
3. Add Library: Sketch → Include Library → Manage
4. Filter with “ezButton”, Select ezButton by ArduinoGetStarted.com Version 1.0.6

Screenshot of the Arduino IDE Library Manager:

The screenshot shows the Arduino IDE Library Manager interface. The search bar at the top contains "ezButton". On the left, the "ezButton" library by "ArduinoGetStarted.com" is listed under "LIBRARY MANAGER" with version 1.0.6 installed. The main area displays the code for "sketch_dec26a.ino" which includes setup() and loop() functions. The "Output" window at the bottom shows the progress of installing the library, with messages like "Downloading ezButton@1.0.6", "ezButton@1.0.6", "Installing ezButton@1.0.6", and "Installed ezButton@1.0.6".

ezButton by ArduinoGetStarted.com Version 1.0.6

5. Test Code “Test1001-Hello RotaryEncoder”



```

/*
This ESP32 code is created by esp32io.com
This ESP32 code is released in the public domain
For more detail (instruction and wiring diagram), visit https://esp32io.com/tutorials/esp32-rotary-encoder
*/
#include <ezButton.h> // the library to use for SW pin

#define CLK_PIN 21 // ESP32 pin GPIO21 connected to the rotary encoder's CLK pin
#define DT_PIN 22 // ESP32 pin GPIO22 connected to the rotary encoder's DT pin
#define SW_PIN 23 // ESP32 pin GPIO23 connected to the rotary encoder's SW pin

#define DIRECTION_CW 0 // clockwise direction
#define DIRECTION_CCW 1 // counter-clockwise direction

int counter = 0;
int direction = DIRECTION_CW;
int CLK_state;
int prev_CLK_state;

ezButton button(SW_PIN); // create ezButton object that attach to pin 7;

void setup() {
  Serial.begin(115200);
  pinMode(CLK_PIN, INPUT);
  pinMode(DT_PIN, INPUT);
  button.setDebounceTime(50); // set debounce time to 50 milliseconds
  // read the initial state of the rotary encoder's CLK pin
  prev_CLK_state = digitalRead(CLK_PIN);
}

void loop() {
  button.loop(); // MUST call the loop() function first

  // read the current state of the rotary encoder's CLK pin
  CLK_state = digitalRead(CLK_PIN);

  // If the state of CLK is changed, then pulse occurred
  // React to only the rising edge (from LOW to HIGH) to avoid double count
  if (CLK_state != prev_CLK_state && CLK_state == HIGH) {
    // if the DT state is HIGH
    // the encoder is rotating in counter-clockwise direction => decrease the counter
    if (digitalRead(DT_PIN) == HIGH) {
      counter--;
      direction = DIRECTION_CCW;
    } else {
      // the encoder is rotating in clockwise direction => increase the counter
      counter++;
      direction = DIRECTION_CW;
    }

    Serial.print("Rotary Encoder:: direction: ");
    if (direction == DIRECTION_CW)
      Serial.print("Clockwise");
    else
      Serial.print("Counter-clockwise");

    Serial.print(" - count: ");
    Serial.println(counter);
  }

  // save last CLK state
  prev_CLK_state = CLK_state;

  if (button.isPressed()) {
    Serial.println("The button is pressed");
  }
}

```