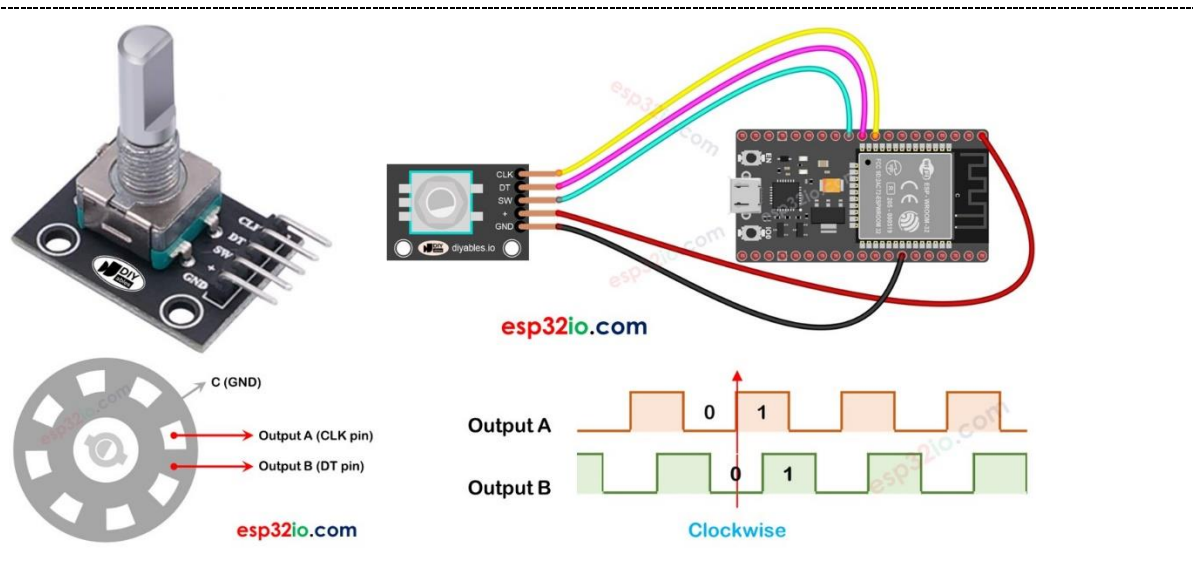


Getting Start ESP32: ESP32 GPIO + ESP32 Interface

Mission 10/12 – ESP32 + Rotary Encoder

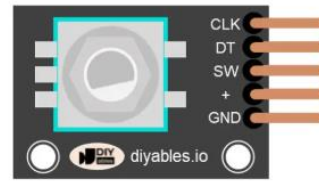
1. Read <https://esp32io.com/tutorials/esp32-rotary-encoder>
2. Read <https://www.allnewstep.com/b/318>

Rotary Encoder เซนเซอร์ตรวจจับการหมุน สำหรับรับข้อมูลโดยการบิดหมุนที่ตัว Rotary จะได้เลขค่าตามที่เรากำหนดไว้ในกาเพิ่มหรือลดค่า ของการหมุนแต่ละลือก ใช้สำหรับกำหนดค่า ขึ้นลง เช่นการปรับวอลุ่มเสียงเพลง



A rotary encoder module has 4 pins:

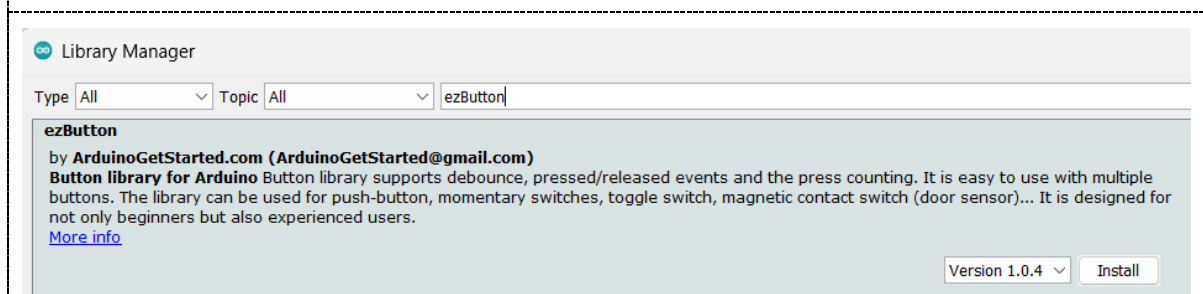
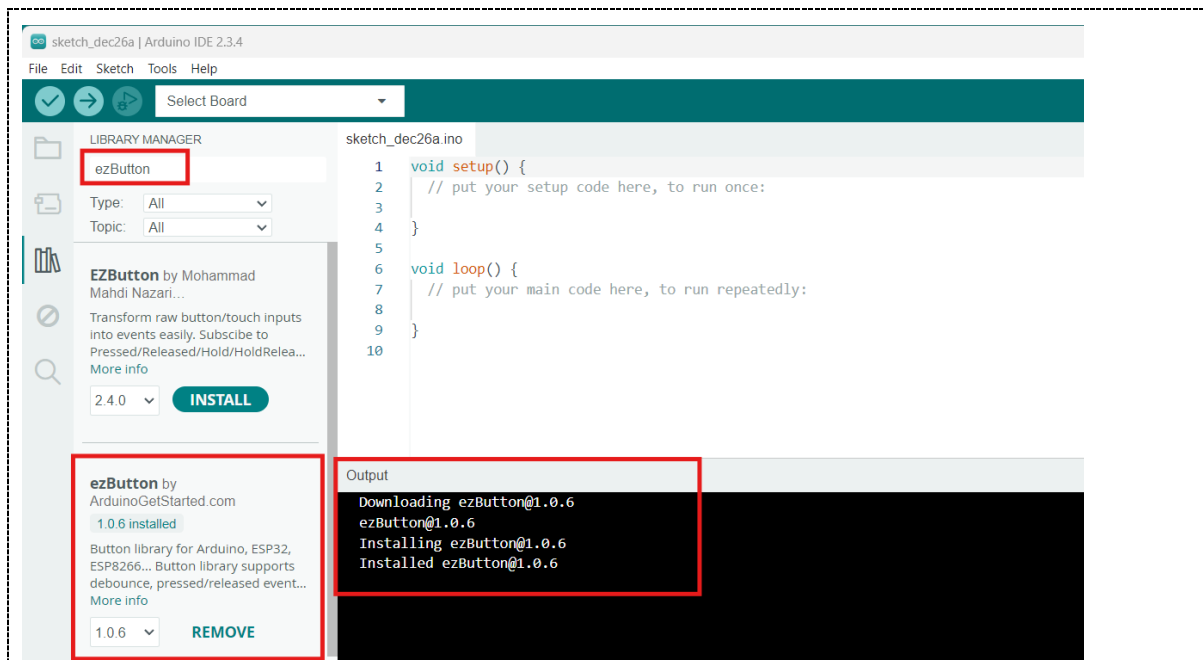
- ◆ **CLK pin (Output A):** is the main pulse that tells us how much rotation has occurred. Whenever you turn the knob by one detent (click) in either direction, the CLK pin outputs a signal that is a completes a full cycle (**LOW** → **HIGH** → **LOW**).
- ◆ **DT pin (Output B):** acts like the CLK 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)



esp32io.com

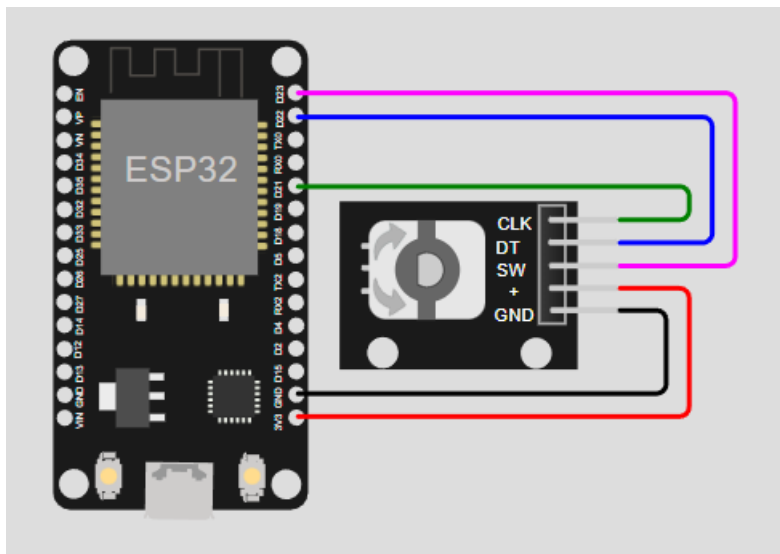
3. Add Library: **Sketch** → **Include Library** → **Manage**

4. Filter with “**ezButton**”, Select **ezButton** by **ArduinoGetStarted.com** Version **1.0.6**



ezButton by **ArduinoGetStarted.com** Version **1.0.6**

5. Test Code "Test1001-Hello RotaryEncoder"



CLK	D21
DT	D22
SW	D23
+	3V3
GND	GND

```
Rotary | Arduino 1.8.19
File Edit Sketch Tools Help

Rotary §
COM14
Send

41 // the e
42 if (digi
43 counte
44 direct
45 } else {
46 // the
47 counte
48 direct
49 }
50
51 Serial.p
52 if (dire
53 Serial
54 else
55 Serial
56
57 Serial.p
58 Serial.p
59 }
60
61 // save la
62
```

```
14:20:06.695 -> Rotary Encoder:: direction: Counter-clockwise - count: -8
14:20:07.162 -> Rotary Encoder:: direction: Counter-clockwise - count: -9
14:20:07.627 -> Rotary Encoder:: direction: Counter-clockwise - count: -10
14:20:08.047 -> Rotary Encoder:: direction: Counter-clockwise - count: -11
14:20:08.372 -> Rotary Encoder:: direction: Clockwise - count: -10
14:20:08.747 -> Rotary Encoder:: direction: Clockwise - count: -9
14:20:09.074 -> Rotary Encoder:: direction: Counter-clockwise - count: -10
14:20:09.074 -> Rotary Encoder:: direction: Counter-clockwise - count: -11
14:20:09.074 -> Rotary Encoder:: direction: Counter-clockwise - count: -12
14:20:09.121 -> Rotary Encoder:: direction: Clockwise - count: -11
14:20:09.584 -> The button is pressed
14:20:10.050 -> The button is pressed
14:20:10.470 -> Rotary Encoder:: direction: Clockwise - count: -10
14:20:10.938 -> The button is pressed
14:20:11.355 -> Rotary Encoder:: direction: Counter-clockwise - count: -11
14:20:11.403 -> Rotary Encoder:: direction: Counter-clockwise - count: -12
```

Autoscroll Show timestamp Newline 115200 baud Clear output

```

/*
  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");
}
}

```