



Incremental Photoelectric Rotary Encoder - 400P/R

SKU: SEN0230



Introduction

This is an industrial incremental photoelectric rotary encoder with aluminum material, metal shell and stainless steel shaft. It generates AB two-phase orthogonal pulse signal through the rotation of the grating disk and optocoupler. 400 pulses/round for each phase, and 1600 pulses/round for dual-phase 4 times output. This rotary encoder supports max 5000 r/min speed. And it can be used for speed, angle, angular velocity and other data measurement.

The photoelectric rotary encoder has a NPN open collector output. It could work with Microcontroller with internal pull-up resistors directly. And it is using 750L05 voltage regulator chip, which has a DC4.8V-24V wide range power input, compatible with Arduino, STM32, PLC and other types of microcontrollers.

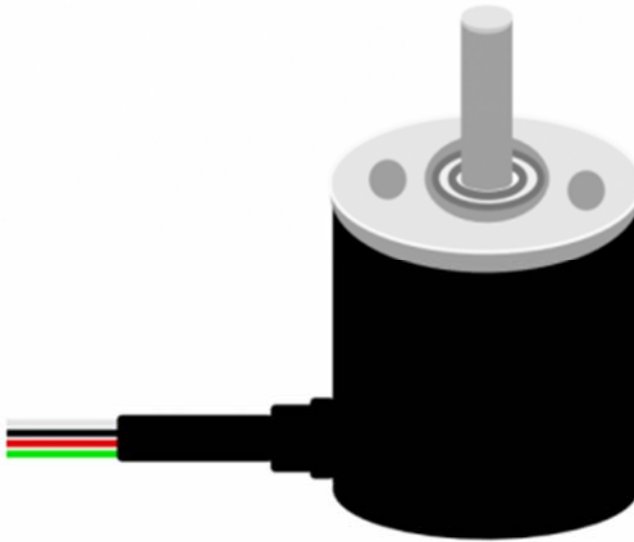


Note: NPN open collector output needs pull-up resistors for the oscilloscope display.

Specification

- Supply Voltage: 4.8V ~ 24v
- Encoder Body Size: $\Phi 39 \times 36.5\text{mm}$
- Output Shaft Diameter: $\Phi 6 \times 13\text{mm}$
- Outside Shaft Platform: $\Phi 20 \times 4.85\text{ mm}$
- Fixing Holes Screws: M3

Board Overview



Incremental Photoelectric Rotary Encoder - 400P/R

Num	Label	Description
White	VCC	Power +
Black	GND	Power -
Red	A	Pulse A (Need pull-up Resistor)
Green	B	Pulse B (Need pull-up Resistor)

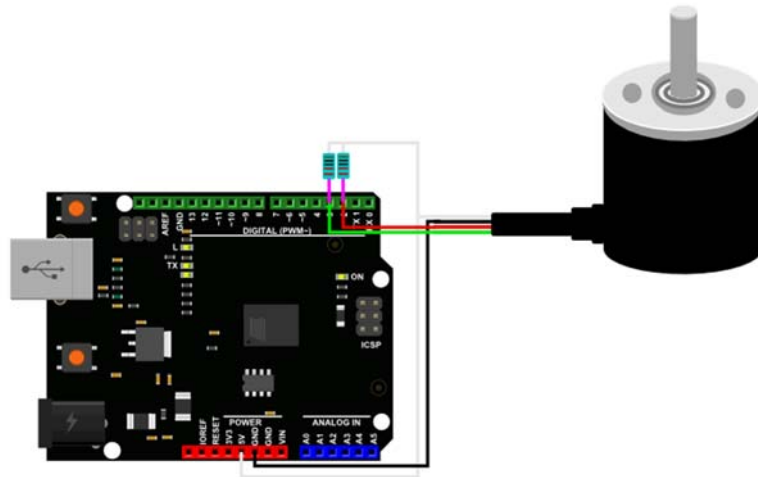
Tutorial

Direction & Interrupt count

Requirements

- **Hardware**
DFRduino UNO (or similar) x 1
Incremental Photoelectric Rotary Encoder
2x 1K Resistor
M-M/F-M/F-F Jumper wires
- **Software**
Arduino IDE, Click to Download Arduino IDE from Arduino®
<https://www.arduino.cc/en/Main/Software%7C>

Connection Diagram



Arduino Connection

Sample Code

```
/*  
Two phase quadrature encoder(Incremental)  
*  
To determine motor with encode (CW OR CCW)  
*/
```

```

@author Dong
@version V1.0
@date 2016-5-26

All above must be included in any redistribution
* *****/
#define A_PHASE 2
#define B_PHASE 3
unsigned int flag_A = 0; //Assign a value to the token bit
unsigned int flag_B = 0; //Assign a value to the token bit
/** * */
void setup() {
    pinMode(A_PHASE, INPUT);
    pinMode(B_PHASE, INPUT);
    Serial.begin(9600); //Serial Port Baudrate: 9600
    attachInterrupt(digitalPinToInterrupt( A_PHASE), interrupt, RISING); //Inte
errupt trigger mode: RISING
}
void loop() {

    Serial.print("CCW: ");
    Serial.println(flag_A);
    Serial.print("CW: ");
    Serial.println(flag_B);
    delay(1000); // Direction judgement

}
void interrupt() // Interrupt function
{ char i;
    i = digitalRead( B_PHASE);
    if (i == 1)
        flag_A += 1;
    else
        flag_B += 1;
}

```

