

# Rotary Encoder Module

## How Encoders work

For how encoders work, read the associated manual with the product [SR1230](#).

## Software implementation

Using encoders on microcontrollers is an easy task. We can use “pin change interrupts” to let the UNO count and keep track of the turning. (*for other devices, look at pin-change interrupts or events for your chosen controller*).

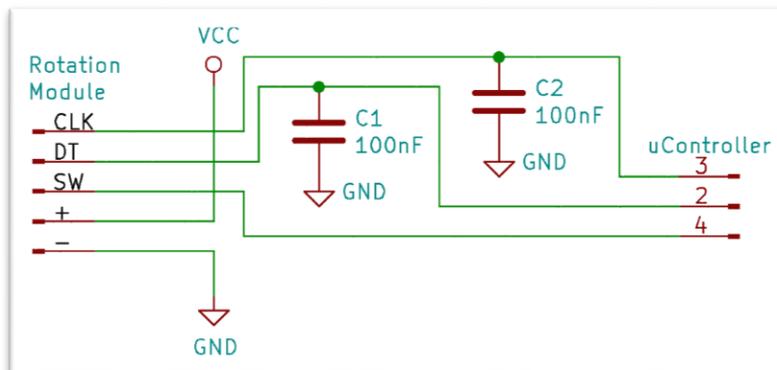
Connect the following:

Rotation Module	Uno Connection
CLK	Pin 3 (for hardware interrupt)
DT	Pin 2 (for hardware interrupt)
SW	Pin 4 (or any input pin)
+	VCC (5V)
-	GND

(Note, for UNO boards, pin change interrupts can only be on pins 2 and 3)

*Note: We recommend 100-200nF capacitors connected between A/B lines to C. You will get unreliable and erratic behaviour without these capacitors.*

Connection between the module and the Arduino is suggested below.



# Rotary Encoder Module

Each “notch” that you can feel while rotating the encoder corresponds with one whole phase change. To track this, we can look at just the falling edge of one of the pins, and check to see if the other pin matches. Try the below code (swap enc\_clk and enc\_dt if it's not quite as expected)

```
#define enc_dt 2
#define enc_clk 3
#define button 4

volatile int encoderValue = 0;

void setup() {
  Serial.begin(9600);
  pinMode(enc_dt, INPUT_PULLUP);
  pinMode(enc_clk, INPUT_PULLUP);
  pinMode(button, INPUT);
  attachInterrupt(digitalPinToInterrupt(enc_clk), encoder, FALLING);
}

void loop() {
  Serial.println(encoderValue);
  delay(100);
}

void encoder() {
  if (digitalRead(enc_clk) == digitalRead(enc_dt)) {
    encoderValue++;
  }
  else {
    encoderValue--;
  }
}
```

We must use `volatile int` and `digitalPinToInterrupt()` in order to use interrupt functions; if you would like to read more, check out:

<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>

