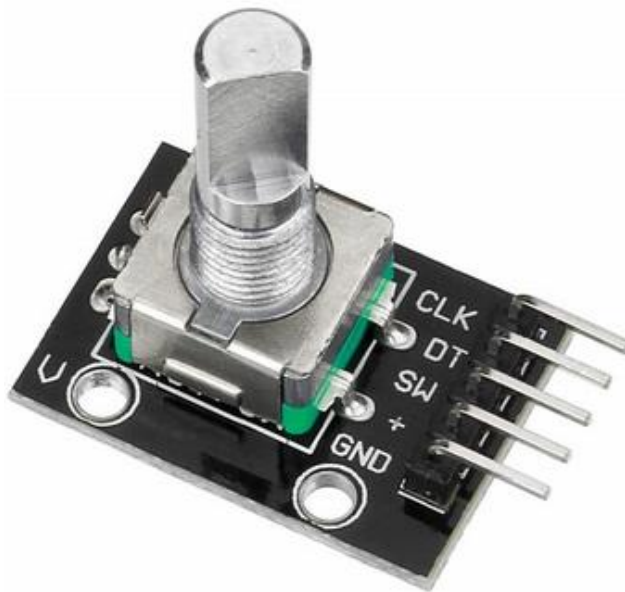


## Rotary Encoder for Arduino/Raspberry

The KY-040 rotary encoder is a rotary input device (as in knob) that provides an indication of how much the knob has been rotated AND what direction it is rotating in. It's a great device for stepper and servo motor control. You could also use it to control devices like digital potentiometers.



SKU: [ASS-1058](#)

### **Brief Data:**

- Operating voltage: 5V.
- Pulses/360° Rotation: 20.
- Output: 2-bit gray code
- Mechanical Angle: 360° continuous.
- With built in push button switch (push to operate)
- Dimensions: (30 x 18 x 30) mm.
- Compatible with Arduino/Raspberry Pi controller board.

## Rotary Encoder Basics

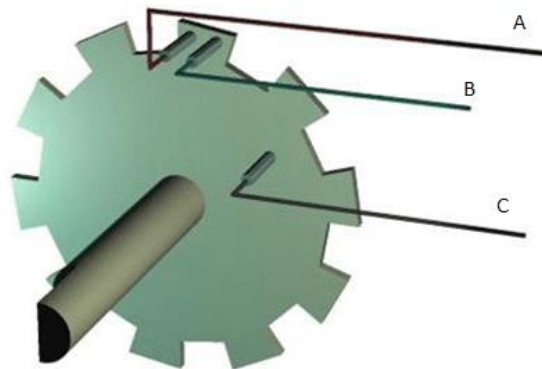
A rotary encoder has a fixed number of positions per revolution. These positions are easily felt as small “clicks” you turn the encoder. The KY-040 module has thirty of these positions. On one side of the switch there are three pins. They are normally referred to as A, B and C. In the case of the KY-040, they are oriented as shown. Inside the encoder there are two switches. Once switch connects pin A to pin C and the other switch connects pin B to C.



In each encoder position, both switches are either opened or closed. Each click causes these switches to change states as follows:

- If both switches are closed, turning the encoder either clockwise or counterclockwise one position will cause both switches to open
- If both switches are open, turning the encoder either clockwise or counterclockwise one position will cause both switches to close.

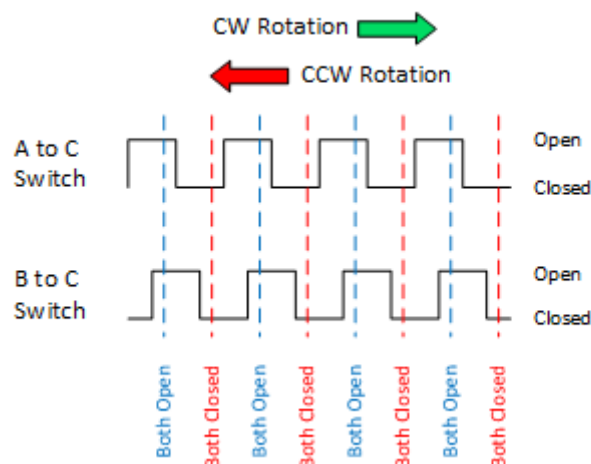
The illustration below is representative of how the switch is constructed.



As you can see, the angular position of the A terminal and the B terminal is such that:

- Rotating the switch clockwise will cause the switch connecting A and C to change states first.
- Rotating the switch counterclockwise will cause the switch connecting B and C to change states first.

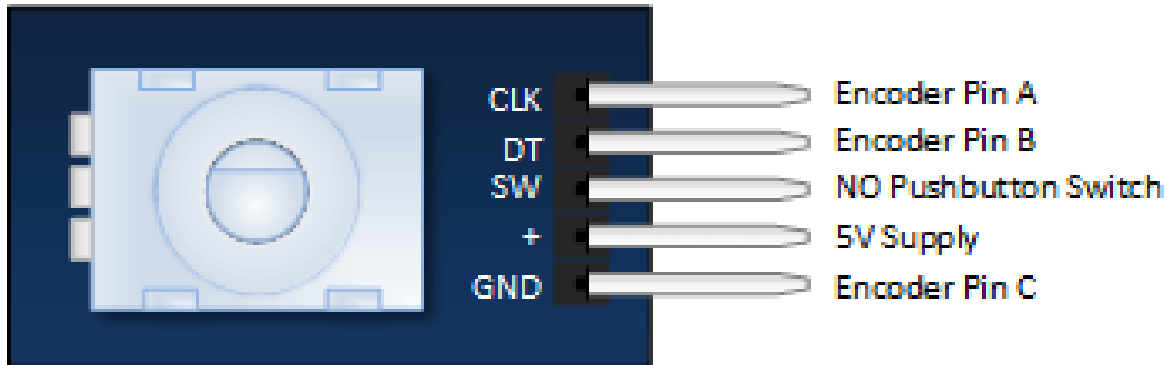
If we were to represent the opening and closing of the switches as wave forms, it would look something like this.



Essentially, determining which switch changed states first is how the direction of rotation is determined. If A changed states first, the switch is rotating in a clockwise direction. If B changed states first, the switch is rotating in a counter clockwise direction.

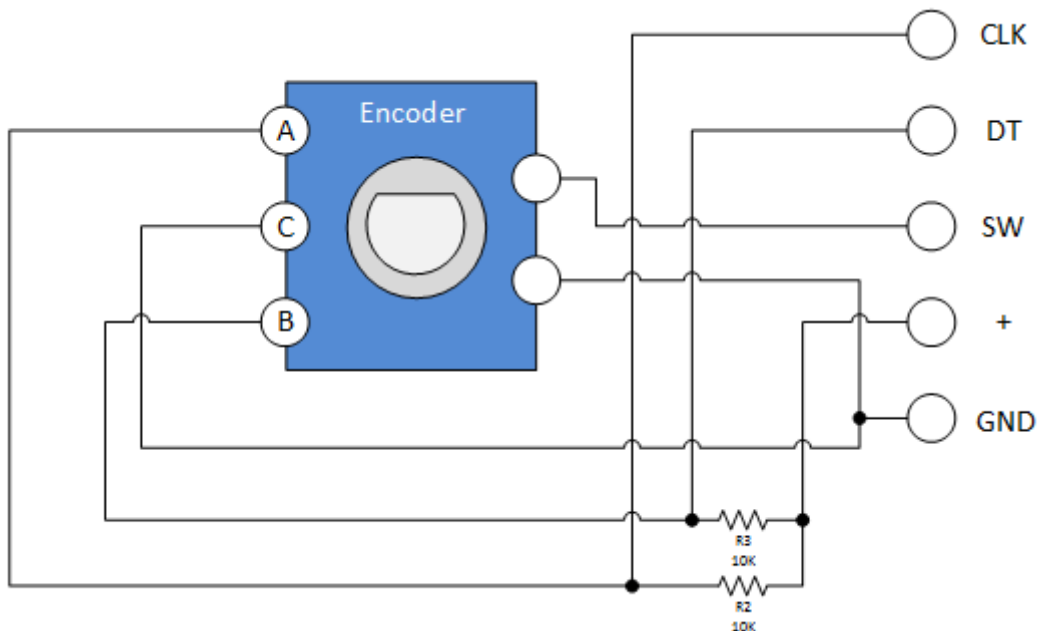
### **Pin Assignment:**

The pin outs for this rotary encoder are identified in the illustration below.



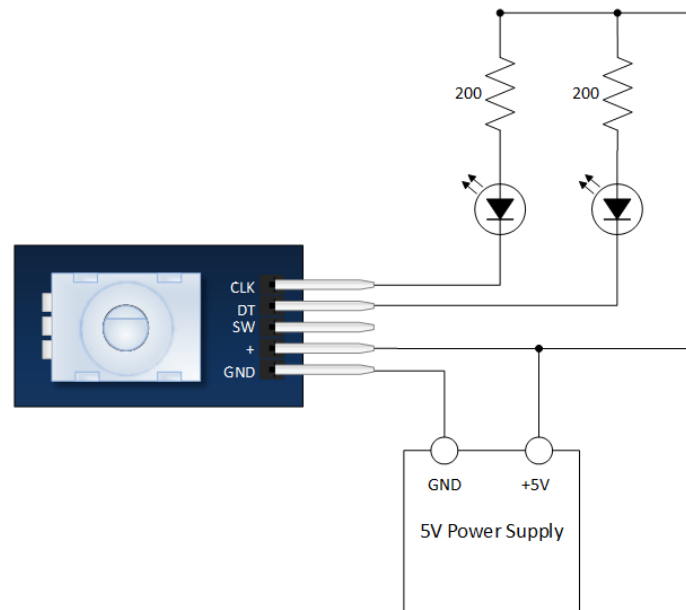
The module is designed so that a low is output when the switches are closed and a high when the switches are open. The low is generated by placing a ground at Pin C and passing it to the CLK and DT pins when switches are closed. The high is generated with a 5V supply input and pull-up resistors, such that CLK and DT are both high when switches are open. Note previously mentioned is the existence of a push button switch that is integral to the encoder. If you push on the shaft, a normally open switch will close. The feature is useful if you want to change switch function. For example, you may wish to have the ability to between coarse and fine adjustments.

### **Rotary Encoder Schematic:**



## Keyes KY-040 Evaluation Circuit:

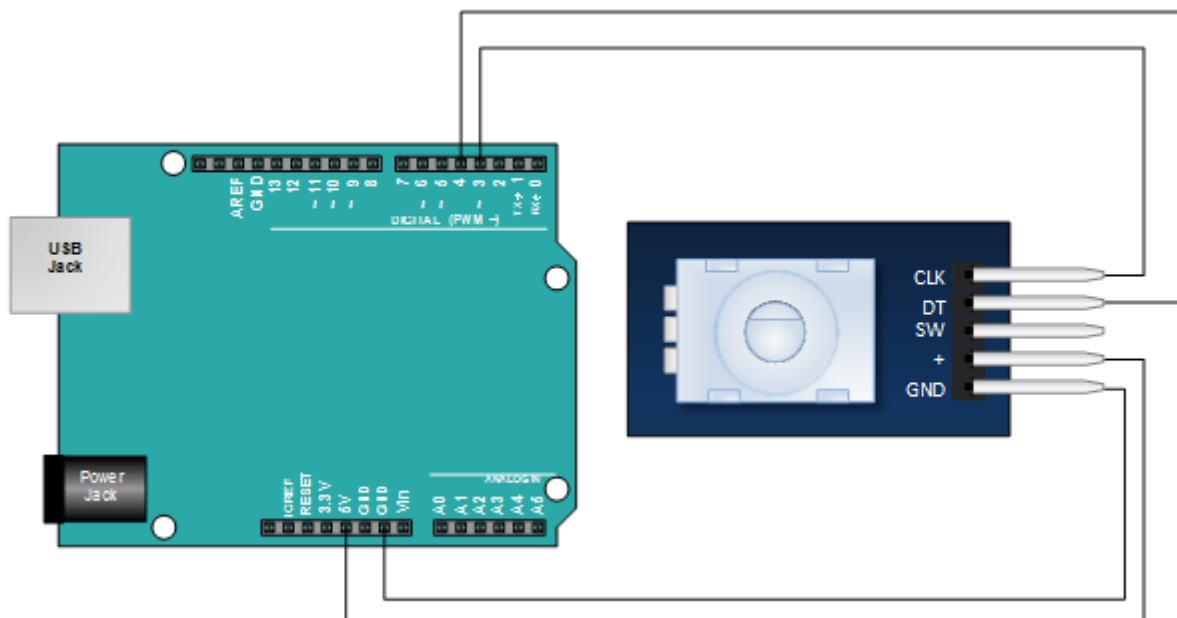
Successfully implementing the Rotary Encoder into any project requires a clear understanding of everything that has been discussed thus far. If you're still a little fuzzy, you may wish to throw together the evaluation circuit illustrated below:



VERY SLOWLY rotates the encoder shaft both clockwise and counterclockwise. Notice which LEDs change state first with rotation.

## KY-040 Arduino Tutorial:

Setup the KY-040 Rotary module to Arduino Uno controller board as below diagram.



Upload the following sketch to Arduino Uno board:

This is a simple sketch that shows how to count the encoder position and how to determine direction of rotation. It has no switch de-bounce, nor does it use interrupts. A fully developed application might need to incorporate these in order to make it robust.

```
/*=====
// Author : Handson Technology
// Project : Arduino Uno with KY040 Rotary Encoder Module
// Description : Reading rotary encoder Value with Arduino Uno
// Source-Code : KY040_Rotary.ino
// Module: KY040
//=====
*/

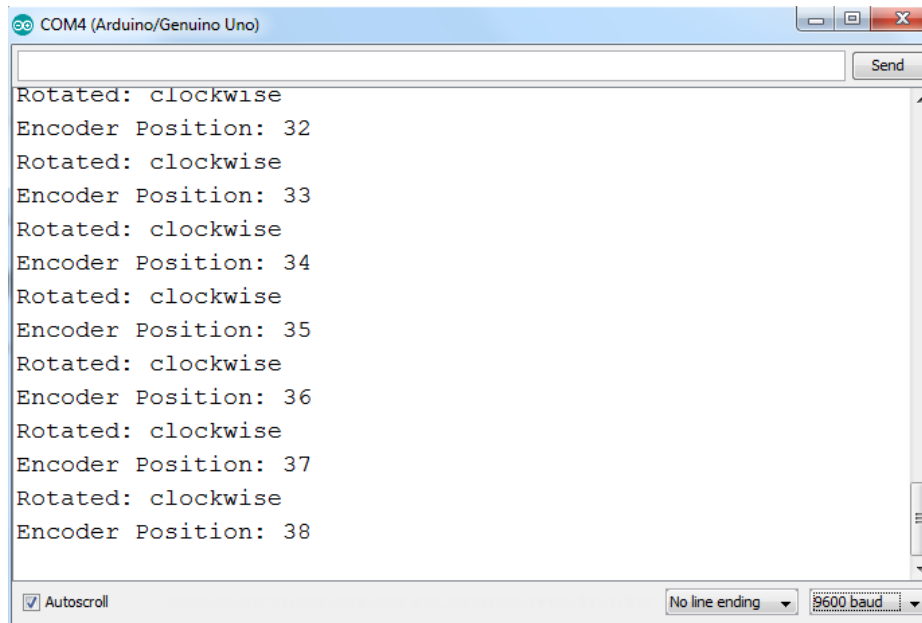
int pinA = 3; // Connected to CLK on KY-040
int pinB = 4; // Connected to DT on KY-040
int encoderPosCount = 0;
int pinALast;
int aVal;
boolean bCW;

void setup() {
  pinMode (pinA,INPUT);
  pinMode (pinB,INPUT);
  /* Read Pin A
  Whatever state it's in will reflect the last position
  */
  pinALast = digitalRead(pinA);
  Serial.begin (9600);
}

void loop() {
  aVal = digitalRead(pinA);
  if (aVal != pinALast){ // Means the knob is rotating
    // if the knob is rotating, we need to determine direction
    // We do that by reading pin B.
    if (digitalRead(pinB) != aVal) { // Means pin A Changed first - We're Rotating
Clockwise.
      encoderPosCount ++;
      bCW = true;
    } else { // Otherwise B changed first and we're moving CCW
      bCW = false;
      encoderPosCount--;
    }
    Serial.print ("Rotated: ");
    if (bCW){
      Serial.println ("clockwise");
    }else{
      Serial.println("counterclockwise");
    }
    Serial.print("Encoder Position: ");
    Serial.println(encoderPosCount);

  }
  pinALast = aVal;
}
```

Open up the Serial Monitor, turn the rotary shaft clockwise and counter clockwise and observe the result:



The screenshot shows the Serial Monitor window for COM4 (Arduino/Genuino Uno). The window contains the following text:

```
Rotated: clockwise  
Encoder Position: 32  
Rotated: clockwise  
Encoder Position: 33  
Rotated: clockwise  
Encoder Position: 34  
Rotated: clockwise  
Encoder Position: 35  
Rotated: clockwise  
Encoder Position: 36  
Rotated: clockwise  
Encoder Position: 37  
Rotated: clockwise  
Encoder Position: 38
```

At the bottom of the window, there are controls for "Autoscroll" (checked), "No line ending", and "9600 baud".



# Handsontec.com

*We have the parts for your ideas*

---

HandsOn Technology provides a multimedia and interactive platform for everyone interested in electronics. From beginner to diehard, from student to lecturer. Information, education, inspiration and entertainment. Analog and digital, practical and theoretical; software and hardware.



open source  
hardware

HandsOn Technology support Open Source Hardware (OSHW) Development Platform.

*Learn : Design : Share*

*[www.handsontec.com](http://www.handsontec.com)*

The Face behind our product quality...

In a world of constant change and continuous technological development, a new or replacement product is never far away – and they all need to be tested.

Many vendors simply import and sell without checks and this cannot be the ultimate interests of anyone, particularly the customer. Every part sell on Handsotec is fully tested. So when buying from Handsotec products range, you can be confident you're getting outstanding quality and value.

We keep adding the new parts so that you can get rolling on your next project.



[www.handsontec.com](http://www.handsontec.com)

[Breakout Boards & Modules](#)



[Connectors](#)



[www.handsontec.com](http://www.handsontec.com)

[Electro-Mechanical Parts](#)



[www.handsontec.com](http://www.handsontec.com)

[Engineering Material](#)



[www.handsontec.com](http://www.handsontec.com)

[Mechanical Hardware](#)



[Electronics Components](#)

P



[www.handsontec.com](http://www.handsontec.com)

[Power Supply](#)



[Arduino Board & Shield](#)

Tools & Accessory



[www.handsontec.com](http://www.handsontec.com)

[Tools & Accessory](#)