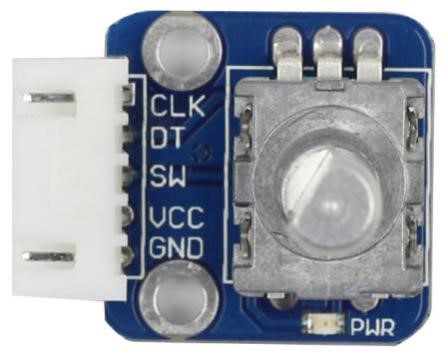
**Introduction**

A rotary encoder is an electro-mechanical device that converts the angular position or motion of a shaft or axle to analog or digital code. Rotary encoders are usually placed at the side which is perpendicular to the shaft. They act as sensors for detecting angle, speed, length, position, and acceleration in automation field.

****

**Components**

- 1 \* SunFounder Uno board

- 1 \* USB cable

- 1 \* Rotary encoder module

- 1 \* 5-Pin anti-reverse cable

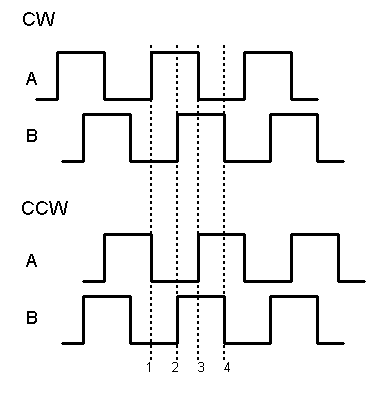
-1 \* Breadboard

**Principle**

A rotary encoder is an electronic switch with a set of regular pulses with strictly timing sequence. When used with IC, it can achieve increment, decrement, page turning and other operations such as mouse scrolling, menu selection, and so on. There are mainly two types of rotary encoders: absolute and incremental (relative) encoders. An incremental one is used in this experiment.

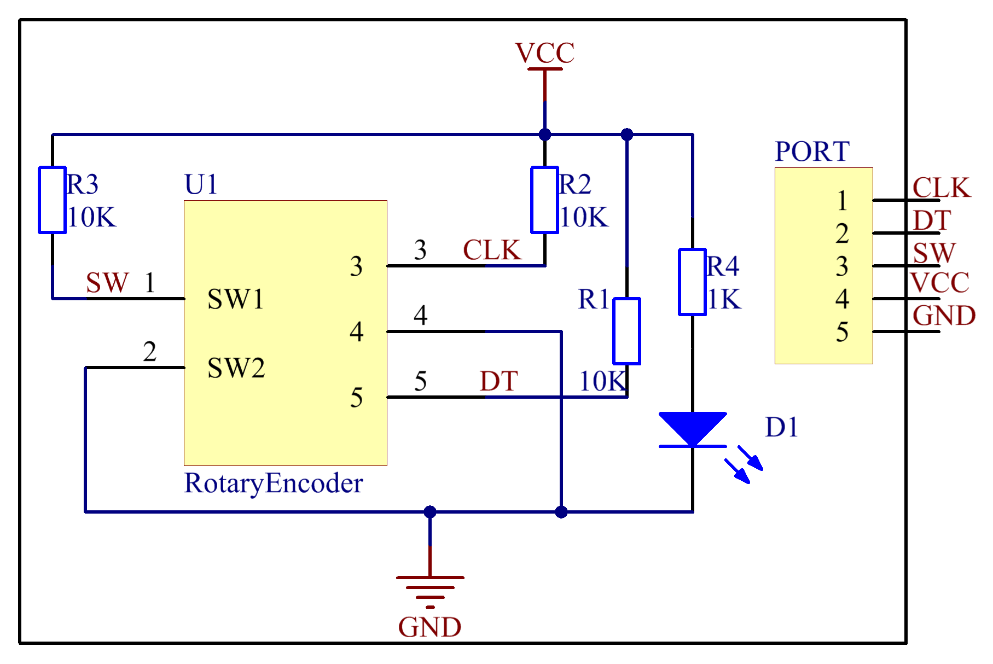


Most rotary encoders have 5 pins with three functions of turning left & right and pressing down. Pin 1 and pin 2 are switch wiring terminals used to press. Pin 4 is generally connected to ground. Pin 3 and pin 5 are first connected to a pull-up resistor and then to VCC and they can generate two-phase square waves whose phase difference is 90°. Usually these waves are called channel A and channel B as shown below:



As shown above, when channel A changes from high level to low level, if channel B is high level, it indicates the rotary encoder spins clockwise (CW); if at that moment channel B is low level, it means spins counterclockwise (CCW). So if we read the value of channel B when channel A is low level, we can know in which direction the rotary encoder rotates.

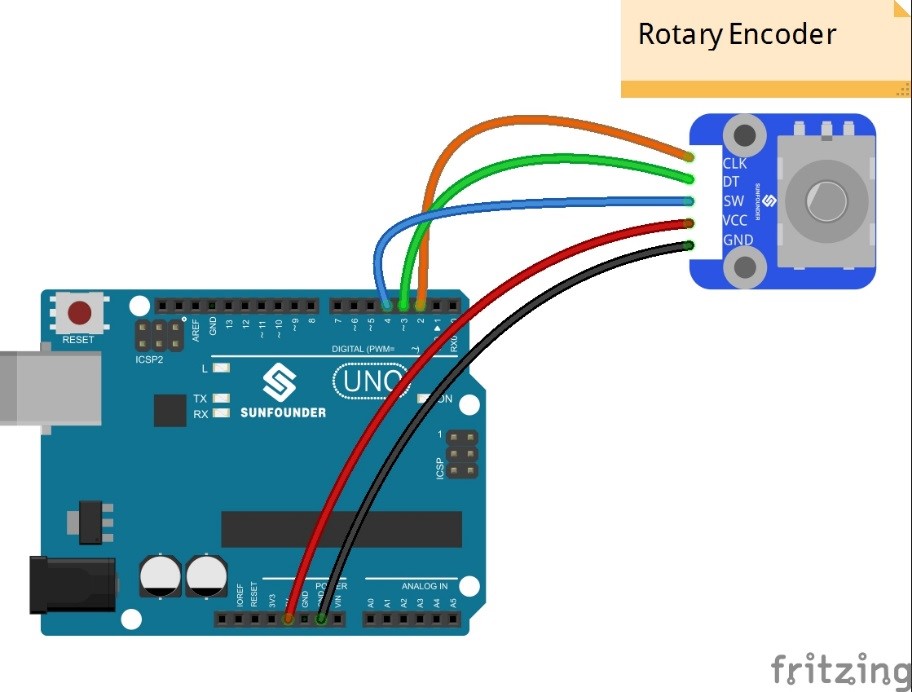
**Principle:**See the schematic diagram of the Rotary Encoder module below. From it we can see that pin 3 of the rotary encoder, namely CLK on the module, is channel B. Pin 5, which is DT, is channel A. To know the rotational direction of the recorder, just read the value of CLK and DT.



This method is concluded by observing the output waveforms of the CLK and DT pin with an oscilloscope by rotating the rotary encoder. You can have a try.

**Experimental Procedures**

**Step 1:**Build the circuit

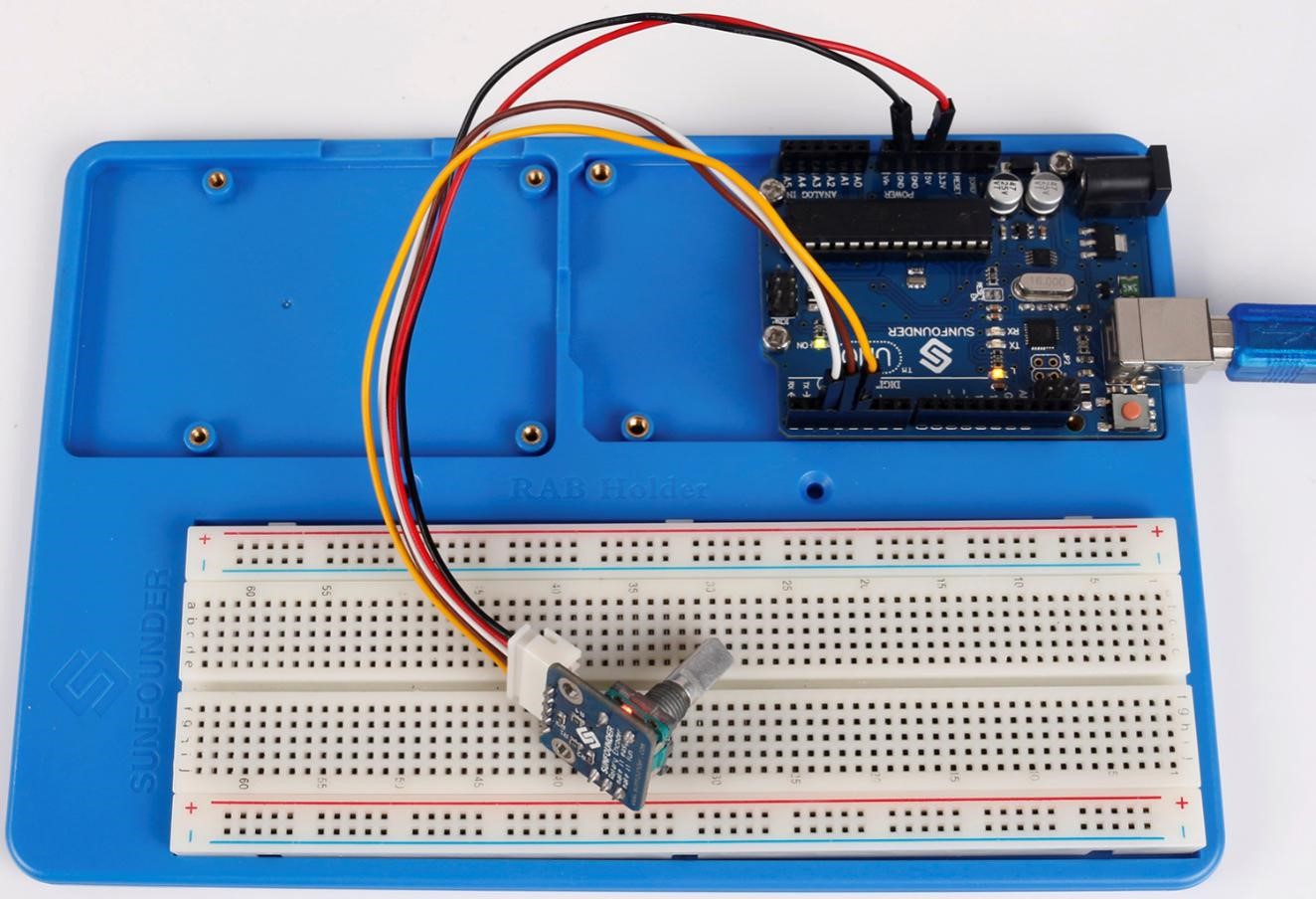


**Step 2**: Open the code file

**Step 3:** Select correct Board and Port

**Step 4:** Upload the sketch to the SunFounder Uno board

Open **Serial Monitor** and you will see the angular displacement of the rotary encoder printed on the window. Spin the shaft of the rotary encoder clockwise, and the angular displacement will decrease; spin it counterclockwise, the value increase. Press it down, and the value will be reset and restore to the initial state.



**Code**

|  |
| --- |
| //Rotary Encoder  /\*You will see the angular displacement of the rotary encoder printed on Serial Monitor.  When you turn the rotary encoder clockwise, the angular displacement is increased;  when turn it counterclockwise, the displacement is decreased.  If you press the switch on the rotary encoder, the readings will return to zero.\*/  //Email: support@sunfounder.com  //Website: www.sunfounder.com  //2015.5.7  const int clkPin= 2; //the clk attach to pin2  const int dtPin= 3; //the dt attach to pin3  const int swPin= 4 ;//the number of the button  int encoderVal = 0;  void setup()  {  //set clkPin,dePin,swPin as INPUT  pinMode(clkPin, INPUT);  pinMode(dtPin, INPUT);  pinMode(swPin, INPUT);  digitalWrite(swPin, HIGH);  Serial.begin(9600); // initialize serial communications at 9600 bps  }  void loop()  {  int change = getEncoderTurn();  encoderVal = encoderVal + change;  if(digitalRead(swPin) == LOW)//if button pull down  {  encoderVal = 0;  }  Serial.println(encoderVal); //print the encoderVal on the serial monitor  }  int getEncoderTurn(void)  {  static int oldA = HIGH; //set the oldA as HIGH  static int oldB = HIGH; //set the oldB as HIGH  int result = 0;  int newA = digitalRead(dtPin);//read the value of clkPin to newA  int newB = digitalRead(clkPin);//read the value of dtPin to newB  if (newA != oldA || newB != oldB) //if the value of clkPin or the dtPin has changed  {  // something has changed  if (oldA == HIGH && newA == LOW)  {  result = (oldB \* 2 - 1);  }  }  oldA = newA;  oldB = newB;  return result;  } |