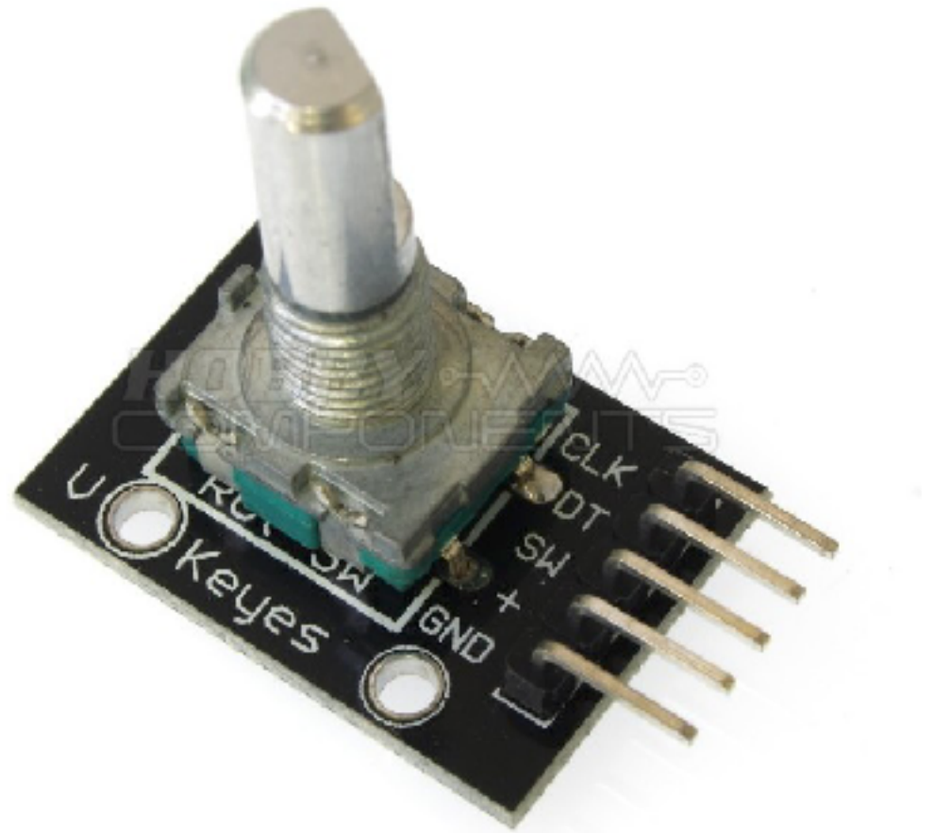
BARSicle

4. Arduino
Rotary Encoder

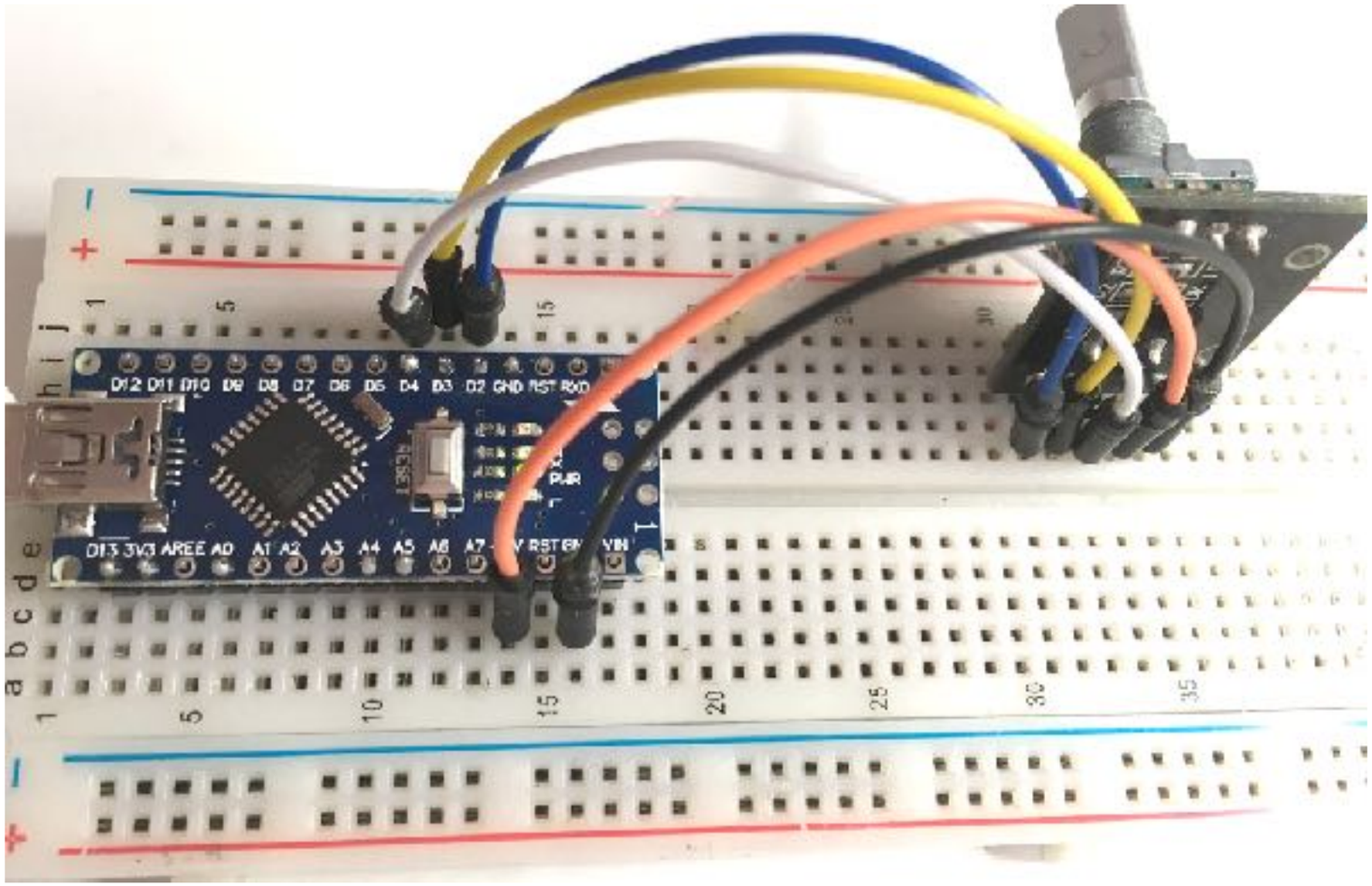
This is Rotary Encoder

- Produces 20 output “clicks” per rotation (DT, CLK)
- Has push button switch (SW)
- Outputs two codes



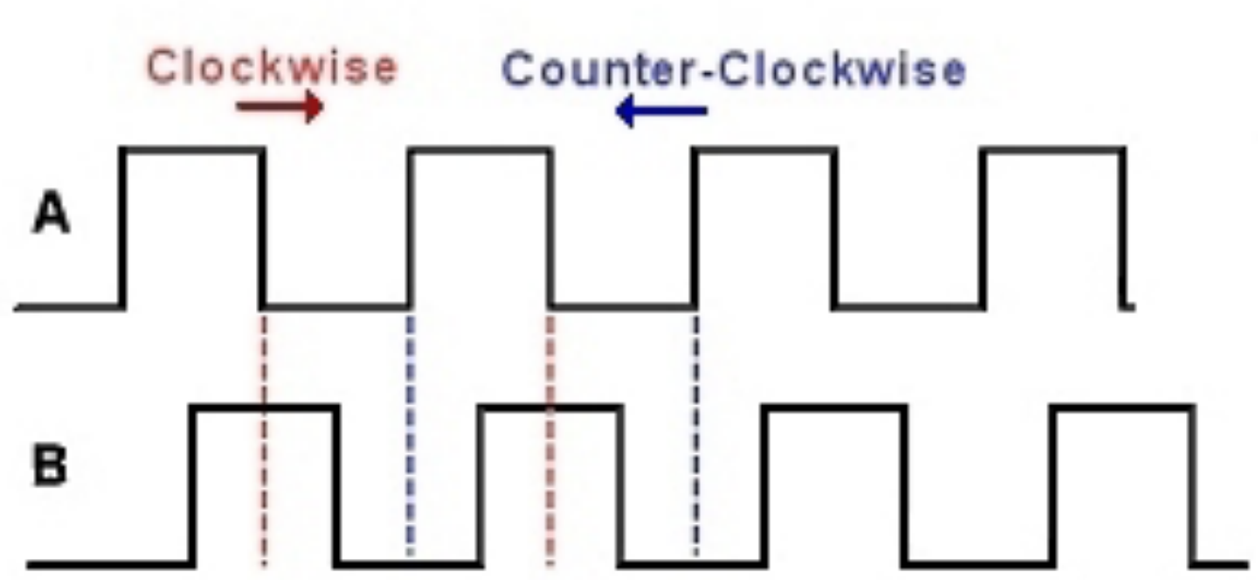
Wiring

- Encoder -> Nano
  - GND -> GND
  - + -> +5V
  - SW -> D4
  - DT -> D3
  - CLK -> D2



Rotary.h

- *Rotary.h* is a library for encoders, two modes
    - Interrupt mode - processor interrupted from whatever it is doing to execute a service routine, faster response
    - Polling mode - go around a loop checking if the encoder has been turned

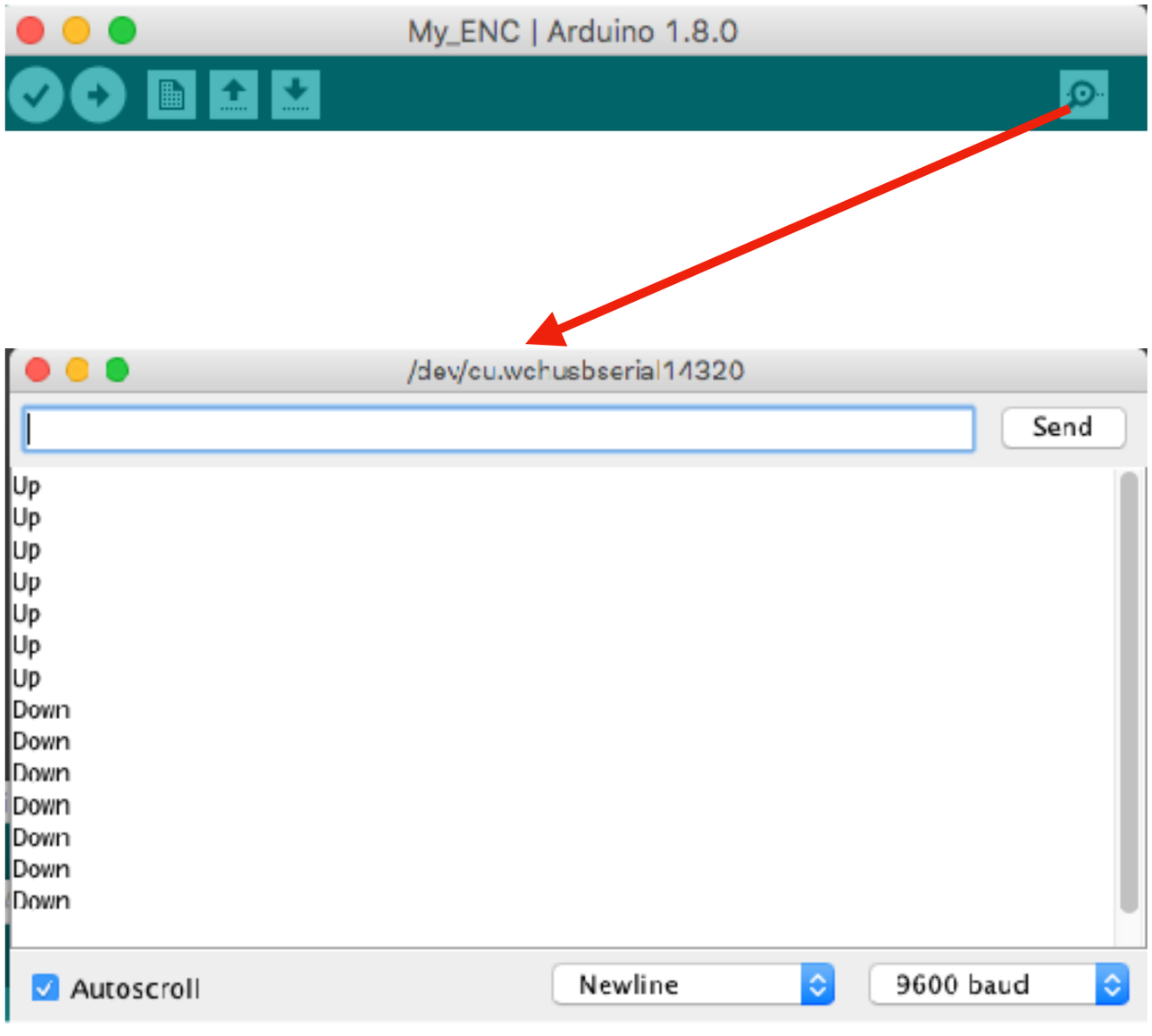


ENC - Polling

- *File > Sketchbook > My_ENC*
- *Open Monitor window*

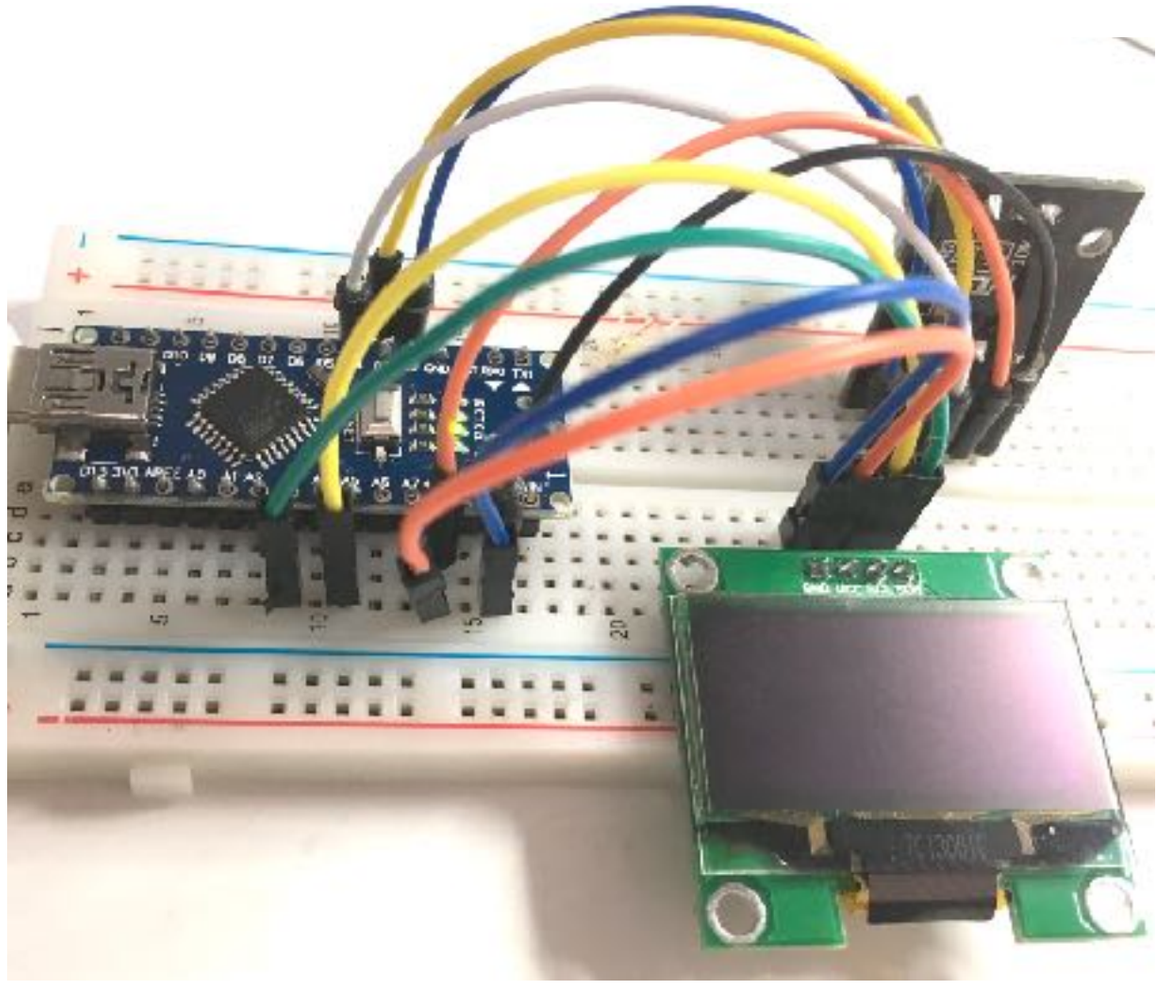
My_ENC

```
// ENCODER

#include "Rotary.h"

Rotary enc = Rotary(2, 3);

void setup() {
  Serial.begin(9600);
  Serial.println("Ready");
}

void loop() {
  unsigned char result;

  result = enc.process();
  if (result) {
    if (result == DIR_CW) {
      Serial.println("Up");
    }
    if (result == DIR_CCW) {
      Serial.println("Down");
    }
  }
}
```



Using ENC & OLED

- Encoder
  - GND -> GND
  - + -> +5V
  - SW -> D4
  - DT -> D3
  - CLK -> D2
- OLED
  - GND -> GND
  - VCC -> +5V
  - SCL -> A5
  - SDA -> A4

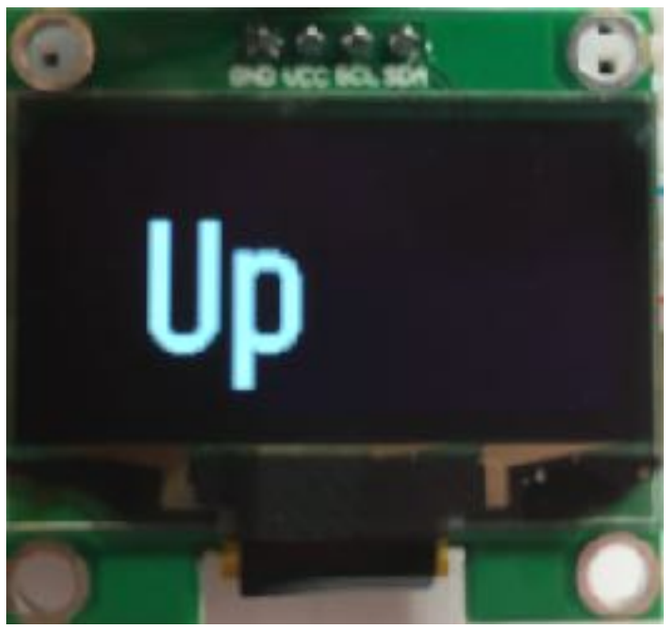


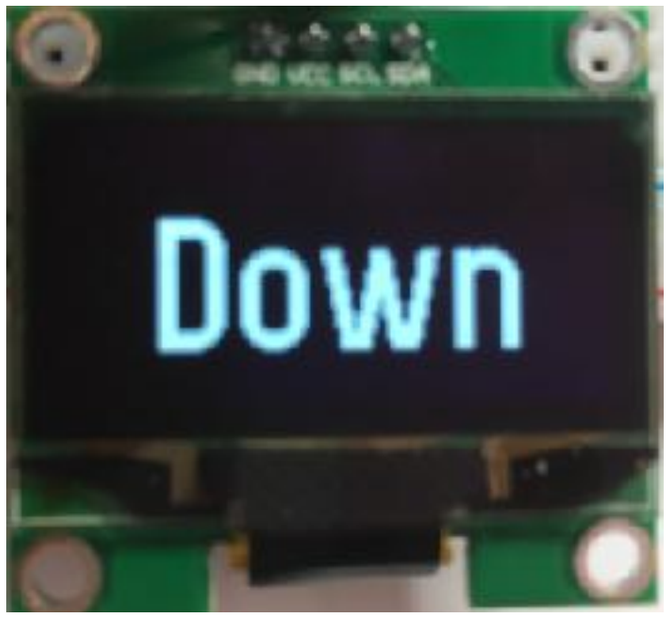
- *File > Sketchbook > My_ENC_OLED*

OLED Display

My_ENC_OLED

```
 2
 3 #include "Oled.h"
 4 #include "Rotary.h"
 5
 6 Rotary enc = Rotary(2, 3);
 7
 8 char m[10] = "Ready";
 9
10 void setup() {
11   oled.begin();
12
13   dispUpdate();
14 }
15
16 void loop() {
17   unsigned char result;
18
19   result = enc.process();
20   if (result == DIR_CW) {
21     strcpy(m, "Up");
22     dispUpdate();
23   }
24   if(result == DIR_CCW) {
25     strcpy(m, "Down");
26     dispUpdate();
27   }
28 }
29
30 void dispUpdate() {
31   oled.firstPage();
32   do {
33     dispMsgUL(20, 20, m);
34   } while ( oled.nextPage() );
35 }
36
```





Well done

Nano + OLED + Encoder