```
// ***** NORMAL ENCODER PART *****

// Alex Curtis
// Encoders
// Finished

// To practice using a 5-pin rotary encoder.
// Pin 3 is connected to CLK
// Pin 6 is connected to DT
// GND is connected to GND
// 5v is connected to the +
// A 6 pF capacitor is placed between the GND and CLK pins

// Pins used
#define CLK_PIN 3
#define DT_PIN 6

int position = 0;   // Stores a count that represents net rotation
String spin;        // Stores the direction to print
int lastClk = HIGH; // Stores the previous state of the clk input
int lastDt = HIGH;  // Stores the previous state of the dt input

void setup() {
  // Start serial
  Serial.begin(9600);

  // Set the clk and dt pins as inputs with the pullup resistors
enabled
  pinMode(DT_PIN, INPUT_PULLUP);
  pinMode(CLK_PIN, INPUT_PULLUP);

  // Print an initial message
  Serial.print("Start ");
  Serial.println(position);
  //    Serial.println("<Direction, lastCLK, lastDT, ");
}

void loop() {

  // Read the current values of clk and dt
  int clk = digitalRead(CLK_PIN);
  int dt = digitalRead(DT_PIN);
```

```
  // If dt changed to a logic 1
  if(dt != lastDt && dt == LOW) {

    // If dt is logic 0(inverted bc of pullup resistor), the encoder
was rotated counterclockwise
    if (clk != dt) {
      position--;
      spin = "CCW";
    }
    // Otherwise, the encoder was rotated clockwise
    else {
      position++;
      spin = "CW";
    }

    // Print spin direction and count
    Serial.print("Turned: ");
    Serial.println(spin);

    // Print STATES for testing (Inverted for pullup)
    Serial.print("Last, Current: ");
    Serial.print(!lastClk);
    Serial.print(!lastDt);

    Serial.print("→");
    Serial.print(!clk);
    Serial.println(!dt);

    Serial.print("Position: ");
    Serial.println(position);

    Serial.println();
  }

  // Save current input to compare with the next iteration of the loop
  lastClk = clk;
  lastDt = dt;

  // Wait 1ms to help with bounce
  delay(1);
}
```

```
// **** ISR PART*****

// Alex Curtis
// Encoders with ISR
// Unfinished

// To practice using a 5-pin rotary encoder using an ISR
// Pin 3 is connected to CLK
// Pin 2 is connected to DT
// GND is connected to GND
// 5v is connected to the +
// A 6 pF capacitor is placed between the GND and CLK pins

// Pins used
#define CLK_PIN 3
#define DT_PIN 2

// Volatile: can change at any time. Used for variables shared between
 ISRs and normal functions.
volatile int position = 0;    // Stores a count that represents net
rotation
volatile int clk, dt;         // Stores the current states
volatile int lastClk, lastDt; // Stores the previous states

void updatePosition();      // ISR
void getPosition();

void setup() {
  // Start serial
  Serial.begin(9600);

  // Set the clk and dt pins as inputs with the pullup resistors
enabled
  pinMode(DT_PIN, INPUT_PULLUP);
  pinMode(CLK_PIN, INPUT_PULLUP);

  // Attach interrupt handler to DT_PIN in CHANGE mode
  attachInterrupt(digitalPinToInterrupt(DT_PIN),updatePosition,FALLING
);

  // Print an initial message
```

```
  Serial.print("Start ");
  Serial.println(position);
}

void loop() {

  getPosition();
  Serial.println(position);

}

// ISR to check clk and dt and update position
// Called when DT_PIN changes
void updatePosition() {

  // Read both channels
  clk = digitalRead(CLK_PIN);
  dt = digitalRead(DT_PIN);

  //If dt went from 0 to 1
  if(dt == LOW) {
    // If the new readings are different:
    if (clk != dt) {
      // CCW
      position -= 2; // count-2
    }
    else {
      // CW
      position += 2; // count+2
    }
  }

  // TODO I thought ISRs shouldn't have print statements?
  Serial.println(position);

}

void getPosition() {

  // TODO what is the point of this function? Is it to correct what I
did in the ISR?
  int newClk = digitalRead(CLK_PIN);
```

```
  int newDt = digitalRead(DT_PIN);

  // Only update if the readings are new?
  if (newDt ≠ dt) {
    // If the old readings are the same and the new readings are
different, adjust in the CCW direction
    if (clk == dt && newClk ≠ newDt ) {
      // Went too far CCW, so add 1
      position ++; // TODO inverted to adjust ISR
      // spin = "CCW";
    }
    // If the old readings are different and the new readings are the
same, adjust in the CW direction
    else if(clk ≠ dt && newClk == newDt) {
      // Went too far CW, so subtract 1
      position --;  // TODO inverted to adjust ISR
      // spin = "CW";
    }
  }


}
```