

# Lesson 28 Controlling Stepper Motor With Rotary Encoder

## Introduction

In this lesson, you will learn how to control stepper motors using a rotary encoder. We will use the inexpensive and popular stepper motor that comes with its own control board: the 28BYJ-48 stepper motor with the ULN2003 board.

We will write some code to have the motor move in the direction that we turn the rotary encoder, and will also keep track of how many steps we have taken, so that we can have the motor move back to the starting position by pressing down on the rotary encoder switch.

## Hardware Required

- ✓ 1 \* RexQualis Mega 2560
- ✓ 1 \* 830 tie-points breadboard
- ✓ 1 \* Rotary Encoder Module
- ✓ 1 \* ULN2003 stepper motor driver module
- ✓ 1 \* Stepper motor
- ✓ 1 \* Power supply module
- ✓ 1 \* 9V1A Adapter
- ✓ 12 \* F-M Jumper Wires

## Principle

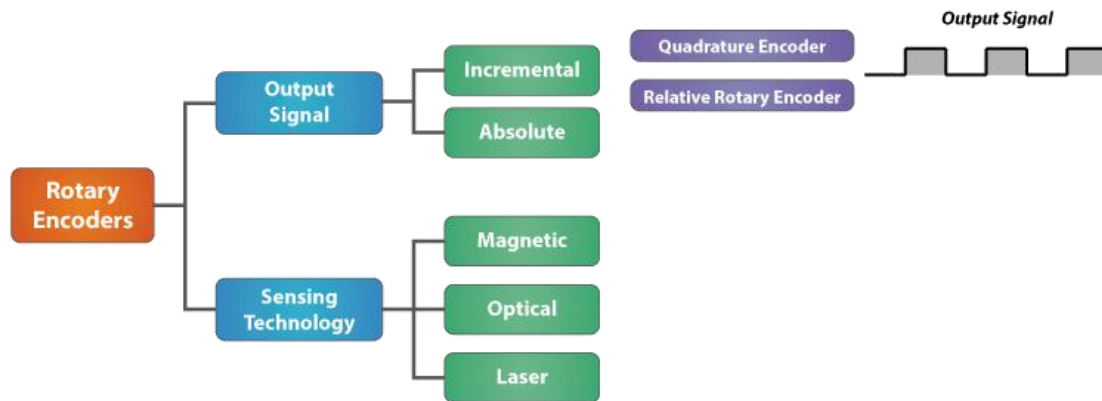
### Rotary encoder

A rotary encoder is a type of position sensor which is used for determining the angular position of a rotating shaft. It generates an electrical signal, either analog or digital, according to the rotational movement.

There are many different types of rotary encoders which are classified by either Output Signal or Sensing Technology. The particular rotary encoder

that we will use in this tutorial is an incremental rotary encoder and it's the simplest position sensor to measure rotation.

This rotary encoder is also known as quadrature encoder or relative rotary encoder and its output is a series of square wave pulses.



## Code interpretation

```
#include <Stepper.h>
```

```
#define STEPS 32
```

```
// Number of steps for one revolution of Internal shaft
```

```
// 2048 steps for one revolution of External shaft
```

```
volatile boolean TurnDetected; // need volatile for Interrupts
```

```
volatile boolean rotationdirection; // CW or CCW rotation
```

```
const int PinCLK=2; // Generating interrupts using CLK signal
```

```
const int PinDT=3; // Reading DT signal
```

```
const int PinSW=4; // Reading Push Button switch
```

```
int RotaryPosition=0; // To store Stepper Motor Position
```

```
int PrevPosition; // Previous Rotary position Value to check accuracy
```

```
int StepsToTake; // How much to move Stepper
```

```
// Setup of proper sequencing for Motor Driver Pins
```

```
// In1, In2, In3, In4 in the sequence 1-3-2-4
```

```
Stepper small_stepper(STEPS, 8, 10, 9, 11);
```

```
// Interrupt routine runs if CLK goes from HIGH to LOW
```

```
void isr () {
```

```
    delay(4); // delay for Debouncing
```

```
    if (digitalRead(PinCLK))
```

```
        rotationdirection= digitalRead(PinDT);
```

```
    else
```

```
        rotationdirection= !digitalRead(PinDT);
```

```
    TurnDetected = true;
```

```
}
```

```
void setup () {
```

```
    pinMode(PinCLK,INPUT);
```

```
    pinMode(PinDT,INPUT);
```

```
    pinMode(PinSW,INPUT);
```

```
    digitalWrite(PinSW, HIGH); // Pull-Up resistor for switch
```

```
    attachInterrupt (0,isr,FALLING); // interrupt 0 always connected to  
    pin 2 on Arduino UNO
```

```
}
```

```
void loop () {
```

```
    small_stepper.setSpeed(700); //Max seems to be 700
```

```
    if (!(digitalRead(PinSW))) { // check if button is pressed
```

```
        if (RotaryPosition == 0) { // check if button was already  
        pressed
```

```
        } else {
```

```
            small_stepper.step(-(RotaryPosition*50));
```

```
            RotaryPosition=0; // Reset position to ZERO
```

```
        }
```

```

    }

    // Runs if rotation was detected

    if (TurnDetected) {

        PrevPosition = RotaryPosition; // Save previous position in
variable

        if (rotationdirection) {

            RotaryPosition=RotaryPosition-1;} // decrease Position by 1

else {

            RotaryPosition=RotaryPosition+1;} // increase Position by 1

        TurnDetected = false; // do NOT repeat IF loop until new
rotation detected

        // Which direction to move Stepper motor

        if ((PrevPosition + 1) == RotaryPosition) { // Move motor CW

            StepsToTake=50;

            small_stepper.step(StepsToTake);

        }

        if ((RotaryPosition + 1) == PrevPosition) { // Move motor CCW

            StepsToTake=-50;

            small_stepper.step(StepsToTake);

        }

    }

    digitalWrite(8, LOW);

    digitalWrite(9, LOW);

    digitalWrite(10, LOW);

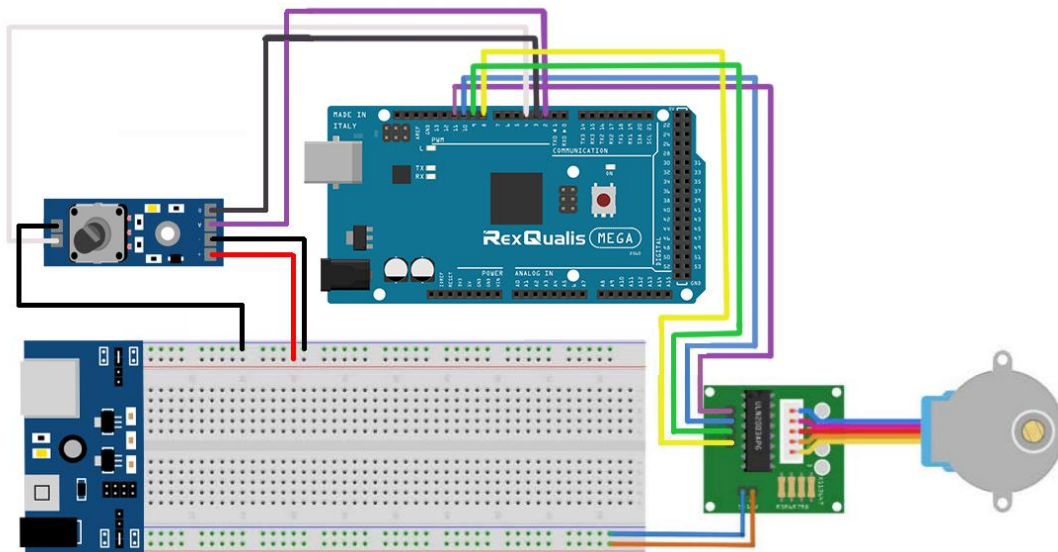
    digitalWrite(11, LOW);

}

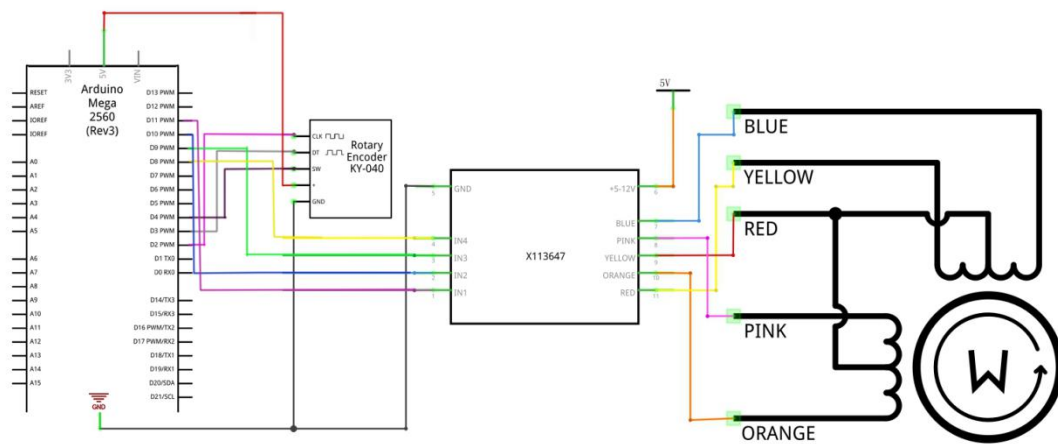
```

# Experimental Procedures

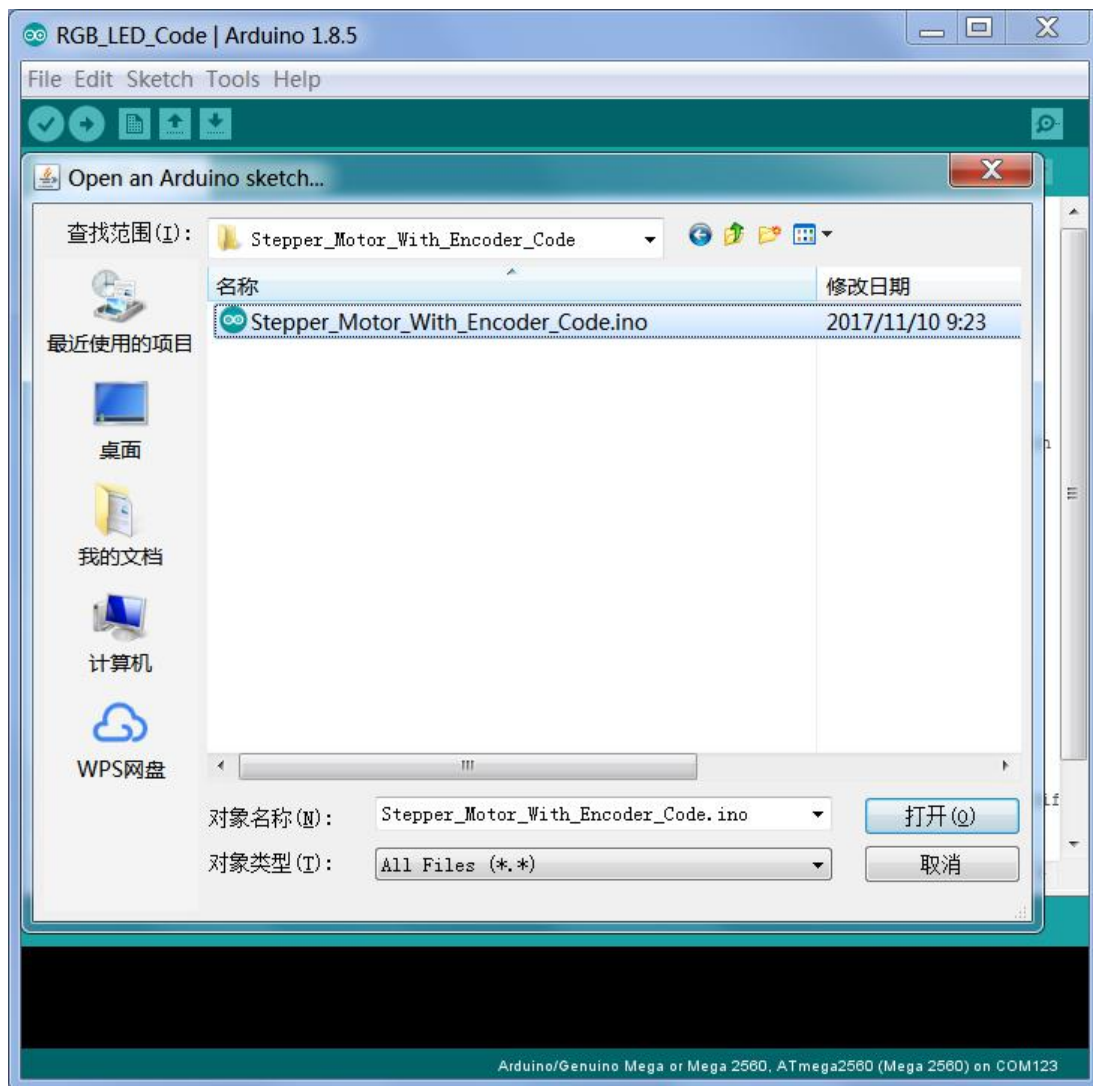
## Step 1: Build the circuit



## Schematic Diagram

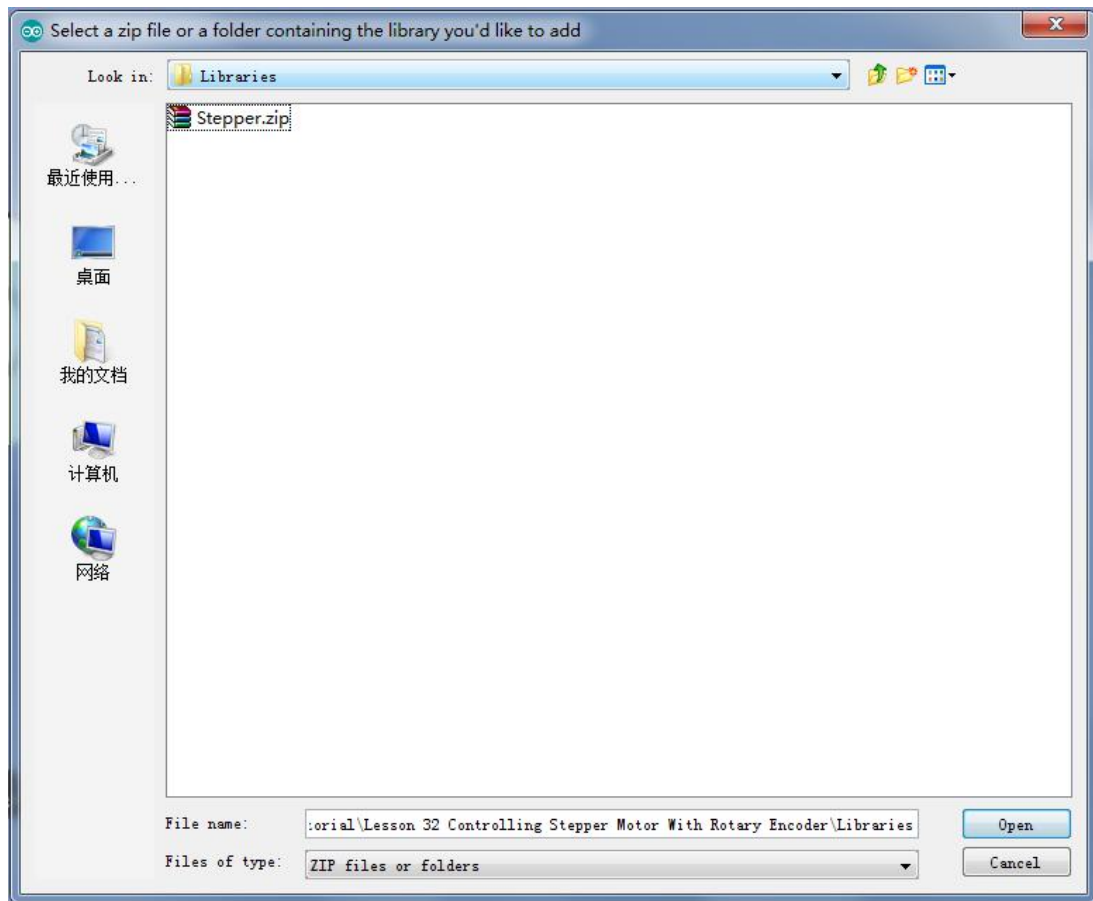


## Step 2: Open the code: Stepper\_Motor\_With\_Encoder\_Code



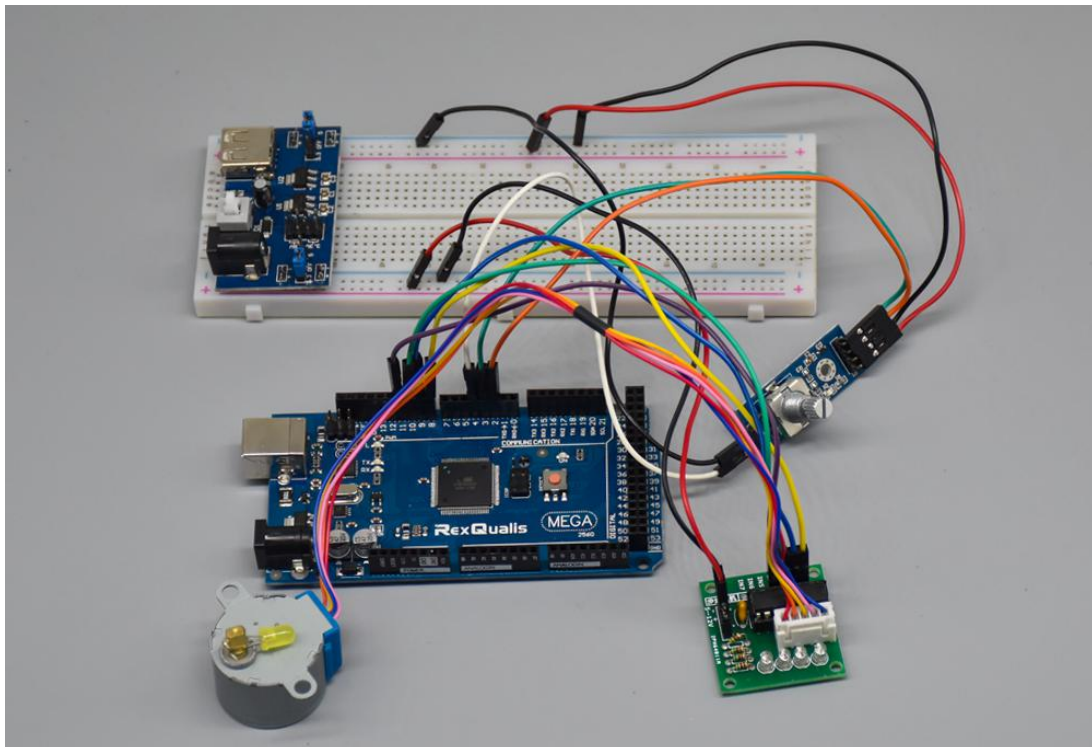
**Step 3: Attach Arduino Mega 2560 board to your computer via USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.**

**Step 4: Load the Library: Stepper**



**Step 5: Upload the code to the RexQualis Mega 2560 board.**

**Then, you can see that when you turn the Rotary Encoder, the Stepper motor will follow.**



**You can see the video of the experiment results on YouTube:**

**[https://youtu.be/M\\_hl5Sr4P-Y](https://youtu.be/M_hl5Sr4P-Y)**

**If it isn't working, make sure you have assembled the circuit correctly, verified and uploaded the code to your board. For how to upload the code and install the library, check Lesson 0 Preface.**