Part 3

lesson 8

Controlling a Stepper Motor With Rotary Encoder

Overview

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.

The 28BYJ-48 motor is not very fast or very strong, but it' s great for beginners to start experimenting with controlling a stepper motor with an Arduino.

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.

Component Required:

(1) x Elegoo Uno R3

(1) x 830 tie-points breadboard

(1) x Rotary Encoder Module

(1) x ULN2003 stepper motor driver module

(1) x Stepper motor

(1) x Power supply module

(1) x 9V1A Adapter

(9) x F-M wires (Female to Male DuPont wires)

(1) x M-M wire (Male to Male jumper wire)

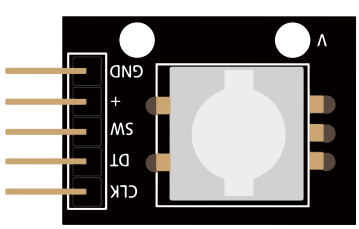
Component Introduction

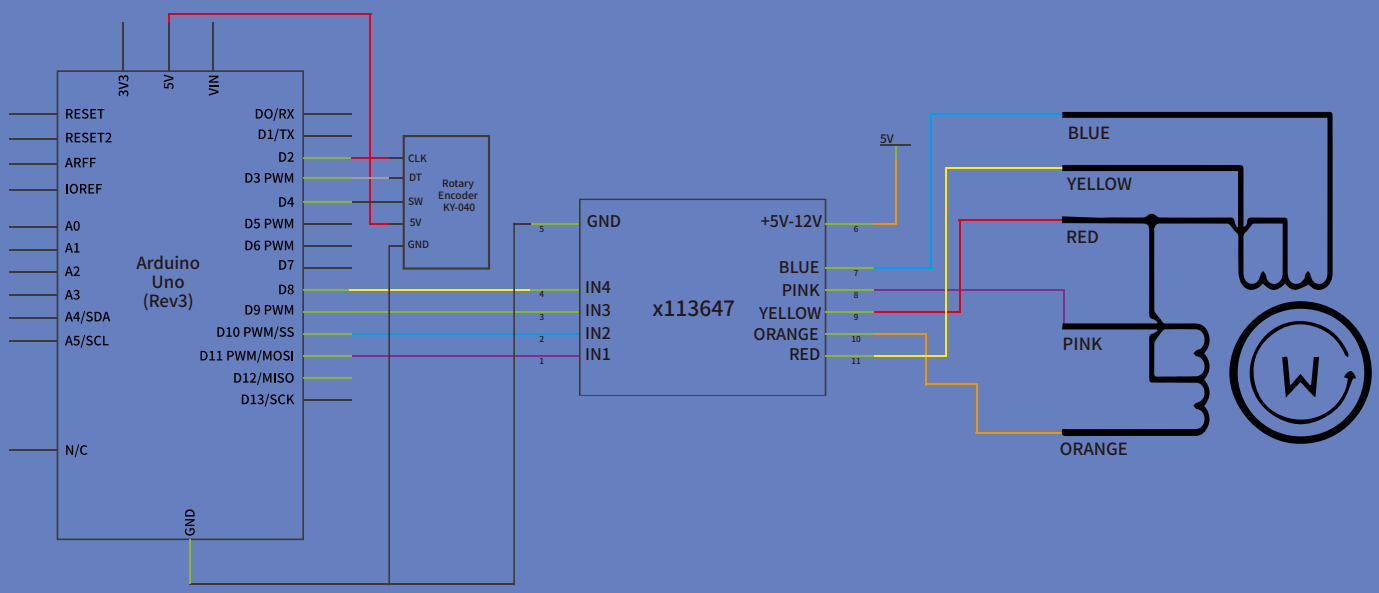
Rotary encoder

A rotary encoder, also called a shaft encoder, is an electro-mechanical device that converts the angular position or motion of a shaft or axle to an analog or digital code.

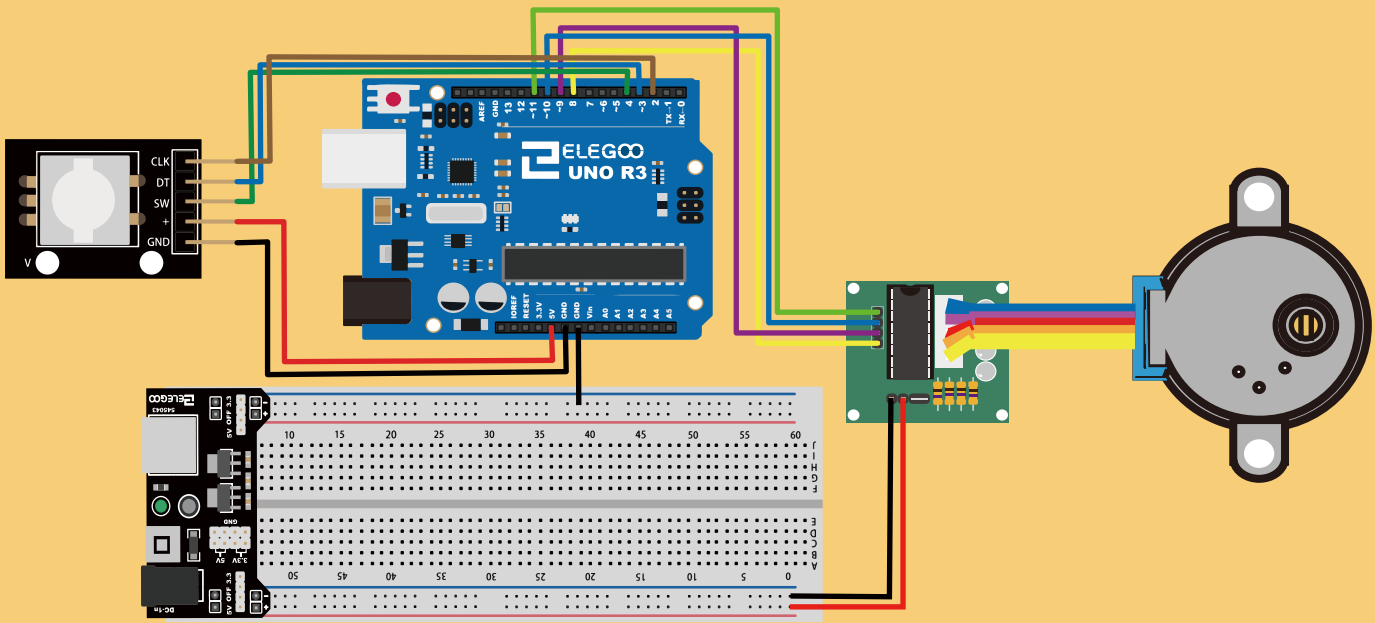
There are two main types: absolute and incremental (relative). The output of absolute encoders indicates the current position of the shaft, making them angle transducers. The output of incremental encoders provides information about the motion of the shaft, which is typically further processed elsewhere into information such as speed, distance and position.

Rotary encoders are used in many applications that require precise shaft unlimited rotation—including industrial controls, robotics, special purpose photographic lenses, computer input devices (such as opt mechanical mice and trackballs), controlled stress rheometers, and rotating radar platforms.





Connection Schematic



We are using 4 pins to control the Stepper and 3 pins for the rotary encoder module. Pins 8-11 are controlling the Stepper motor and pins 2-4 are receiving information from the rotary encoder.

We connect the 5V and Ground from to UNO to the rotary encoder and as a precaution, use a breadboard power supply to power the stepper motor since it can use more power that the UNO can provide.

We also connect the UNO Ground to the ground of the breadboard, to serve as a reference.

Wiring diagram

Code

- After wiring, please open the program in the code folder **With_Encoder** and click UPLOAD to upload the program. See Lesson 5 of part 1 for details about program uploading , if there are any errors.
- Before you can run this, make sure that you have installed the **< Stepper >** library or re-install it, if necessary. Otherwise, your code won't work.
- For details about loading the library file, see Lesson 5 of part 1.
- We are using some variables to store the current position, since we want to keep track of the position of the stepper motor so we can make it move back to the starting position.
- We also included some error checking code to make sure that the rotary encoder is not missing steps, since that would make our motor position inaccurate.