Programming For Engineers

Controlling a Stepper Motor Using Arduino

by

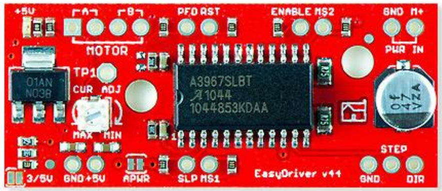
Wan Azhar Wan Yusoff[1], Ahmad Fakhri Ab. Nasir[2]
Faculty of Manufacturing Engineering
wazhar@ump.edu.my[1], afakhri@ump.edu.my[2]

Chapter’s Information

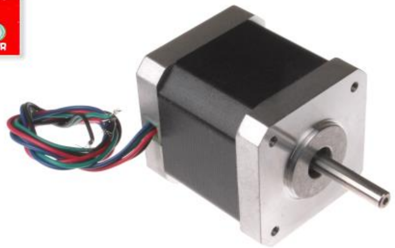
- Purpose
  - The purpose of this writing is to guide students to control a stepper motor from ADUINO UNO with the help of EASY DRIVER stepper motor driver.
- Required materials

  We require the following materials in order to perform this project:

  a) A PC with ARDUINO IDE
  b) ARDUINO UNO board
  c) ARDUINO USB to PC cable
  d) 12V Stepper Motor
  e) EASY DRIVER stepper motor driver
  f) 12V DC power supply with 2A minimum
  g) Prototype breadboard
  h) Necessary jumper cable



www.sparkfun.com/products/12779

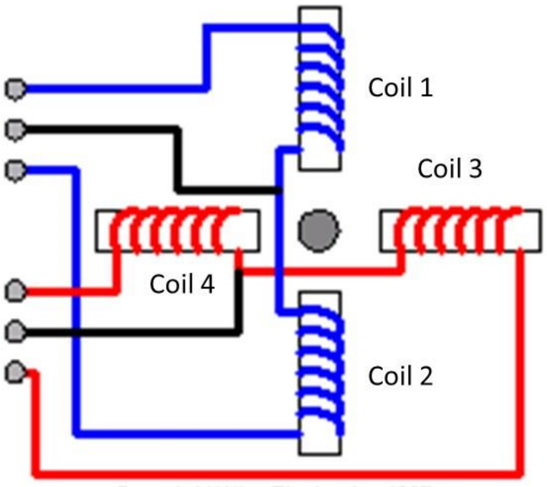


https://upload.wikimedia.org/wikipedia/commons/8/83/Nema_17_Stepper_Motor.jpg

Project Background

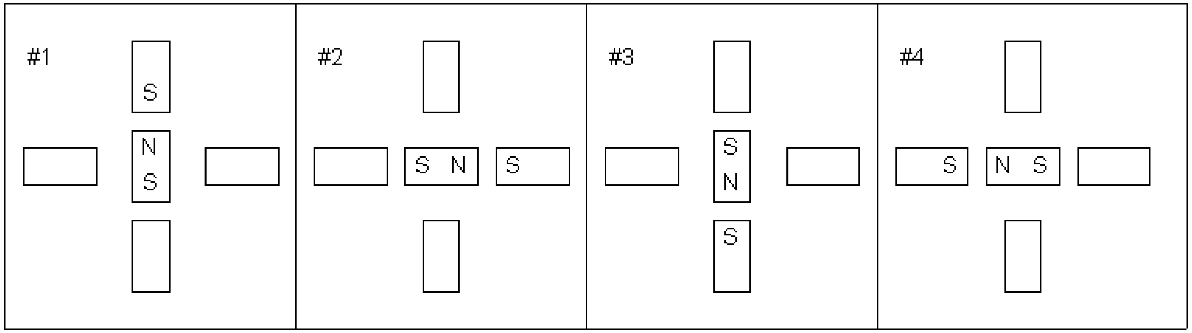
- In the last lab, we managed to send signal from ARDUINO to the real world although at that time, LEDs are our real world. We discovered that turning ON and OFF LEDs through programming are interesting but they are not very useful except to become indicator (or to please our eyes!)
- In this lab, we are going to control the motion of a small stepper motor through our ARDUINO board and C programming. Just like the DC motor, stepper motor requires higher current than LEDs. In such case, we simply cannot draw the required currents from our ARDUINO board. Drawing too much current from ARDUINO board is certainly not a smart thing to do. We need an additional power source to "drive" the motor. The circuit to "drive" the motor is called driver circuit.
- Referring to the figure below, stepper motor is built by creating magnetic field around magnets. To rotate the shaft, we must energize each magnetic field in such a sequence that will cause the shaft to repel or attract to the North-South magnetic field.

Project Background



http://homepage.divms.uiowa.edu/~jones/step/

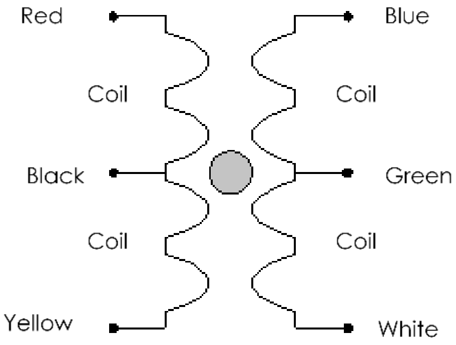
- One example of a stepper-motor sequence in given below.



Project Background

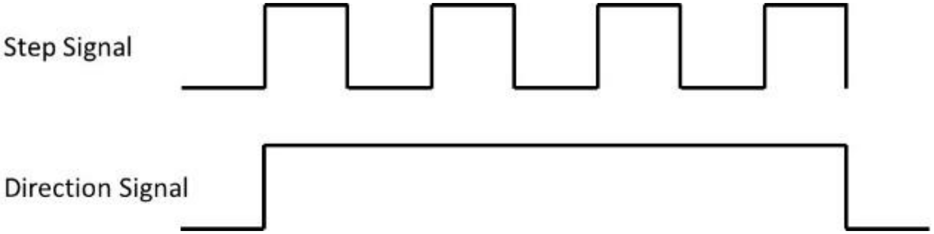
- Suppose we want to rotate the shaft clockwise. Referring to #1 in the diagram, let's say the shaft and the magnetic coil are in the 0 degree position. If we energize the right magnetic field and create a South Pole facing the shaft, it will cause the shaft to rotate to the right (#2). Likewise, if we energize the bottom magnetic coil to South Pole (#3), we cause the shaft to attract to the magnetic coil and therefore rotates clockwise. This is how stepper motor works.
- The schematic diagram of the stepper motor is given below. Usually, the wires that come out from stepper motor are of different colors and each color represents specific coil position. For our unipolar motor, we have black, green, red, yellow, white and blue wires. We will not use the black and green wires (two wires). We will connect the rest of the wires to motor driver.

Project Background



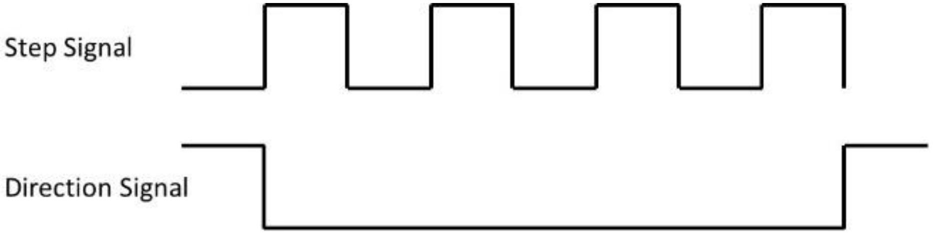
http://homepage.divms.uiowa.edu/~jones/step/

- The circuit to implement the above sequence is implemented by the stepper motor driver. We have to instruct the driver two information: (1) the direction of rotation and (2) the number of step to rotate. The signal to implement the two information is shown below.



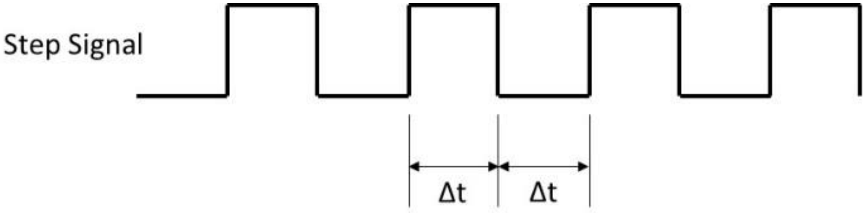
Project Background

- When we give one ON-OFF signal (that means 5V and then 0V) as our step signal, the driver will rotate the motor one step. Therefore, if we want the motor to rotate 27 steps, we have to give 27 ON-OFF signal. Furthermore, if we send the direction signal as ON signal (5V), the motor will move in forward direction. If we want the motor to rotate in the opposite direction, we should send the direction signal the OFF signal (0V). This is shown below.



Project Background

- Furthermore, we can also control the speed of our rotation. Referring to the figure below, we control the speed of the motor by controlling the delay time between ON-OFF of our step signal. If the delay time is big, then we rotate the motor slowly. If the delay time is small, we rotate the motor faster. Remember, the smaller the delay time the faster the motor.



- Referring to the diagram on the right, here are what the EASY DRIVER pins are for:

Project Background

a. We connect the white and blue wires to Phase A of the motor driver. It does not matter which wire goes to which pin as long as the connection is at Phase A.

b. We connect the red and yellow wires to Phase B of the motor driver. It does not matter which wire goes to which pin as long as the connection is at Phase B.

c. We connect the positive DC supply to PWR pin of the driver. The supply can be from 6 V to 30 V.

d. We connect the ground of DC supply to GND pin of the driver (that is next to PWR pin).

e. We connect the STEP pin of the driver to digital pin 3 of ARDUINO board.

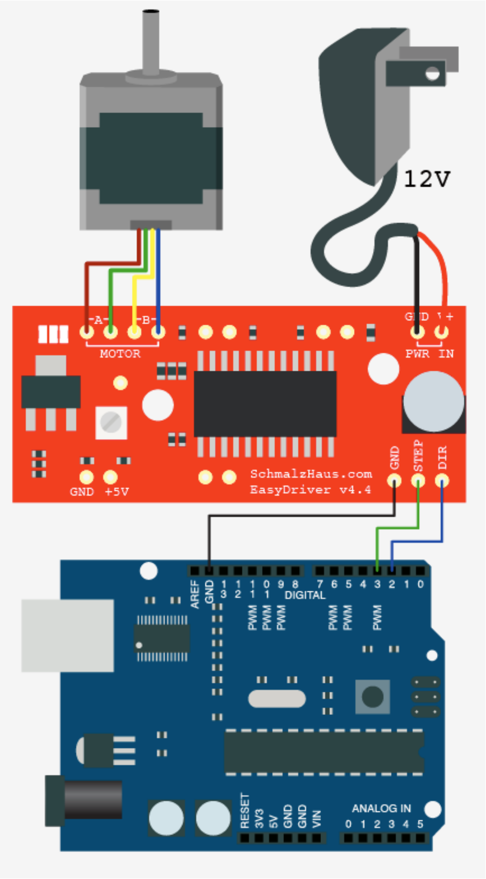
Project Background

f. We connect the DIR pin of the driver to digital pin 2 of ARDUINO board.

g. Finally, we connect the GND of the driver (that is next to the step pin) to ARDUINO ground pin.

Step-by-step Actions

(1) Create Electrical Connection. We will create the connection between ARDUINO digital pins with EASY DRIVER like on the right figure. We supply the 12V from our 12V power supply. For safety reason, try to take 12V from ARDUINO board first. If the motor moving, you can use this 5V. If not you have to take 12V from power supply. You have to take extra careful if using power supply. Make sure all connections are in proper manner. Safety first!



http://bildr.org/2011/06/easydriver/

Step-by-step Actions

(2) We would like to forward the motor for 360 degree, reverse the motor for 360 degree and stop the motor. Below is our ARDUINO program.

```
int pinStep = 3;
int pinDir = 2;
int oneRevStep = 1600; //one rev is 1600 steps
int speed = 2; //for 2 miliseconds delay
void setup() {
    pinMode(pinStep,OUTPUT);
    pinMode(pinDir,OUTPUT);
    Serial.begin(9600); }
void loop() {
    move_forward(oneRevStep,speed);
    delay(1000);
    move_backward(oneRevStep,speed);
    delay(1000); }
```

Step-by-step Actions

```
void move_forward(int step, int speed)
{
    int i;
    digitalWrite(pinDir,LOW);
    for (i=0;i<step;i++)
    {
        digitalWrite(pinStep,HIGH);
        delay(speed);
        digitalWrite(pinStep,LOW);
        delay(speed);
    }
    digitalWrite(pinDir,LOW);
    digitalWrite(pinStep,LOW);
}
void move_backward(int step, int speed)
{
    int i;
```

Step-by-step Actions

```
    digitalWrite(pinDir,HIGH);
    for (i=0;i<step;i++)
    {
        digitalWrite(pinStep,HIGH);
        delay(speed);
        digitalWrite(pinStep,LOW);
        delay(speed);
    }
    digitalWrite(pinDir,LOW);
    digitalWrite(pinStep,LOW);
}
```

Please notice the use of delay(speed) in order to control the speed of the motor. In our program, we use speed = 2 miliseconds. Furthermore, the motor that we use requires 1600 steps in order to complete one revolution.

Exploration

- Perform the followings?
  1. Make the motor rotate in “forward” direction for 180 degree ONLY after we click a switch.
  2. When we click the switch AGAIN, the motor rotate in “reverse” direction to its original position.
  3. Lower the speed of the stepper motor at least two times.

Reflections

- We have learn how to:
  - Program and download program using ARDUINO UNO.
  - Use digital pins to read status of a switch.
  - Use digital pins to send out signals to control a stepper motor.
  - Use EASY DRIVER stepper motor.
  - Program the ARDUINO to rotate stepper motor in either direction.
  - Able to control the speed of the stepper motor.