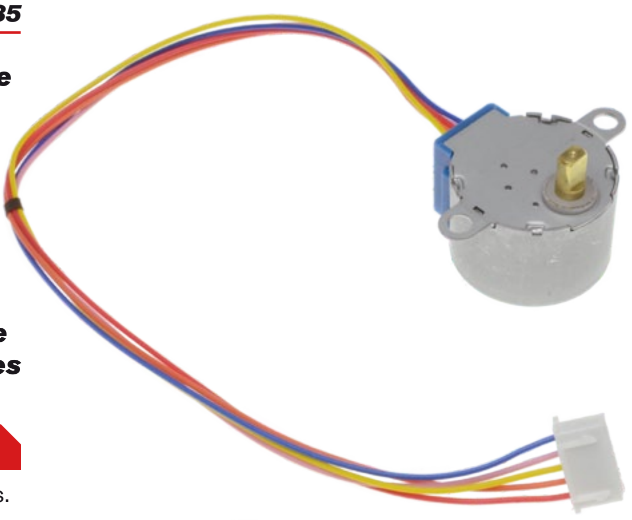
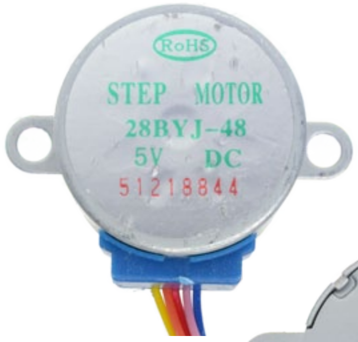
Geared Stepper Motor 4 Phase 5 Wire — MO1735

- ***Rotation angle of motor is proportional to the input pulse***
- ***Full torque at standstill (if the windings are energized)***
- ***Precise positioning and repeatability of movement***
- ***Excellent response to starting/stopping/reversing***
- ***Very reliable (no contact brushes in the motor)***
- ***Open-loop control makes the motor easier to control***
- ***It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft***
- ***A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses***



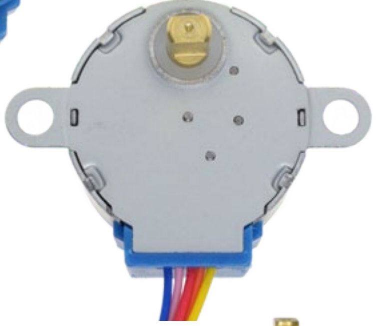
Description

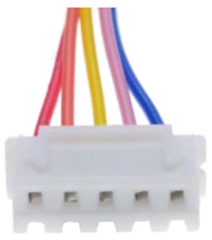
A stepper motor converts electrical pulses into discrete mechanical movements. The shaft or spindle of a stepper motor rotates in discrete step increments when electrical command pulses are applied to it in the proper sequence. One of the most significant advantages of a stepper motor is its ability to be accurately controlled in an open loop system. Open loop control means no feedback information about position is needed. This type of control eliminates the need for expensive sensing and feedback devices such as optical encoders. Your position is known simply by keeping track of the input step pulses.



Specifications

Rated voltage	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC Resistance	50Ω±7%(25°C)
Idle In-traction Frequency	>600Hz
Idle Out-traction Frequency	>1000Hz
Self-positioning Torque	>34.3mN.m (120Hz)
Friction Torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K (120Hz)
Noise	<35dB (120Hz, No load, 10cm)





The Arduino programming environment comes with a function library for controlling a stepper motor. To use the library, in the Arduino Editor from the top menu bar: Sketch > Import Library > Stepper.

1. The example code assumes that the stepper is being controlled by Arduino pins 8, 9, 10 and 11, but you can use any set of four pins.
2. The "#define STEPS 100" line defines the number of steps per rev. A 3.75 deg motor has 96 steps/rev while a 7.2 deg motor has 48 steps/rev.
3. The "Stepper stepper(STEPS, 8, 9, 10, 11)" line is where you enter the four pins used to control the stepper.
4. The "stepper.setSpeed(x)" command sets the motor speed to x rpm.
5. The "stepper.step(x)" command turns the motor x steps at the speed last set in the stepper.setSpeed() command. The motor turns one direction for positive x and the reverse direction for negative x.

Test Code

```
int Pin0 = 10;
int Pin1 = 11;
int Pin2 = 12;
int Pin3 = 13;
int _step = 0;
boolean dir = true;
void setup()
{
 pinMode(Pin0, OUTPUT);
 pinMode(Pin1, OUTPUT);
 pinMode(Pin2, OUTPUT);
 pinMode(Pin3, OUTPUT);
}
 void loop()
{
 switch(_step){
   case 0:
     digitalWrite(Pin0, LOW);
     digitalWrite(Pin1, LOW);
     digitalWrite(Pin2, LOW);
     digitalWrite(Pin3, HIGH);
   break;
   case 1:
     digitalWrite(Pin0, LOW);
     digitalWrite(Pin1, LOW);
     digitalWrite(Pin2, HIGH);
     digitalWrite(Pin3, HIGH);
   break;
   case 2:
     digitalWrite(Pin0, LOW);
     digitalWrite(Pin1, LOW);
     digitalWrite(Pin2, HIGH);
     digitalWrite(Pin3, LOW);
   break;
   case 3:
     digitalWrite(Pin0, LOW);
     digitalWrite(Pin1, HIGH);
     digitalWrite(Pin2, HIGH);
     digitalWrite(Pin3, LOW);
   break;
   case 4:
     digitalWrite(Pin0, LOW);
     digitalWrite(Pin1, HIGH);
     digitalWrite(Pin2, LOW);
     digitalWrite(Pin3, LOW);
   break;
   case 5:
     digitalWrite(Pin0, HIGH);
     digitalWrite(Pin1, HIGH);
     digitalWrite(Pin2, LOW);
     digitalWrite(Pin3, LOW);
   break;
     case 6:
     digitalWrite(Pin0, HIGH);
     digitalWrite(Pin1, LOW);
     digitalWrite(Pin2, LOW);
     digitalWrite(Pin3, LOW);
   break;
   case 7:
     digitalWrite(Pin0, HIGH);
     digitalWrite(Pin1, LOW);
     digitalWrite(Pin2, LOW);
     digitalWrite(Pin3, HIGH);
   break;
   default:
     digitalWrite(Pin0, LOW);
     digitalWrite(Pin1, LOW);
     digitalWrite(Pin2, LOW);
     digitalWrite(Pin3, LOW);
   break;
 }
 if(dir){
   _step++;
 }else{
   _step--;
 }
 if(_step>7){
   _step=0;
 }
 if(_step<0){
   _step=7;
 }
 delay(1);
}
```