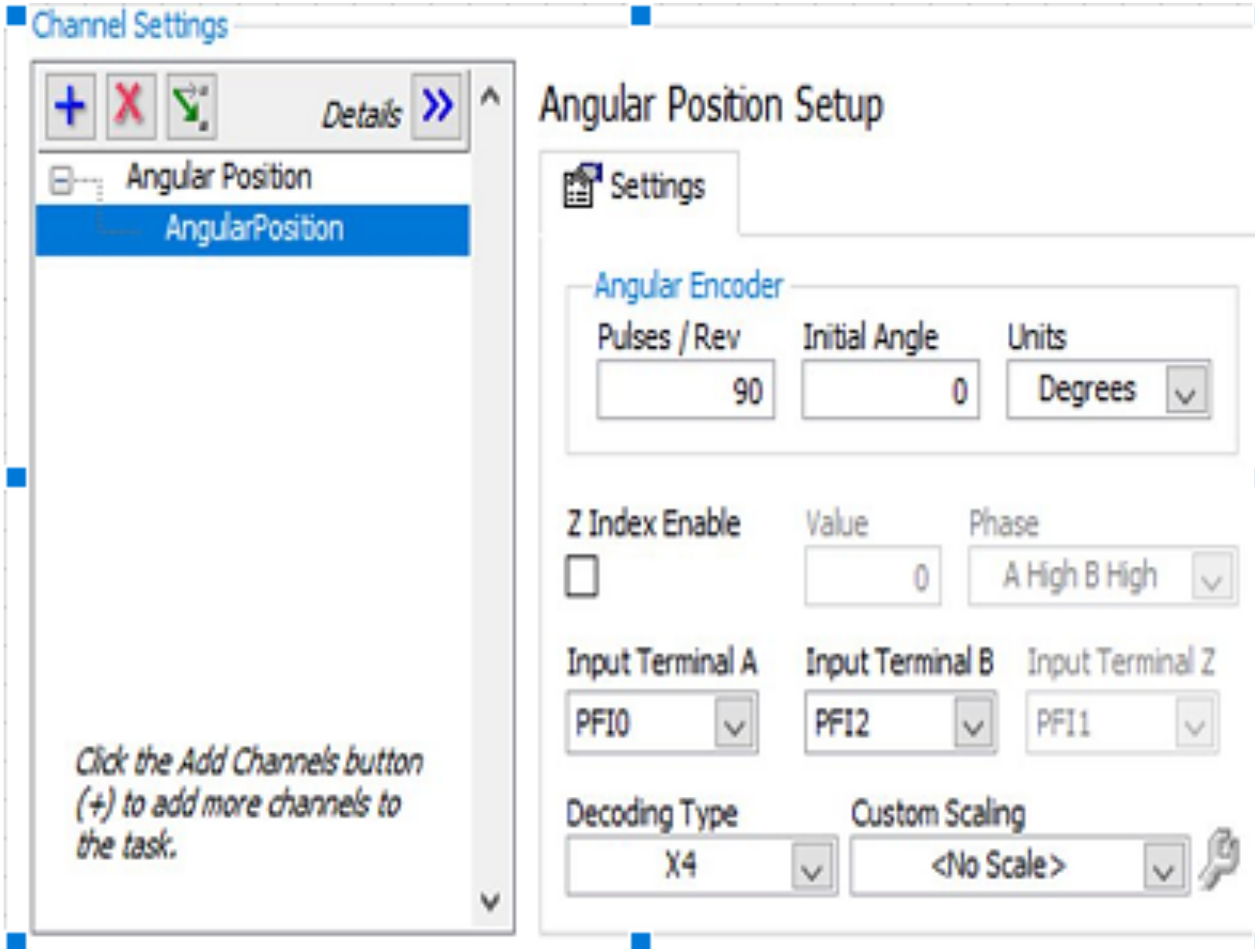
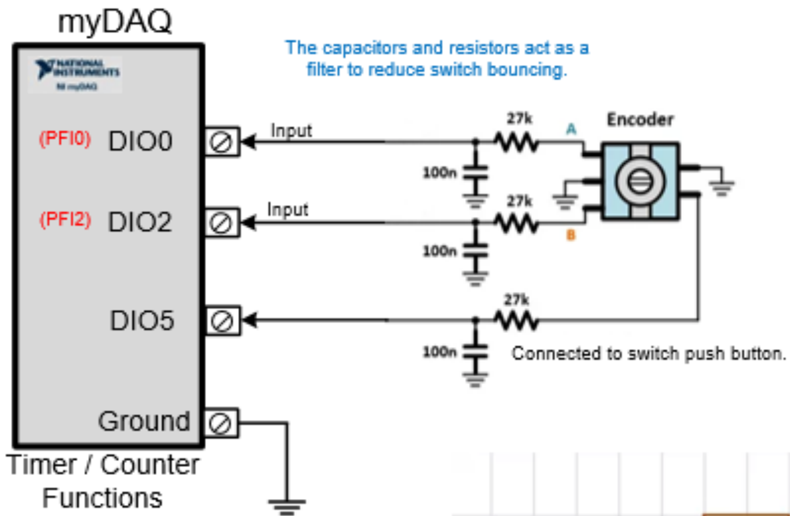
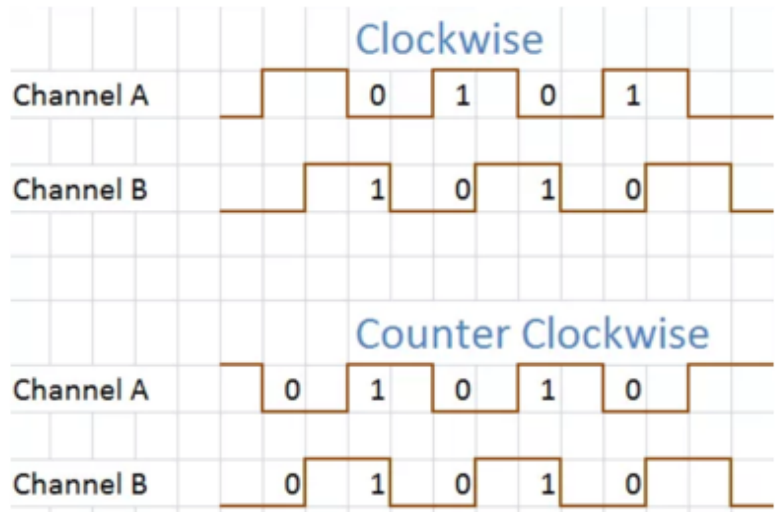
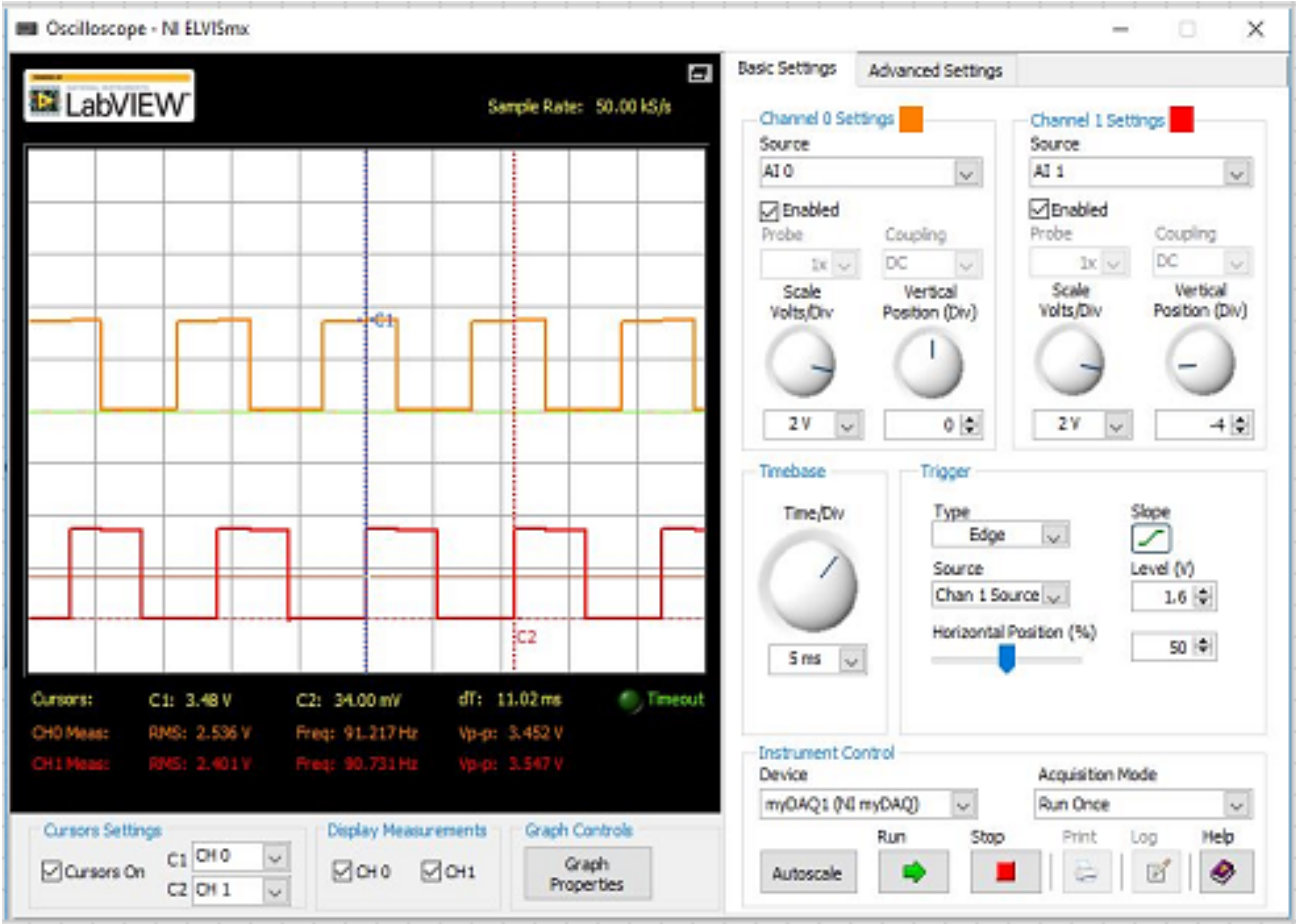

Channel Settings
Details
Angular Position
AngularPosition
Click the Add Channels button (+) to add more channels to the task.
Angular Position Setup
Settings
Angular Encoder
Pulses / Rev
90
Initial Angle
0
Units
Degrees
Z Index Enable
Value
0
Phase
A High B High
Input Terminal A
PFI0
Input Terminal B
PFI2
Input Terminal Z
PFI1
Decoding Type
X4
Custom Scaling
<No Scale>

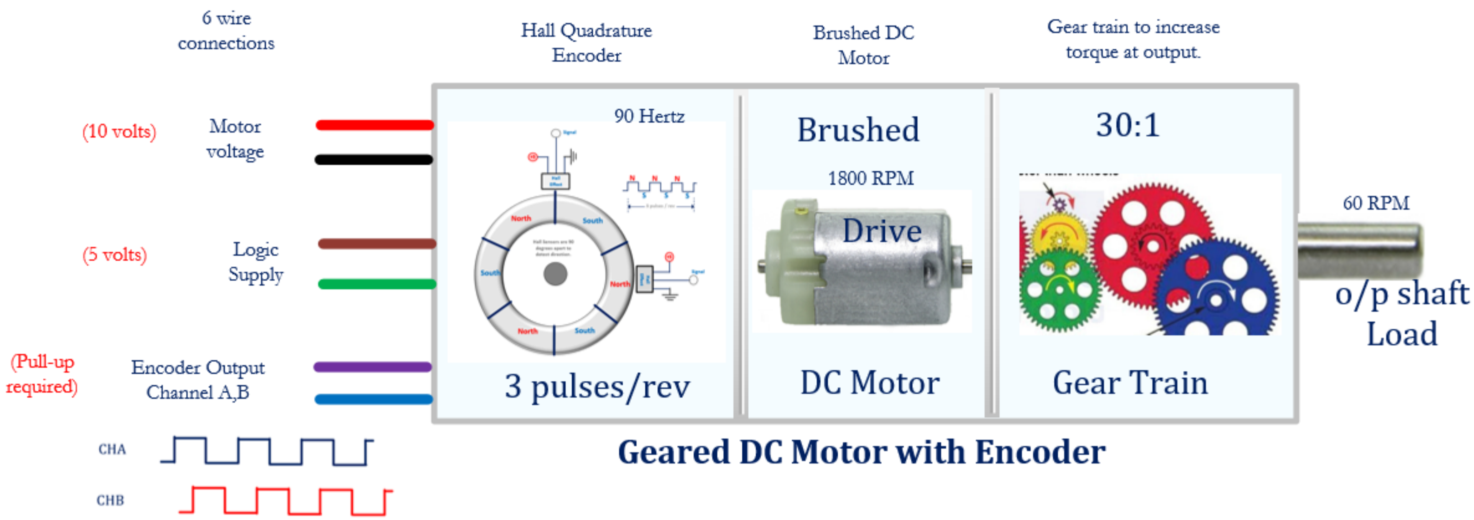


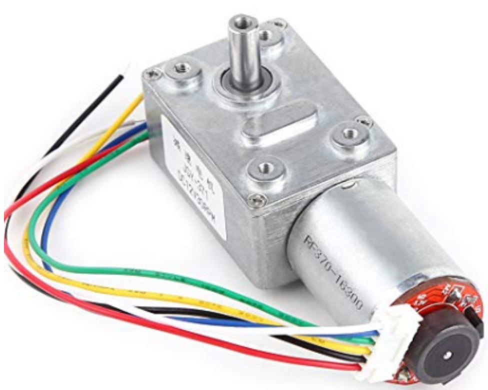
myDAQ
(PFI0) DIO0
(PFI2) DIO2
DIO5
Ground
Timer / Counter
Functions
The capacitors and resistors act as a
filter to reduce switch bouncing.
Input
Input
27k
27k
27k
100n
100n
100n
A
B
Encoder
Connected to switch push button.



Clockwise
Channel A
0 1 0 1
Channel B
1 0 1 0
Counter Clockwise
Channel A
0 1 0 1 0
Channel B
0 1 0 1 0


Oscilloscope - NI ELVISmx
LabVIEW
Sample Rate: 50.00 kS/s
Cursors: C1: 3.48 V C2: 34.00 mV dT: 11.02 ms
Timeout
CH0 Meas: RMS: 2.536 V Freq: 91.217 Hz Vp-p: 3.452 V
CH1 Meas: RMS: 2.401 V Freq: 90.731 Hz Vp-p: 3.547 V
Basic Settings
Advanced Settings
Channel 0 Settings
Channel 1 Settings
Source
AI 0
AI 1
Enabled
Probe
Coupling
DC
Scale Volts/Div
Vertical Position (Div)
2 V
0
-4
Timebase
Time/Div
5 ms
Trigger
Type
Edge
Source
Chan 1 Source
Slope
Level (V)
1.6
Horizontal Position (%)
50
Instrument Control
Device
myDAQ1 (NI myDAQ)
Acquisition Mode
Run Once
Autoscale
Run
Stop
Print
Log
Help
Cursors Settings
Cursors On
C1 CH 0
C2 CH 1
Display Measurements
CH 0
CH 1
Graph Controls
Graph Properties





High Torque Turbine Worm Geared Motor Reducer Motor with Encoder DC 12V 10/20/30/40/100RPM(10RPM)

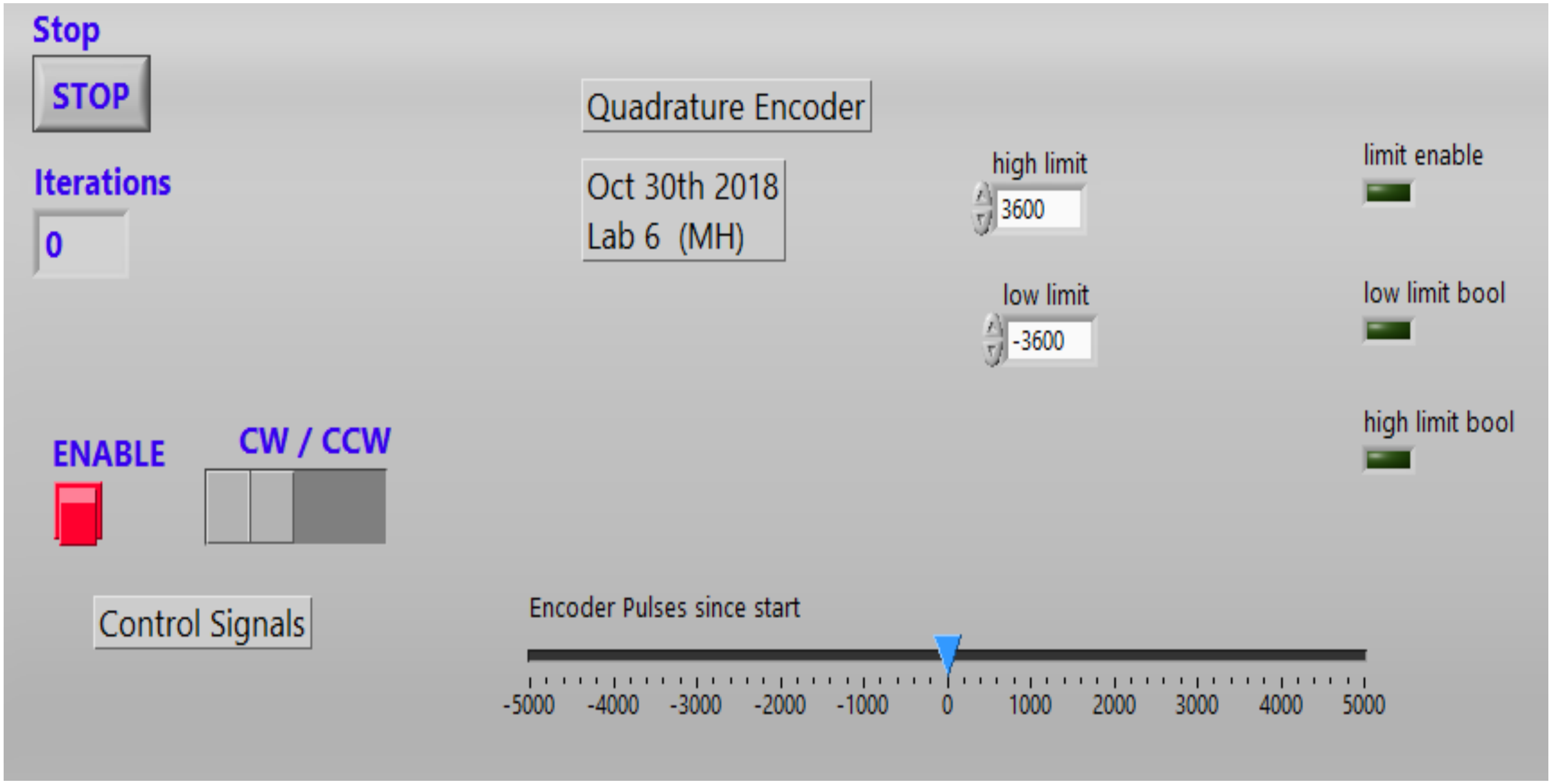
by Walfront

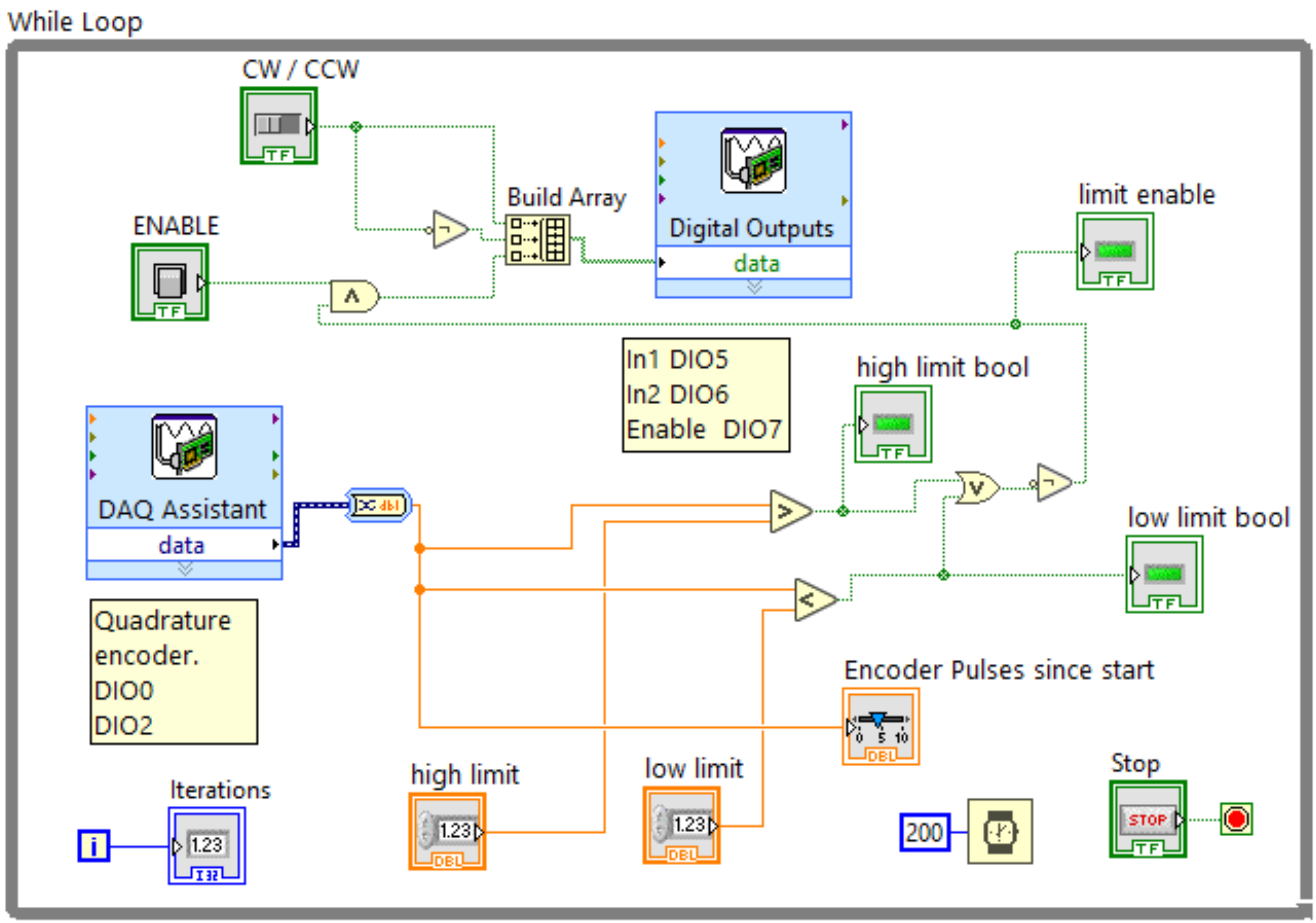
★★★★★ 1 customer review

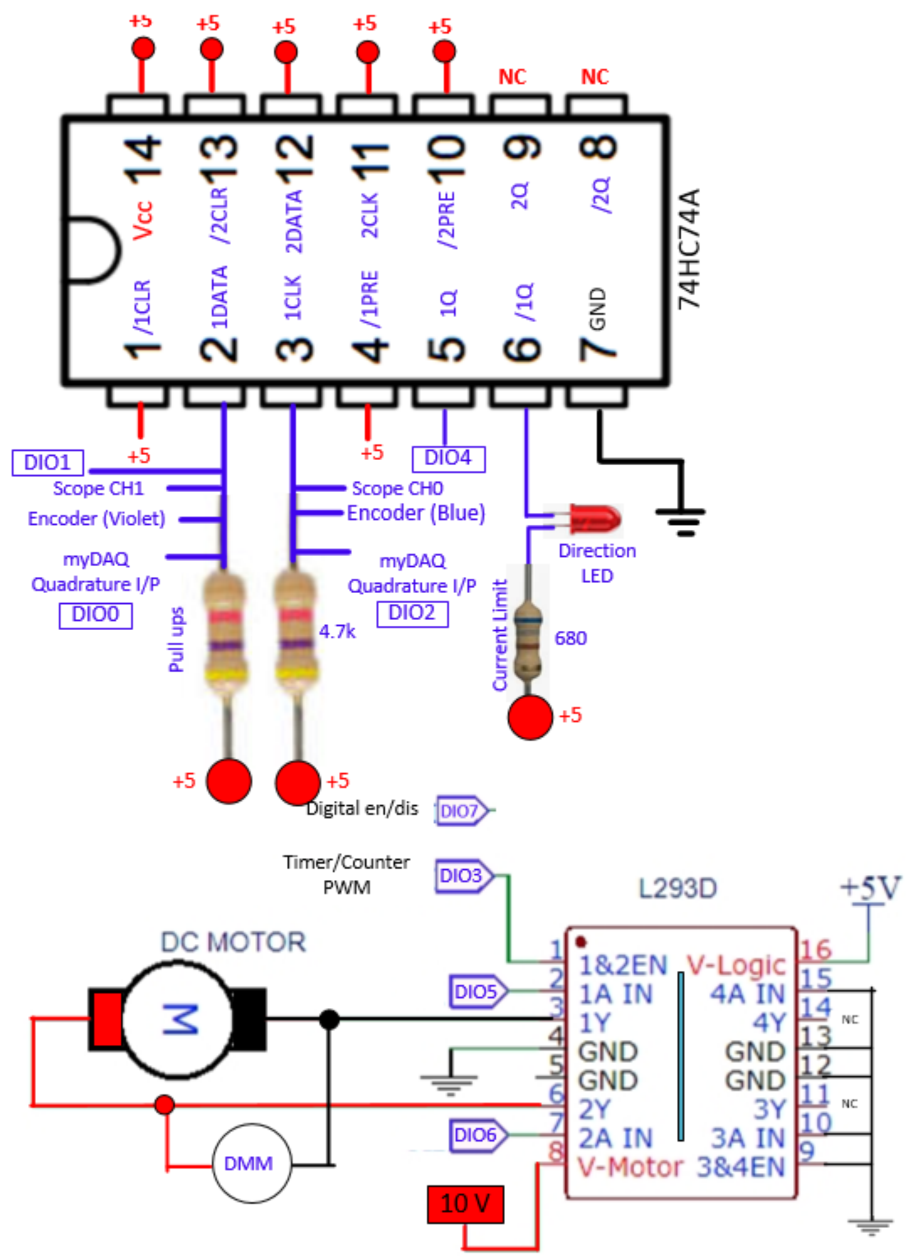
Price: CDN$ 22.99 ✓prime | FREE One-Day

2 new from CDN$ 22.99

- Two-phase reduction gear motor with encoder
- Using precious metals carbon brush
- Long service life, low noise, large torque
- Stable and reliable performance
- Can rotate and reversal, work more accurate


Stop
STOP
Iterations
0
Quadrature Encoder
Oct 30th 2018
Lab 6 (MH)
high limit
3600
low limit
-3600
limit enable
low limit bool
high limit bool
ENABLE
CW / CCW
Control Signals
Encoder Pulses since start
-5000 -4000 -3000 -2000 -1000 0 1000 2000 3000 4000 5000


While Loop
CW / CCW
Build Array
Digital Outputs
data
ENABLE
limit enable
In1 DIO5
In2 DIO6
Enable DIO7
high limit bool
DAQ Assistant
data
low limit bool
Quadrature encoder.
DIO0
DIO2
Encoder Pulses since start
high limit
low limit
Stop
Iterations
200



In Part A of lab 6:

DIO7 controls the L293D enable signal
DIO5 connects to IN1 on the L293D
DIO6 connects to IN2 on the L293D

In Part B of Lab 6:

DIO7 controls the L293D enable signal
DIO5 connects to IN1 on the L293D to control mode
DIO6 connects to IN2 on the L293D to control mode
DIO0 connects to one of the motor quadrature encoders
DIO2 connects to the other motor quadrature encoder signal

In Part C of Lab 6:

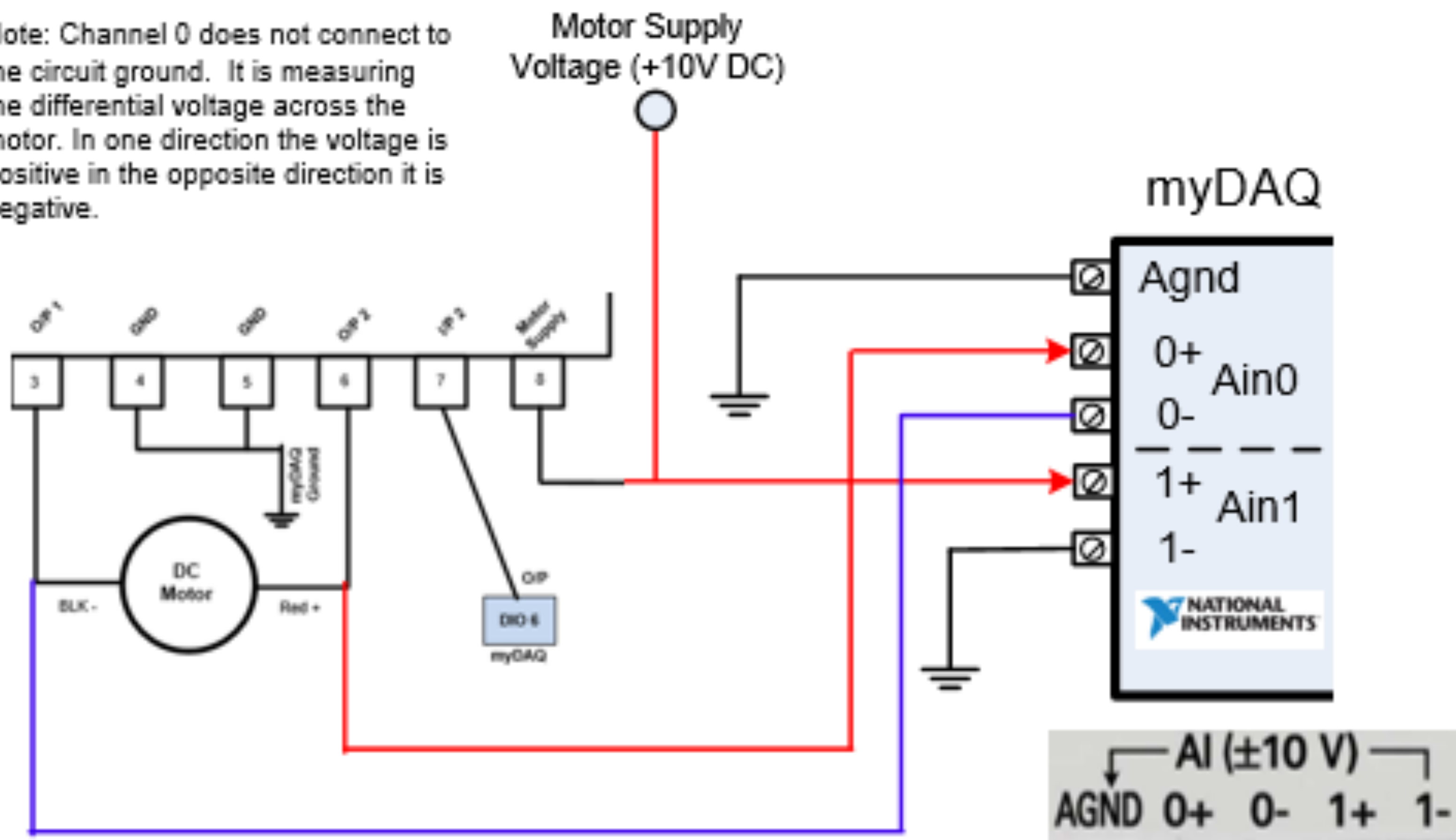
DIO3 (PWM) control the L293 Enable signal
DIO5 connects to IN1 on the L293D to control mode
DIO6 connects to IN2 on the L293D to control mode

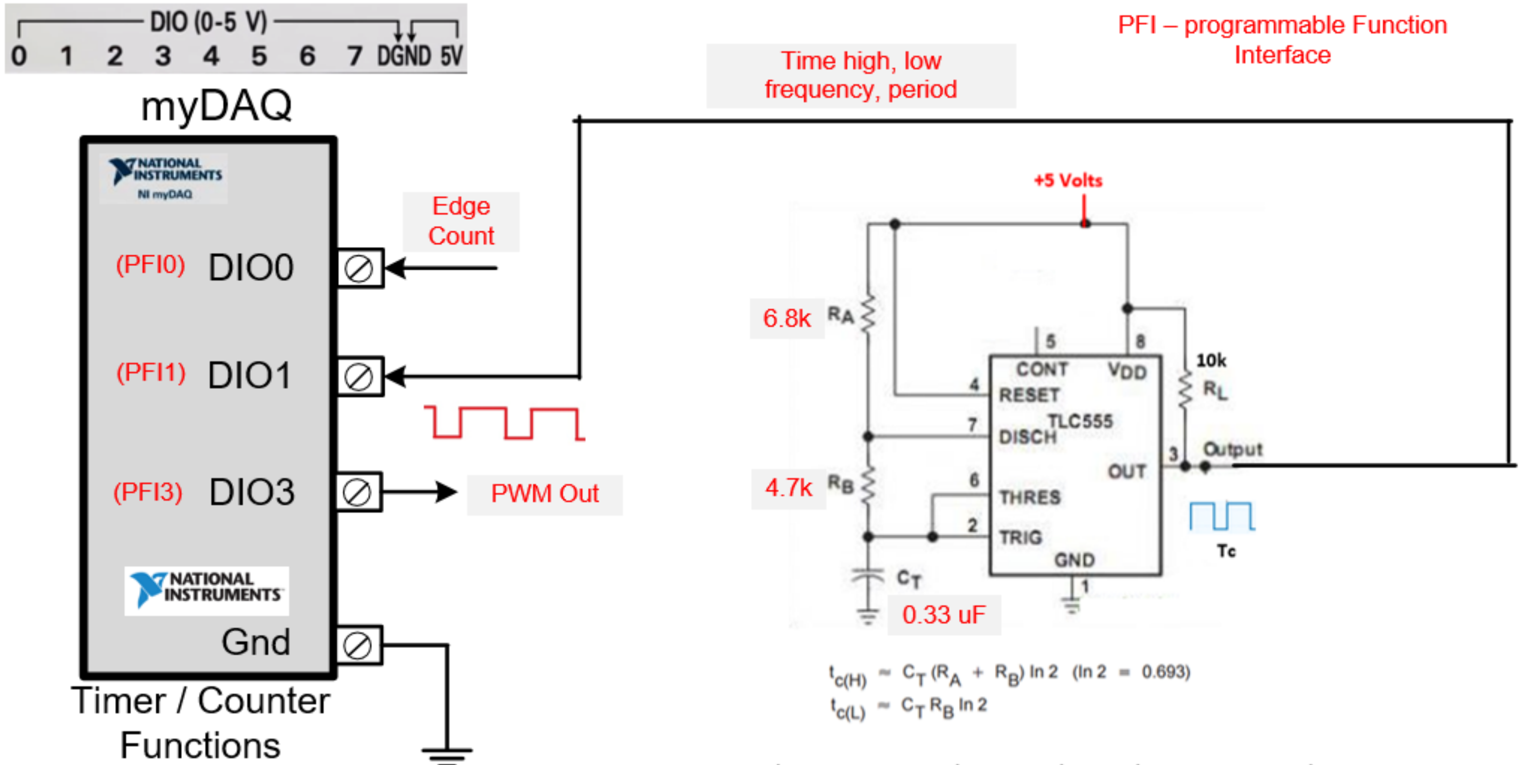
In Part D of Lab 6:

myDAQ DIO7 controls the L293D enable signal
myDAQ DIO5 connects to IN1 on the L293D to control mode
myDAQ DIO6 connects to IN2 on the L293D to control mode
myDAQ DIO4 connects to the Q output of the 74HC74A to read direction
myDAQ DIO1 connects to the one of the motor quadrature encoder signal to measure speed

The 74HC74A data connects to one of the motor quadrature encoders to measure speed and determine direction.
The 74HC74A connects to the other motor quadrature encoder signal to determine direction.

Note: Channel 0 does not connect to the circuit ground. It is measuring the differential voltage across the motor. In one direction the voltage is positive in the opposite direction it is negative.





$t_{c(H)} \approx C_T (R_A + R_B) \ln 2 \quad (\ln 2 = 0.693)$

$t_{c(L)} \approx C_T R_B \ln 2$

The capacitor charges through RA + RB and discharges through RB only.

ALGONQUIN COLLEGE

Class Notes:

DC Stepper and Servo Motor Controls

CAM8302E Fall 2018

Slide Index (Total 62 slides):

DC Motor Types

Brushed DC

Advantages

- Cheapest and simplest motor
- Speed linear to applied voltage
- Simple motor control

Disadvantages

- High maintenance
- Low life-span (due to physical wear on brushes)

Brushless DC

Advantages

- High efficiency
- Little to no maintenance
- Long life span
- High output power per frame size

Disadvantages

- More complicated motor control
- Large initial costs

Stepper

Advantages

- Accurate position control
- Excellent low speed torque
- Long life

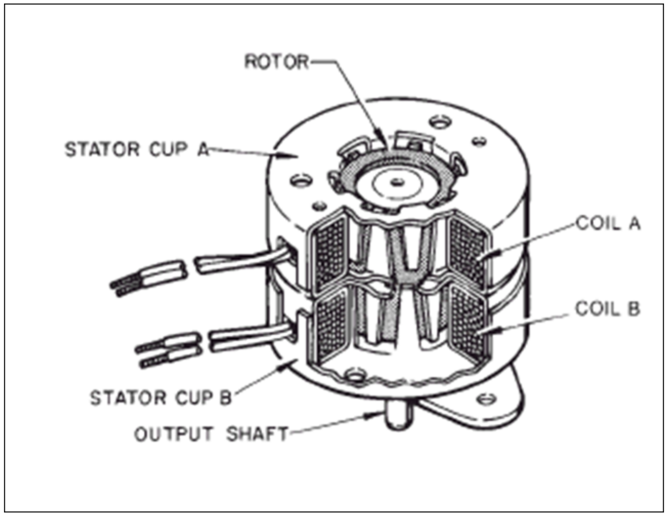
Disadvantages

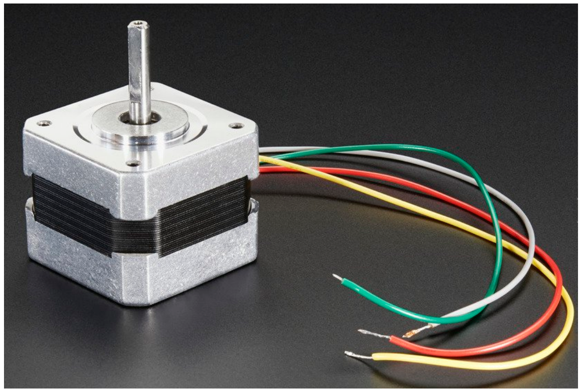
- Low efficiency
- Prone to resonances, noise, and torque ripple
- Cannot accelerate loads rapidly

Stepper Motor

A stepping motor is a motor which has the capability of:

- rotating in either direction
- starting or stopping in various positions
- moving the rotor in precise angular increments for each step applied
- the motor can be stopped so that is not easily rotated (holding torque).





https://www.youtube.com/watch?v=vxxnPJBxG3M Microchip Stepper Motors

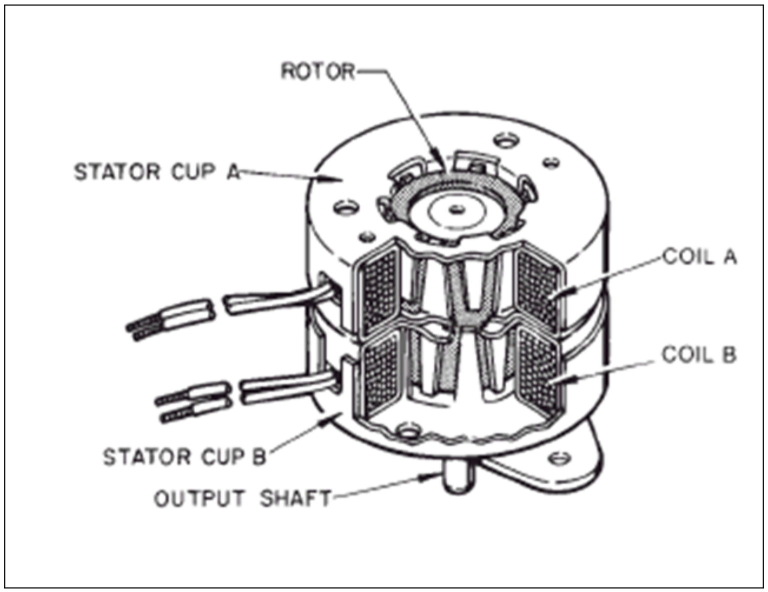
https://www.youtube.com/watch?v=TWMai3oirnM How To Mechatronics

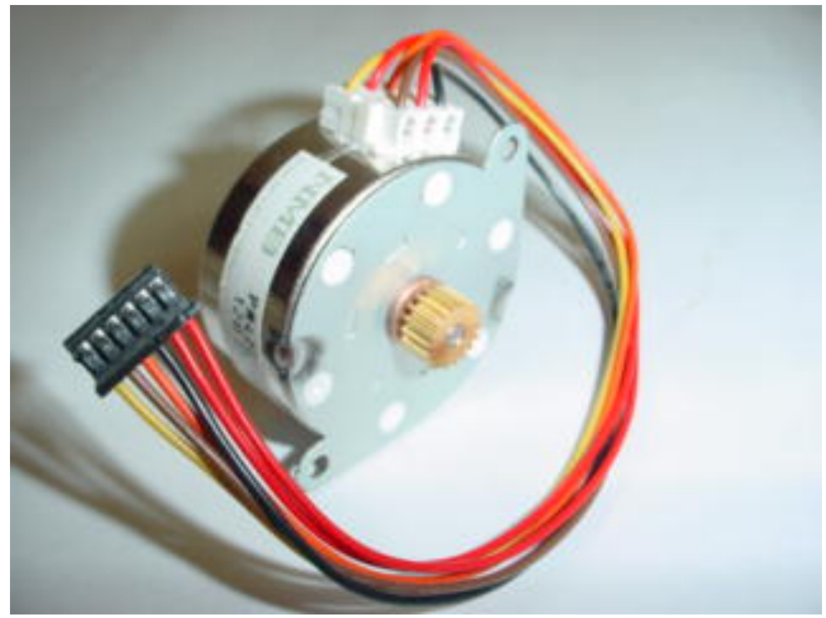
https://www.youtube.com/watch?v=KbDPgxHpgAA Stepper with Arduino

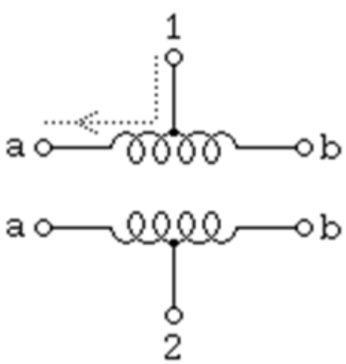
http://howtomechatronics.com/projects/diy-vending-machine-arduino-based-mechatronics-project/ How To Mechatronics

Vending Machine

Stepper Motors

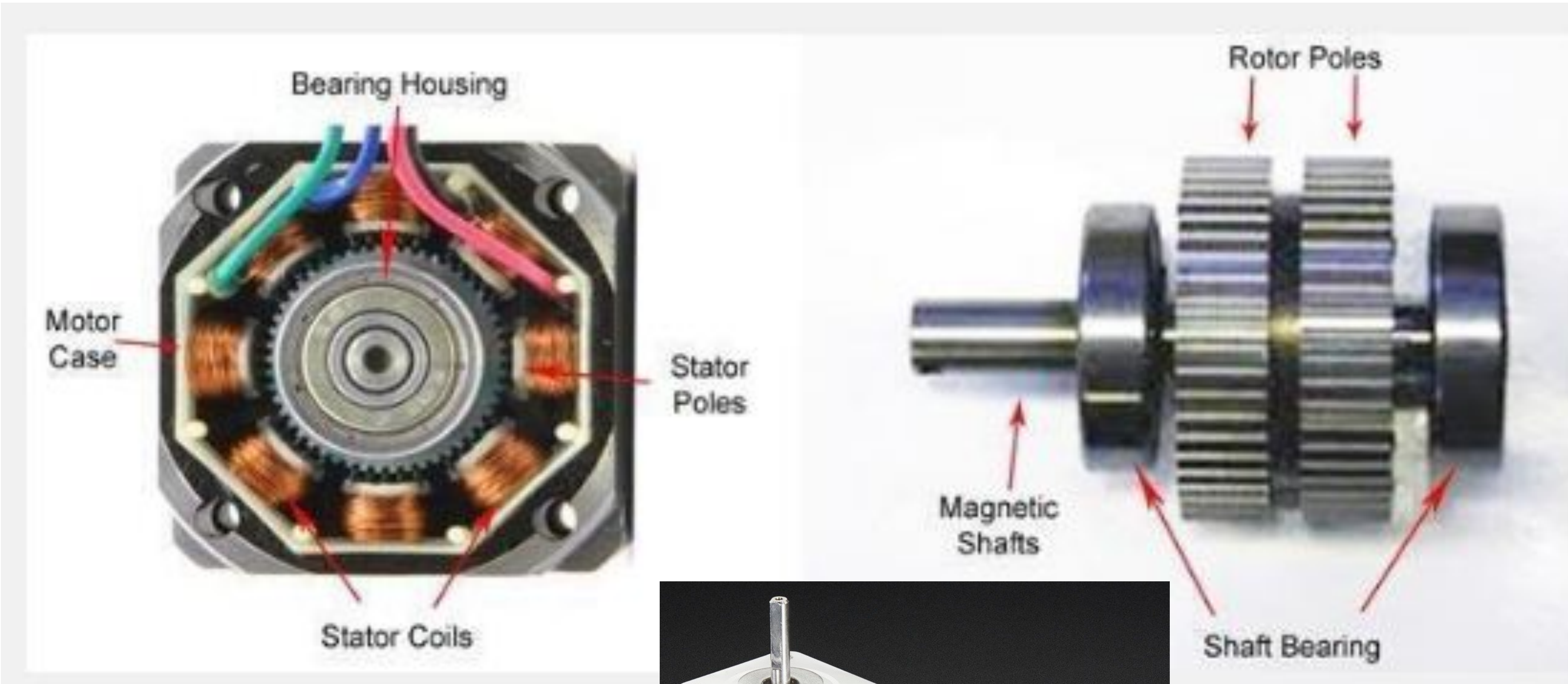


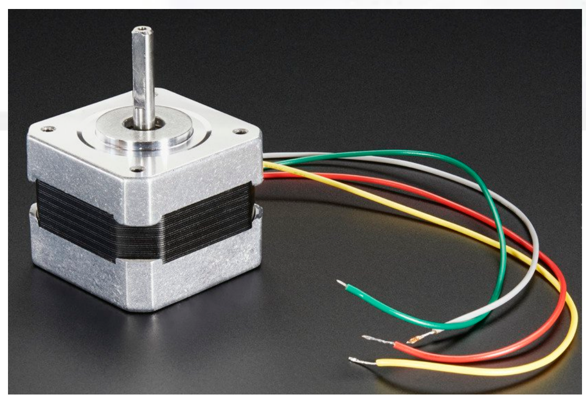




Permanent Magnet   Unipolar Stepper Motor
Unipolar – current flow in only one direction
Bipolar – current flow in two directions

Inside A Stepper Motor

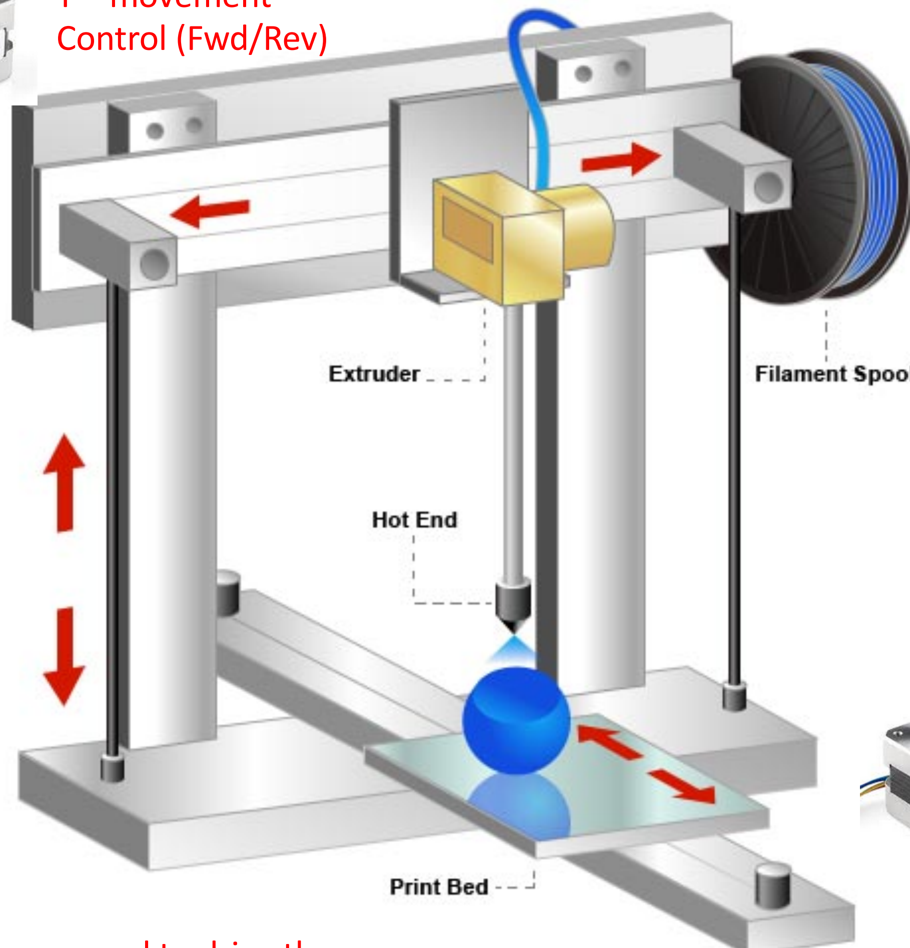




Stator – Parts that do not rotate.
Rotor – Rotating parts.
Pole/Phase – coil of wire.

3D Printer Parts for Position

Y – movement
Control (Fwd/Rev)



Stepper motors are used in 3D printer to control the X, Y and Z axis position.

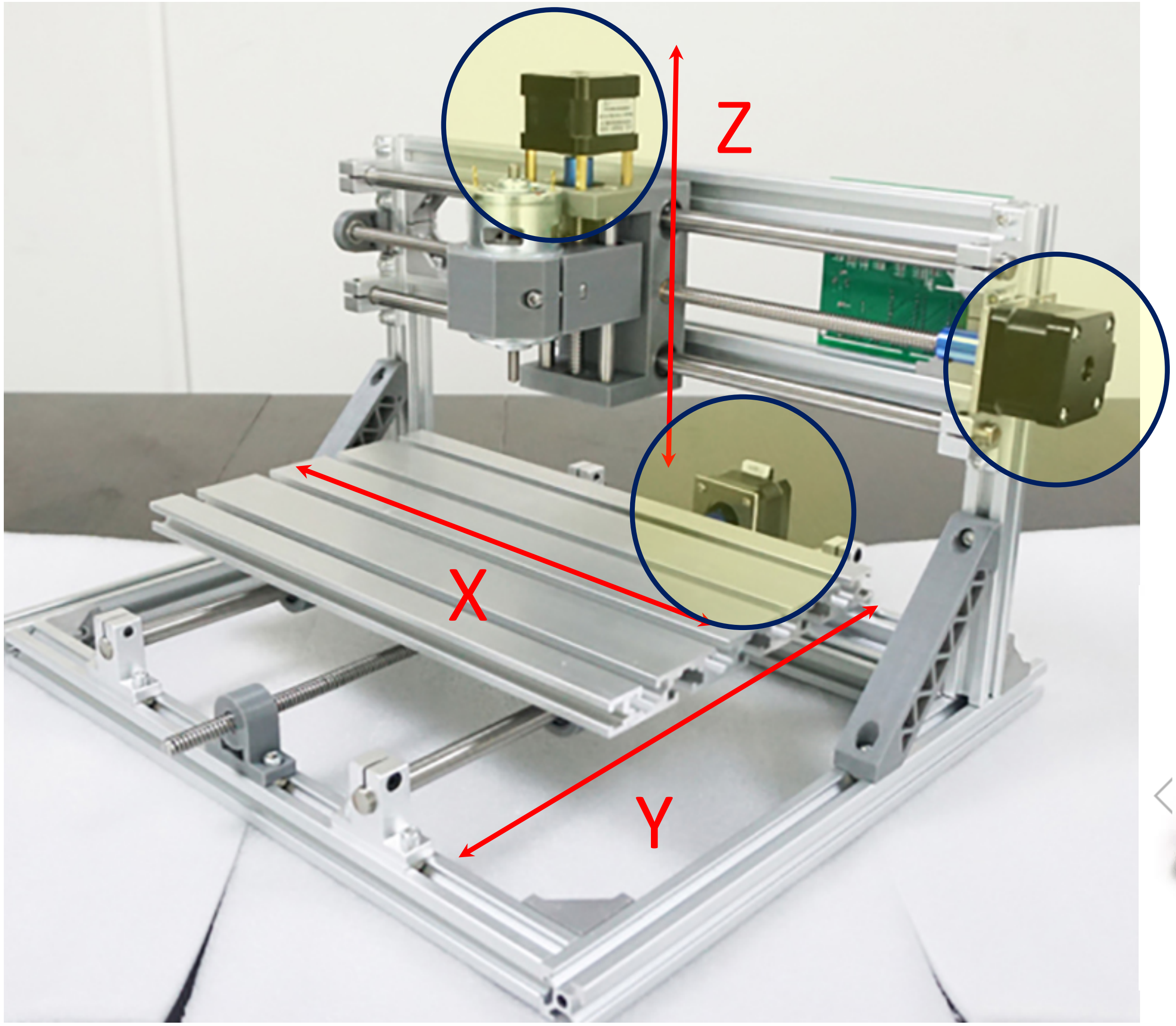
The motors are connected to a belt drive or a lead screw.

Z – movement
Control (Up/Down)

4th stepper used to drive the filament to the print head.

X – movement
Control (Left/Right)

CNC Using 3 Stepper Motors

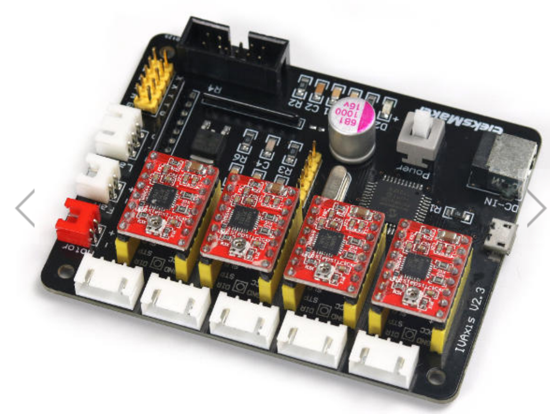


CNC – computer numerical control

Laser Etcher

PCB etching

Controller for 4 stepper motors.



3D Printer Parts

There are four types of filament: metal, ceramic, composite, and plastic. Plastic, the most common type, can be subdivided into five categories:

- Acrylonitrile Butadiene Styrene (ABS)
- Polylactic Acid (PLA)
- Polyvinyl Alcohol (PVA)
- Polycarbonate (PC)
- High-Density Polyethylene (HDPE)

ABS is the most common type of plastic filament. **PLA** is a runner-up, being both biodegradable and available in soft or hard form. **PVA**, a dissolvable substance found in lubricants and adhesives, has a small, but growing, niche. **PC**, used in media discs and bullet-resistant glass, is still in the developmental stage. **HDPE**, a type of moisture-resistant cheap plastic found in bottles and pipes, has little use because of its proneness to warping and shrinkage.

—

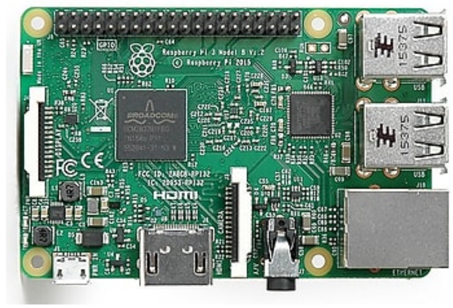
3D printers measure resolution by tenths to hundredths of a millimeter (mm). The more minute the resolution, the longer the printing process. A small object set at 0.3 mm resolution may take 15 min., while the same object may take 3 hrs. to print when set at a 0.1 mm resolution. Of course, it takes a capable printer to put out such a precise resolution–in other words, it costs more money.

Stepper YouTube Videos

https://www.youtube.com/watch?v=vxxnPJBxG3M	Microchip Video
https://www.youtube.com/watch?v=MHdz3c6KLrg	Part 1 operation
https://www.youtube.com/watch?v=t-3VnLadIbc	Part 2 operation
https://www.youtube.com/watch?v=blJxz28HuQI	3D Printer
https://www.youtube.com/watch?v=dmk6zIkj7WM	Anaheim Automation
https://www.youtube.com/watch?v=eyqwLiowZiU	Stepper Animation

DC Stepper Motor

- Stepper motors are easy to control with digital / microcontroller boards such as the Arduino or the Raspberry PI.

- Stepper motors provide very precise angular positioning.

- Speed and direction are easily controlled with digital circuits .

- Stepper motors have no feedback and may miss steps under high torque conditions.

- Stepper motors often use a switch or sensor to determine a known home position.

DC Stepper Motor

Types of Step Motors

There are three basic types of step motors in common use:

- Active rotor: permanent magnet (PM)
- Reactive rotor: variable reluctance (VR)
- Combination of VR and PM: Hybrid (HY)

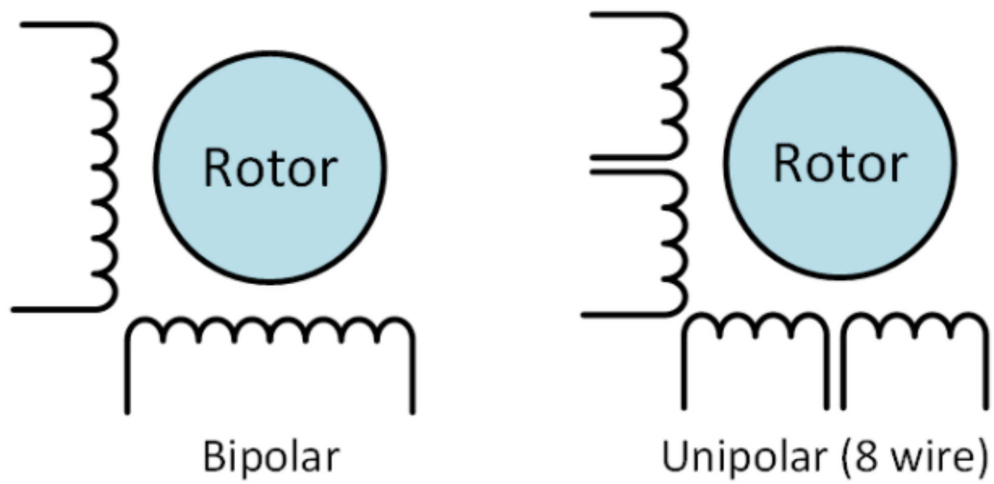
These are brushless electrical machines which rotate in fixed angular increments when connected to a sequentially switched DC current.

The Hybrid type is most common for higher torque and higher speed applications.

Stepper Motors Types

Coils and Phases

A stepper motor may have any number of coils. But these are connected in groups called "phases". All the coils in a phase are energized together.



* May have 5, 6 or 8 wires.

Unipolar vs. Bipolar

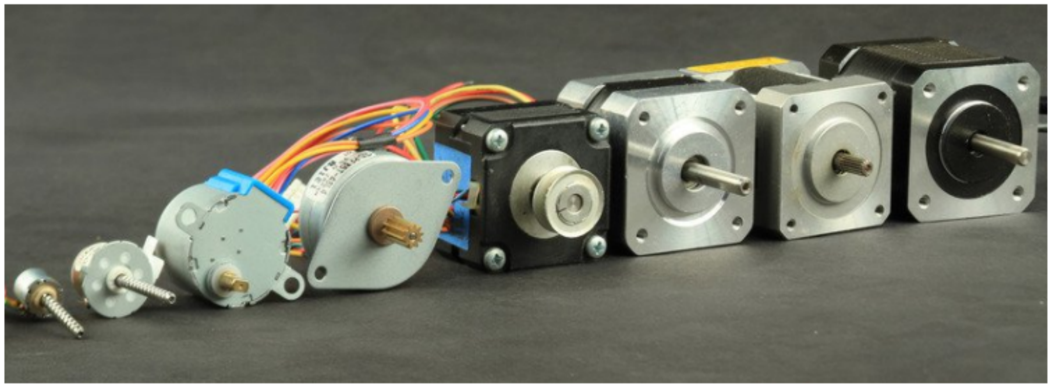
Unipolar drivers, always energize the phases in the same way. One lead, the "common" lead, will always be negative. The other lead will always be positive. Unipolar drivers can be implemented with simple transistor circuitry. The disadvantage is that there is less available torque because only half of the coils can be energized at a time.

Bipolar drivers use H-bridge circuitry to actually reverse the current flow through the phases. By energizing the phases with alternating the polarity, all the coils can be put to work turning the motor.

Types of Steppers

by Bill Earl

There are a wide variety of stepper types, some of which require very specialized drivers. For our purposes, we will focus on stepper motors that can be driven with commonly available drivers. These are: Permanent Magnet or Hybrid steppers, either 2-phase bipolar, or 4-phase unipolar.



Motor Size

One of the first things to consider is the work that the motor has to do. As you might expect, larger motors are capable of delivering more power. Stepper motors come in sizes ranging from smaller than a peanut to big NEMA 57 monsters.

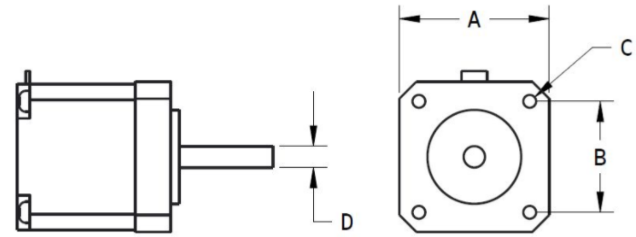
Most motors have torque ratings. This is what you need to look at to decide if the motor has the strength to do what you want.

NEMA 17 is a common size used in 3D printers and smaller CNC mills. Smaller motors find applications in many robotic and animatronic applications. The larger NEMA frames are common in CNC machines and industrial applications.

The NEMA numbers define standard faceplate dimensions for mounting the motor. They do not define the other characteristics of a motor. Two different NEMA 17 motors may have entirely different electrical or mechanical specifications and are not necessarily interchangeable.

adafruit

NEMA – national electrical manufacturers association



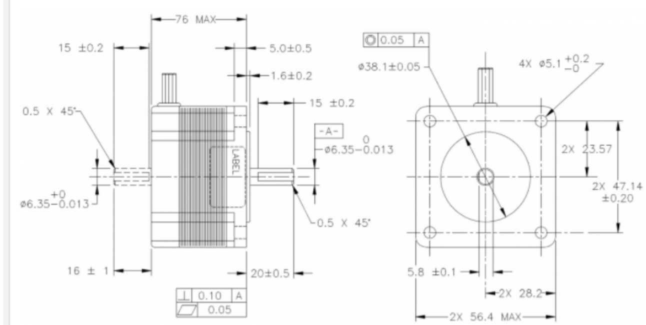
SIZE	A	B	C	D (Dia)
NEMA 11	28.2	23	M2.5 Thread	5
NEMA 14	35.2	26	M3 Thread	5
NEMA 17	42.3	31	M3 Thread	5
NEMA 23	56.4	47.1	5.5 Dia	6.35
NEMA 34	86	69.6	5.5 Dia.	14
NEMA 42	110	89	8.5	19

What do NEMA sizes mean?

OCT 28

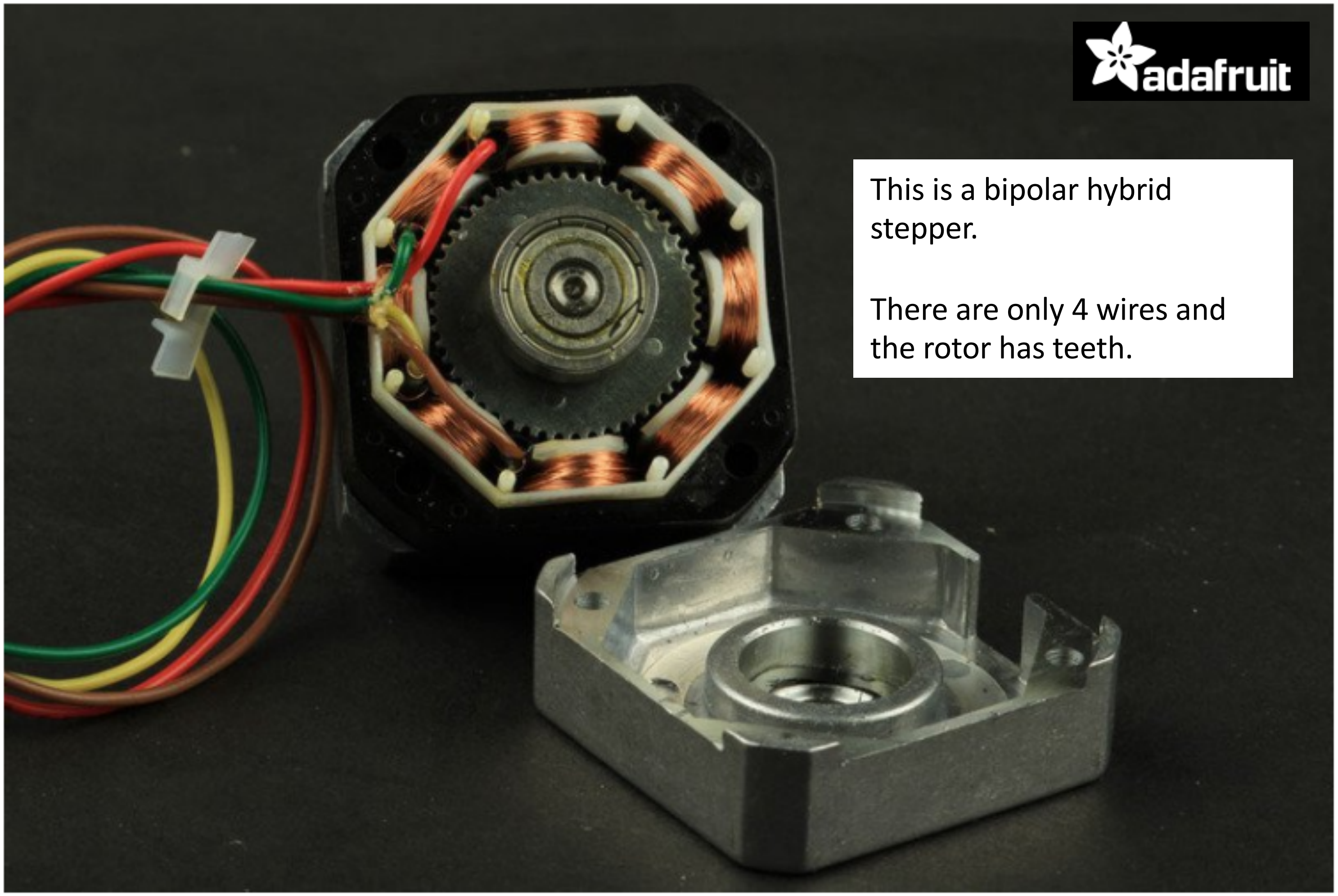
by Jeff Kordik

Step motors are categorized by frame size, such as "size 11" or "size 23". Ever wonder how that came to be or what it means? The National Electrical Manufacturers Association sets standards for many electrical products, including step motors. Generally speaking, "size 11" mean the mounting face of the motor is 1.1 inches square. So a Size 23 step motor is, wait for it, 2.3 inches square. Or 56.4 mm as shown below.



Wiring

There are many variations in stepper motor wiring. For our purposes, we will focus on steppers that can be driven with commonly available drivers. These are Permanent Magnet or Hybrid steppers wired as 2-phase bipolar, or 4-phase unipolar.



Note: This is a modified unipolar stepper motor. The white wire was removes and the PCB trace cut.

a. **Orange to Violet** __________ **ohms**
b. **Violet to blue** __________ **ohms**
c. **Yellow to orange** __________ **ohms**

Each coil should be about 240 ohms.

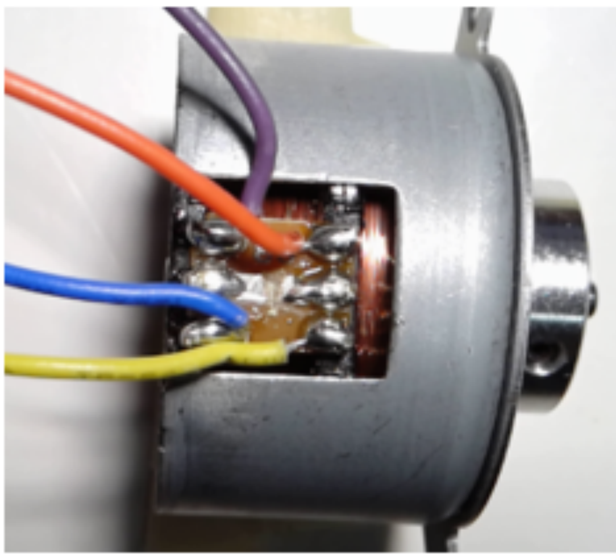


Figure 2 Modified Stepper

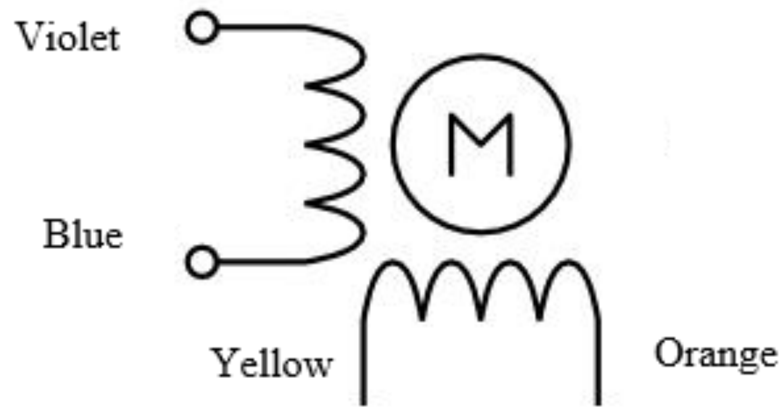
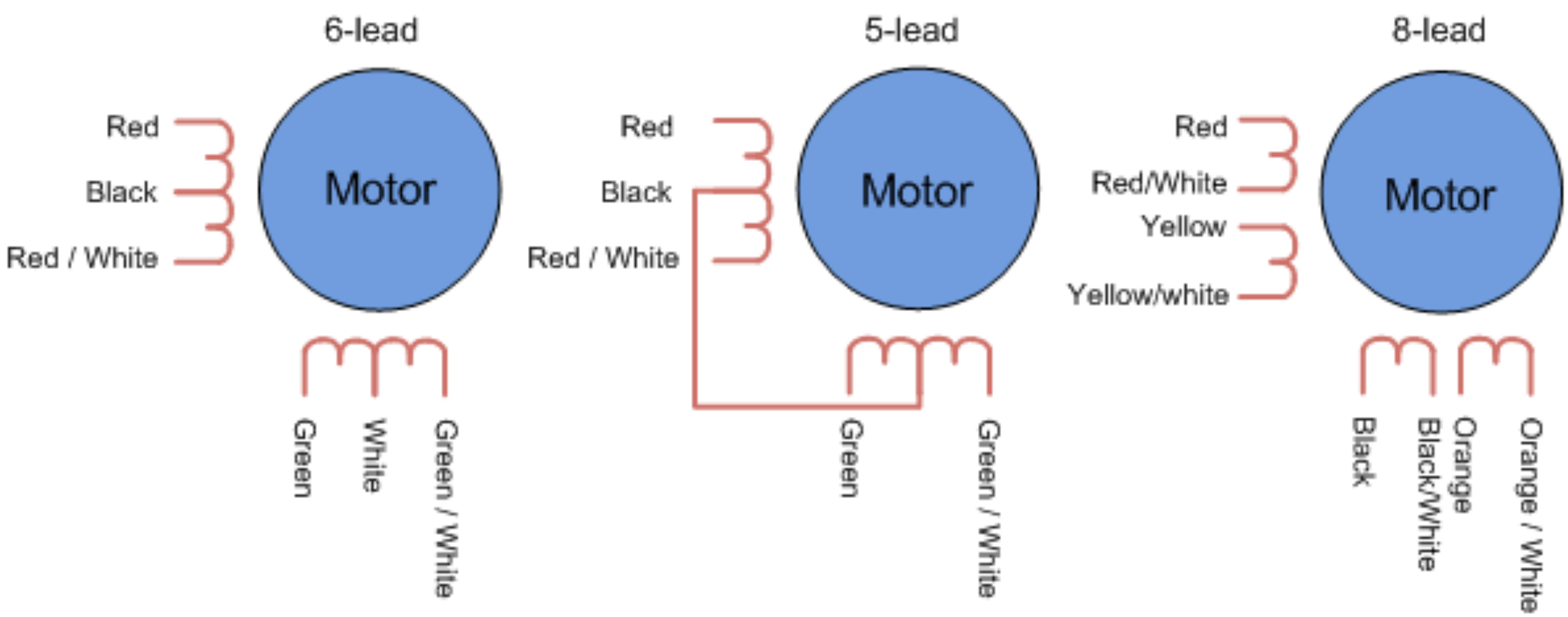


Figure 3 Modified Bipolar Wiring

Unipolar stepper motors have 5, 6 or 8 leads.

They all use two coils with a centre tap.

Some steppers have the wires connected internally, others are connected externally.



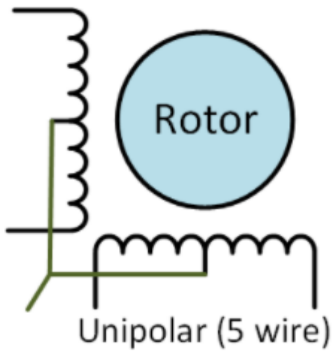
Unipolar Stepper Motor

Unipolar vs. Bipolar

Unipolar drivers, always energize the phases in the same way. One lead, the "common" lead, will always be negative. The other lead will always be positive. Unipolar drivers can be implemented with simple transistor circuitry. The disadvantage is that there is less available torque because only half of the coils can be energized at a time.

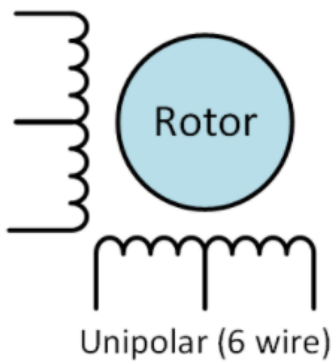
Bipolar drivers use H-bridge circuitry to actually reverse the current flow through the phases. By energizing the phases with alternating the polarity, all the coils can be put to work turning the motor.

A two phase bipolar motor has 2 groups of coils. A 4 phase unipolar motor has 4. A 2-phase bipolar motor will have 4 wires - 2 for each phase. Some motors come with flexible wiring that allows you to run the motor as either bipolar or unipolar.



5-Wire Motor

This style is common in smaller unipolar motors. All of the common coil wires are tied together internally abd brought out as a 5th wire. This motor can only be driven as a unipolar motor.

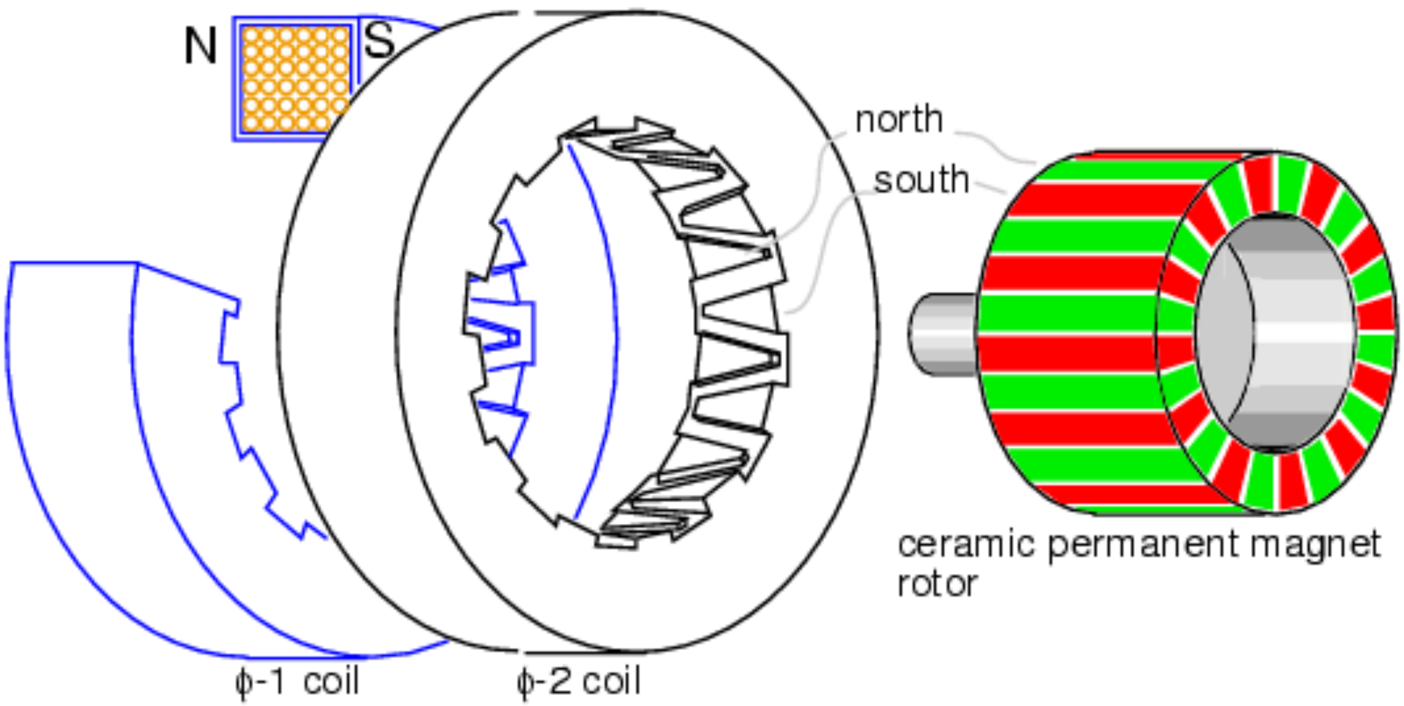


6-Wire Motor

This motor only joins the common wires of 2 paired phases. These two wires can be joined to create a 5-wire unipolar motor.

Or you just can ignore them and treat it like a bipolar motor!

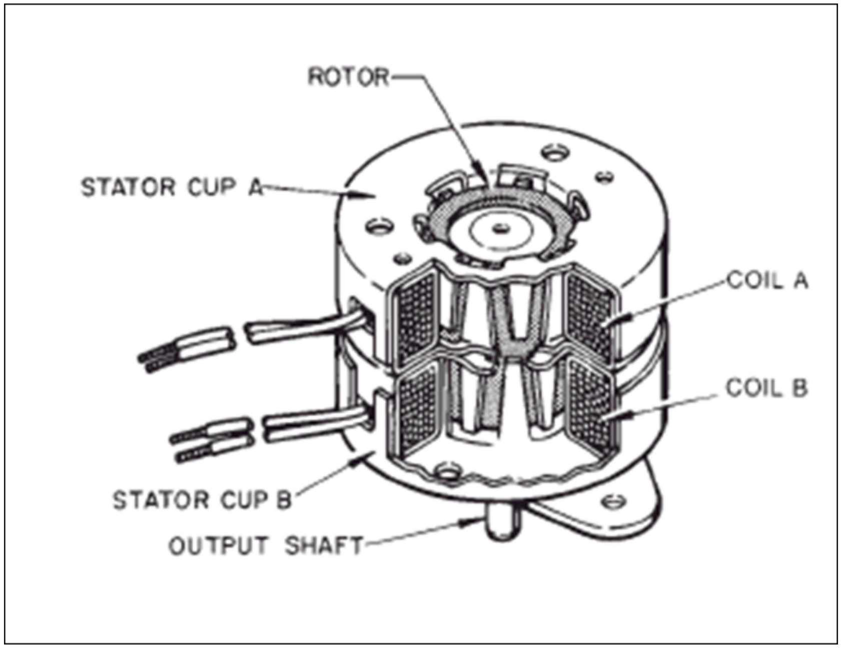
Stepper Motor



Permanent magnet stepper motor, 24-pole can-stack construction.

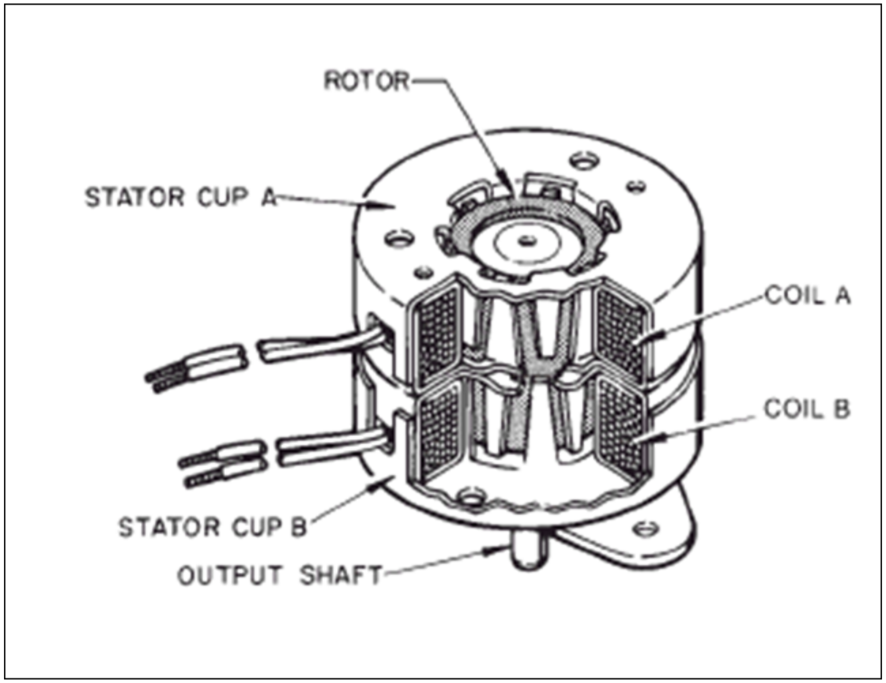
Stepper Motor

- A stepping motor is a motor which has the capability of rotating in either direction, starting or stopping in various positions, and moving the rotor in precise angular increments for each step applied. The motor can be stopped so that is not easily rotated (holding torque).



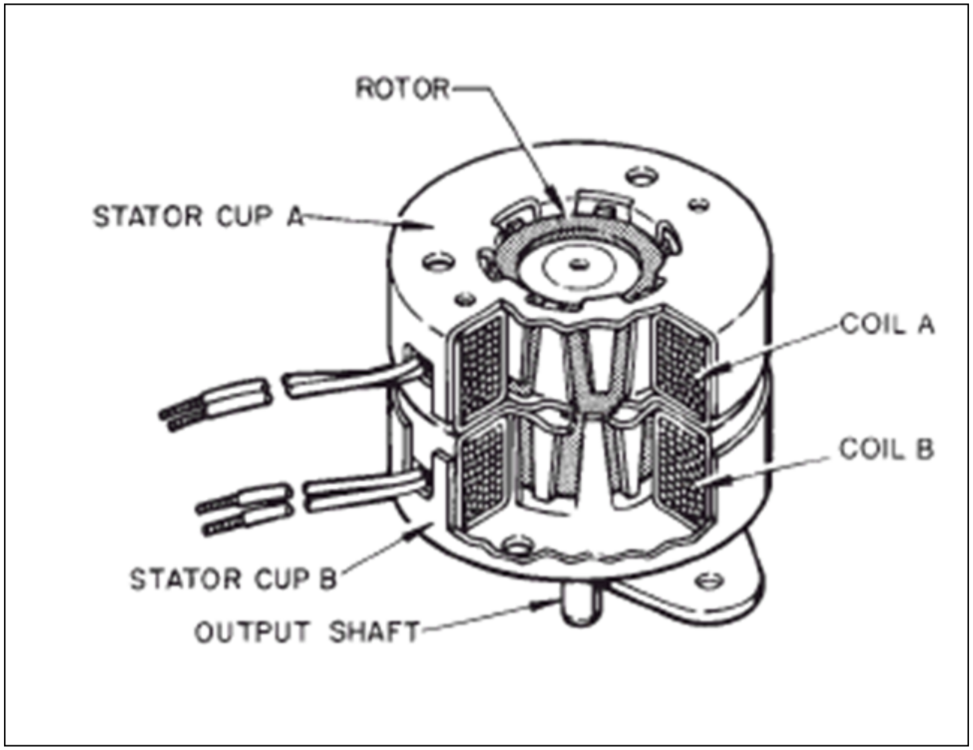
Stepper Motor Characteristics

- Two Coils of fine gauge wire – Unipolar has a centre tap.
- Permanent magnet rotor
- No Brushes
- Good Holding Torque (not turning)
- Rotates at precise angular increments. 1.8 deg, 5.0 deg. 7.5 deg.
- Stops and starts quickly
- Digital Control of direction and speed.
- No Feedback circuit is required.



Stepper Motor Applications

- Drive Conveyor Belts
- Robotics
- Mechanical Positioning
- Ventilation/Lighting Control
- Camera Controls
- Pick & Place Machines
- Assembly Lines
- Industrial Control
- CNC and 3D printers



Stepper Motor

Permanent-magnet (PM) Stepper Motors

The **permanent-magnet stepper motor** operates on the reaction between a permanent-magnet rotor and an electromagnetic field. Figure 6-18 shows a basic two-pole PM stepper motor. The rotor shown in Figure 6-18(a) has a permanent magnet mounted at each end. The stator is illustrated in Figure 6-18(b). Both the stator and rotor are shown as having teeth. The teeth on the rotor surface and the stator pole faces are offset so that there will be only a limited number of rotor teeth aligning themselves with an energized stator pole. The number of teeth on the rotor and stator determine the step angle that will occur each time the polarity of the winding is reversed. The greater the number of teeth, the smaller the step angle.

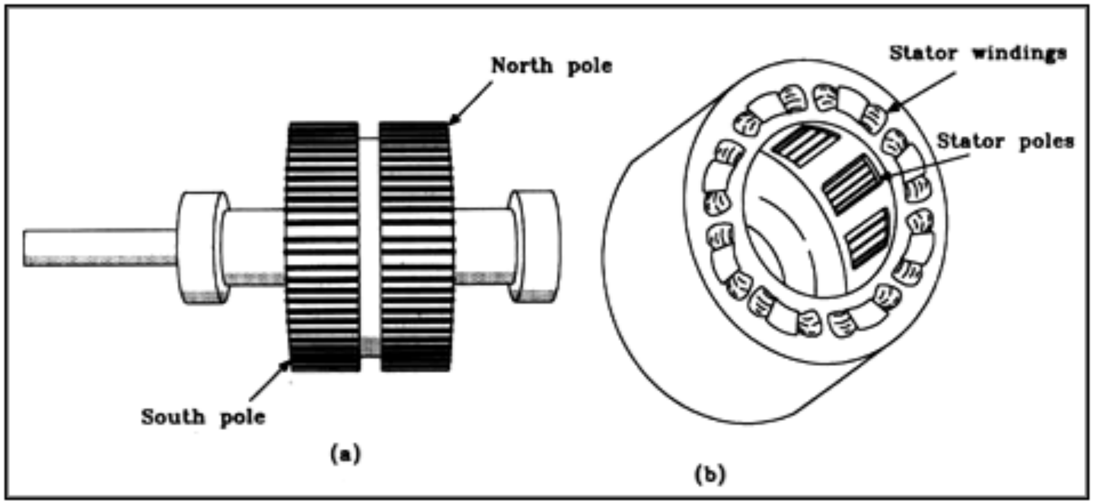


Figure 6-18 Components of a PM stepper motor: (a) Rotor; (b) stator

From NI.com

NI – National Instruments (LabVIEW)

Bipolar Stepper Motor

Bipolar Motors

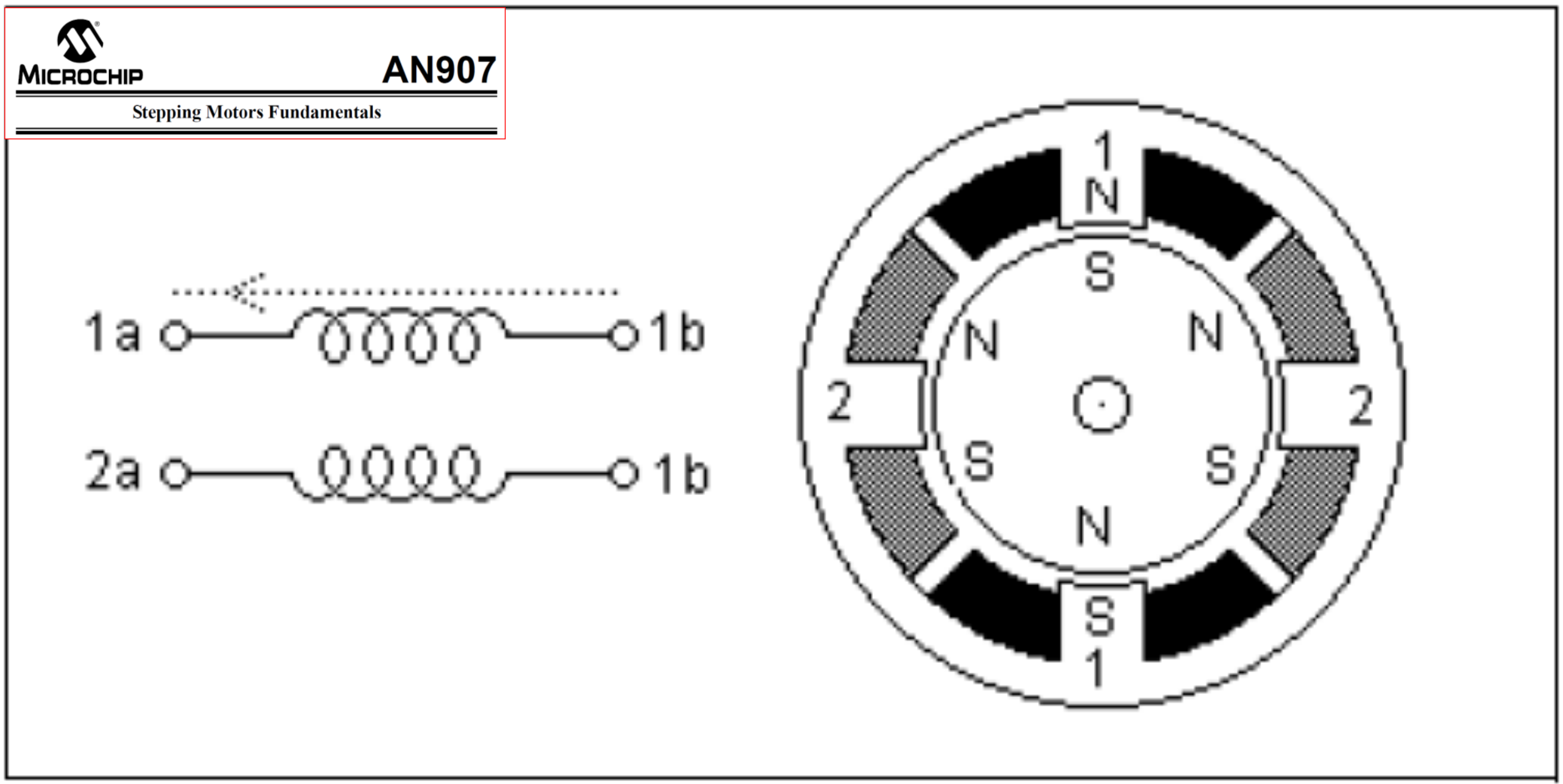
MICROCHIP AN907

Stepping Motors Fundamentals

Bipolar stepping motors are composed of two windings and have four wires. Unlike unipolar motors, bipolar motors have no center taps. The advantage to not having center taps is that current runs through an entire winding at a time instead of just half of the winding. As a result, bipolar motors produce more torque than unipolar motors of the same size. The draw back of bipolar motors, compared to unipolar motors, is that more complex control circuitry is required by bipolar motors.

Bipolar Stepper Motor

FIGURE 3: BIPOLAR STEPPER MOTOR



Unipolar Stepper Motor

Unipolar Motors

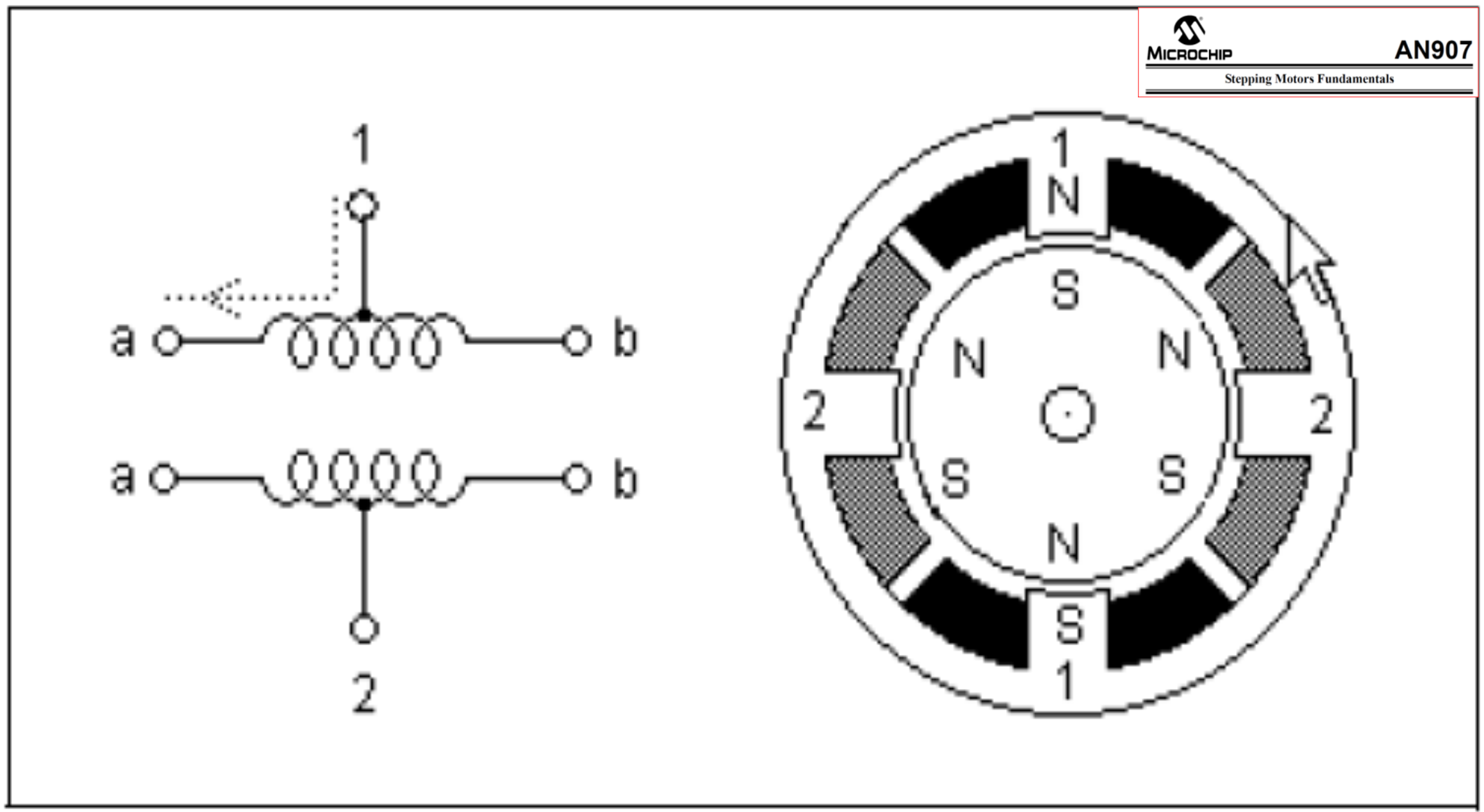
MICROCHIP AN907

Stepping Motors Fundamentals

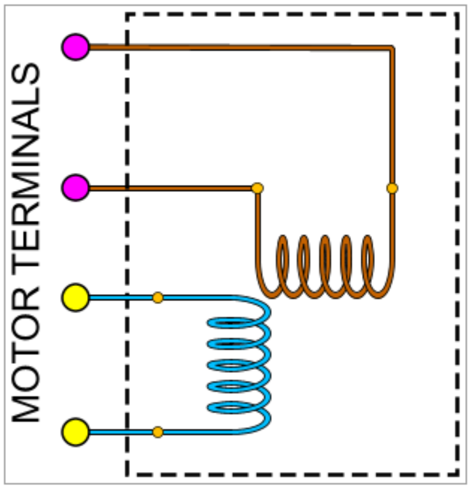
Unipolar stepping motors are composed of two windings, each with a center tap. The center taps are either brought outside the motor as two separate wires (as shown in Figure 2) or connected to each other internally and brought outside the motor as one wire. As a result, unipolar motors have 5 or 6 wires. Regardless of the number of wires, unipolar motors are driven in the same way. The center tap wire(s) is tied to a power supply and the ends of the coils are alternately grounded.

Unipolar Stepper Motor

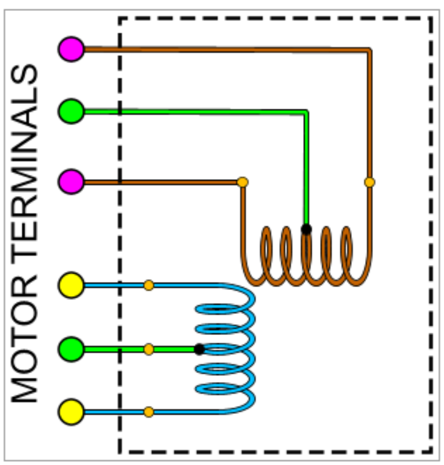
FIGURE 2: UNIPOLAR STEPPER MOTOR



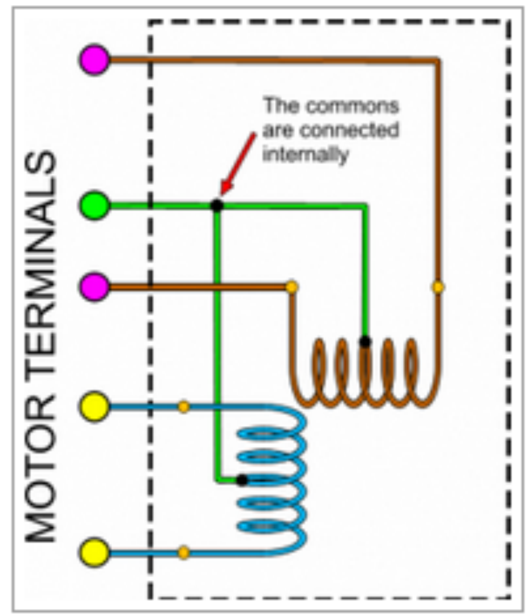
Stepper Motor Wiring



Bipolar
Stepper
Motor
4 Wires



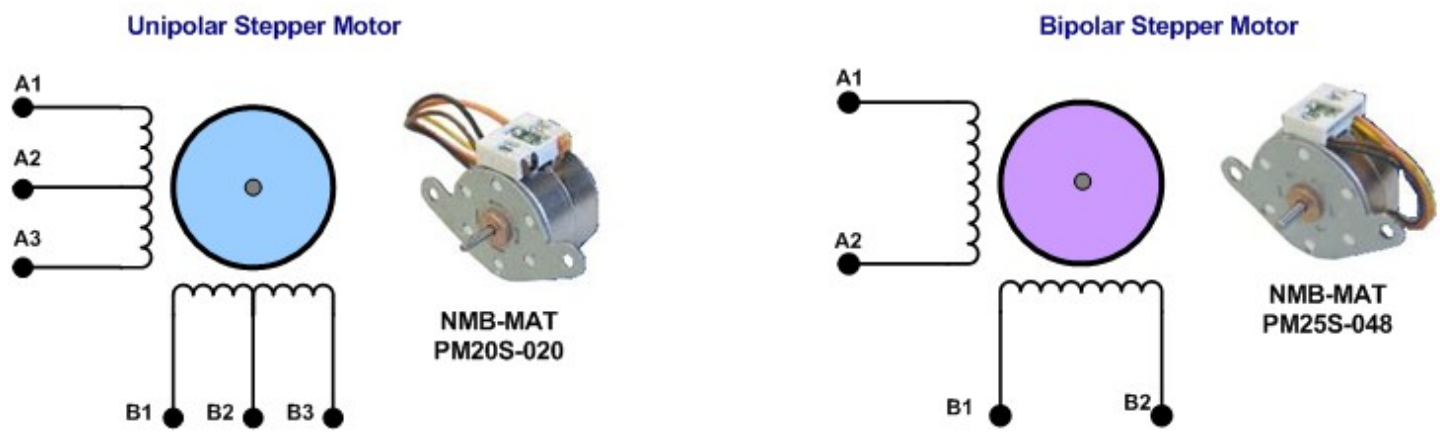
Unipolar
Stepper
Motor
6 Wires



A unipolar motor with 5 terminals

Unipolar
Stepper
Motor
5 Wires

Bipolar/Unipolar stepper motors.



The Unipolar and Bipolar Stepper Motor Windings

Unipolar stepper motors require a simpler drive circuit, bipolar stepper motors have higher torque capabilities.

Unipolar – current flow in only one direction

Bipolar – current flow in two directions

DESIGNSPARK

Stepper motors and drives, what is full step, half step and microstepping?

Full step and half step

Stepper drives control how a stepper motor operates, there are three commonly used excitation modes for stepper motors, full step, half step and microstepping. These excitation modes have an effect on both the running properties and torque the motor delivers.

A stepper motor converts electronic signals into mechanical movement each time an incoming pulse is applied to the motor. Each pulse moves the shaft in fixed increments. If the stepper motor has a 1.8° step resolution, then in order for shaft to rotate one complete revolution, in full step operation, the stepper motor would need to receive 200 pulses, 360° ÷ 1.8 = 200.

There are two types of full step excitation modes.

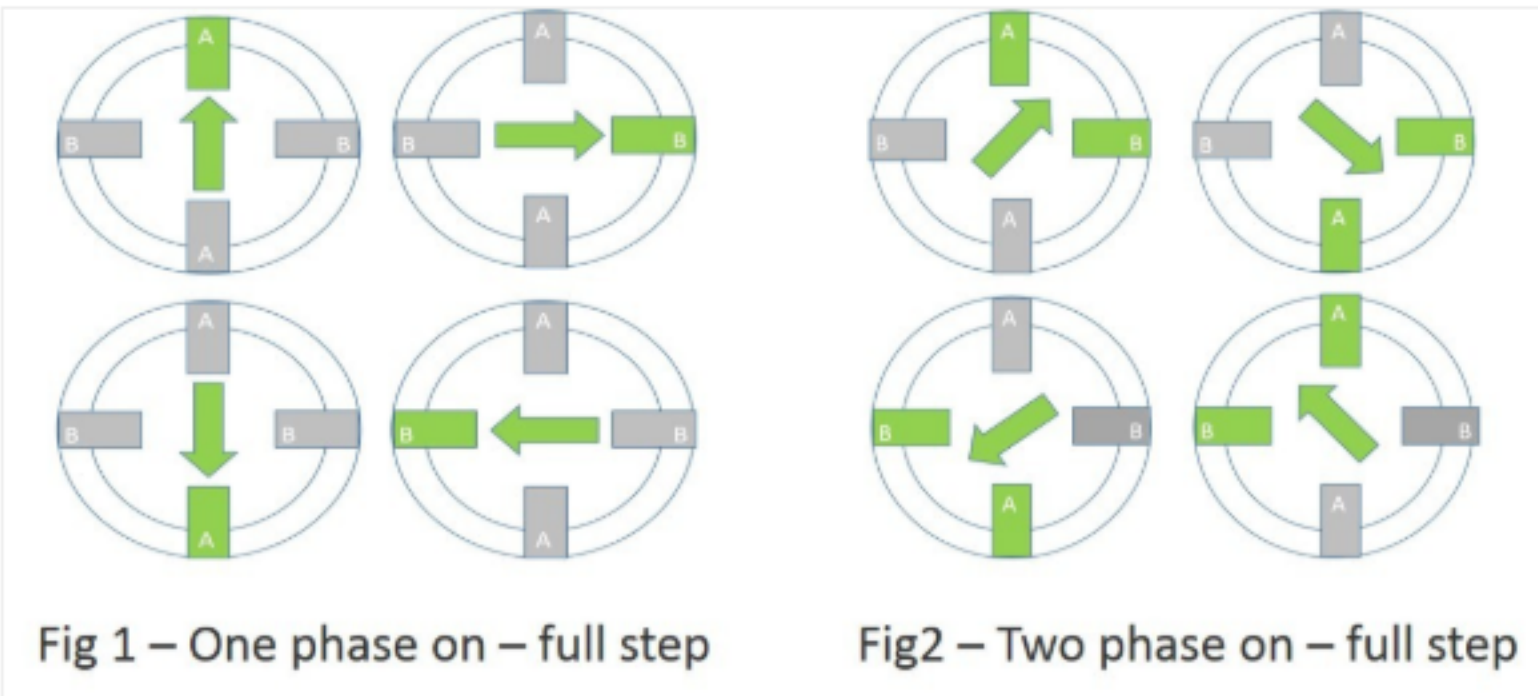
In one-phase on - full step, Fig1, the motor is operated with only one phase energized at a time. This mode requires the least amount of power from the driver of any of the excitation modes.

In two-phase on - full step, Fig2, the motor is operated with both phases energized at the same time. This mode provides improved torque and speed performance. Two-phase on provides about 30% to 40% more torque than one phase on, however it requires twice as much power from the driver.

In one-phase on - full step, Fig1, the motor is operated with only one phase energized at a time. This mode requires the least amount of power from the driver of any of the excitation modes.

In two-phase on - full step, Fig2, the motor is operated with both phases energized at the same time. This mode provides improved torque and speed performance. Two-phase on provides about 30% to 40% more torque than one phase on, however it requires twice as much power from the driver.

DESIGNSPARK



Half step produces about 15% less torque than two phase on - full step, however modified half stepping eliminates the torque decrease by increasing the current applied to the motor when a single phase is energized. See Fig3

DESIGNSPARK

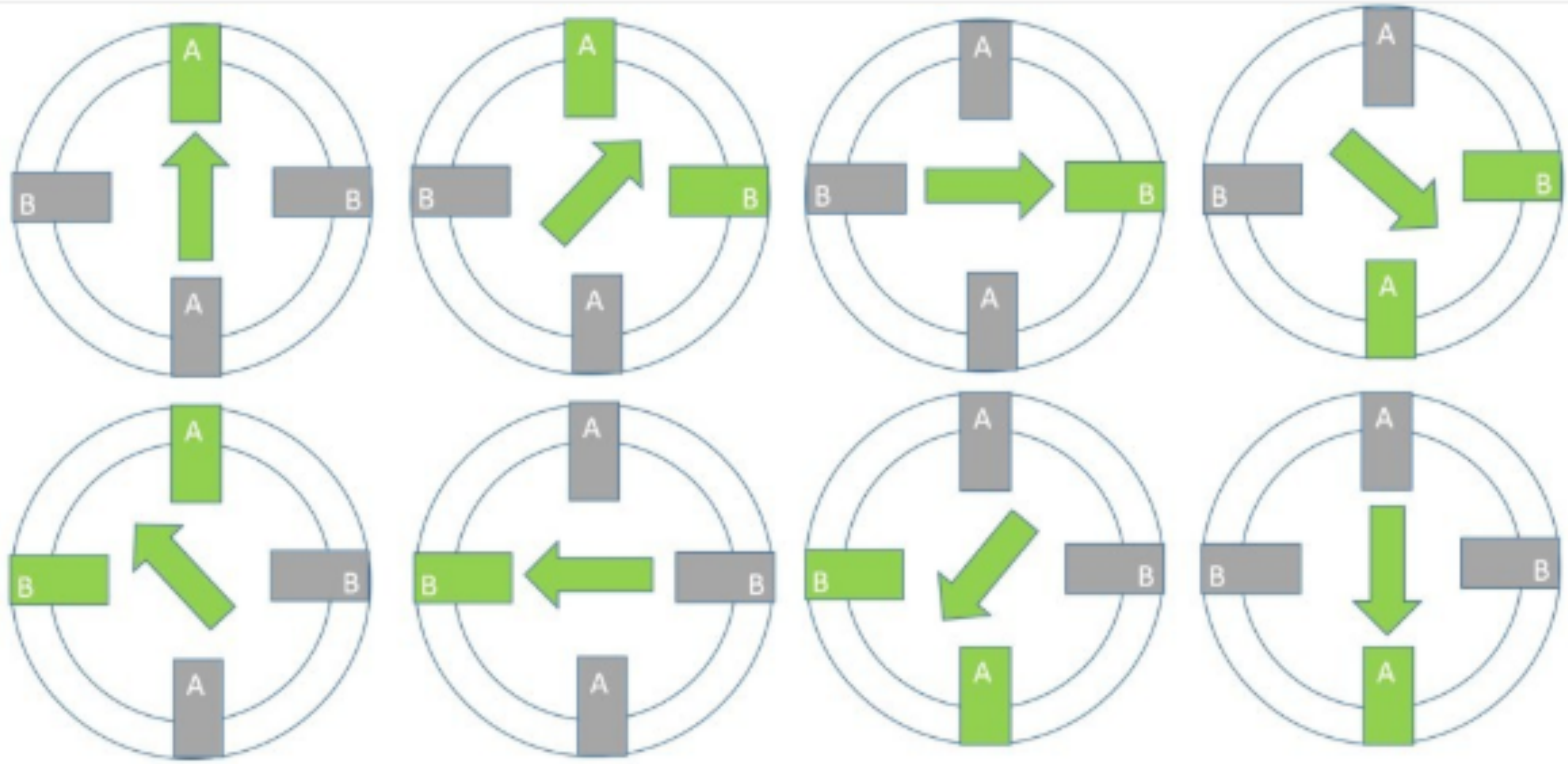


Fig3 - One-two phase on - half step

Microstepping for greater control and smoother operation

Microstepping can divide a motor's basic step by up to 256 times, making small steps smaller. A Micro drive uses two current sinewaves 90° apart, this is perfect for enabling smooth running of the motor. You will notice that the motor runs is quietly and with no real detectable stepping action.

By controlling direction and amplitude of the current flow in each winding, the resolution increases and the characteristics of the motor improve, giving less vibration and smoother operation. Because the sinewaves work together there is a smooth transition from one winding to the other. When current increases in one it decreases in the other resulting in a smooth step progression and maintained torque output. See Fig4

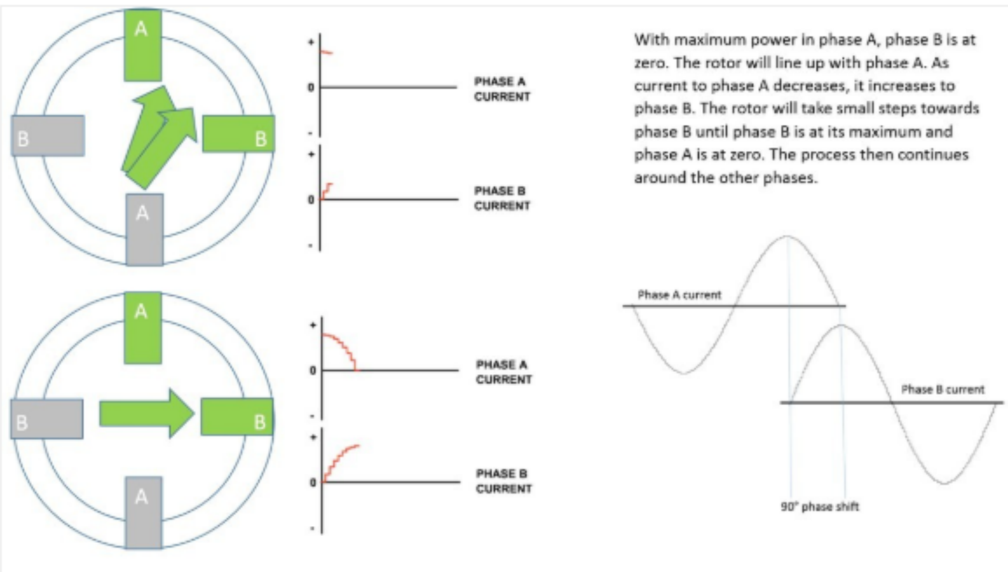
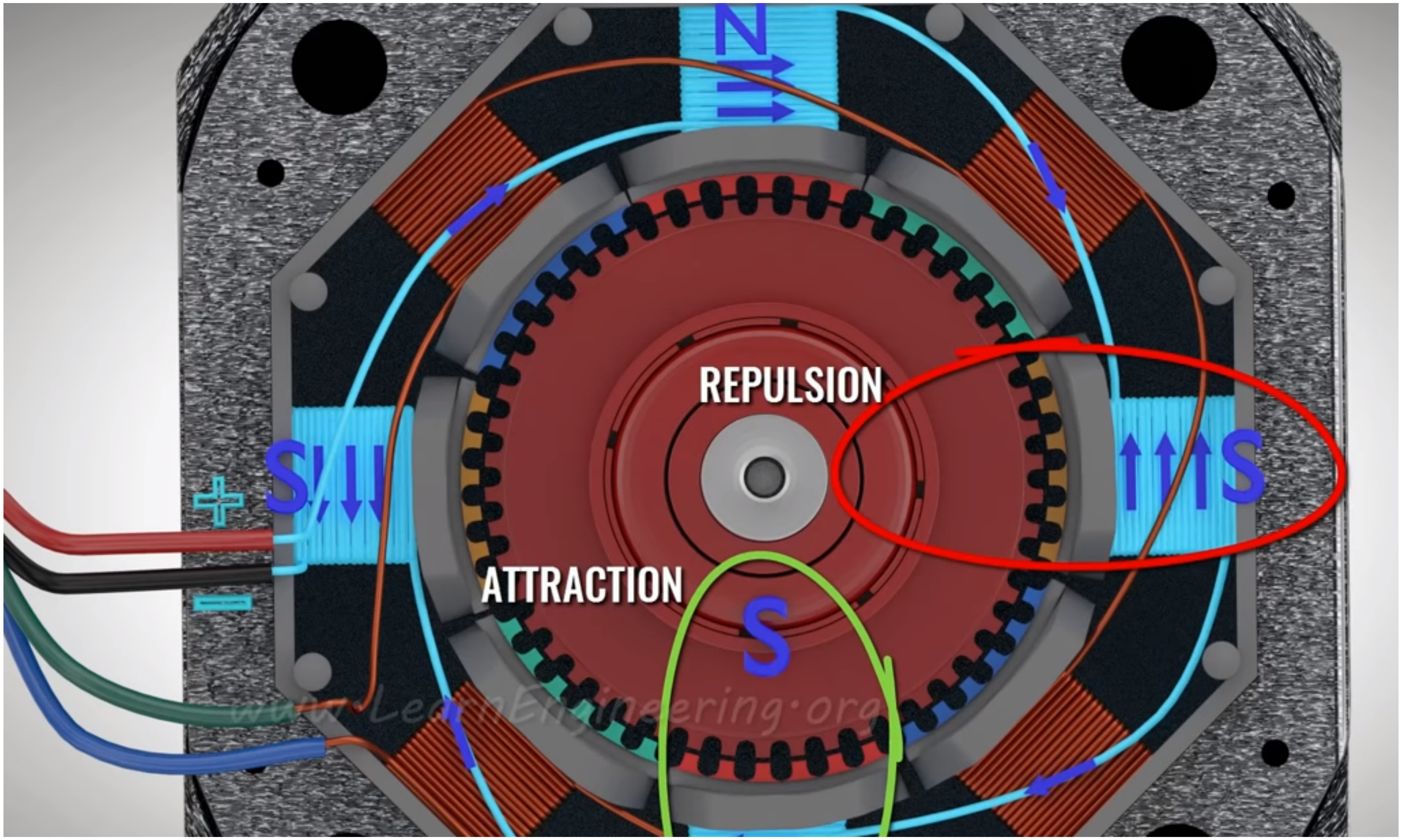


Fig 4 - Microstepping

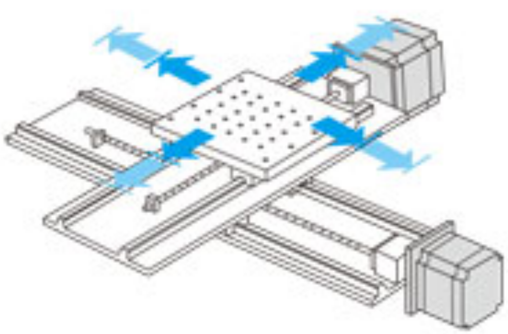


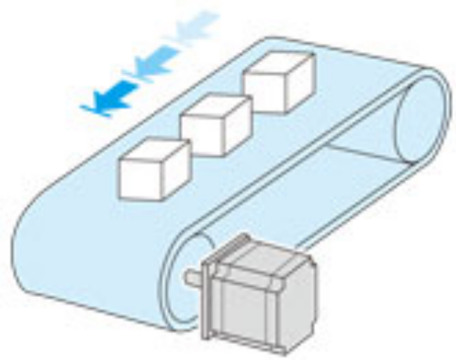
Two Coils, 48 teeth, rotor is magnetized with north and south poles.

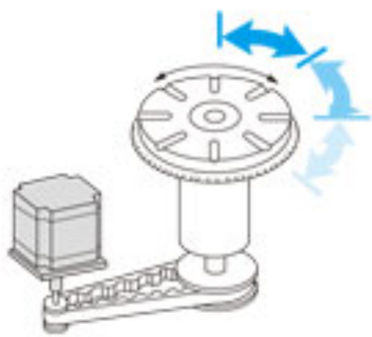
Stepper Motor Applications

Uses for Stepper Motors

While stepper motors can be used for a wide-variety of applications, they obviously excel in positioning systems (such as in the pick and place machines that precisely place the components on commercially produced PCBs). Some simple illustrations (courtesy Tamagawa Seiki Co.) of how stepper motors might be used in a commercial or industrial application can be seen below:







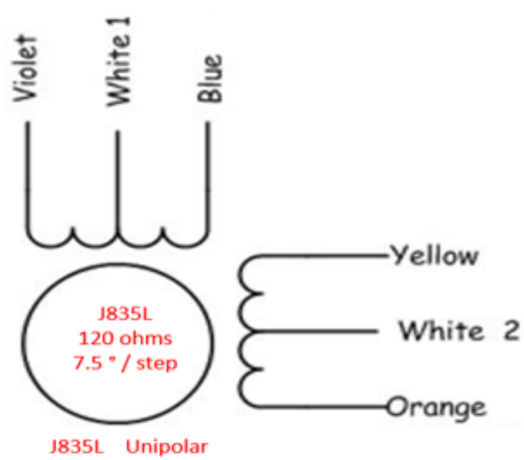
Stepper Motors – Unipolar Drive

Unipolar Full and Half Step Mode

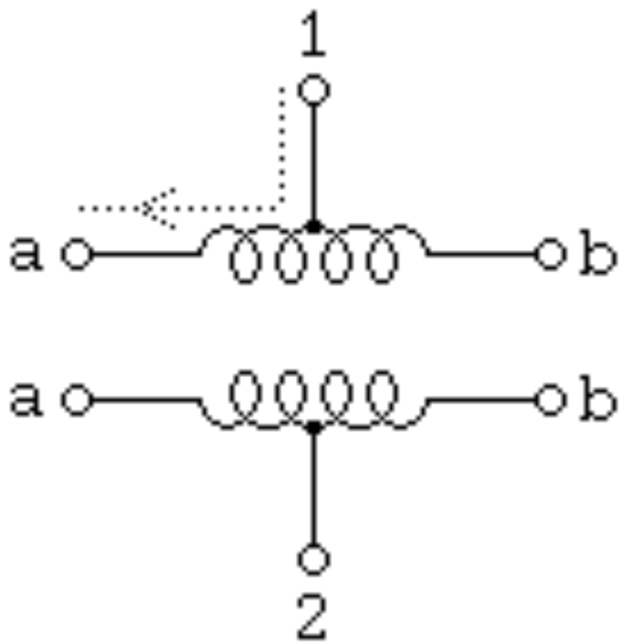
The Unipolar motor can be operated in full step or half step.

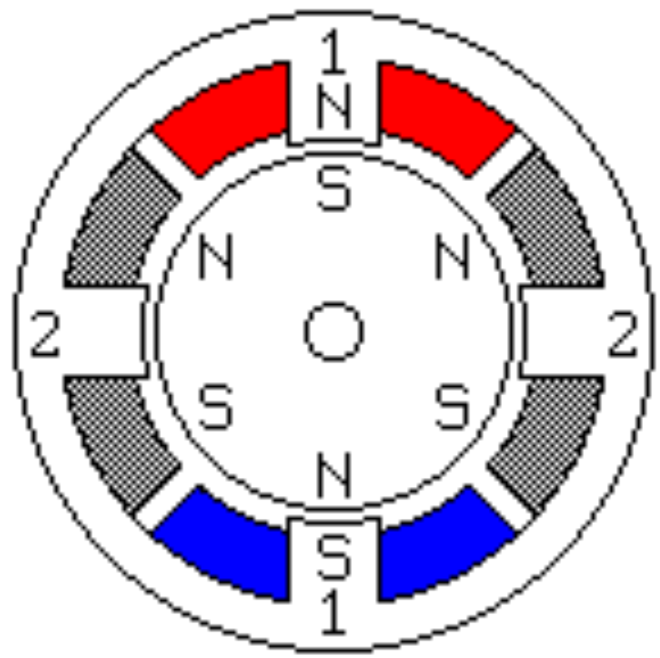
In full step mode one side of each of the two centre tapped coils is turned on.
The motor moves 7.5 degrees / step. There are 4 different discrete steps.

In half step mode two coils are energized and then one is energized in sequence. In this mode the step angle is half the angle as in full step mode. The drawback is that the half step mode has less torque. In half step mode there are 8 distinct steps.



Half Step Animation





Half Step Sequence:

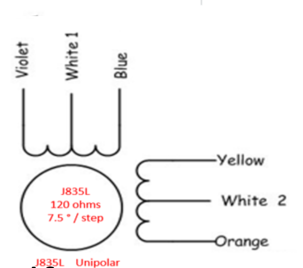
Animation of half step sequence. Centre tap pins 1 and 2 are connected to a positive voltage and the a and b pins are grounded.

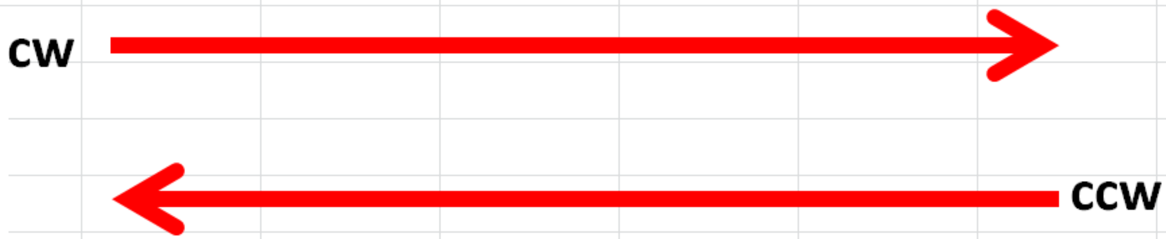
Half and Full Step Sequence Unipolar

Full step – always two coils turned on. Has better torque but less step resolution.

L1 Violet (1)	OFF	OFF	ON	ON	OFF	OFF	ON	ON
L2 Orange (2)	OFF	ON	ON	OFF	OFF	ON	ON	OFF
L3 Blue (1)	ON	ON	OFF	OFF	ON	ON	OFF	OFF
L4 Yellow (2)	ON	OFF	OFF	ON	ON	OFF	OFF	ON
	STEP 1	STEP 2	STEP 3	STEP 4	STEP 1	STEP 2	STEP 3	STEP 4

1 step sequence





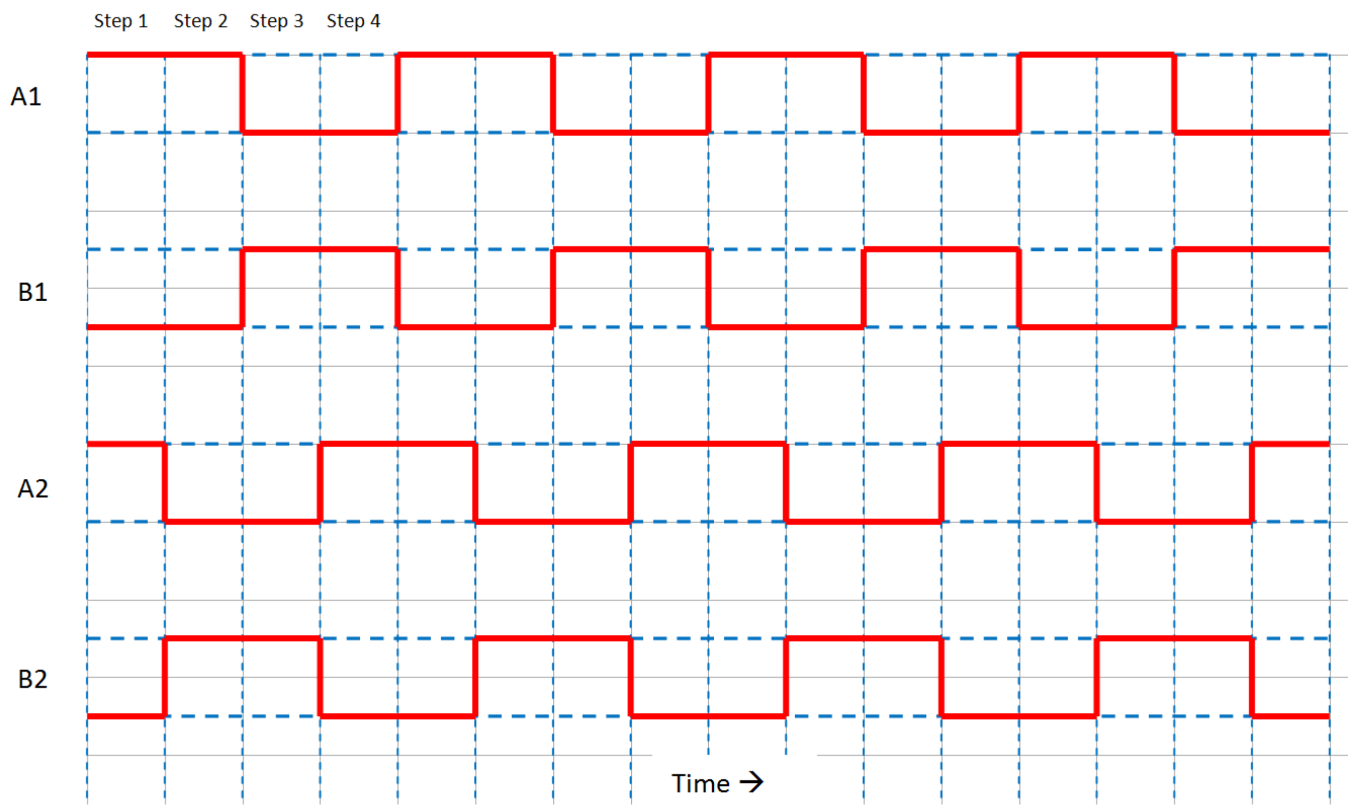
Half step – one or two coils turned on. Has reduced torque but greater step resolution.

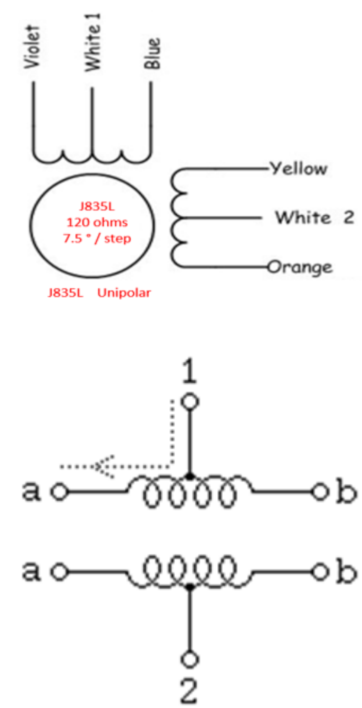
L1 Violet (1)	OFF	OFF	OFF	OFF	OFF	ON	ON	ON
L2 Orange (2)	OFF	OFF	OFF	ON	ON	ON	OFF	OFF
L3 Blue (1)	OFF	ON	ON	ON	OFF	OFF	OFF	OFF
L4 Yellow (2)	ON	ON	OFF	OFF	OFF	OFF	OFF	ON
	STEP 1	STEP 2	STEP 3	STEP 4	STEP 5	STEP 6	STEP 7	STEP 8

8 step sequence

Stepper Full Step Sequence Unipolar Motor

Full Step Sequence Timing Diagram:





Full Step Sequence:

With a 7.5 degrees/step motor this sequence will take 48 steps to complete one revolution(360 degrees). The full step sequence must be repeated 12 time for 1 revolution.

Stepper Motor

The half-step sequence is shown below.

STEP	L1	L2	L3	L4
1	H	L	L	L
2	H	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

Half step sequence has the highest resolution and is most stable at higher speeds.

Half step twice the resolution as full step. (degrees / step)

Unipolar Half Step Mode

Timing Example 1:

Determine the time required per step to rotate a stepper motor at 80 RPMs in half step sequence.

Step 1: With a 7.5 degrees/step motor this sequence will take 96 steps to complete one revolution.(360 degrees ÷ 3.75/step = 96).

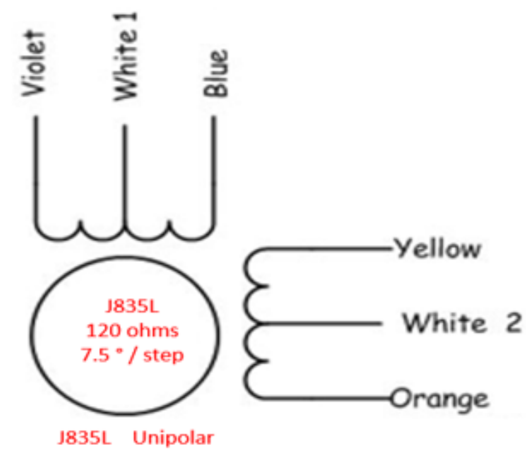
Step 2: Convert the RPMs to RPS(rotations per second) divide by 60.
80 RPM/60 = 1.333 RPS

Step 3: Determine the number of steps / rev. in half step mode. = 96

Step 4: Convert the RPS to time / rotation (1/RPS) 1/1.333 RPS = 0.75 sec/rev

Step 5: Divide the time/rev by the steps/rev 0.75 sec/rev ÷ 96 steps/rev = 7.81 ms

Time / step = 7.81 ms./step Result 96 steps x 7.81 ms/step = 0.75 seconds.



Unipolar Full/Half Step Sequence

Full / Half Step Sequence Timing another example:

Full 4 Step and Half 8 Step Pattern:

Full-stepping

Clockwise Rotation

Index	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	0
8	0	0	1	1

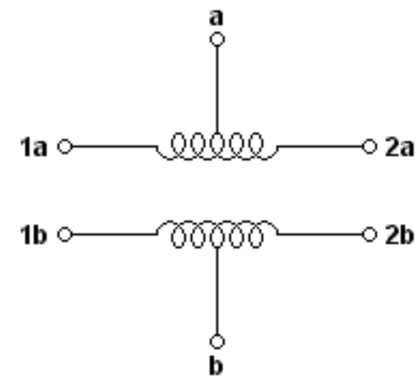
Alternate Full Step Sequence
(Provides more torque)

Half-stepping

Clockwise Rotation

Index	1a	1b	2a	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	1	0	0	0
10	1	1	0	0
11	0	1	0	0
12	0	1	1	0
13	0	0	1	0
14	0	0	1	1
15	0	0	0	1
16	1	0	0	1

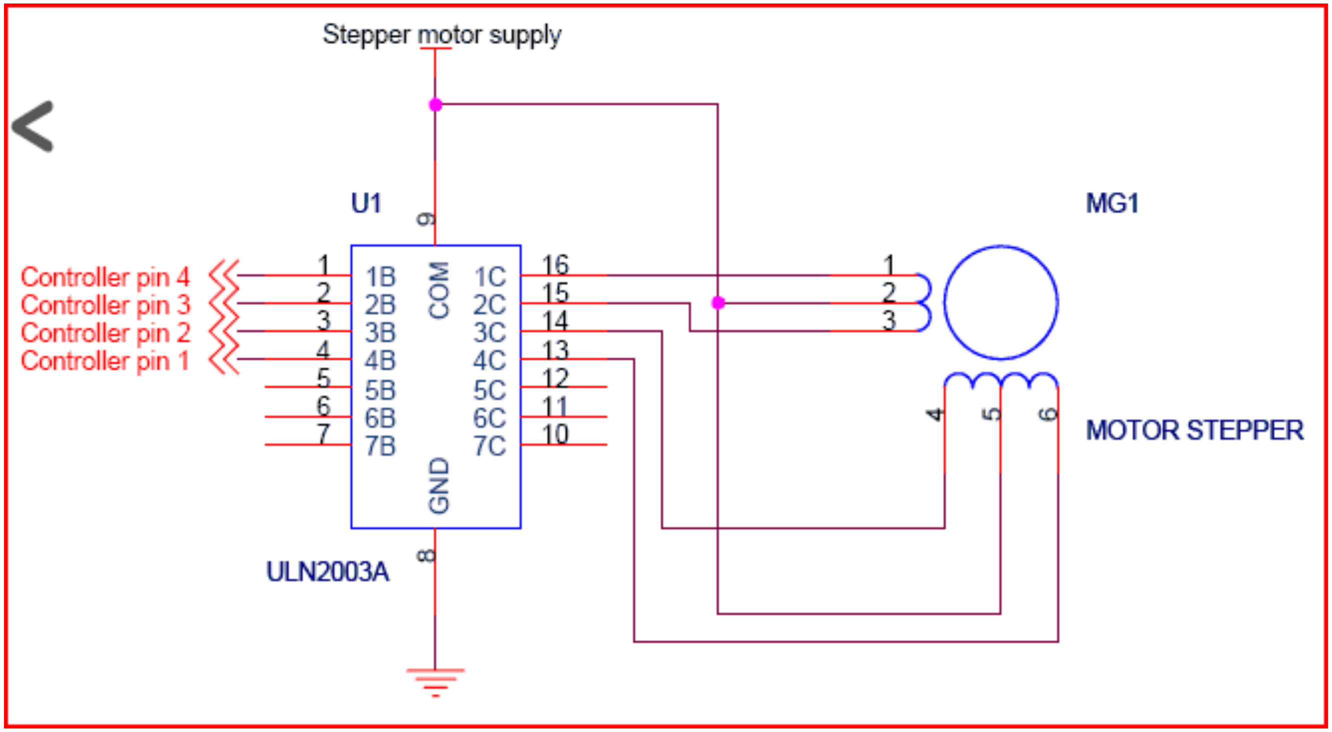
Half Step Sequence

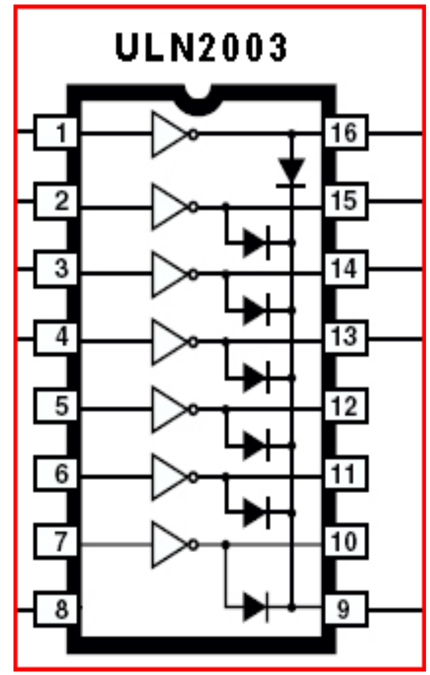


In full step mode coils are on for 2 steps then off for 2 steps, this mode produces more torque.

In half step mode a coil is on for three steps then off for five. This mode has twice the resolution for a given stepper motor.

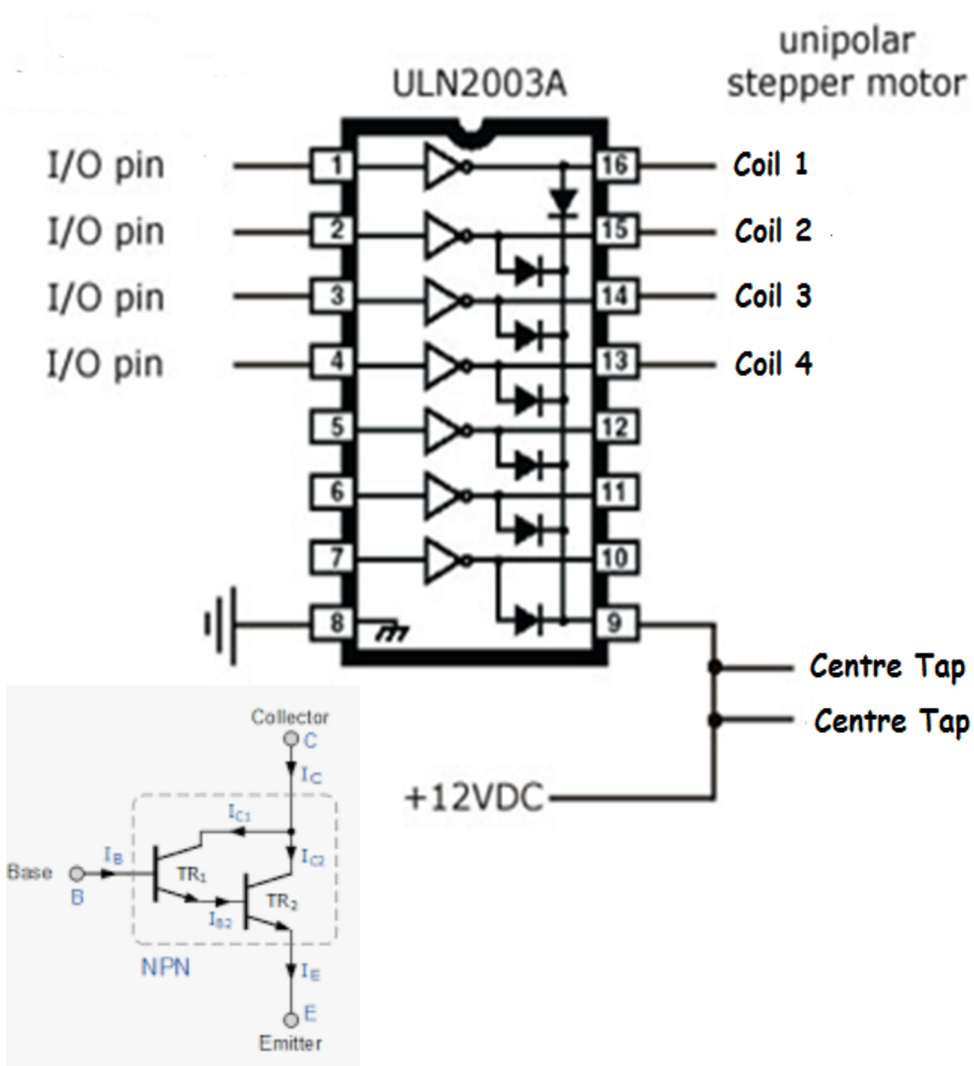
Unipolar Stepper Motor Driver





The ULN2003A is a driver IC. The input is a low voltage, low current digital signal. The IC is capable of driving output loads of higher currents and voltages.

Unipolar Stepper Motor Driver



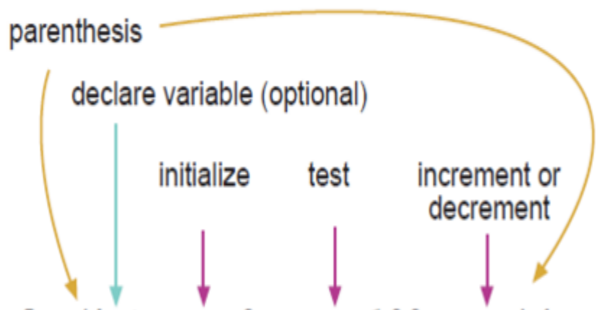
ULN 2003A Driver IC

This IC drives inductive loads such as DC motors, solenoids and relays.

The output of the circuit includes suppression diodes to prevent damage to the electronics. The device that appears as an inverter is actually a Darlington Pair transistor drive circuit. When the input is a logic high the output is driven to 0 volts. The Darlington Pair transistor has a much greater Beta than a single NPN transistor.

The input is typically less than 1 mA. Output can handle at least 100 mA per channel.

For Loop – Used in Stepper Program

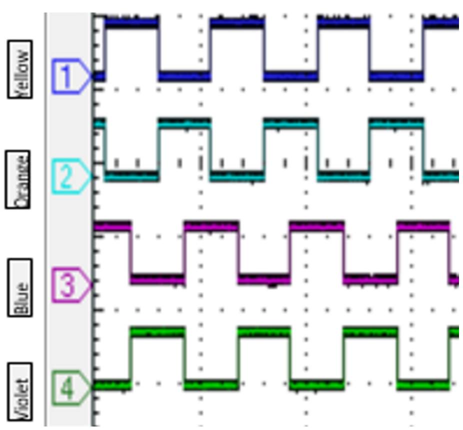



```
for(int x = 0; x < 100; x++){
   println(x);  // prints 0 to 99
}
```

The **initialization** happens first and exactly once. Each time through the loop, the **condition** is tested; if it's true, the statement block, and the **increment** is executed, then the **condition** is tested again. When the **condition** becomes false, the loop ends.

Arduino Control of a Unipolar Motor in Full Step

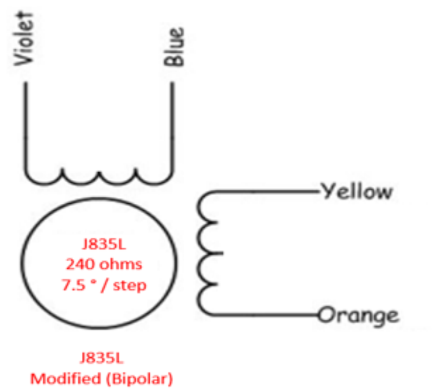
```
/* Stepper Control Unipolar Full Step */

int orange = 8;
int blue = 9;
int yellow = 10;
int violet = 11;

int delayTime = 20;   // milliseconds

void setup() {
  pinMode(orange, OUTPUT);
  pinMode(blue, OUTPUT);
  pinMode(yellow, OUTPUT);
  pinMode(violet, OUTPUT);
}
```

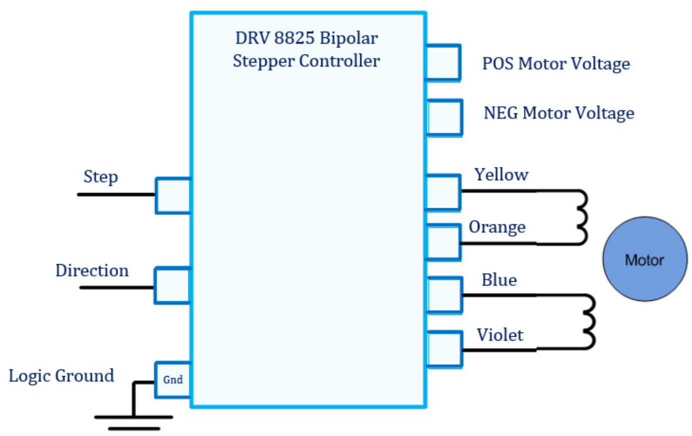



```
 for (unsigned int x = 0; x < 12; x++)

{
  digitalWrite(orange, HIGH);
  digitalWrite(blue, LOW);
  digitalWrite(yellow,LOW);
  digitalWrite(violet, HIGH);
  delay(delayTime);

  digitalWrite(orange, LOW);
  digitalWrite(blue, LOW);
  digitalWrite(yellow, HIGH);
  digitalWrite(violet, HIGH);
  delay(delayTime);

  digitalWrite(orange, LOW);
  digitalWrite(blue, HIGH);
  digitalWrite(yellow, HIGH);
  digitalWrite(violet, LOW);
  delay(delayTime);

  digitalWrite(orange, HIGH);
  digitalWrite(blue, HIGH);
  digitalWrite(yellow, LOW);
  digitalWrite(violet, LOW);
  delay(delayTime);

}
delay(1000);
}
```

Stepper Motor – Bipolar Drive

Bipolar Stepper Motor





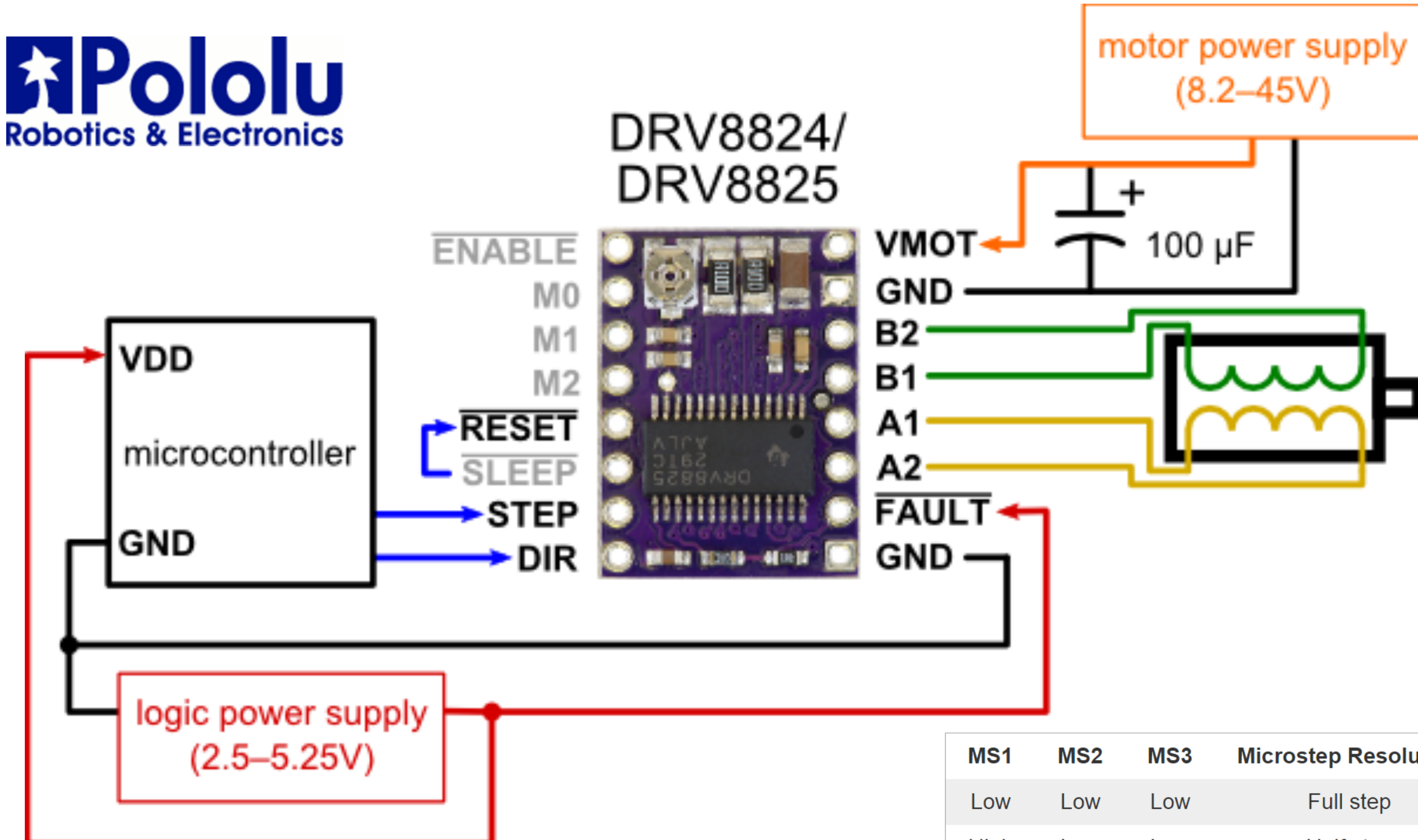
The Bipolar stepper motor has only 4 wires.

Since the full coil is energized it has a higher torque then the unipolar motor.

The J835L motor used in the lab was modified to operate as a bipolar so that it can be controlled with a bipolar motor controller module. The white wire was cut on both coils.

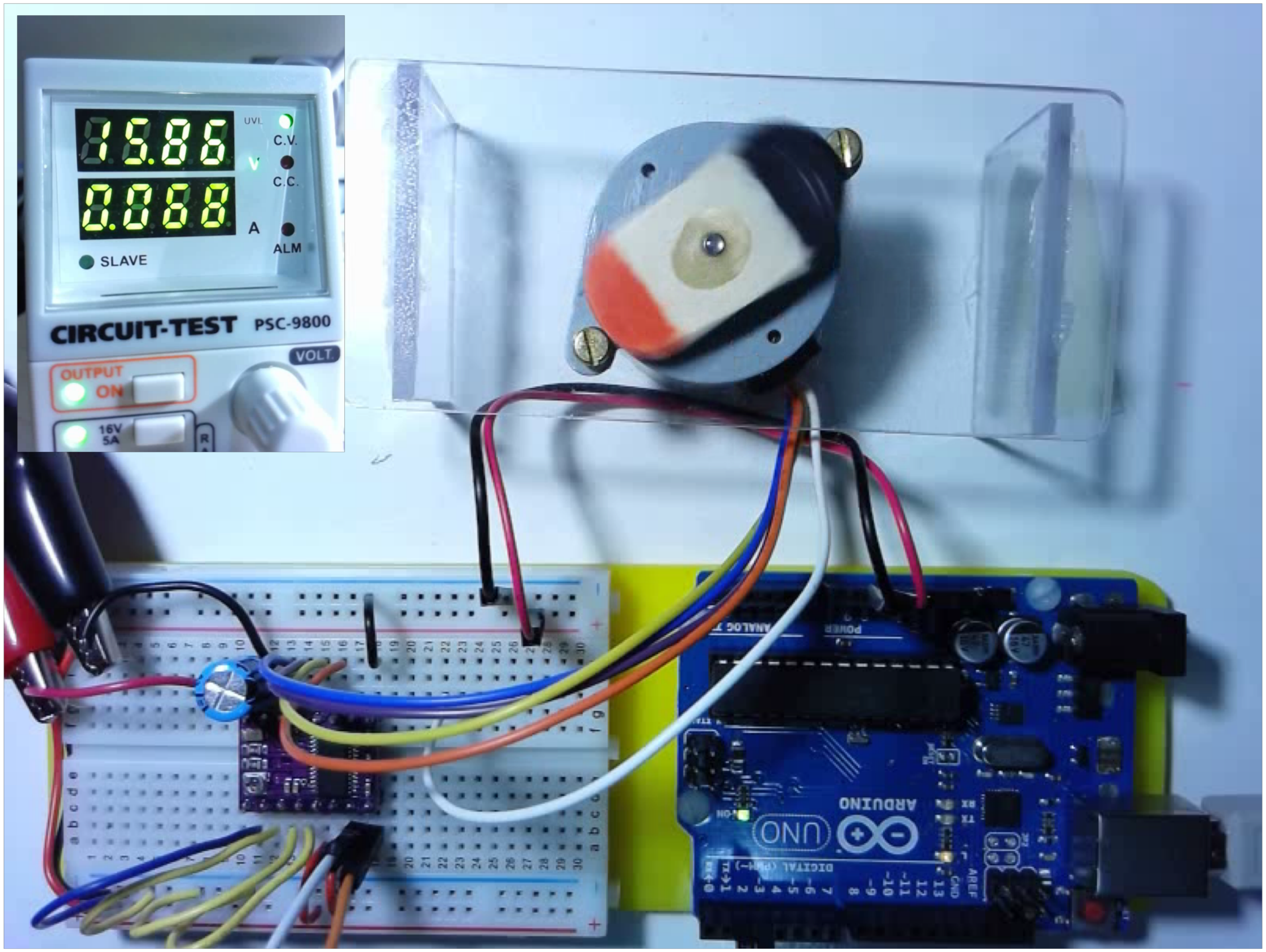
The DRV8825 is a bipolar stepper motor driver. The driver requires only a step and direction control signal.

Bipolar DC Stepper Motor Controller

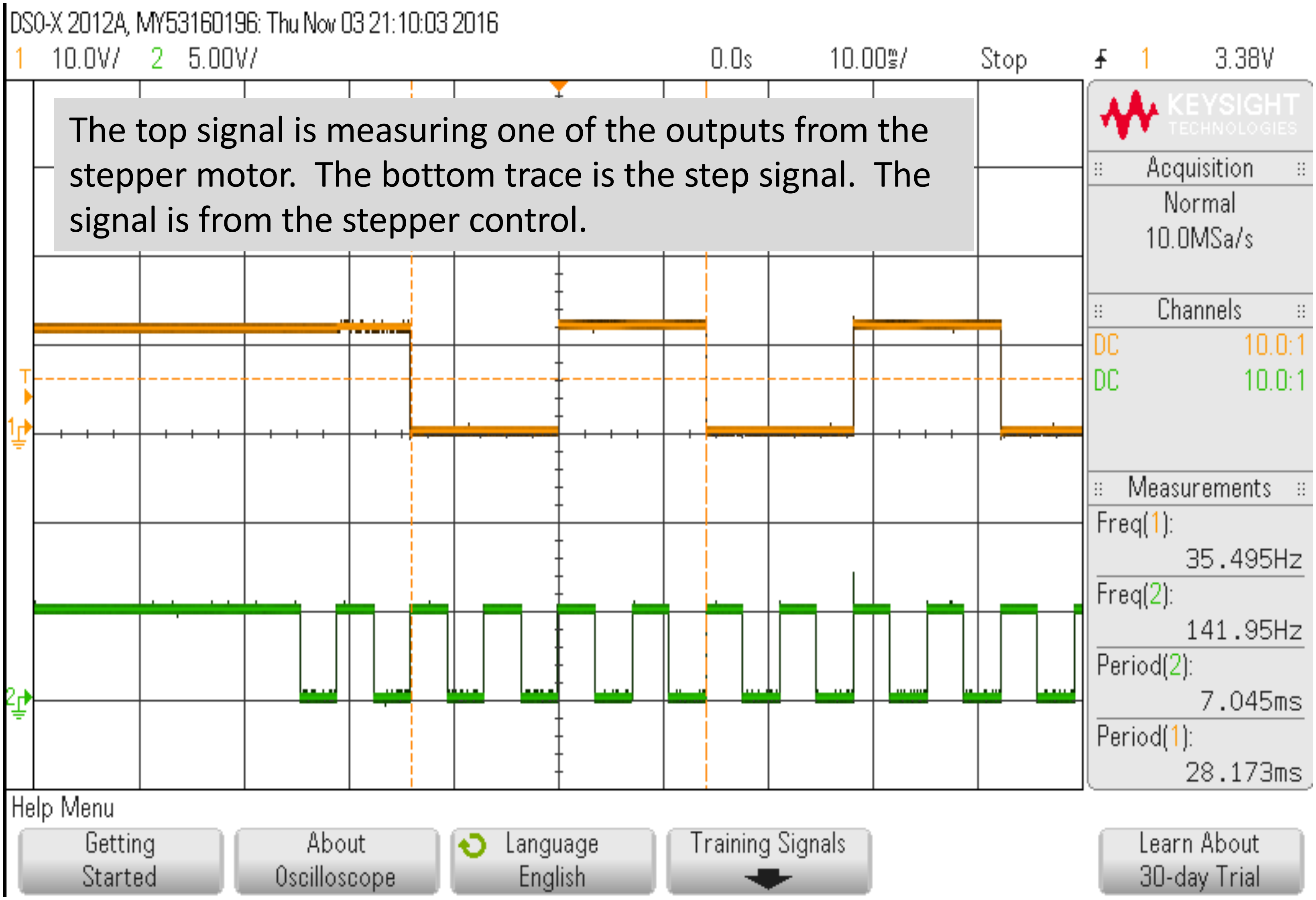


MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

Bipolar DC Stepper Motor Controller



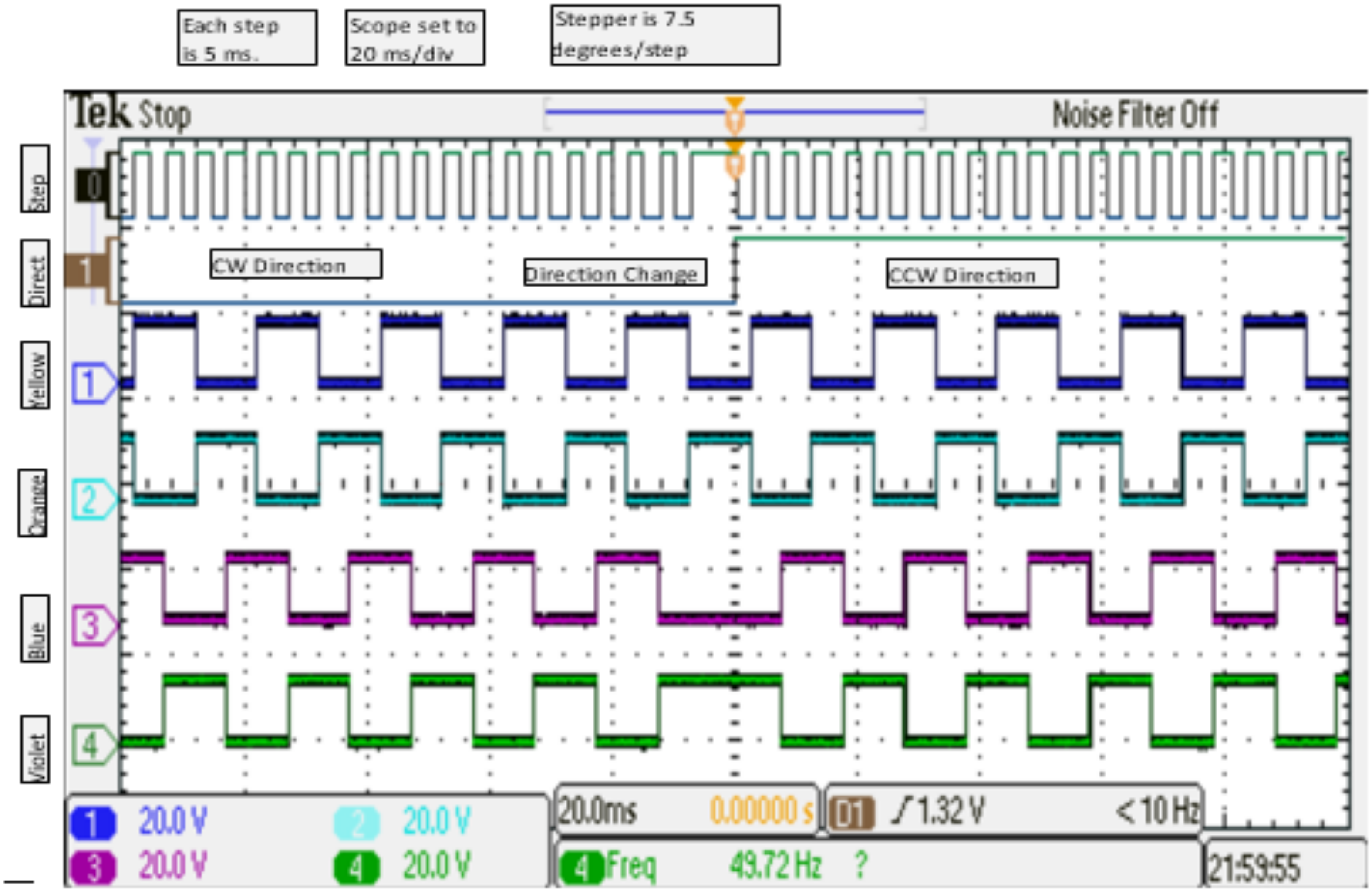
The top signal is measuring one of the outputs from the stepper motor. The bottom trace is the step signal. The signal is from the stepper control.



Stepper_FWD_REV_DRV8825_F2016

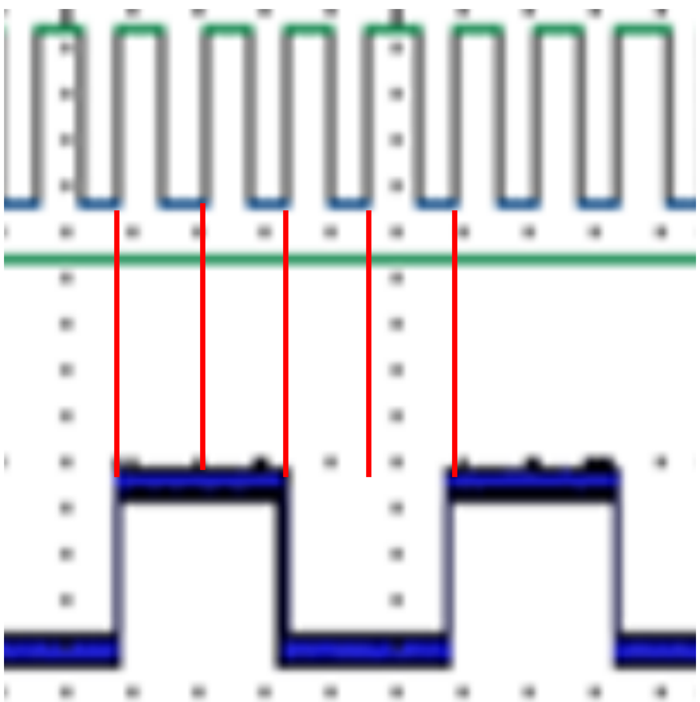
```
/* Stepper Motor Control using a
 *  POlolu DRV8825 Controller
 *  October 31st 2016
 *  Michel Hanbury from CAM8302E
 */
void setup() {
  pinMode(2, OUTPUT);   // step pulse
  pinMode(3, OUTPUT);   // direction
 }

void loop()
{
   digitalWrite(3, LOW); //direction

for (int x=0; x < 200; x++)
{
   digitalWrite(2, LOW);
   delayMicroseconds(400);
   delay(2);

   digitalWrite(2, HIGH);
   delayMicroseconds(400);
   delay(2);
}
delay(1000);


   digitalWrite(3, LOW); //direction

for (int x=0; x < 200; x++)
{
   digitalWrite(2, LOW);
   delayMicroseconds(400);
   delay(2);

   digitalWrite(2, HIGH);
   delayMicroseconds(400);
   delay(2);
}
delay(1000);
}
```

Stepper Half Step Sequence Unipolar Stepper



Bipolar Stepper motor waveforms full step.

Stepper CW and CCW Direction

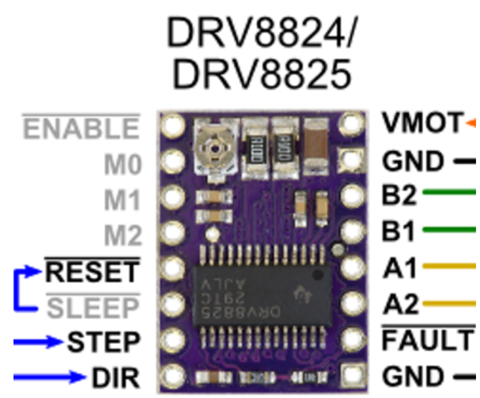


4 steps * 7.5 deg./step = 30 degrees.

Controller clock signal
compared to coil signal.

The output changes on the rising edge of the clock.

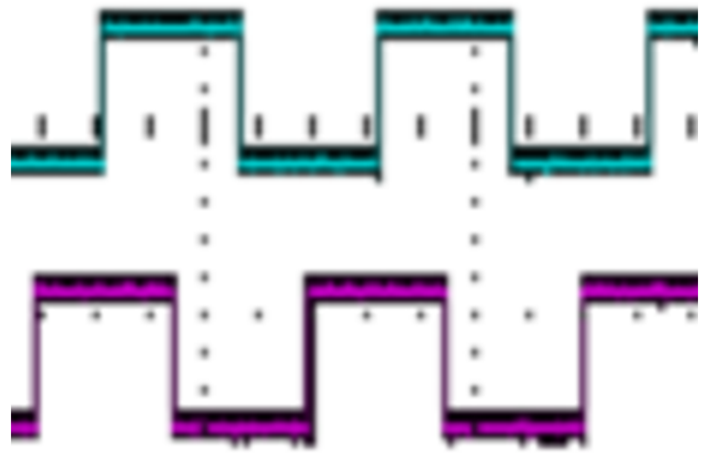
There are 4 clock periods for 4 steps. One high and one low on the coil represents four steps.



Stepper CW and CCW Direction

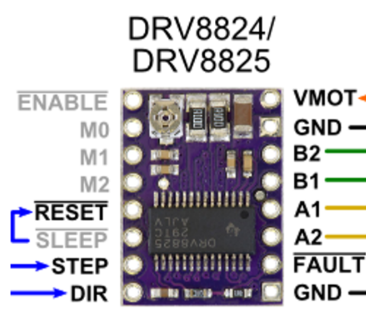
CW direction: (direction = 0)

Lower signal rising edge leads the rising edge of the upper signal by 90 degrees.



CCW direction: (direction = 1)

Top signal rising edge leads the rising edge of the lower signal by 90 degrees.



Interrupts vs Polling

```
reading = digitalRead(inPin);


if (reading == HIGH)
{
digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  }
else
{
 digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
 }
```

The section of code above uses input polling to test the state of a mechanical switch. If the switch is a logic high an LED is turned on otherwise it is turned off. This method is not efficient and most of the loop time is consumed in testing the switch. If the loop if long then if may take time before the switch is tested and an event may be missed.

Interrupts Setup

```
21 void setup() {
22
23   attachInterrupt(0, slow, RISING); // on pin 2
24   attachInterrupt(1, fast, RISING); // on pin 3
25
```

```
101 void fast(){                    // Interrupt service routine
102   delayTime = 10;
103 }
104
105 void slow(){                    // Interrupt service routine
106   delayTime = 20;
107 }
```

Lines 21-24 configure the interrupt. The program is telling the Arduino that when a rising edge occurs on pin 2 (interrupt 0) then stop what the program is currently doing, execute the "slow" routine then return to where it was before the interrupt.

Interrupts vs Polling

Reference Language | Libraries | Comparison | Changes

https://www.arduino.cc/en/Reference/AttachInterrupt

Buy

attachInterrupt()

Description

Digital Pins With Interrupts

The first parameter to attachInterrupt is an interrupt number. Normally you should use digitalPinToInterrupt(pin) to translate the actual digital pin to the specific interrupt number. For example, if you connect to pin 3, use digitalPinToInterrupt(3) as the first parameter to attachInterrupt.

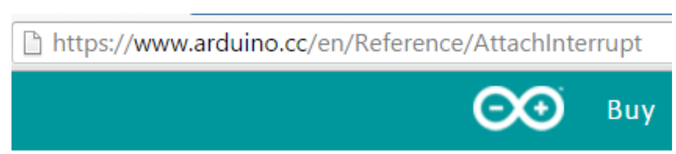
Board	Digital Pins Usable For Interrupts
Uno, Nano, Mini, other 328-based	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21

Using Interrupts

Using Interrupts

Interrupts are useful for making things happen automatically in microcontroller programs, and can help solve timing problems. Good tasks for using an interrupt may include reading a rotary encoder, or monitoring user input.

If you wanted to insure that a program always caught the pulses from a rotary encoder, so that it never misses a pulse, it would make it very tricky to write a program to do anything else, because the program would need to constantly poll the sensor lines for the encoder, in order to catch pulses when they occurred. Other sensors have a similar interface dynamic too, such as trying to read a sound sensor that is trying to catch a click, or an infrared slot sensor (photo-interrupter) trying to catch a coin drop. In all of these situations, using an interrupt can free the microcontroller to get some other work done while not missing the input.



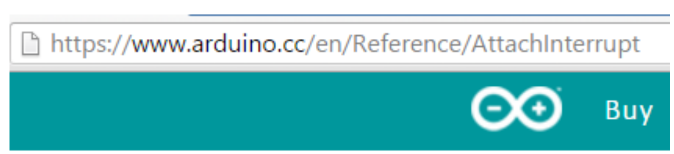
Interrupts Syntax

Syntax

attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);	(*recommended*)
attachInterrupt(interrupt, ISR, mode);	(*not recommended*)
attachInterrupt(pin, ISR, mode) ;	(*not recommended Arduino Due, Zero only*)

Parameters

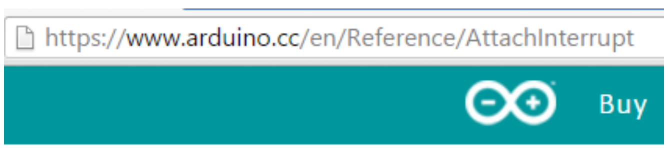
interrupt:	the number of the interrupt (*int*)	
pin:	the pin number	(*Arduino Due, Zero only*)
ISR:	the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an *interrupt service routine.*	



Interrupts Modes

mode: defines when the interrupt should be triggered. Four contstants are predefined as valid values:

- **LOW** to trigger the interrupt whenever the pin is low,
- **CHANGE** to trigger the interrupt whenever the pin changes value
- **RISING** to trigger when the pin goes from low to high,
- **FALLING** for when the pin goes from high to low.

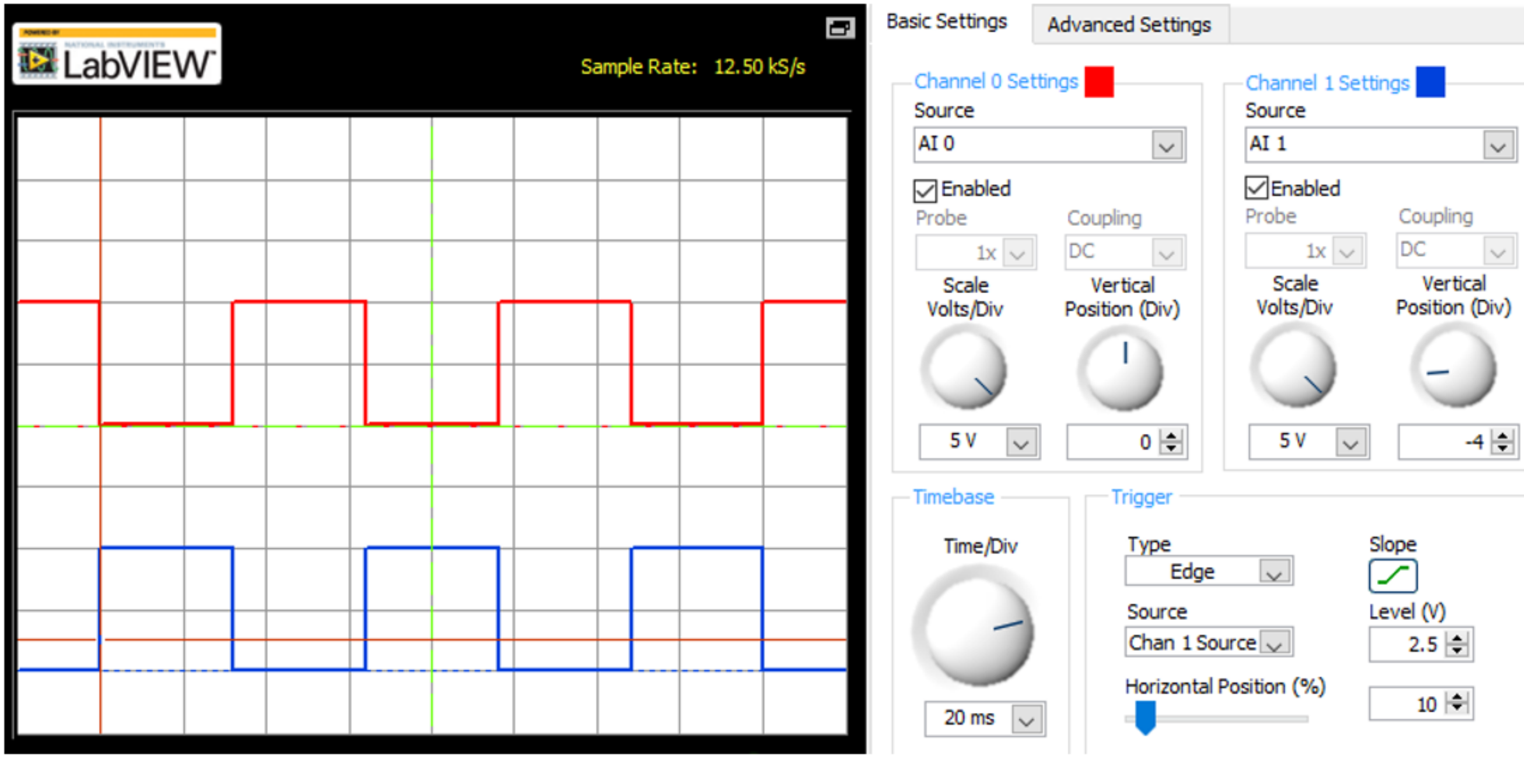


Interrupt Program for Stepper P1

```
#define LED 13

// Volatile required for interrupt variable    //
// Interrupt inputs must be debounced.          //

volatile bool direction_1 = LOW;
volatile int freq = 100;
volatile int x;

void setup() {

  pinMode(LED, OUTPUT);
  attachInterrupt(0, toggle_s, FALLING);  // pin 2 interrupt
  attachInterrupt(1, toggle_d, FALLING);  // pin 3 interrupt

//  Serial.begin(9600);  //turn on serial communication
}
void loop() {
 tone(4,freq); // 31 to 65,535 Hertz
 digitalWrite(5,direction_1);
}
```

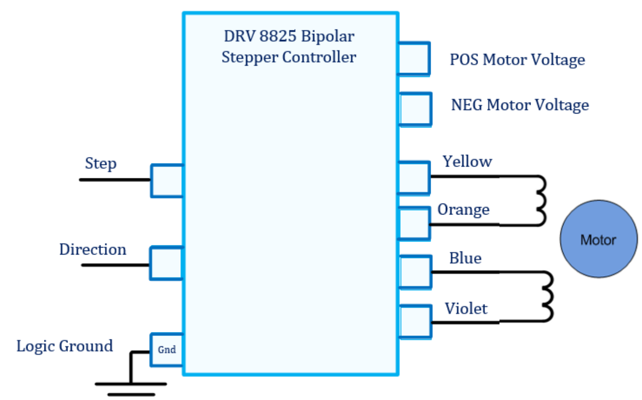
Interrupt Program for Stepper P2

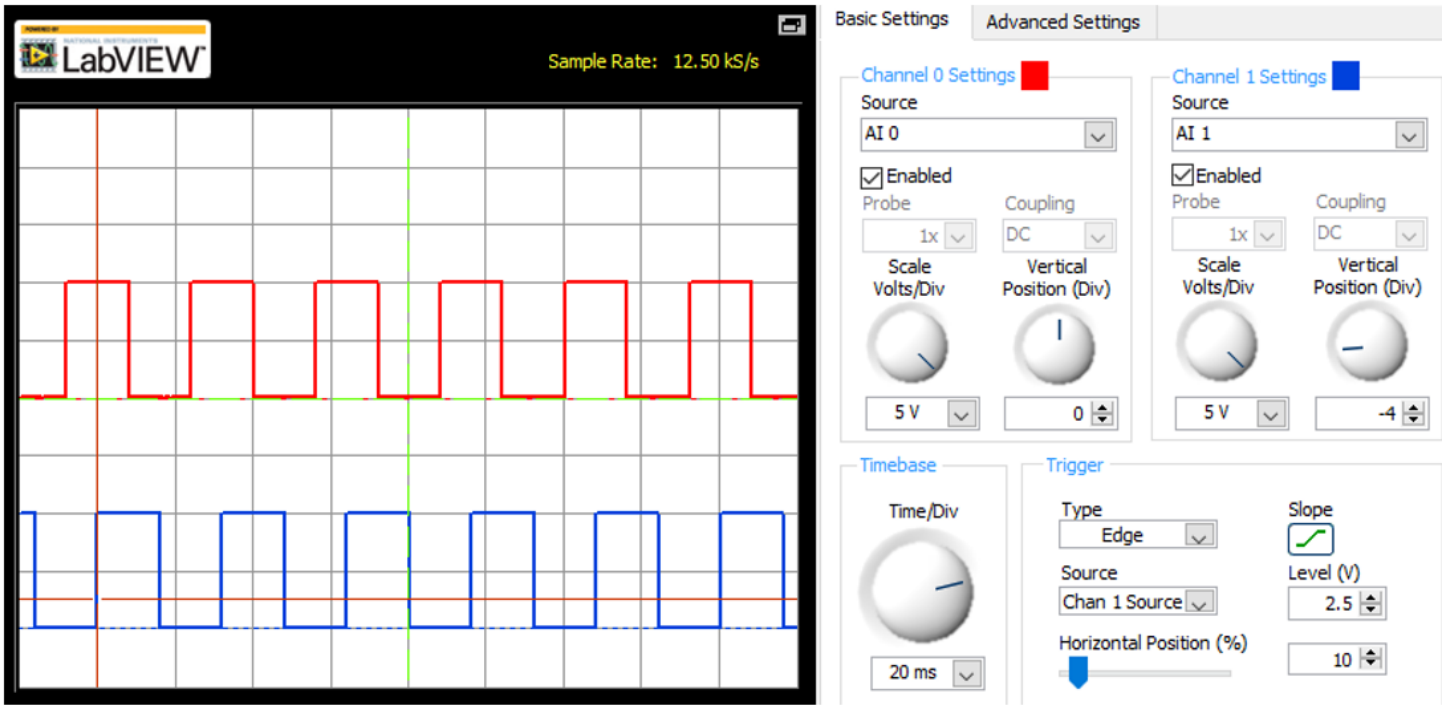
```
void loop() {
 tone(4,freq); // 31 to 65,535 Hertz
 digitalWrite(5,direction_1);
}

void toggle_s() {   // pin 2 interrupt 0 toggle speed
   digitalWrite(LED, HIGH);

   if (freq == 125)
     freq = 100;      // change speed if interrupted
   else if (freq == 100)
     freq = 125;
    x++; // inx x
}
void toggle_d() {   // pin 3 interrupt 1  toggle direction

   detachInterrupt(1);
   digitalWrite(LED, LOW);

    if (direction_1 == HIGH)
      direction_1 = LOW;  // change direction if interrupted
   else if (direction_1 == LOW)
      direction_1 = HIGH;
    x--; // dec x
    attachInterrupt(1, toggle_d, FALLING);
}
```



The two traces above are measuring the signal on the driver outputs connected to the yellow wire and the orange wire.

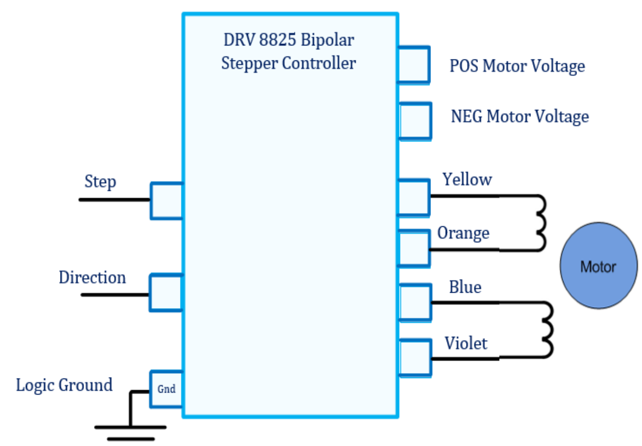
The direction of current flow in the coil changes on every two steps.



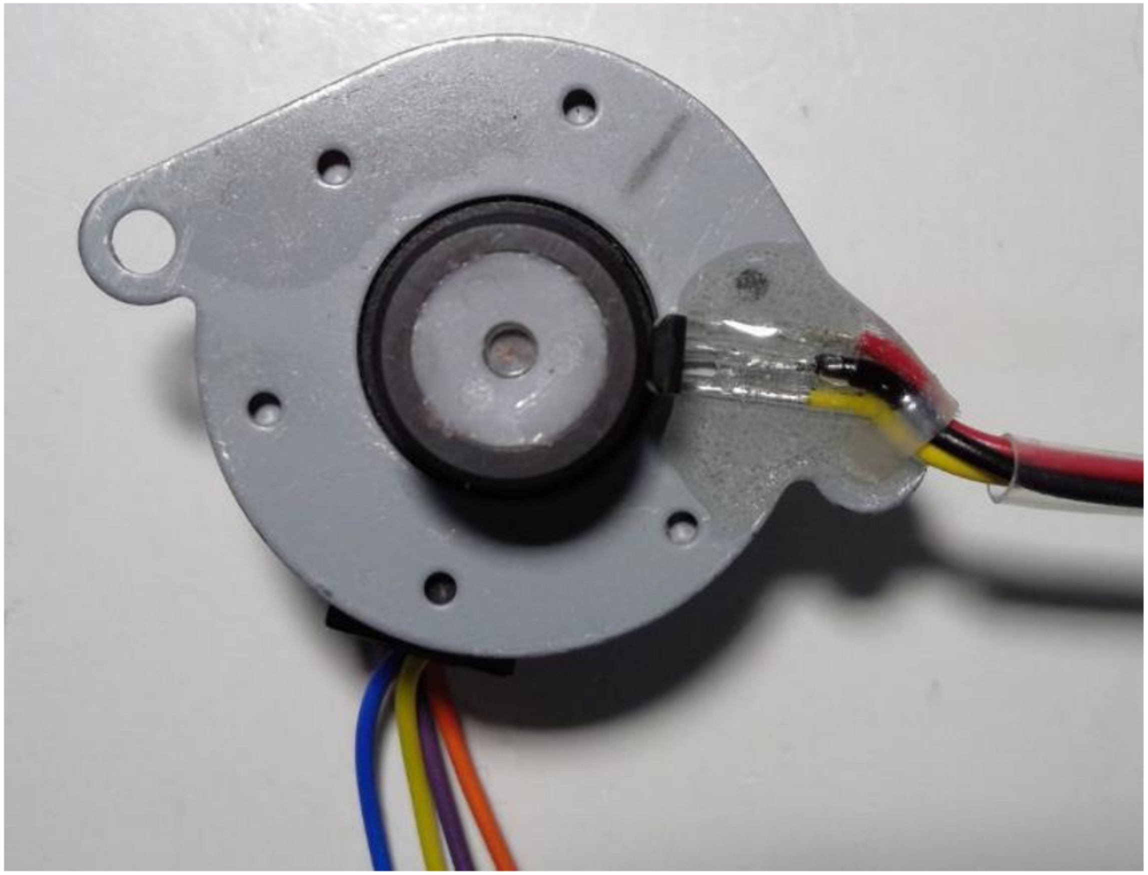


The two traces above are measuring the signal on the driver outputs connected to the yellow wire and the blue wire.

The two signals are 90 degrees out of phase. When the motor is turning in one direction the signal is leading by 90 degrees, in the opposite direction the signal is lagging by 90 degrees to the other coil.



Modified Stepper motor with Hall Effect Sensor and magnet.



240 ohms / coils

7.5 degrees / step

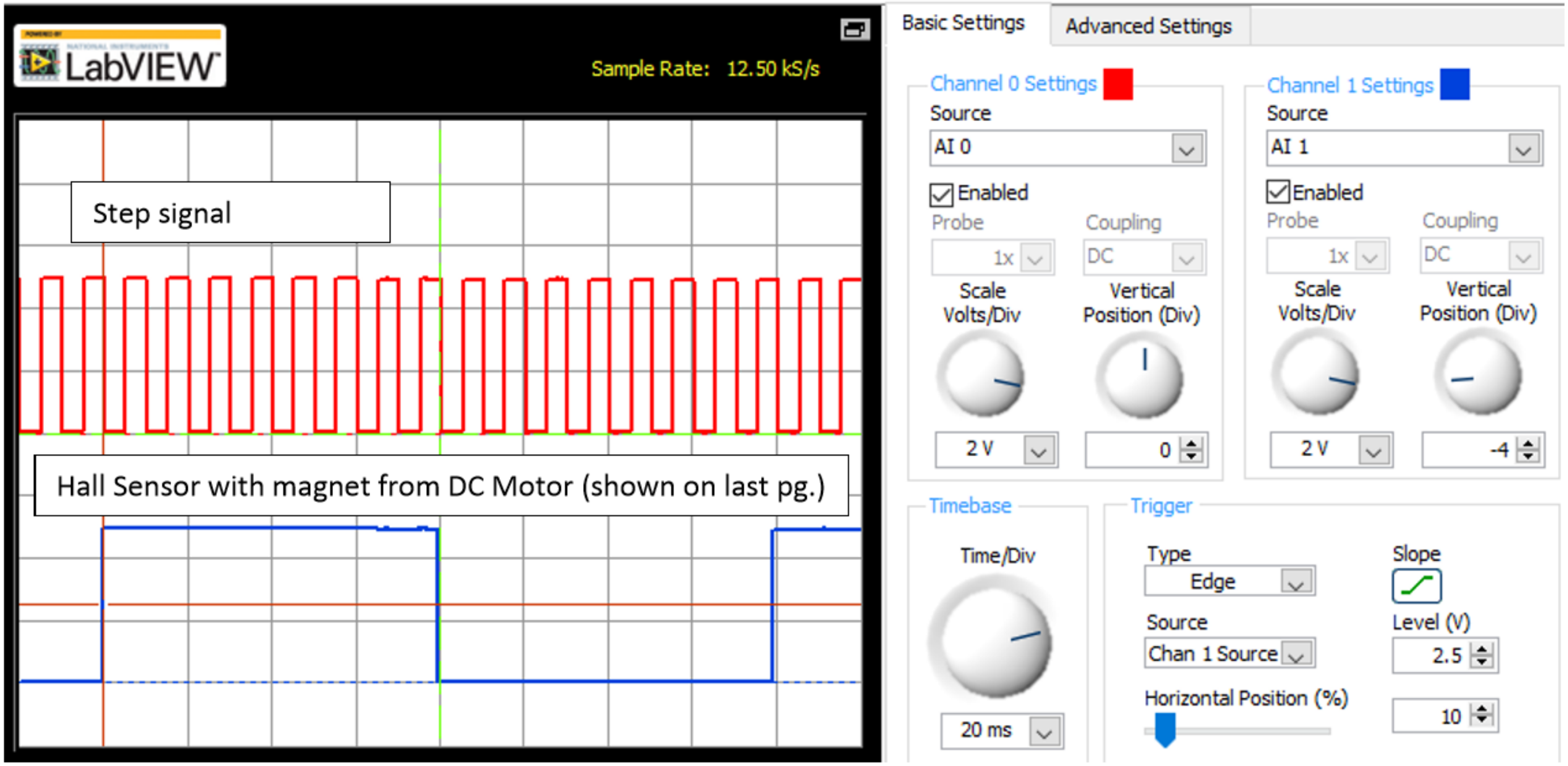
Bi-polar

A Hall Effect sensor is attached near the magnet.

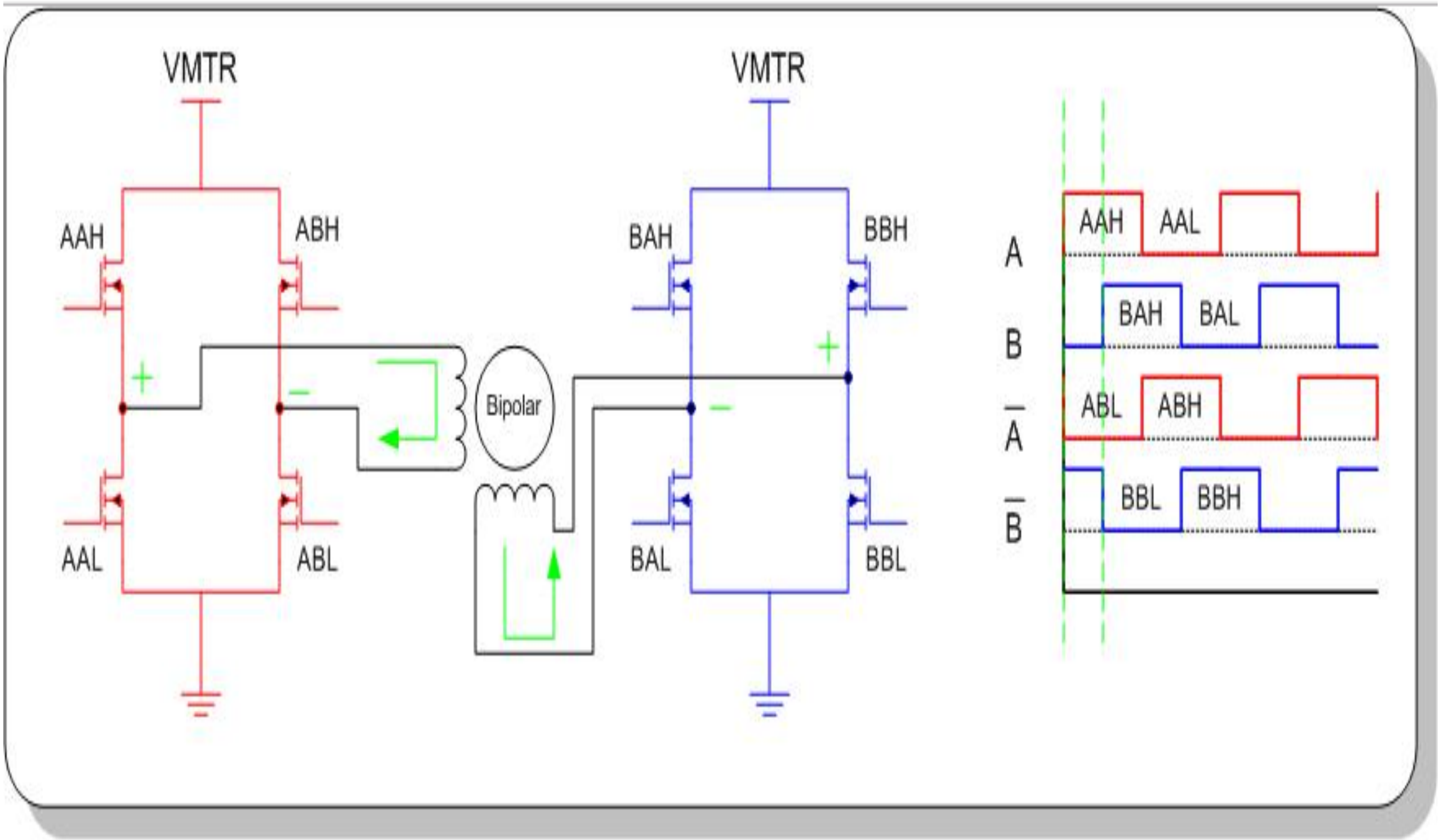
The magnet is the same one used on the DC motor with three NS magnets that will generate 3 pulses per revolution.

The Hall sensor requires a 5 volt supply and a 10 k pull up resistor attached to the open collector output of the Hall sensor.

The Hall Effect sensor, and the magnet were added to measure the RPMs of the stepper motor. The magnet has three N/S poles that generate three pulses per revolution.



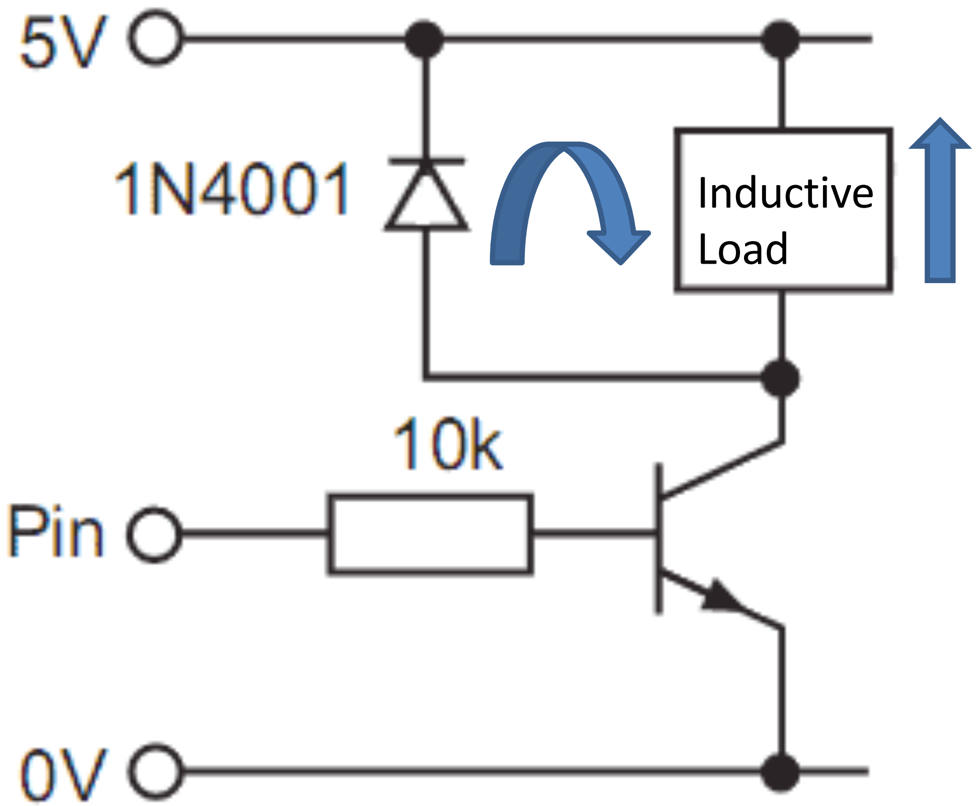
The motor takes 480 ms. per revolution, 1/480 ms. = 2.08 revolutions per second or about 120 RPM.



A unipolar stepper requires a simpler driver circuit because coil current flows in only one direction. The bipolar driver requires a dual H-Bridge. This example uses MOSFET transistor which are much more efficient than bipolar transistors.

Flyback, Flywheel,Transient Suppression Diode.

Back EMF – Back Electromotive Force.

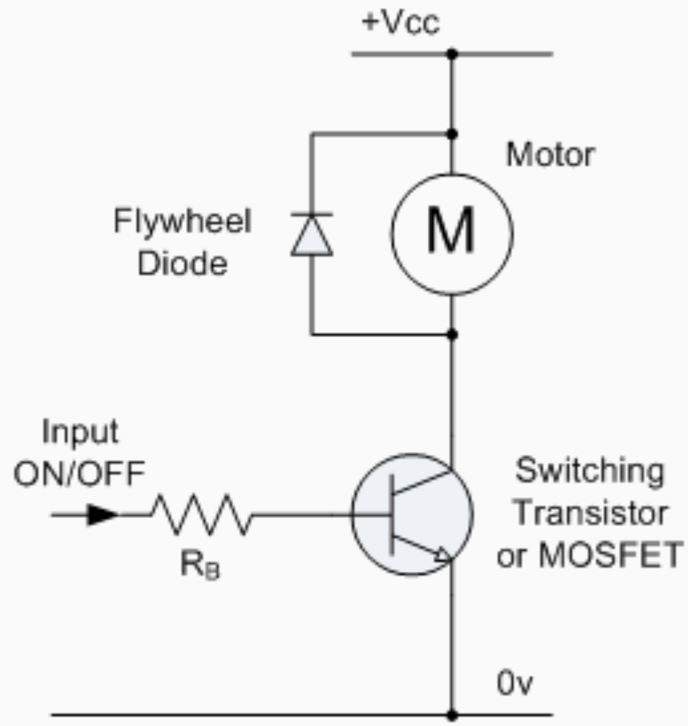


When inductive loads such as relays, solenoids and motors are switched off the stored energy in the magnetic field around the inductor collapses. The collapsing field produces a current in the opposite direction than when the coil is energized. Without the diode a high voltage spike may damage electronic circuits used to drive the coil. Adding the diode provides a path for the current to flow through the diode and the load, preventing damage to the electronic circuits. During normal operation the diode is reverse biased.

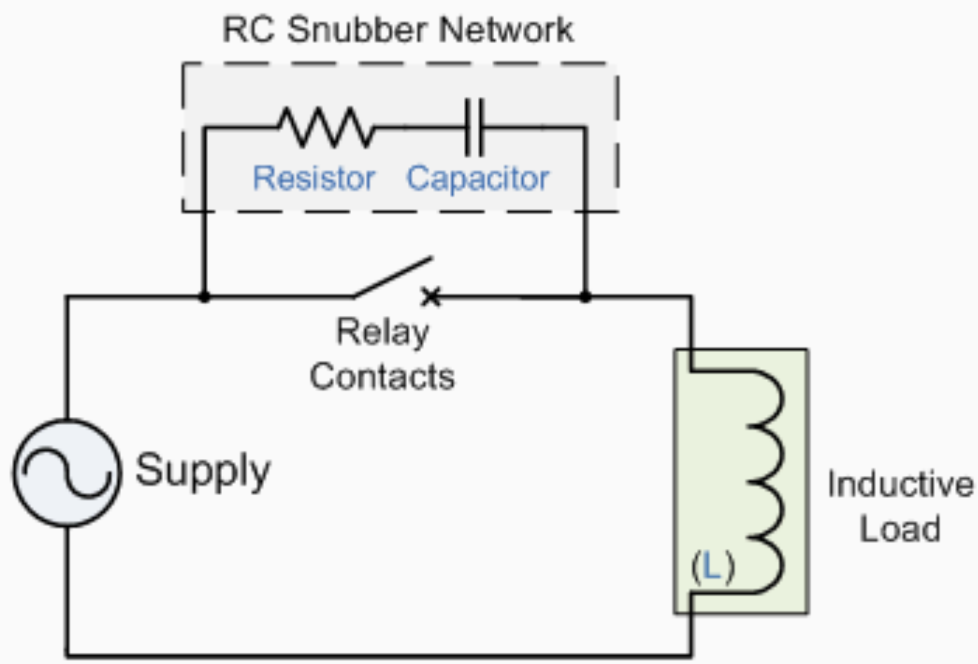
DC Motor Switching and Control

Small DC motors can be switched "On" or "Off" by means of relays, transistors or mosfet circuits. The simplest form of motor control is "Linear" control. This type of circuit uses a bipolar **Transistor as a Switch** (A Darlington transistor may also be used were a higher current rating is required) to control the motor from a single power supply. By varying the amount of base current flowing into the transistor the speed of the motor can be controlled for example, if the transistor is turned on "half way", then only half of the supply voltage goes to the motor. If the transistor is turned "fully ON", then all of the supply voltage goes to the motor and it rotates faster. Then for the linear type of control, power is delivered constantly as shown below.

This circuit shows the connections for a **Uni-directional** (one direction only) motor control circuit. A continuous logic "1" or logic "0" is applied to the input of the circuit to turn the motor "ON" or "OFF" respectively and a flywheel diode is connected across the motor terminals to protect the transistor from any back emf generated by the motor when the transistor turns "OFF". As well as the basic "ON/OFF" control the same circuit can also be used to control the motors rotational speed. By repeatedly switching the motor current "ON" and "OFF" the speed of the motor can be varied between stand still (0 rpm) and full speed (100%). This is achieved by varying the proportion of "ON" time (t_{on}) to the "OFF" time (t_{off}) and this is called "**Pulse Width Speed Control**".

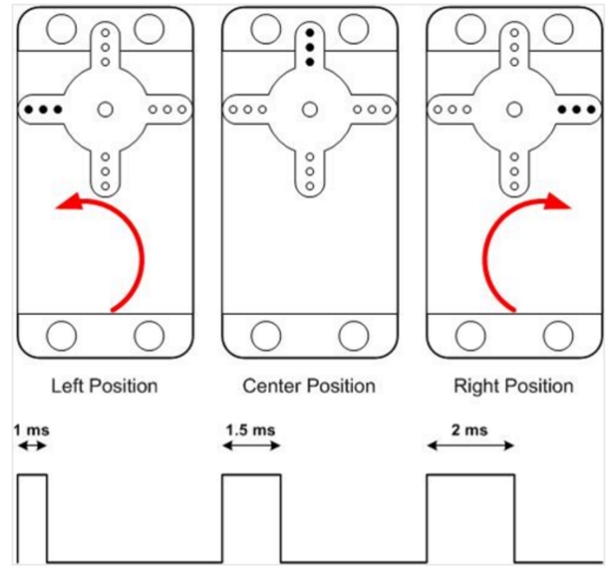


Relay Snubber Circuit



The snubber circuit reduces the peak voltages when an inductive load is removed. If the control is a switch it reduces the spark across the switch. If the control is an electronic circuit (transistor) the snubber prevents damage to the electronics.

Servo Motors



RC Servos has been designed to accept the RC PWM signal which is nothing more than a periodic pulse with a width of anything between 1.0 ms and 2.0 ms. Some systems with more resolution will have allow for pulses in the range of 0.5 ms to 2.5 ms. However, 1 ms to 2 ms is pretty much standard. The idea behind this position protocol is that 1.5 ms commands the servo to go to the center position. A 1.0 ms pulse commands the motor to attempt to reach its leftmost position and 2.0 ms to its rightmost position. Any pulse measuring in between 1.0 ms and 2.0 ms is decoded as a position in between leftmost and rightmost. Since the remote control is analog, practically any position can be attained.

Servo Motor – used in robotics and remote controlled devices.
Position is controlled using PWM signal. The motor has a feedback potentiometer to determine the current position of the output control arm. The servo motor moves CW and CCW with about 180 degrees of movement. The servo has a gear train to give it a greater torque.

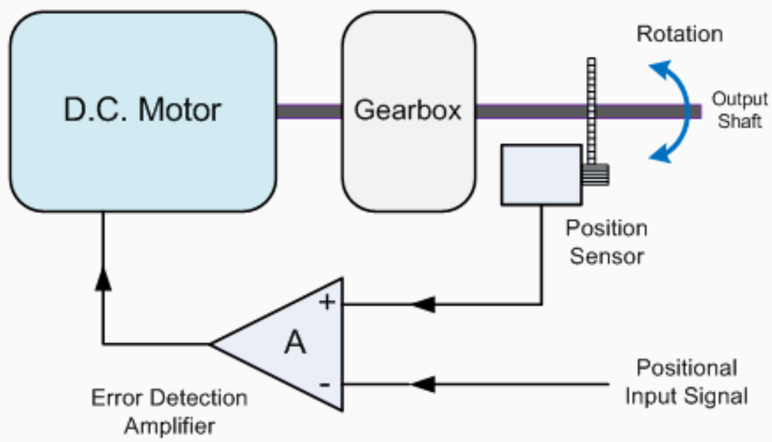
DC Servo Motors

- Another type of DC motor is called a servo motor. Servos are often used for position control.
- A servo motor is controlled using a pulse width modulated signal.

The DC Servo Motor

DC Servo motors are used in closed loop type applications were the position of the output motor shaft is fed back to the motor control circuit. Typical positional "Feedback" devices include Resolvers, Encoders and Potentiometers as used in radio control models such as airplanes and boats etc. A servo motor generally includes a built-in gearbox for speed reduction and is capable of delivering high torques directly. The output shaft of a servo motor does not rotate freely as do the shafts of DC motors because of the gearbox and feedback devices attached.

DC Servo Motor Block Diagram

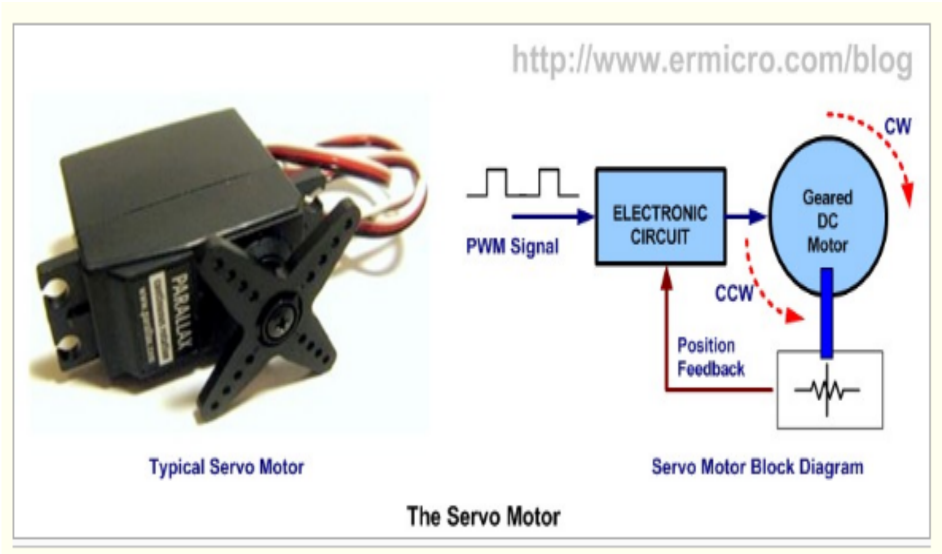


A servo motor consists of a DC motor, reduction gearbox, positional feedback device and some form of error correction. The speed or position is controlled in relation to a positional input signal or reference signal applied to the device. The error detection amplifier looks at this input signal and compares it with the feedback signal from the motors output shaft and determines if the motor output shaft is in an error condition and, if so, the controller makes appropriate corrections either speeding up the motor or slowing it down. This response to the positional feedback device means that the servo motor operates within a "Closed Loop System".

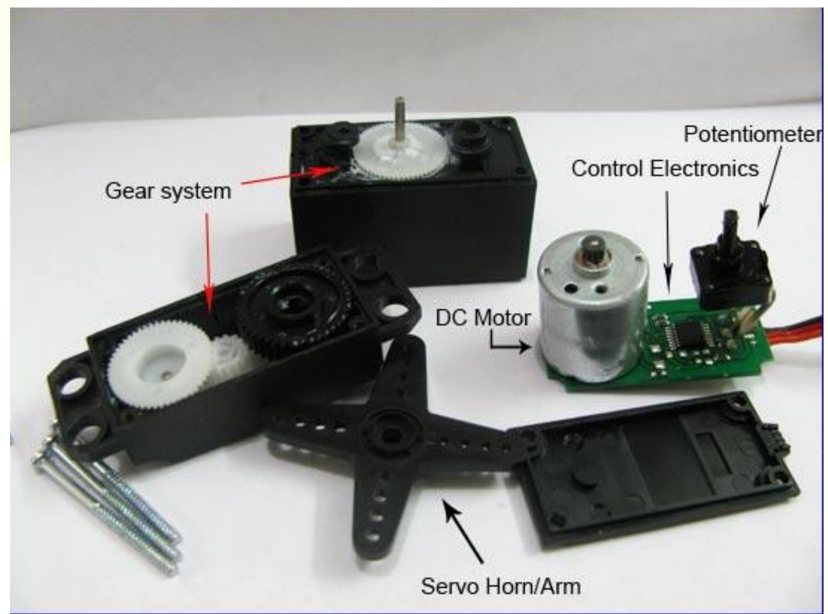
Servo motors are also used in remote control models with most servo motors being able to rotate up to about 180 degrees in both directions making them ideal for accurate angular positioning. However, these RC type servos are unable to continually rotate at high speed like conventional DC motors unless specially modified. A servo motor consist of several devices in one package, motor, gearbox, feedback device and error correction for controlling position, direction or speed. They are controlled using just three wires, Power, Ground and Signal Control.

- A servo motor consists of four main sections:
  - DC motor
  - Gear set
  - Potentiometer for feedback.
  - Electronic Circuit

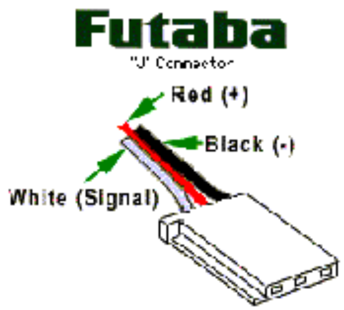
DC Servo Motors



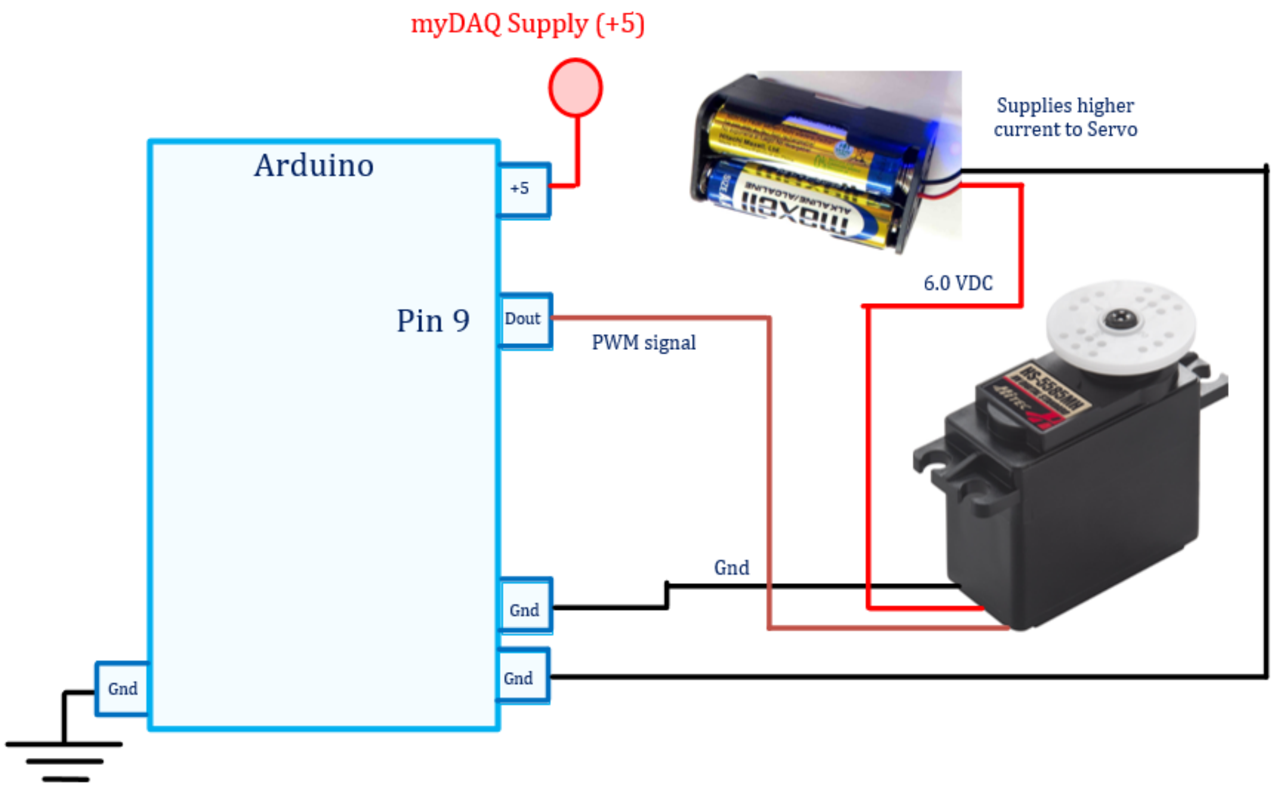
Servo Motor Block Diagram (2nd image)



DC Servo Motors

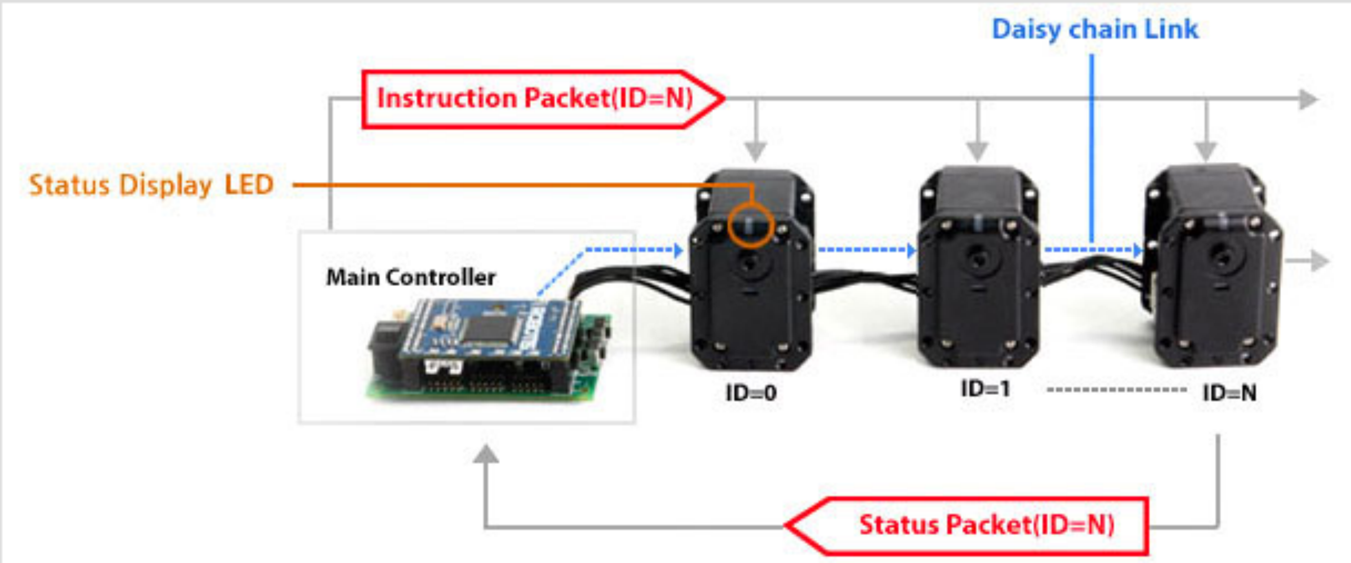


- A servo motor connects to a controller using three wires.
- One of the wires connects to ground a second wire to 4.5 – 6.3 volts DC.
- The 3rd wire is the input signal. The signal originated from a controller producing a PWM output signal.
- The PWM signal has a period of 20 ms. The time high ranges from about 1 to 2 ms.

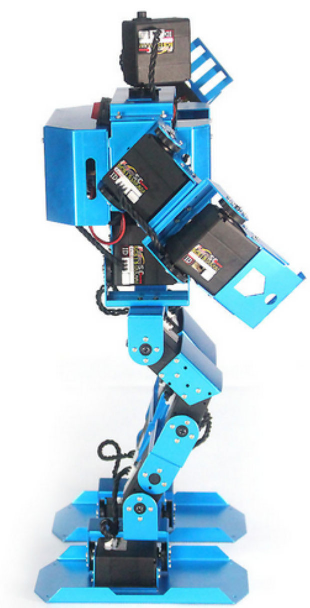


Servo hardware configuration.

Servo Motor Connections to Arduino: Connect the power for the servo to a 4.5 to 6.5 volt battery or DC power supply. The signal controlling the servo motor shaft position is controlled using an Arduino PWM output.



Smart Servos Connected To A Controller Offer A Much Cleaner Solution

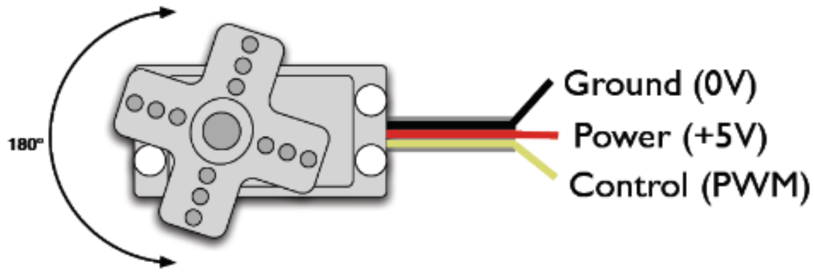


Smart Servo Motor – used in robotics and remote controlled devices.

Smart Servos communicate between a controller and many servos through a serial link.

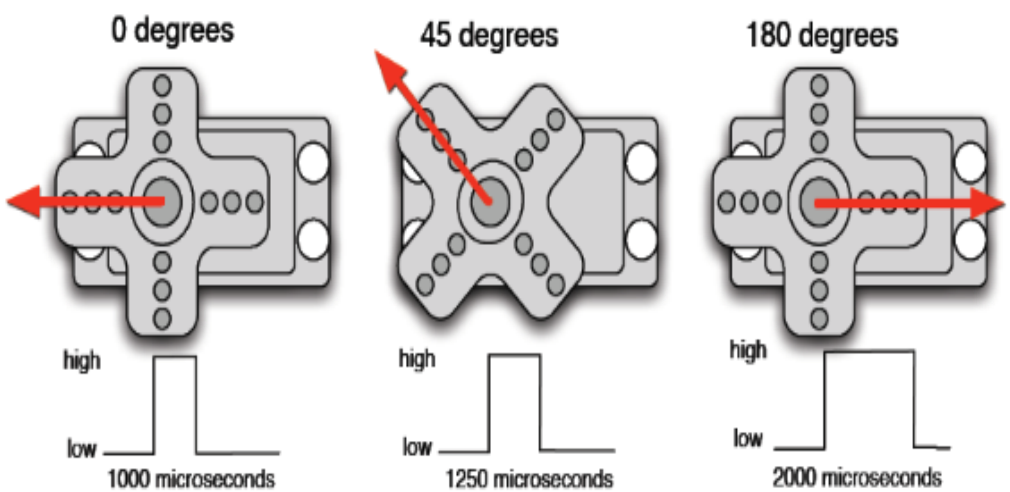
Make the servo movement

Of course you need to connect the servo motor. Now get your servo motor, and you can find three pin hole. They are:



*

As we know, we are going to send a pulse to make the servo move. The pulse should be ranges from 1 to 2 millisecs.

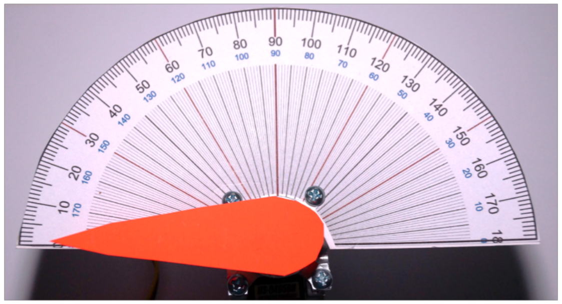


*

servo_motor_test_with_slider_Nov_11_2016

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
// Modified by Michel Hanbury
// For CAM8302E Nov. 12, 2016

#include <Servo.h>    // add servo library

Servo myservo;  // create servo object to control a servo

int slider = 0;    // analog pin used to connect the potentiometer
int Analog_In;     // variable to read the value from the analog pin
int Servo_Angle;
void setup()
{
  Serial.begin (9600);   // 9600 bits/second to display variable data
  myservo.attach(9);     // attaches the servo on pin 9 to the servo object
}

void loop()
{

  Analog_In = analogRead(slider);            // reads the value of the potentiometer (value between 0 and 1023)
  Servo_Angle = map(Analog_IN, 0, 1023, 0, 180);      // scale it to use it with the servo (value between 0 and 180)
                                                      // adjusted for min of 0 deg to 180 deg.
  myservo.write(Servo_Angle);                         // sets the servo position according to the scaled value

    Serial.print("Analog In  ");
    Serial.print(Analog_In);
    Serial.print("    Servo Angle   ");
    Serial.println(Servo_Angle);

    delay(15);                           // waits for the servo to get there
}
```

Servo Motor test program.

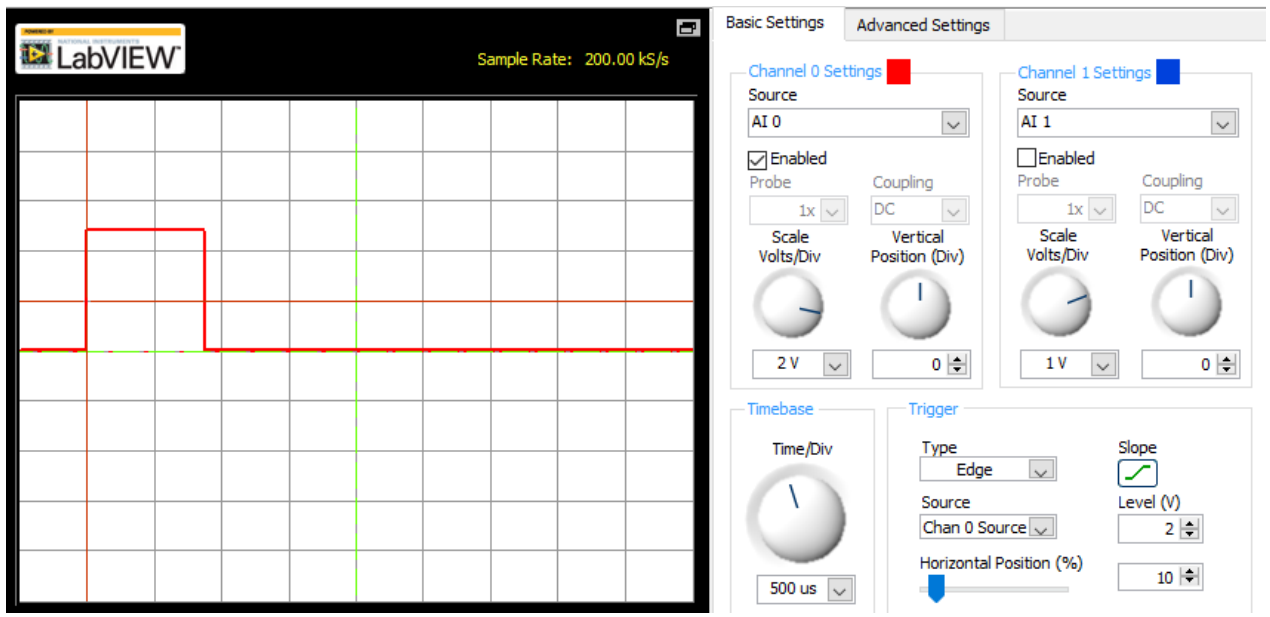


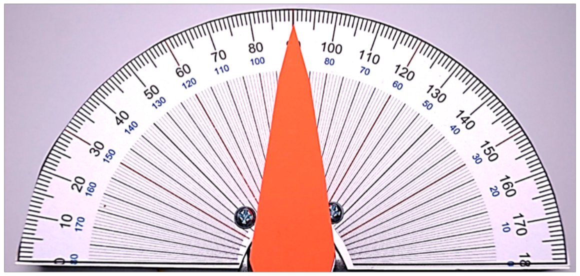
Servo motor fully CCW. There are hard limits to the servo position, do not drive the motor to the maximum or minimum positions.

This screen captures shows a time high of ~0.9 ms.

Program Output

```
Analog In  186    Servo Angle   32
Analog In  186    Servo Angle   32
Analog In  186    Servo Angle   32
```



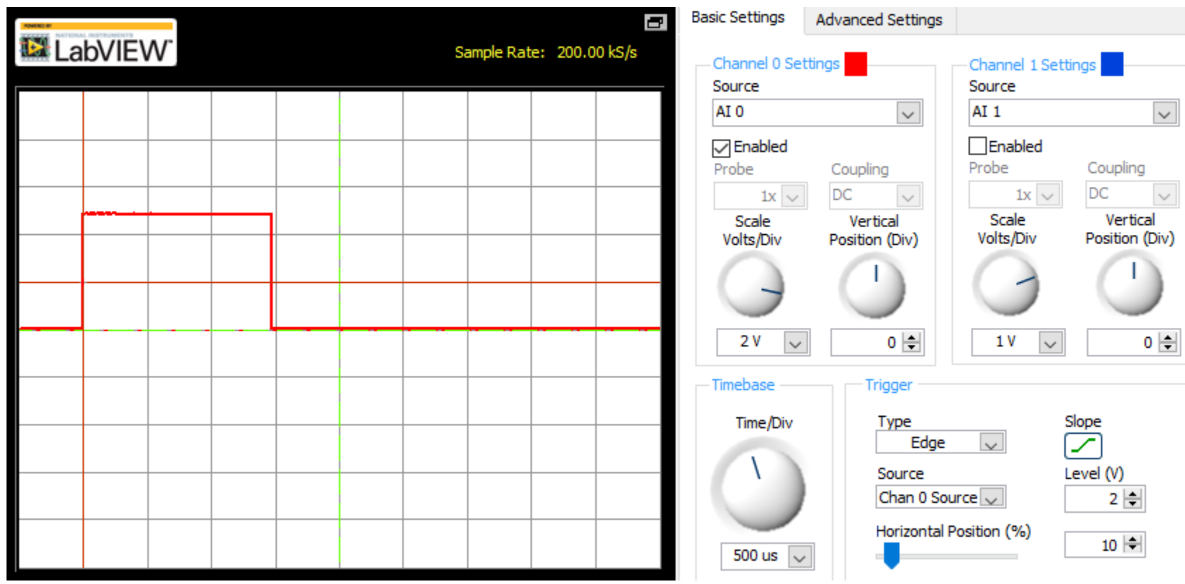


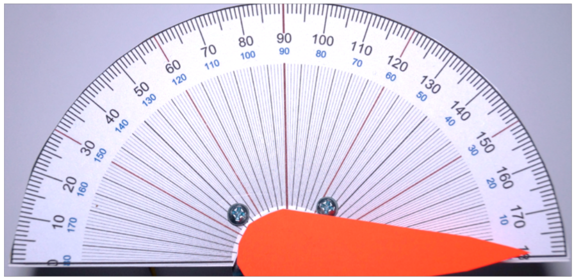
Servo motor at the middle neutral position. This is normally about 1.5 milliseconds.

This screen captures shows a time high of ~1.5 ms.

Program Output

```
Analog In   515    Servo Angle    90
Analog In   515    Servo Angle    90
Analog In   515    Servo Angle    90
```



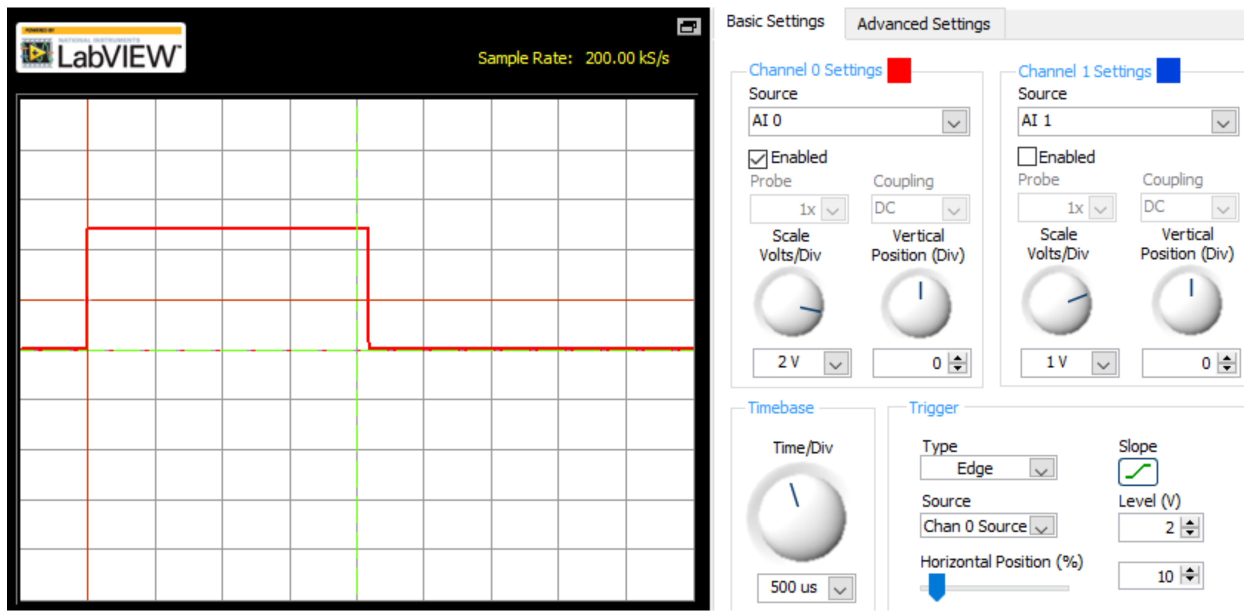


Servo motor fully CW. There are hard limits to the servo position, do not drive the motor to the maximum or minimum positions.

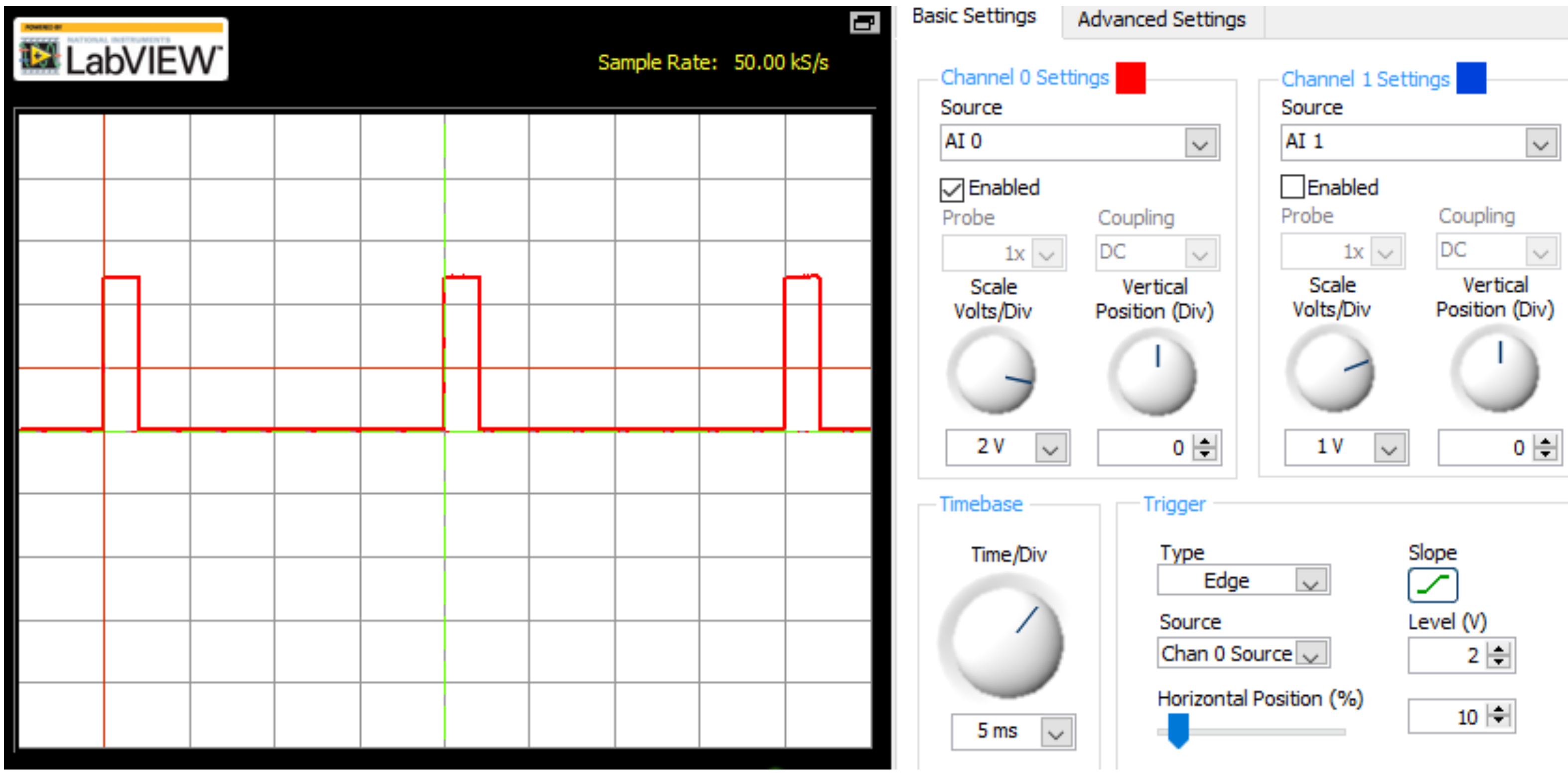
This screen captures shows a time high of ~2.1 ms.

Program Output

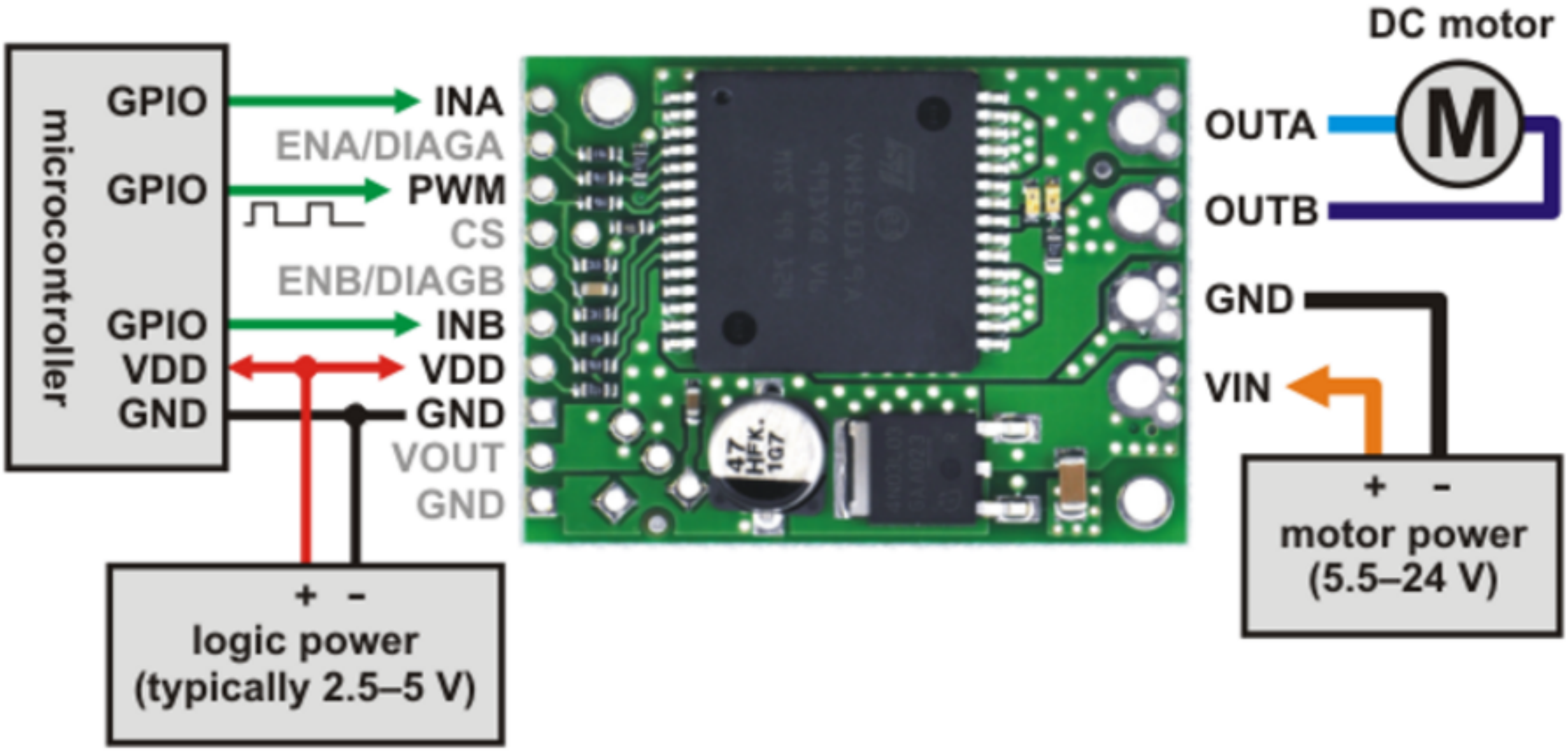
```
Analog In   856     Servo Angle    150
Analog In   855     Servo Angle    150
Analog In   855     Servo Angle    150
```



Screen capture showing the period of the PWM for the Servo. The period equals 20 milliseconds. It is the time high that is changed to control position.



Pololu DC Motor Controller



DC Motor Controller can handle 12 Amps at 24 volts.
Direction and speed control.
Easy interface to PLC, myDAQ or Arduino.

Pololu Robotics & Electronics