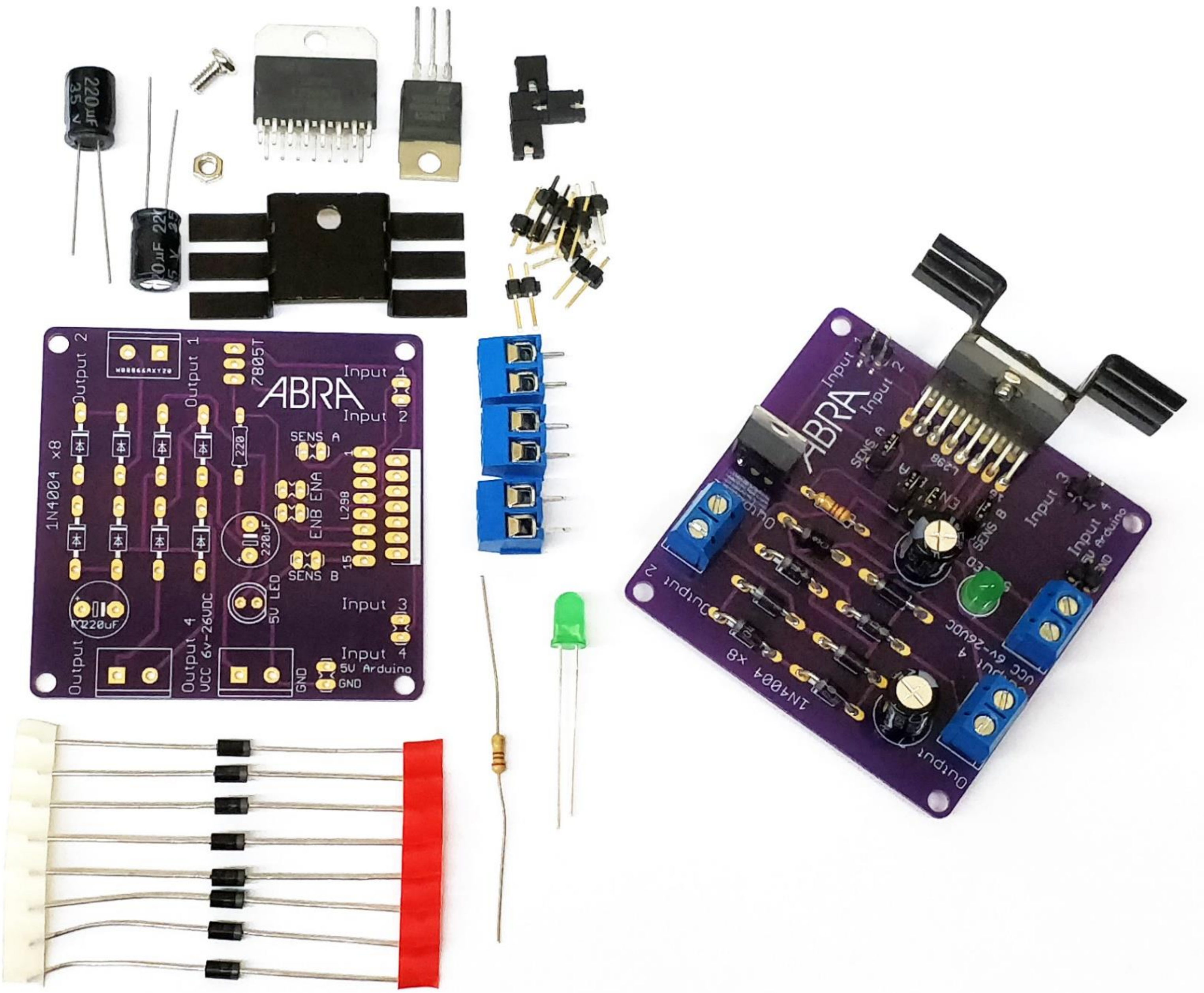
STEM AK-240

Motor Driver Module – Educational D.I.Y Kit



www.abra-electronics.com

Table of Contents

Table of Figures

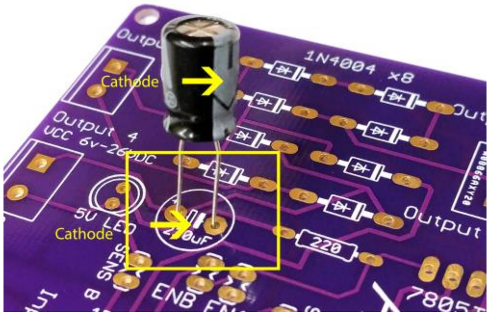


Figure S1

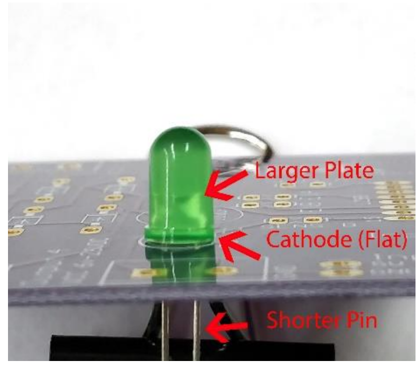


Figure S2

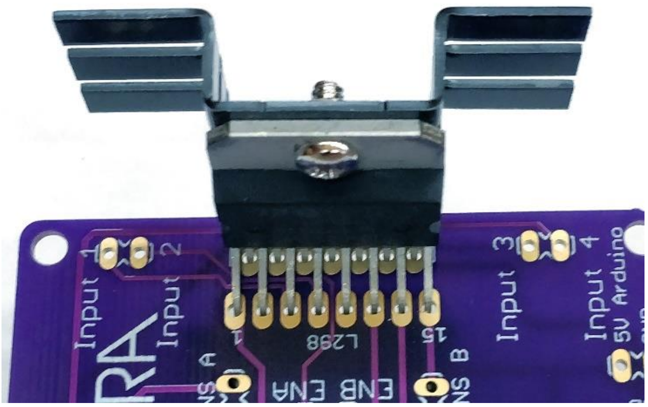


Figure S3

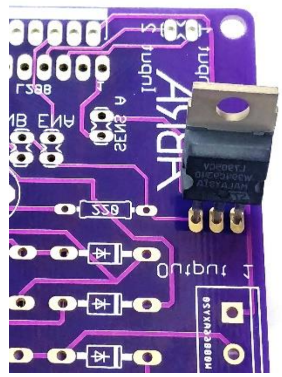


Figure S4

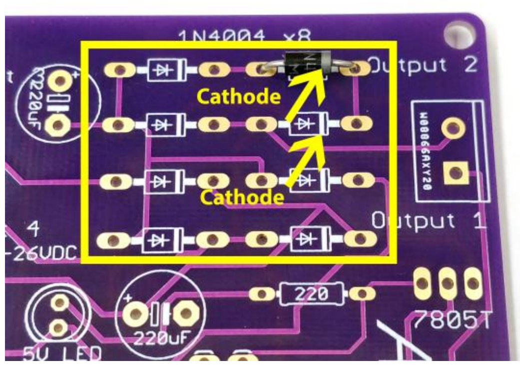


Figure S5

About this Product

This Motor Driver Board was designed to deliver power for up to two motors independently. Input signal can be feed from microcontroller using PWM signal or dc signal. Added functionality to enable or disable motor movement. Current sensing pins are assessable to use AM-110 to measure current and report back to the microcontroller. Additional Pin for powering Arduino or 5v microcontroller using jumper wires rather than screw terminals.

This product was designed for students and beginner hobbyist to learn, sharpen and improve their knowledge in electronics.

For more kits please visit
www.abra-electronics.com/educational-kits-trainers/educational-kits/

Bill of Materials

Custom Board

x1

Header – 2 Pin

x7

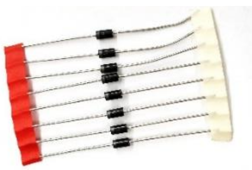
Screw Terminal Block – 2 Pin

x3

220µF Polarized Capacitor

x2

1N4004 Diode



x8

5V regulator 7805

x1

5mm green LED

x1

220Ω Resistor ¼ Watt

x1

L298N Dual H-Bridge + Heatsink

x1

Philips Screw

x1

Hex nut

x1

Shunt/Jumpers

x7

Tools Required/Suggested

Temperature Adjustable Soldering Iron

60/40 or 63/37, Tin-Lead Solder

Flush Cutters

Needle Nose Plier

Phillips Screwdriver

Lint Free Cloth

Soldering Instructions

1. Start by soldering components that have a smaller height clearance. A good example is to solder the following components in the order that they are mentioned:
    a. Diodes
    b. Resistor
    c. LED
    d. Headers pins
    e. Terminal blocks
    f. Capacitors
    g. IC – 7805T and L298N

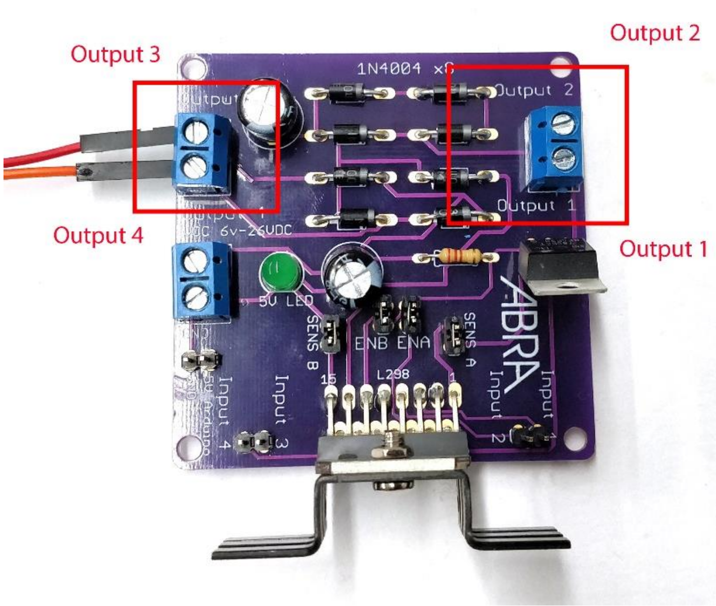
Note: when soldering lower height components, ensure you provide a gap between the board and the component itself before soldering to help prevent heat transfer from the components to the board.

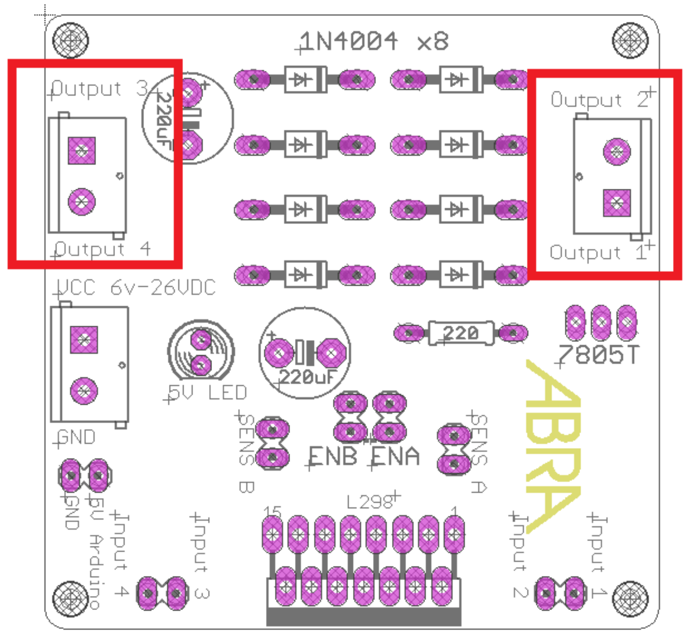
2. After soldering use the flush cutter to help remove excess wires and allow ease of access to solder joints.
3. Pay attention to the polarity of the capacitor, white marking is to indicate that the "-" strip on the capacitor is to be soldered into that location. (*See figure S1*)
4. Please Pay attention to the marking on the board for the LED. The layout in white shows a flat side to the round LED body. This Flat side is the negative pin (Cathode) for the LED. (*See figure S2*)
5. Solder the L298N and insert the heatsink with the fins facing outwards. (*See figure S3*)
6. Insert the 7805T as indicated on the board (*See figure S4*)
7. Clean the board with a lint free cloth and some isopropyl for a clean solder work. (Optional)
8. Solder Diodes with white marker matching with silver band. (*see figure S5*)

Wiring/Interfacing Guideline

BOARD VIEW

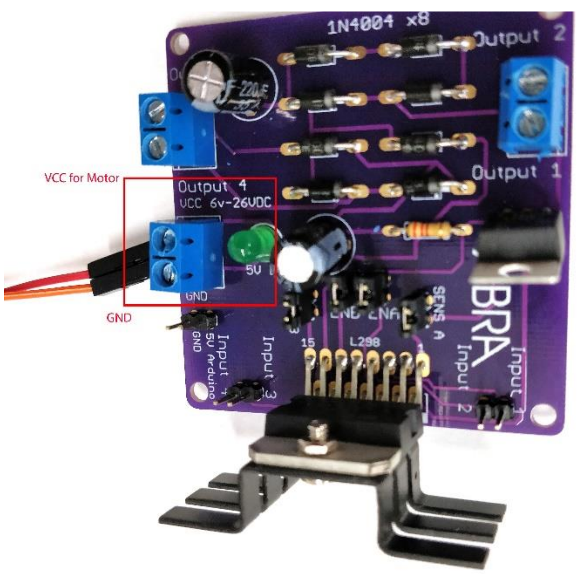
PCB SCHEMATICS VIEW

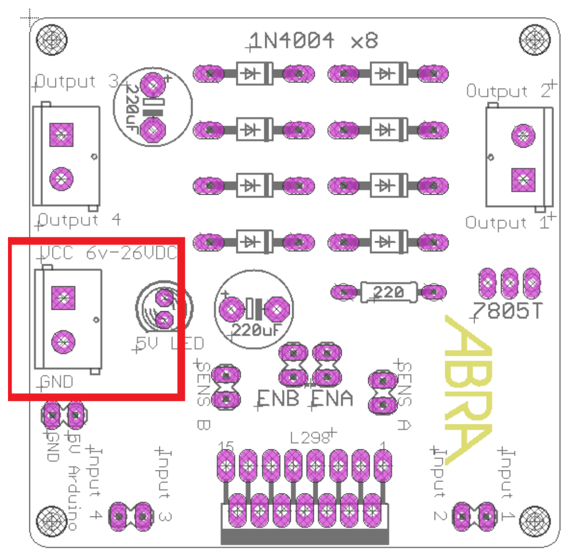




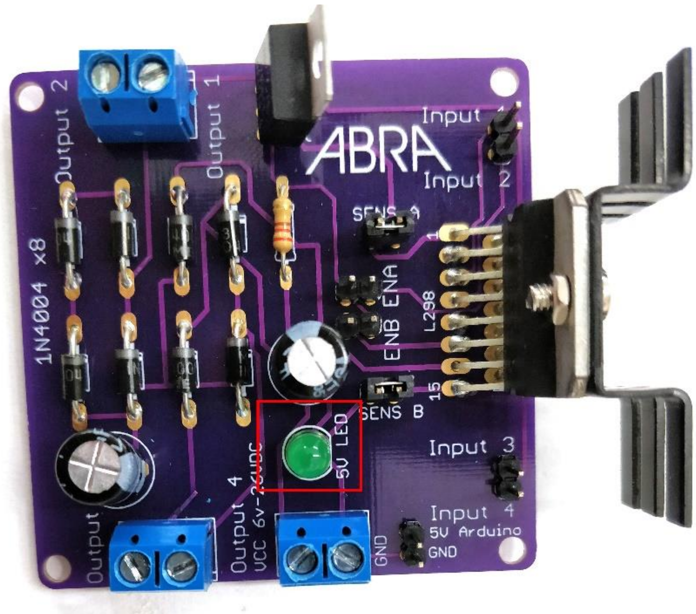
Connect a motor to terminal block output 1 and 2.

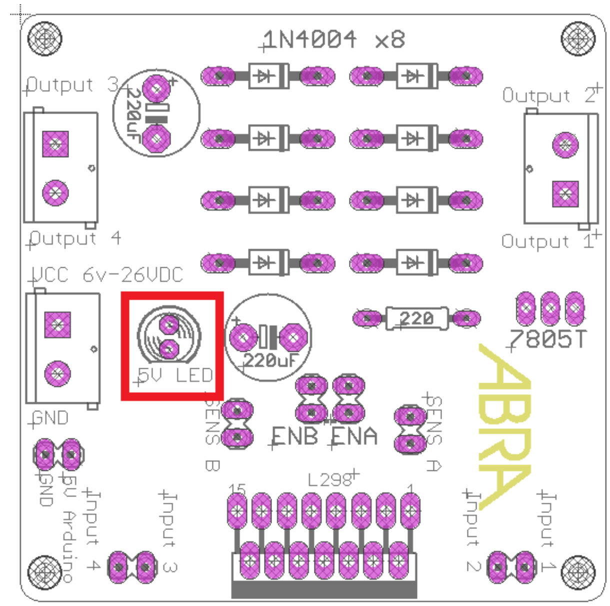
To interface another motor to the board, wire the second motor to terminal block output 3 and 4.



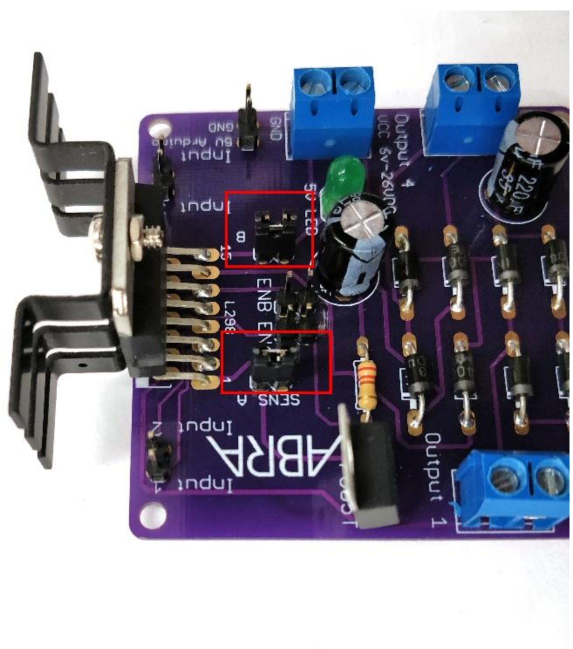


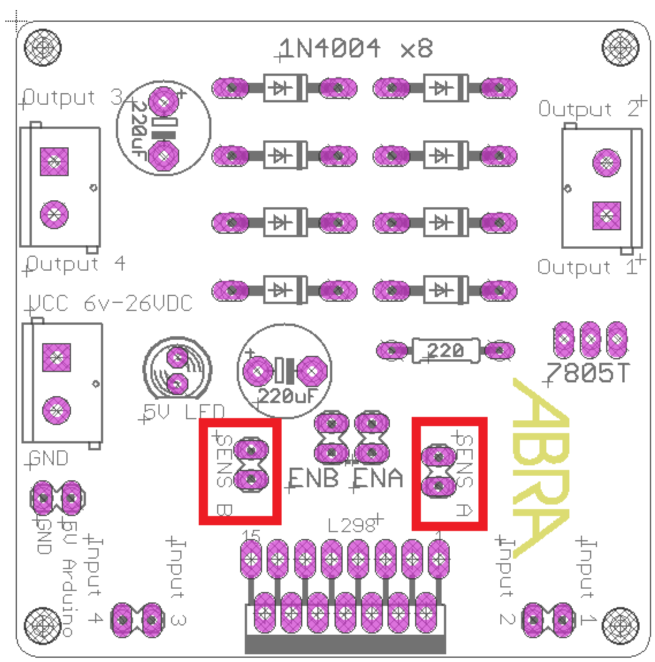
Provide power as indicated on the board with power varying from 6-26VDC and ground. The maximum this board can deliver in terms of current is 1.5amps.



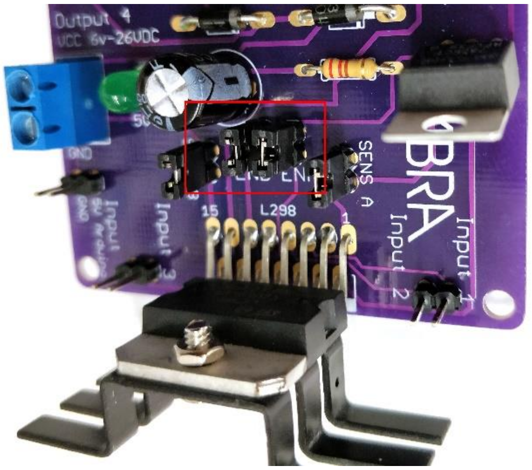


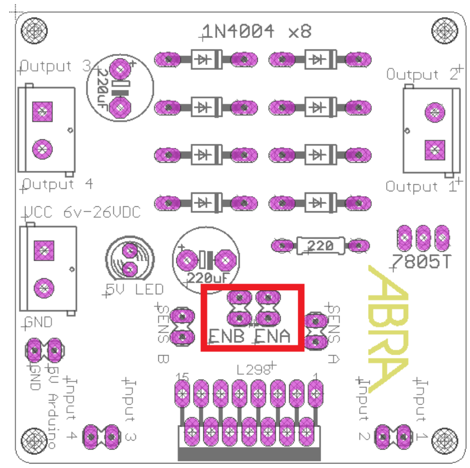
When power is provided the status led will light up.



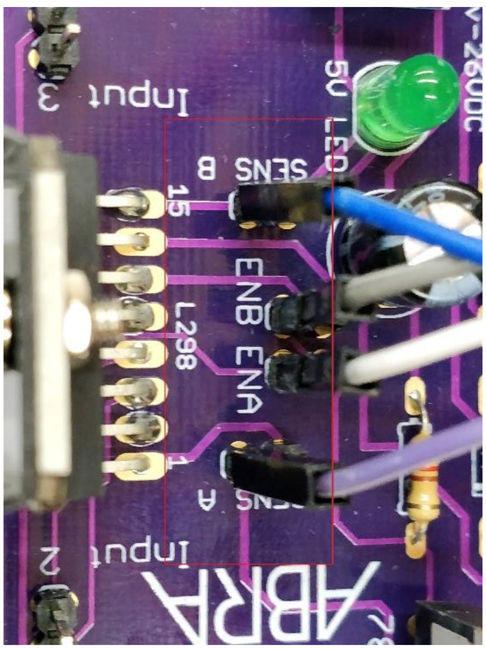


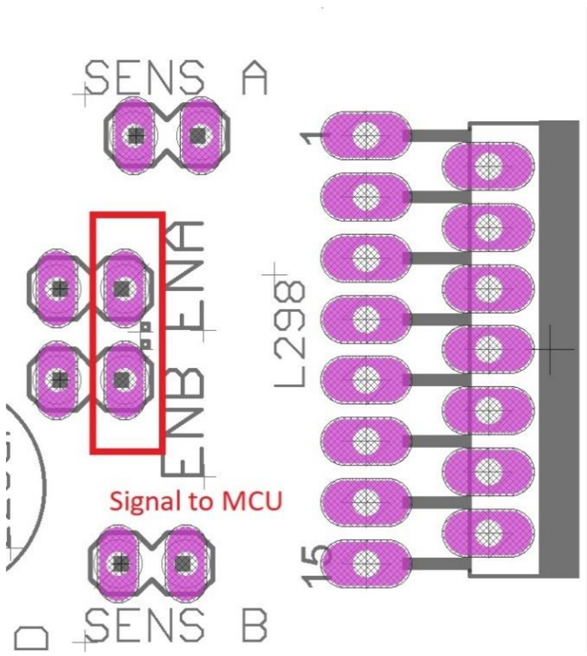
Place jumpers on SENS B and A to disable current sensing capability



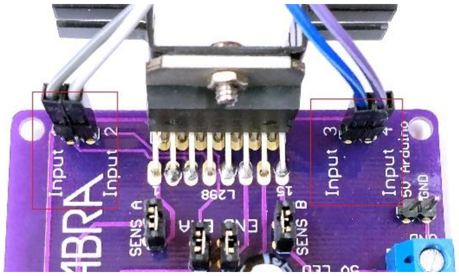


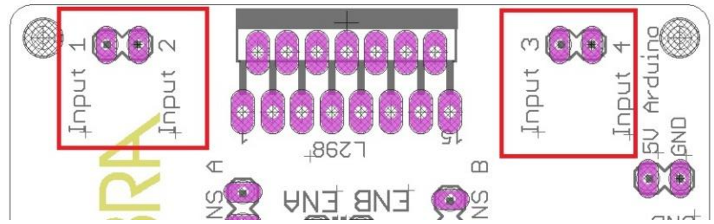
Place jumpers on ENA and ENB to allow motors to move and enable PWM signals to motor.



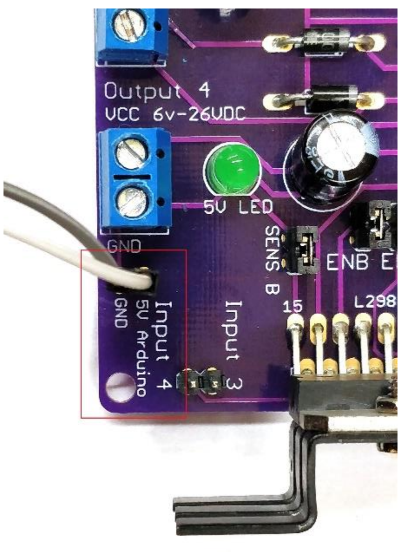


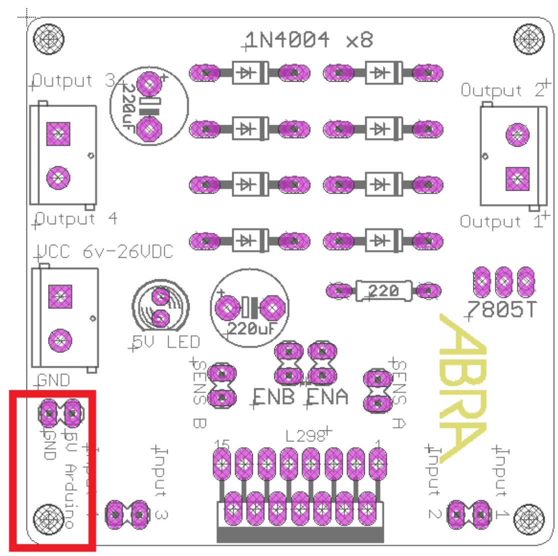
Remove jumpers and connect a jumper cable from the board to your microcontroller to control digitally enable of motor movement.





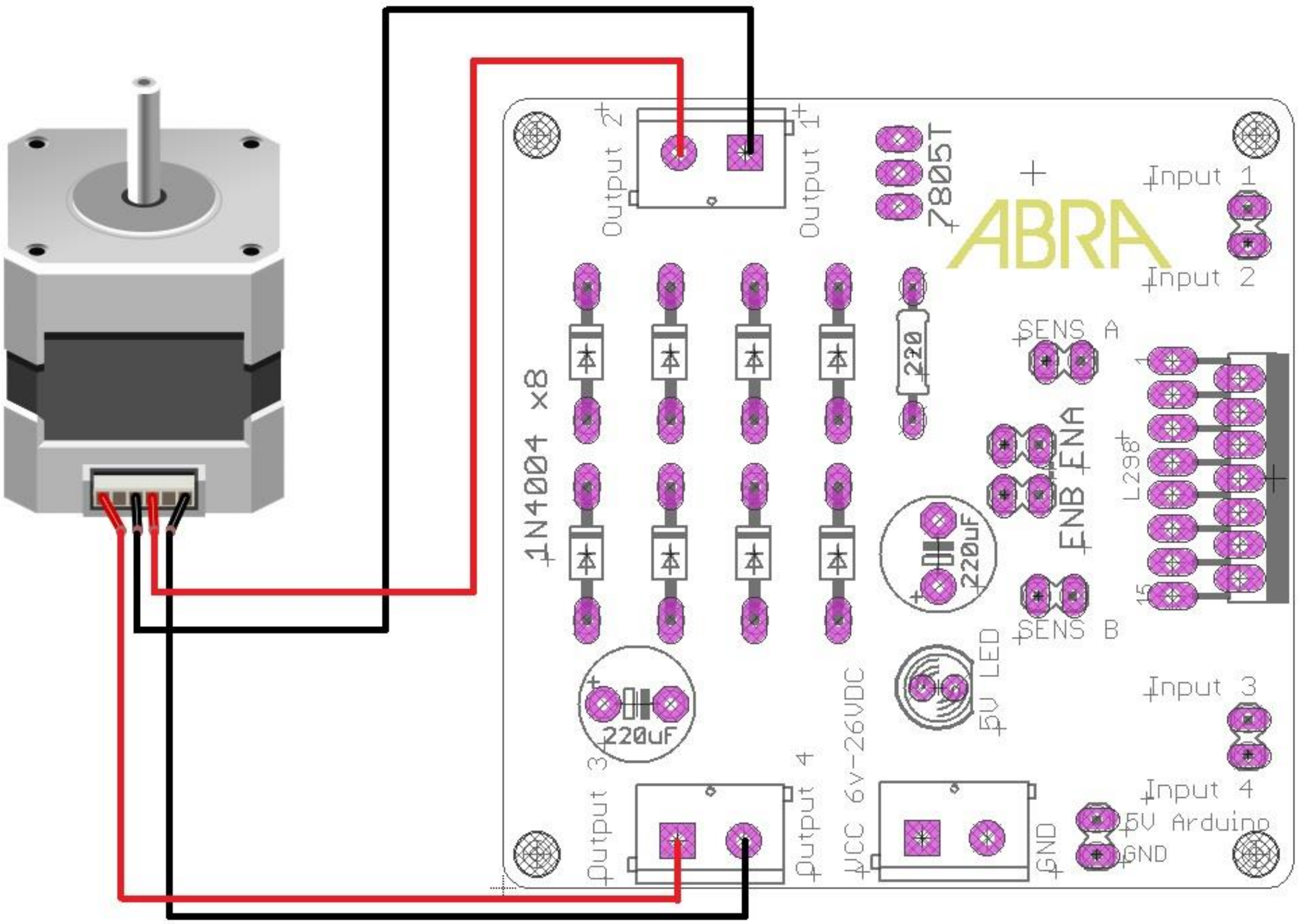
Connect your input signals from your microcontroller to feed PWM signal via the pin header pins located to the right and left side of the L298n





Optional header pins for providing Arduino power to Vin using male to female jumper wires.

Stepper Motor Interfacing

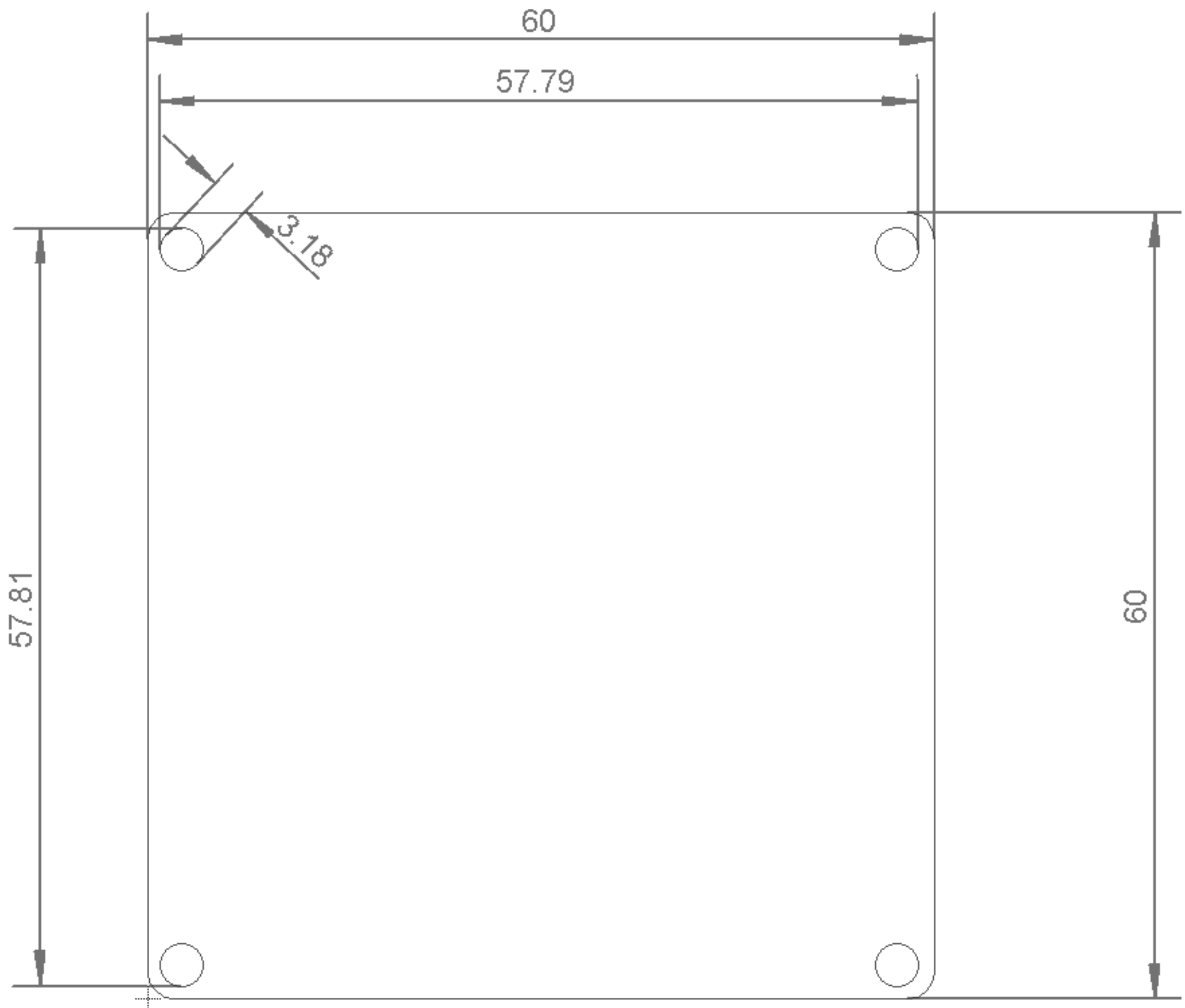


Connect the four leads from your stepper motor as shown in input 1,2,3 and 4, the activation of the coils will be allowing movement at a certain direction. You will only be able to drive one stepper motor.

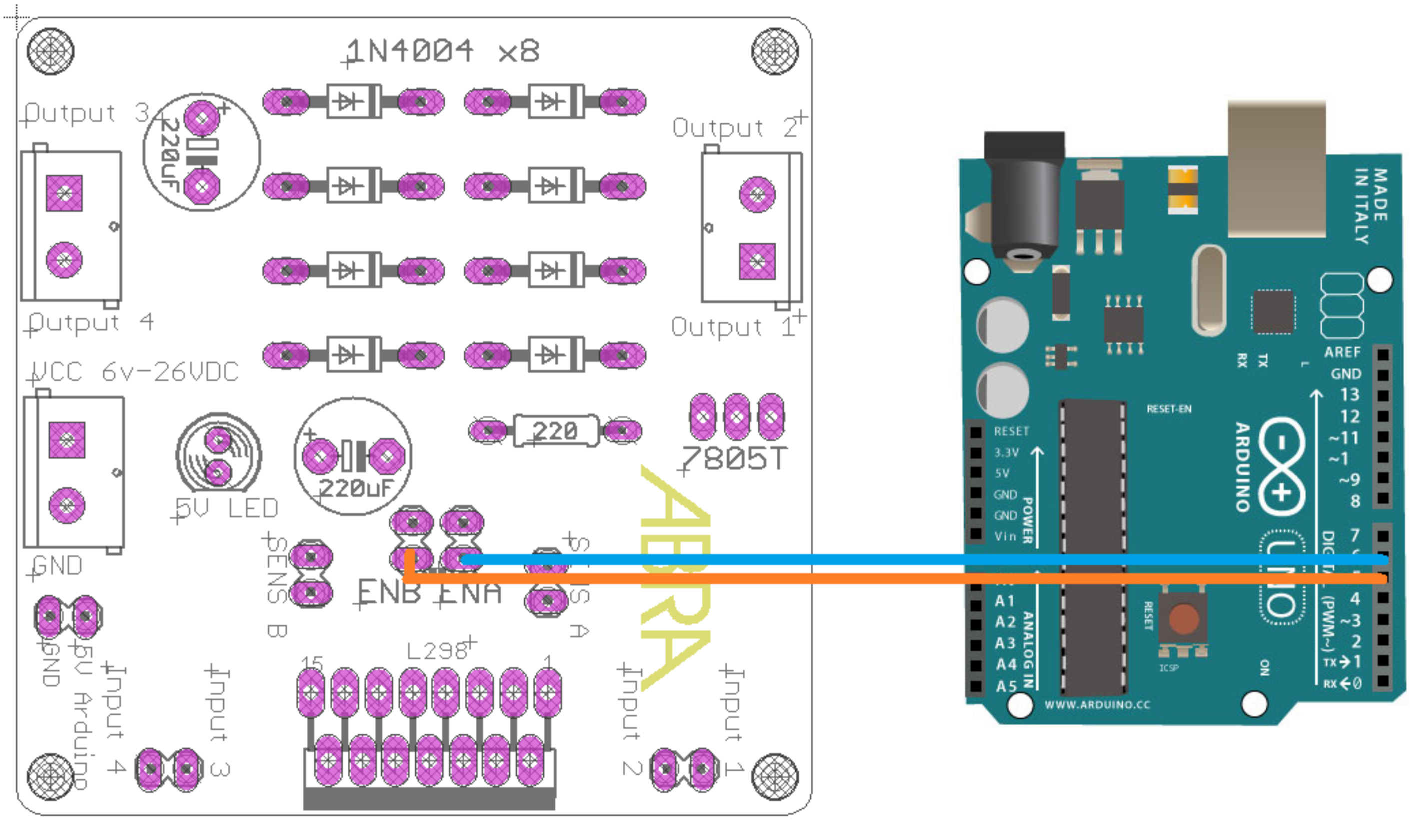
Board Specifications

Absolute Maximum Ratings

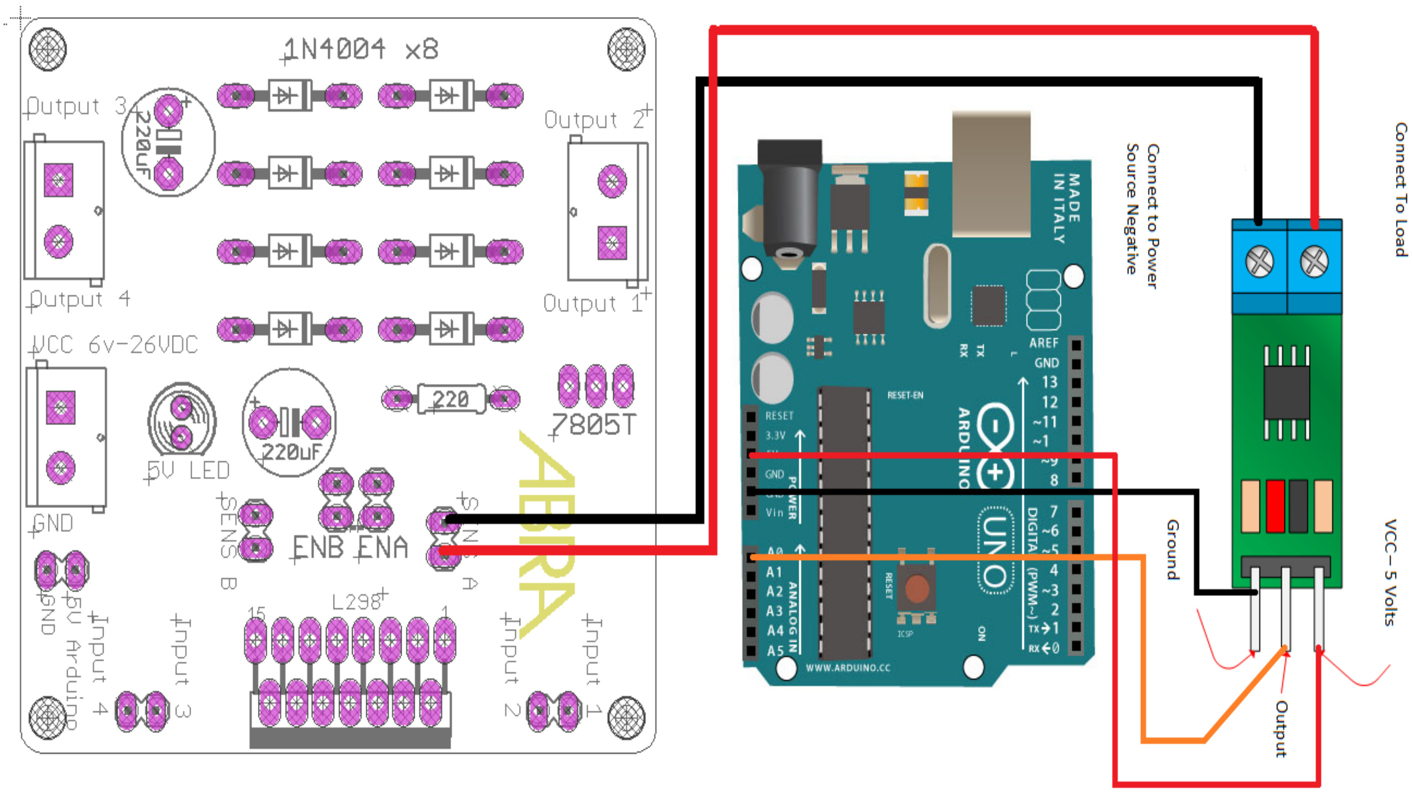
Parameter	Value	Unit
Power Supply	35	V
Arduino Source Power	5	V
Peak Output Current (each channel)	2	A
- Non-Repetitive (t=100µS)	1.5	A
- Repetitive (80% on -20% off; ton = 10ms)	1.2	A
- DC Operation		
Power Dissipation	25	W
Junction Operating Temperature	-25 to 65	ºC
Trace Thickness	2	Oz



Enabling Features



ENABLE MOTOR VIA MICROCONTROLLER



Current Sense to external circuit

Theory of Operation

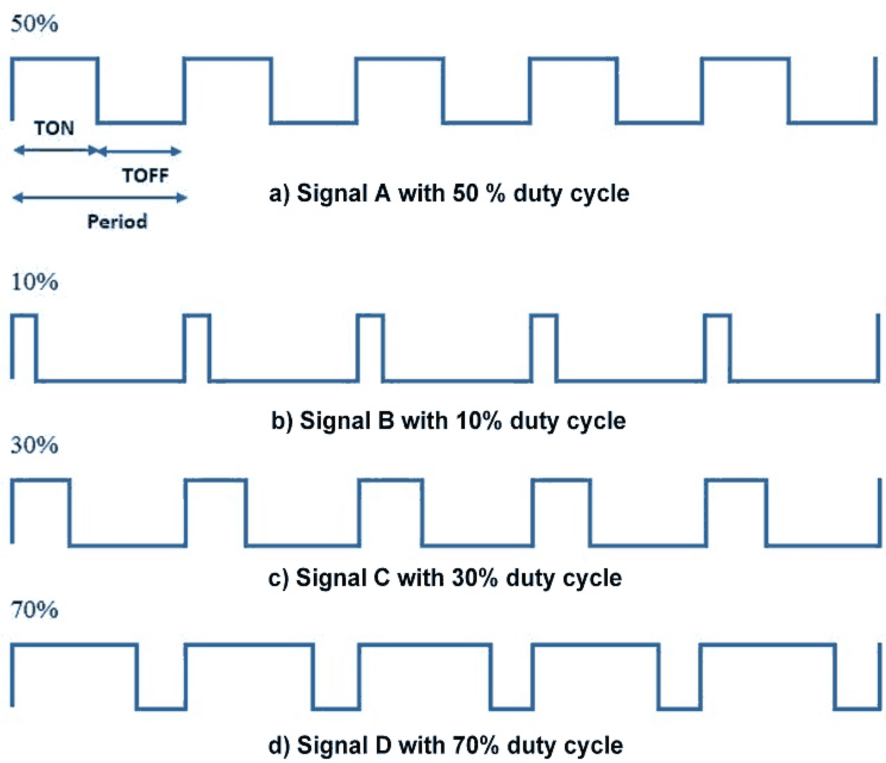
Connecting a wire from the enable pin located on the controller board to your 5v microcontroller PWM will allow to turn the motors on or off at specific time intervals. These time intervals can be done via programming creating a timer and interrupt or delay time function.

A regular digital pin will offer signals from two states 0 or 5vdc, commonly referred to as on or off. If the end user wanted to run a motor for 10 seconds and wants half of the 10 seconds to stop, we would call this a duty cycle function. If this code is left in the loop, it will continuously repeat itself every 10 seconds, 0-5 seconds on, 5-10 seconds off. This on-off interval that repeats is called a pulse width modulation, this specific duty cycle is set to 50% as it is on half the duration of the time specified. You can have different duty cycles as well ranging from 0-100%.

$$Duty\ Cycle = \frac{Ton}{Total\ cycle\ time} \times 100\%$$

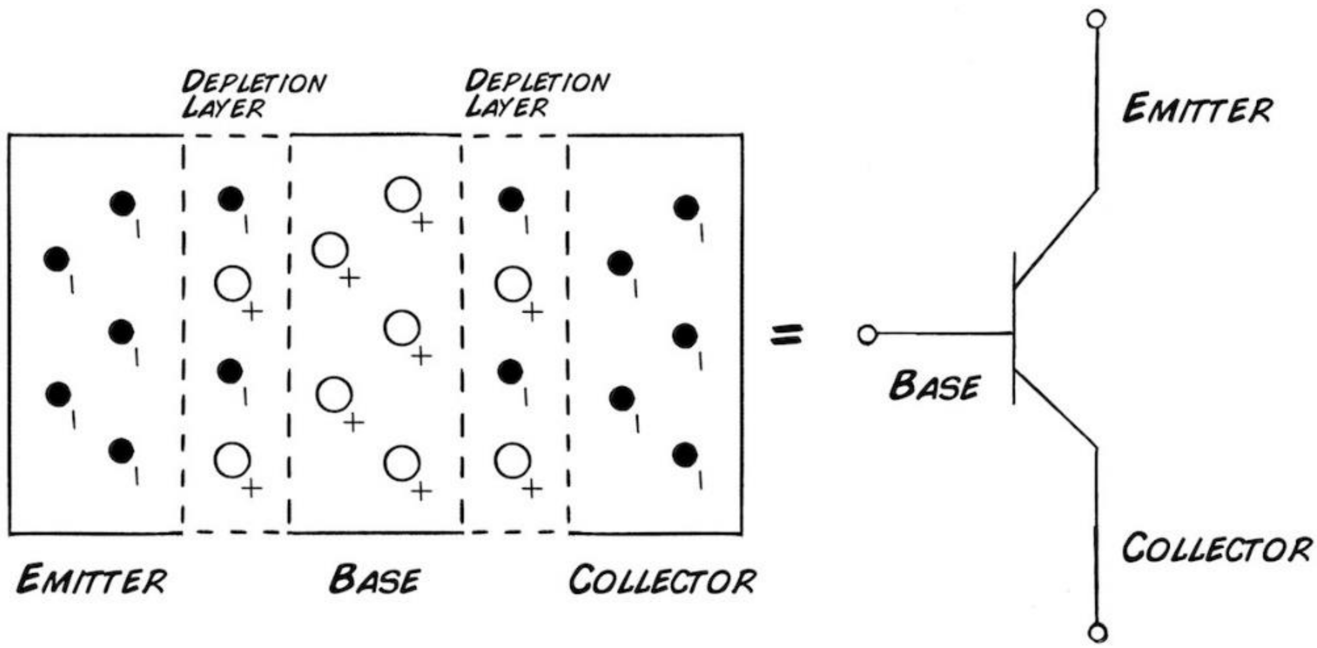
or

$$Ton\ required\ for\ desired\ percentage = \frac{Desired\ duty\ cycle\ [\%]}{100\%} * Total\ cycle\ time$$

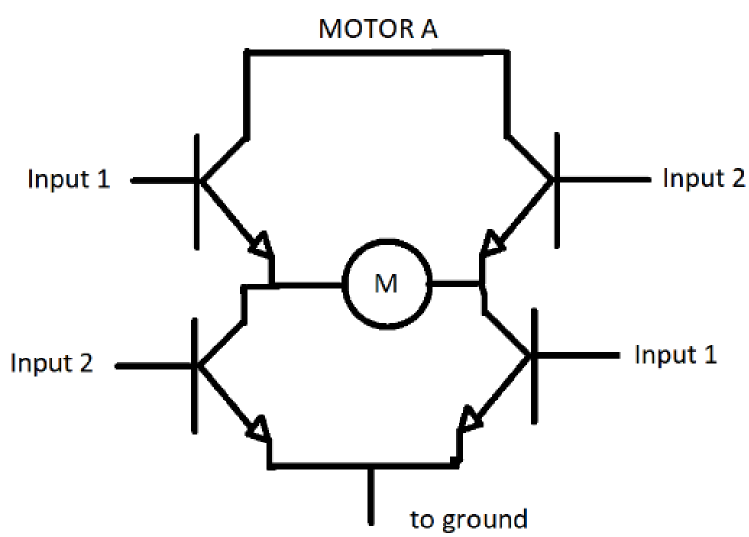


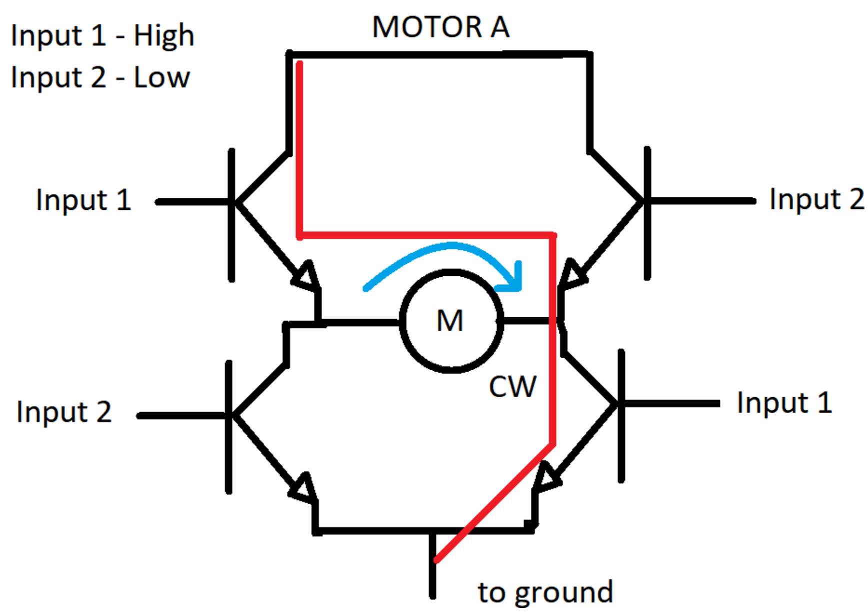
H-Bridge is designed to allow the end user to easily control Bi-Polar motors to spin in either direction depending on how the two input signals are set as. Internally the transistor is like an electronic switch. The symbol of a switch is shown below with two states, either as open or closed. Open prevents power from passing through as the line is cut open and current doesn't flow through, resulting in infinite resistance. Closed circuit allows current to flow through like a race track, resulting in low resistances from the wire or material that the current must pass through. To be able to turn on or off the switch without having to physically touch it, a transistor is used with the microcontroller acting as the arm that pushes or pulls the middle line to create an open or closed circuit.

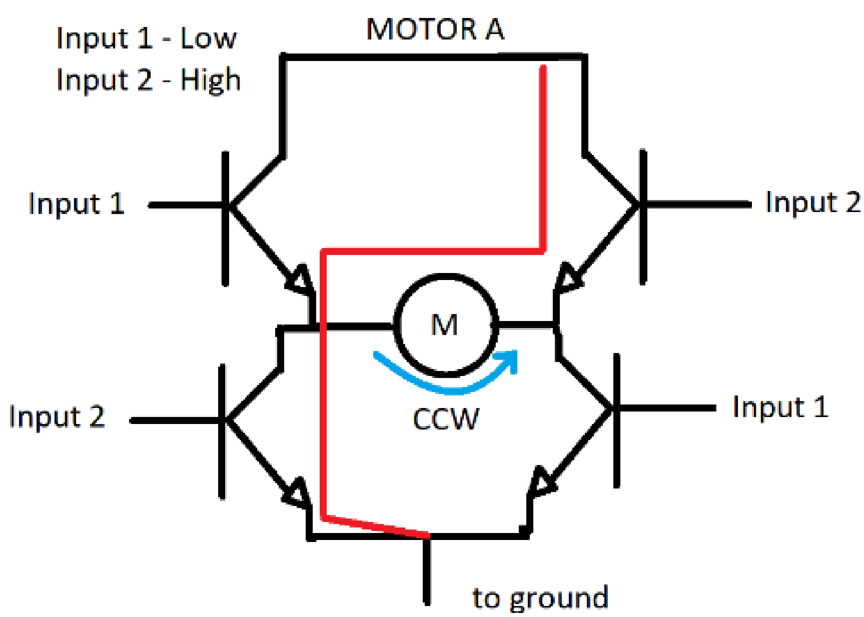
Transistors can get complicated very easily. For this example, we are focusing on NPN transistors the other type is PNP and the materials used are silicone based rather than germanium. We are also looking at a simple TO-220 package-based transistor, as the type and package format can vary to current and voltage needs. When the microcontroller connects to the base the electrons flow into the base region creating an area where there is unequal number of electrons (minus sign). This action will fill the gap called the depletion layer. Since the base region is unstable it will allow electrons from the collector to flow into the emitter creating a circuit. To prevent electrons from going back into the collector a barrier called a diode is built into this tiny device. Depending on the material you will observe the voltage from the emitter will have either a 0.7v drop is silicone and 0.3v if germanium.



The L298N is a h-bridge driver that can drive two motors but requires four input signals. As one side has power applied, the motor will spin in one direction as it tries to find the path to ground. If the other side has power applied the motor will spin the opposite direction. If no power is supplied to either side the motor will not spin and if both sides have power applied the motor will stall, burn or go into circuit protection mode. [try to avoid that]







The manufacture of the L298 have also included two pins that can be accessed to measure current if desired, using an external board the current measured can be feed into the Arduino microcontroller as analog value. These current sensing, proximity sensing devices use a principle called Hall Effect transducer to measure varying voltage level.

If you do not want to use this feature, simply keep the jumpers on the board to prevent said feature.

Codes

Remove or uncomment “//” lines to enable pwm control without jumper and via pwm pins from Arduino microcontroller.

```
//remove "//" from the two define and code to enable pwm control
//#define ENA 6
//#define ENB 5

//wire input pins to the associated arduino pin or change numbers
#define IN1 12
#define IN2 11
#define IN3 10
#define IN4 9

void setup()
{
 //pinMode(ENA, OUTPUT);
 //pinMode(ENB, OUTPUT);

 pinMode(IN1, INPUT);
 pinMode(IN2, INPUT);
 pinMode(IN3, INPUT);
 pinMode(IN4, INPUT);

 //inital setup for safety
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,LOW);
}

void loop()
{
 forward();
 delay(5000); // continue forward for 5 seconds
 reverse();
 delay(5000); // continue reverse for 5 seconds
 left();
 delay(5000); // continue left for 5 seconds
 right();
 delay(5000); // continue right for 5 seconds
 halt();
 delay(10000); // continue halt function for 10 seconds will restart at forward
}
```

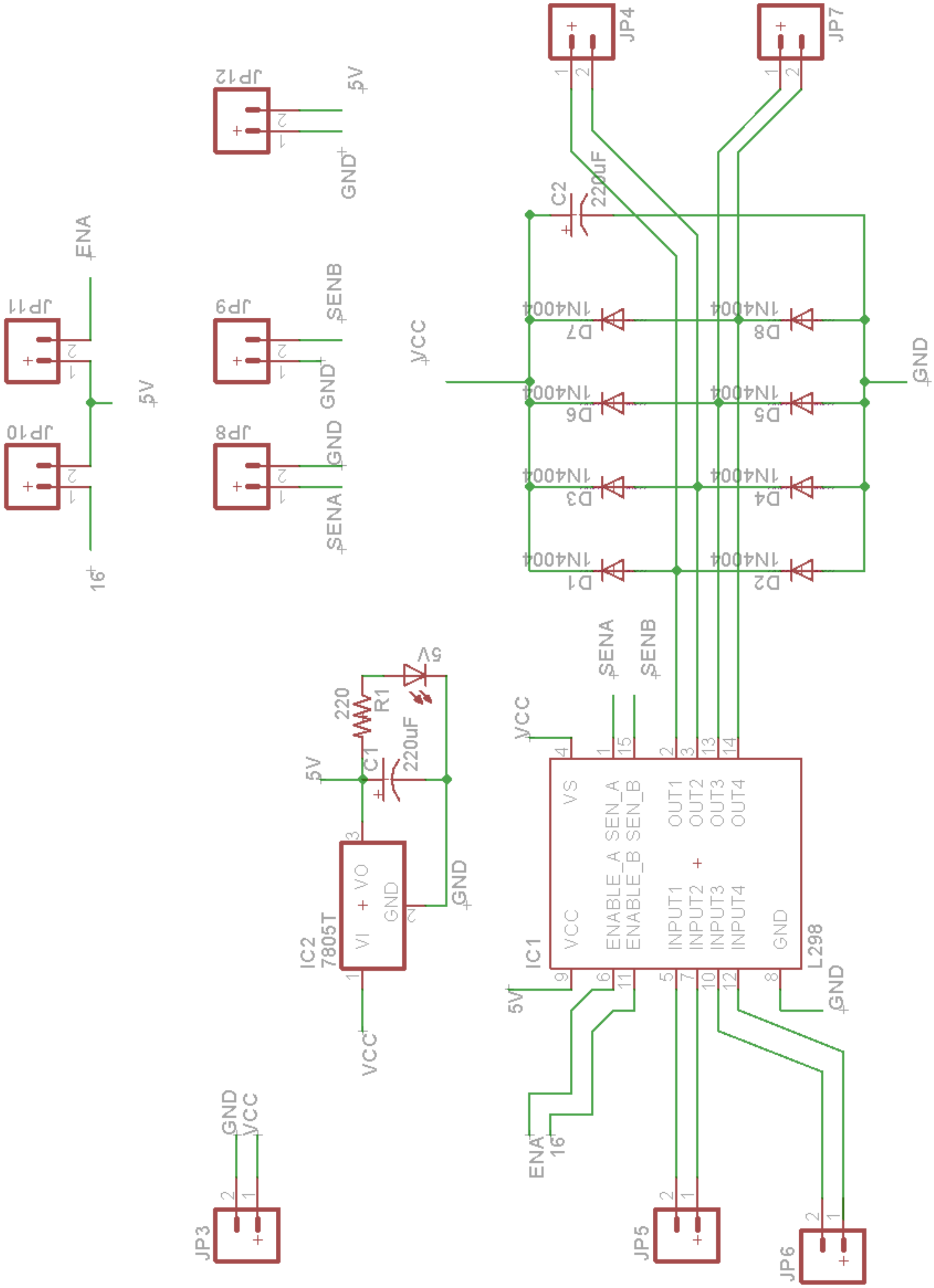
```
void forward()
{
 digitalWrite(IN1,HIGH);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,HIGH);
 digitalWrite(IN4,LOW);
}

void reverse()
{
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,HIGH);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,HIGH);
}

void left()
{
 digitalWrite(IN1,HIGH);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,LOW);
}

void right()
{
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,HIGH);
 digitalWrite(IN4,LOW);
}
void halt()
{
 digitalWrite(IN1,LOW);
 digitalWrite(IN2,LOW);
 digitalWrite(IN3,LOW);
 digitalWrite(IN4,LOW);
}
```

Schematics



JP4
JP7
JP12
JP11
JP10
JP9
JP8
JP3
JP5
JP6
5V
GND
ENA
SENA
SENB
16
VCC
C2
220uF
C1
220uF
R1
220
D1
D2
D3
D4
D5
D6
D7
D8
1N4004
IC2
7805T
VI
VO
IC1
L298
VS
SEN_A
SEN_B
OUT1
OUT2
OUT3
OUT4
ENABLE_A
ENABLE_B
INPUT1
INPUT2
INPUT3
INPUT4