IAT 884

Alternate Output:

Motors, Movement, and Control Voltage

Motors

Types

Standard DC Motors

Speed of rotation is determined by supplied voltage

Servo

Alignment is set by a microcontroller from 0 – 180 degrees*

Stepper Motors

Precision Aligned using a microcontroller

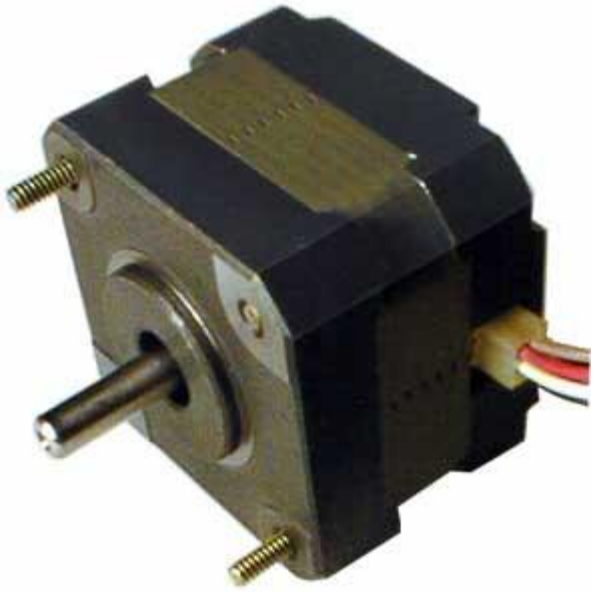
Solenoids

Motors that have a forward and backward (linear) motion

*Servos can be modified to rotate 360 degrees

Motors

Types



Motors

Standard DC Motors

Operation

Speed of rotation is determined by supplied voltage

Motors are all rated for a specific voltage.

Do not exceed this voltage or you will burn out your motor.

Motor Concerns:

Motors produce a magnetic field that can generate back voltage in a circuit which can damage a microcontroller.

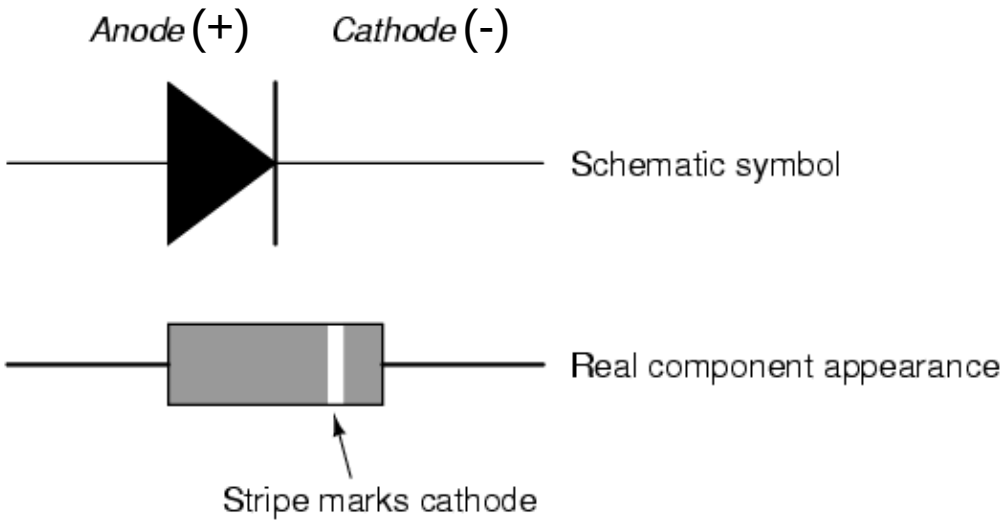
A motor may require a lot of voltage. The Arduino board can only output 5 volts.

Motors

Motors Concerns

Preventing Back Voltage

When a motors stops or changes direction it can send current the wrong way through a circuit. To prevent this you put a Snubber Diode in your circuit.

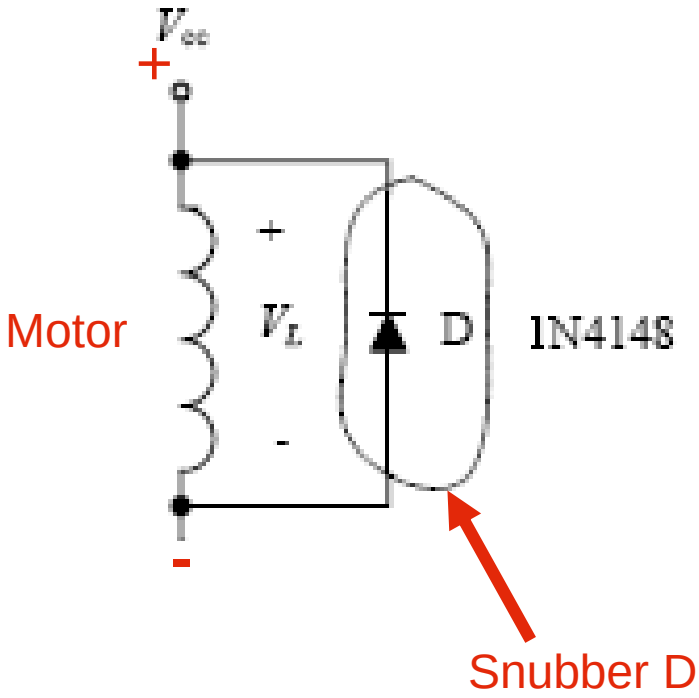


Motors

Motor Concerns

Preventing Reverse (kick) Voltage

A Snubber Diode is a diode places inversely parallel to an inductive load. This provides an alternate path for the current and will protect your valuable microcontroller from surges.



Motors

Motor Concerns

High Current/Voltage Loads

Sometimes a motor, light or other inductive load requires more voltage or current than a microcontroller can provide.

This requires 2 discrete power sources in a circuit.

The Microcontroller output voltage (0V-5V) can be used to proportionally change the voltage supplied to the motor (0V – 12V) by a separate power supply.

To do this we use a transistor.

Motors

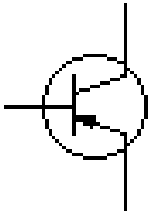
Transistors

Transistors are used to switch or amplify voltage

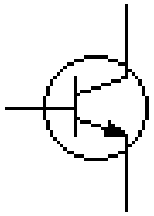
A transistor has three connections, the base, the collector, and the emitter.

There are different types of transistors. Some look different, some act different. Two common types of bipolar transistors are the PNP and the NPN types.

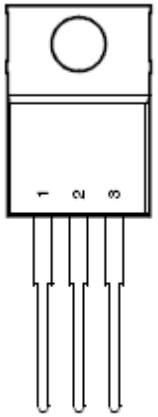
PNP = Switch that is normally closed
NPN = Switch that is normally open



PNP



NPN



B C E

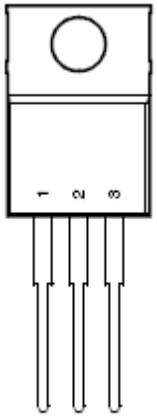
Motors

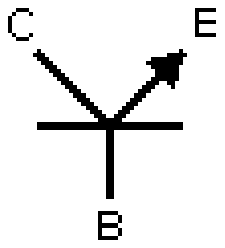
Transistors

Transistors are used to switch or amplify voltage

For our circuits we will use NPN (normally open) transistors.

The Base is connected to a control voltage (e.g. Microcontroller output)
The Collector is attached to the negative terminal of the motor (or light)
The Emitter is connected to the common ground





NPN
Schematic
Symbol

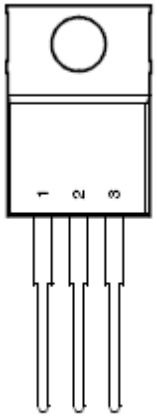
Motors

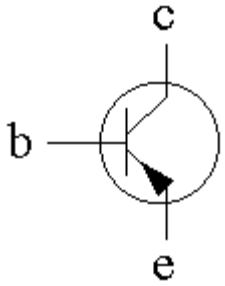
Transistors

Transistors are used to switch or amplify voltage

The other type of transistor is the NPN (normally closed) transistor.

The Base is connected to a control voltage (e.g. Microcontroller output)
The Collector is attached to the positive terminal of the motor (or light)
The Emitter is connected to the power source



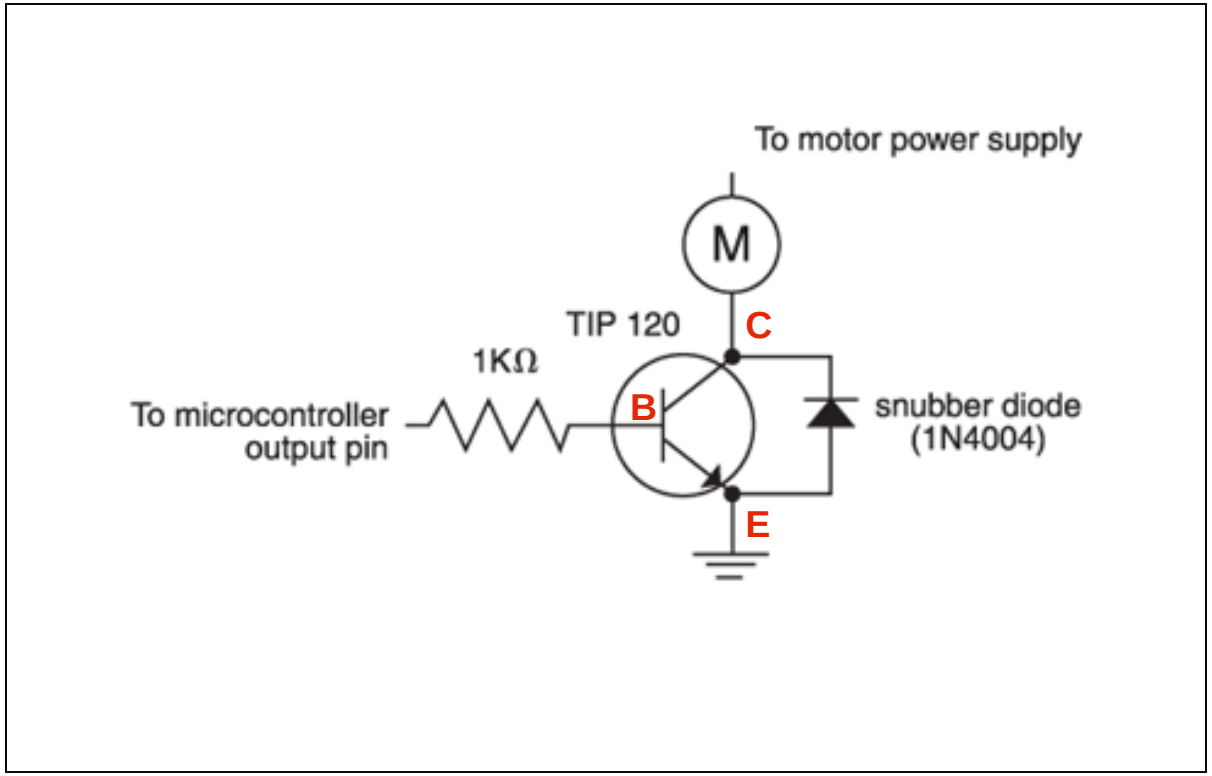


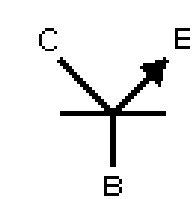
PNP
Schematic
Symbol

Motors

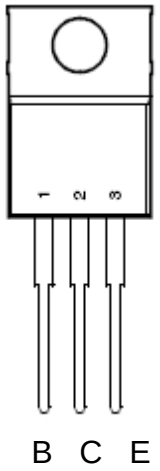
Transistors

Motor Control Circuit





NPN Schematic Symbol

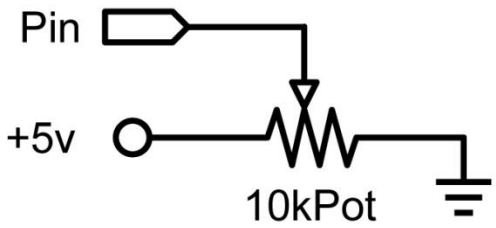


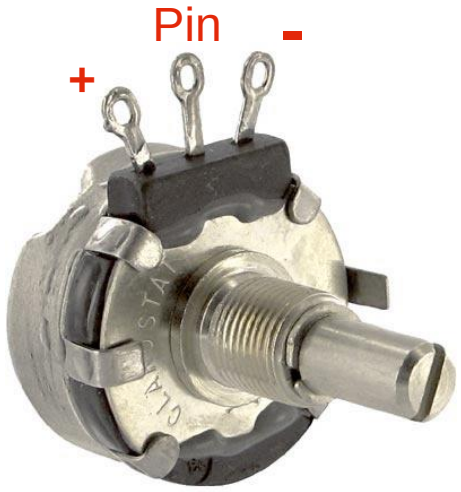
Motors

Potentiometer

Variable Resistor

A potentiometer is a rotary voltage divider. It functions like a variable resistor changing the output voltage when the knob is turned.



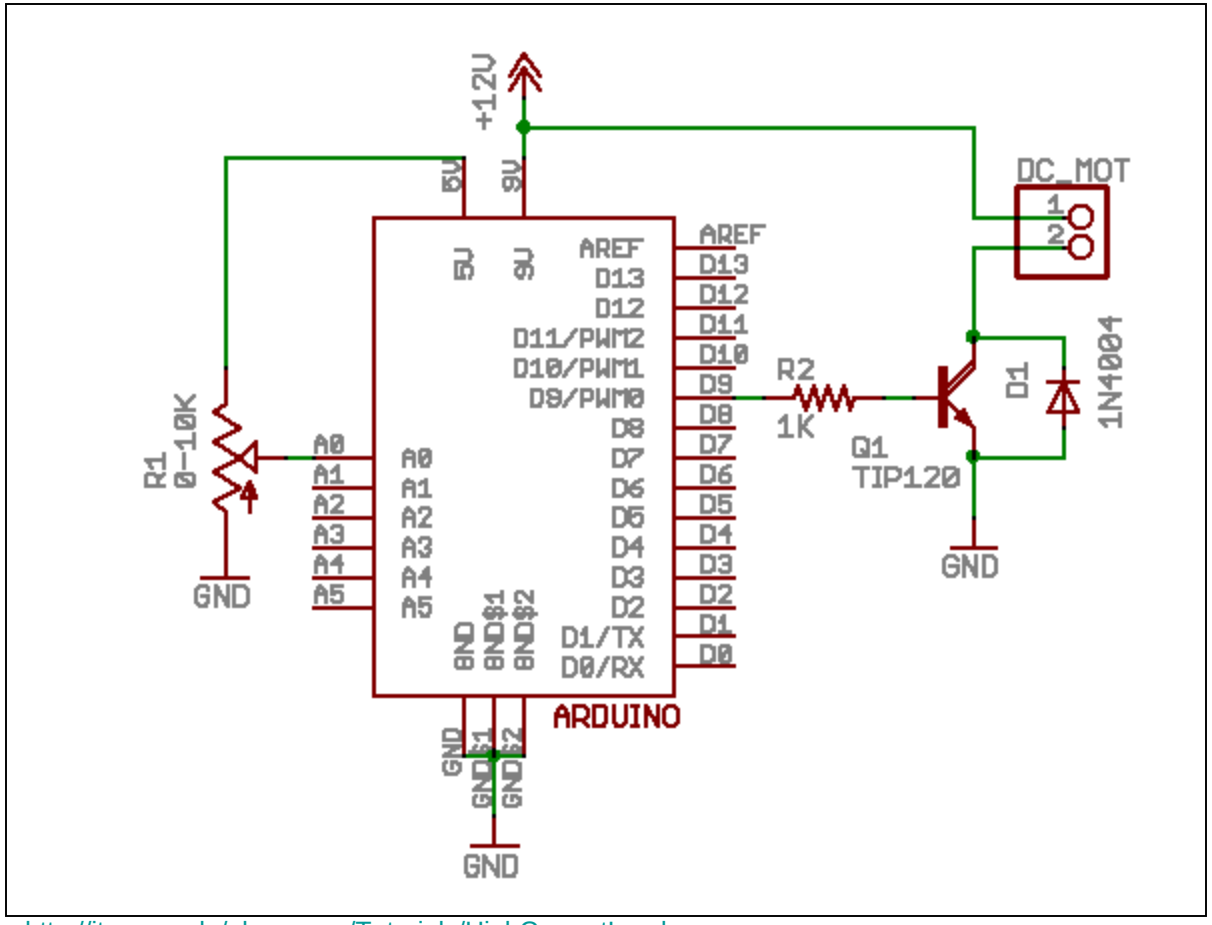


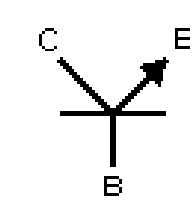
http://itp.nyu.edu/physcomp/Tutorials/HighCurrentLoads

Motors

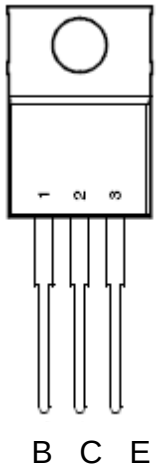
Transistors

Example: Use a potentiometer to control motor speed





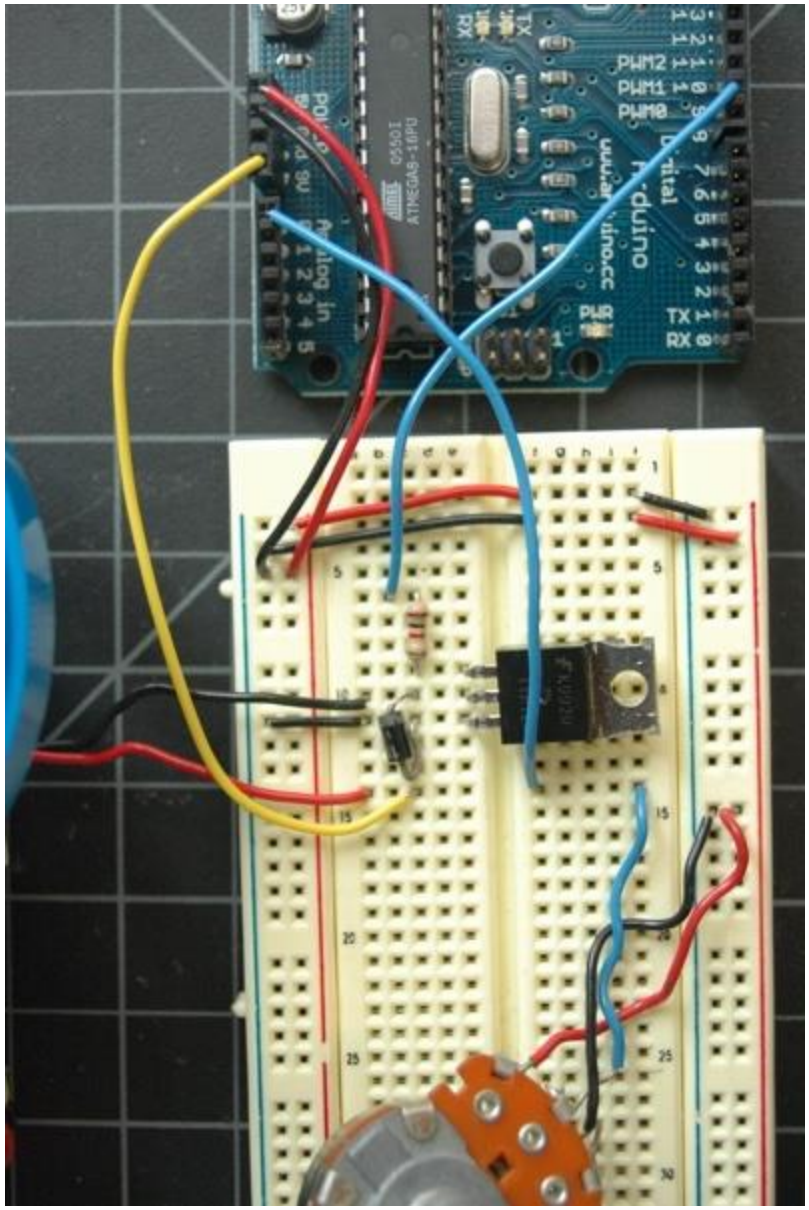
NPN Schematic Symbol



http://itp.nyu.edu/physcomp/Tutorials/HighCurrentLoads

Motors
Transistors

Arduino Circuit



Source: http://itp.nyu.edu/physcomp/Tutorials/HighCurrentLoads

Motors

Transistors

Example: Use a potentiometer to control motor speed

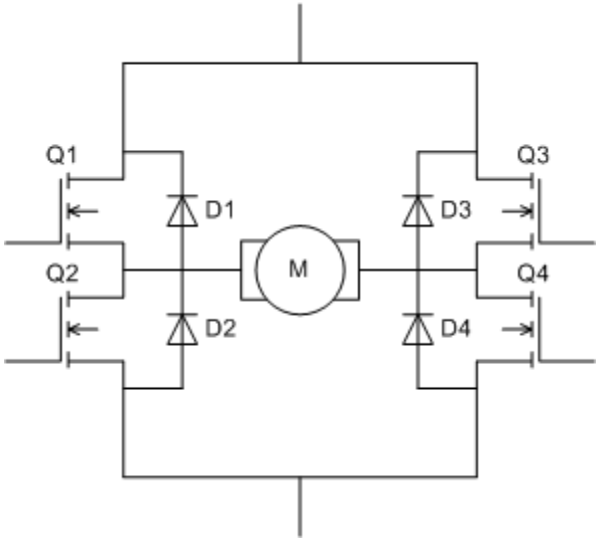
```
int potPin = 0;           // Analog in 0 connected to the potentiometer
int transistorPin = 9;    // connected to the base of the transistor
int potValue = 0;         // value returned from the potentiometer

void setup() {
  // set  the transistor pin as output:
  pinMode(transistorPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // read the potentiometer, convert it to 0 - 255:
  potValue = analogRead(potPin) / 4;
  Serial.println(potValue, DEC);
  // use that to control the transistor:
  analogWrite(9, potValue);
}
```

Motors

H-Bridge Circuits

Bidirectional Motor Control

Transistors are great if we only need to move a motor in one direction.
H-Bridge Circuits allow us to change a motor's direction on demand.
An H-Bridge is just 4 transistors wired together.

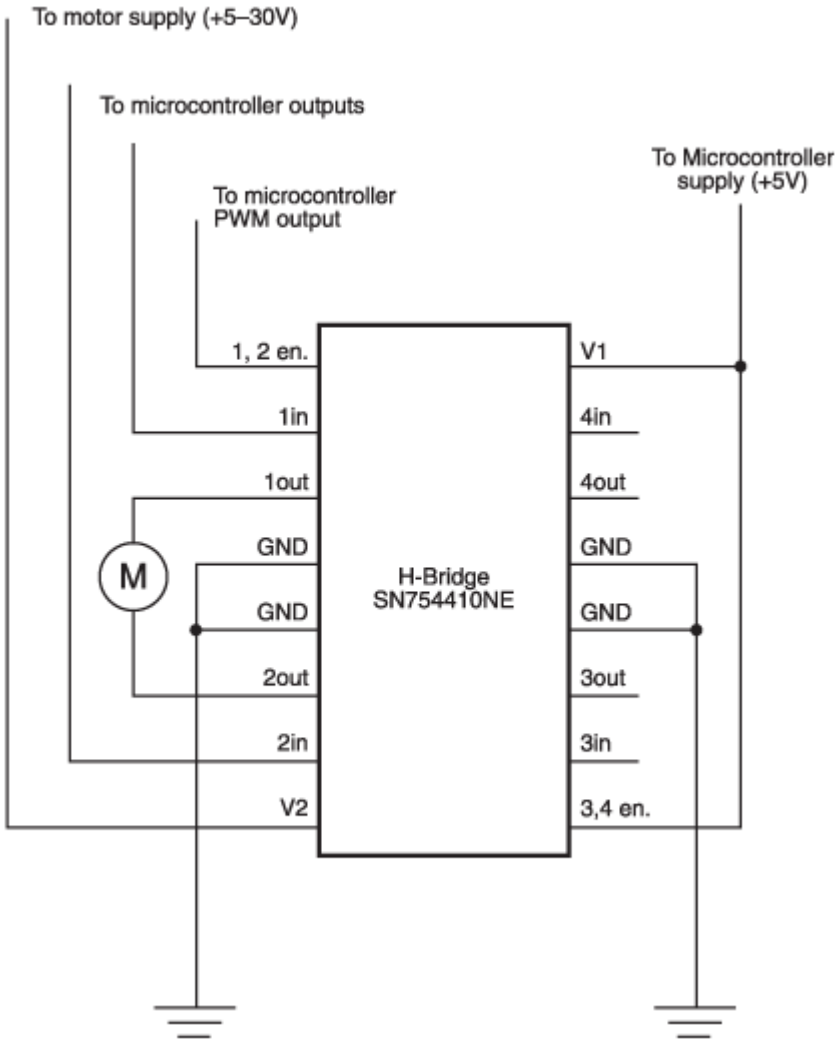


H-Bridge Circuit

Motors

H-Bridge Circuits

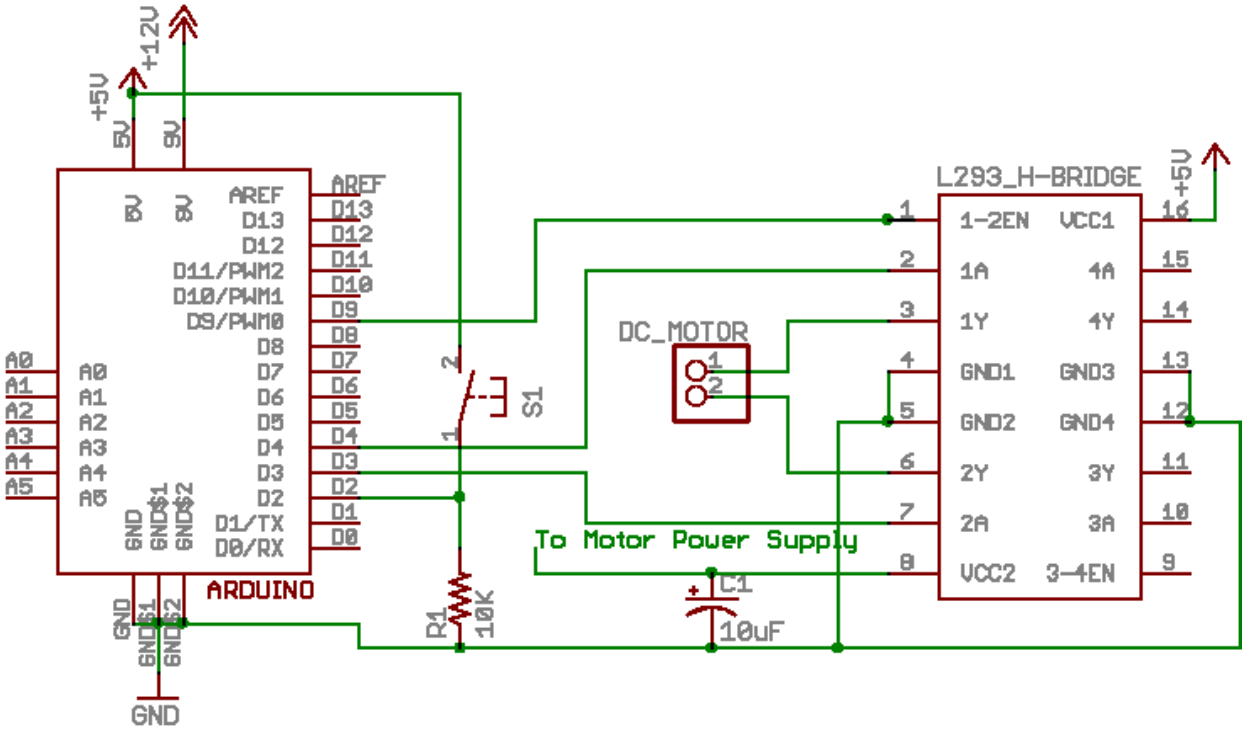
Wiring Scheme



Motors

H-Bridge Circuits

Connecting to an Arduino



Source: http://itp.nyu.edu/physcomp/Labs/DCMotorControl

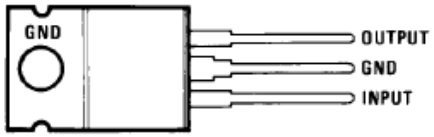
Motors

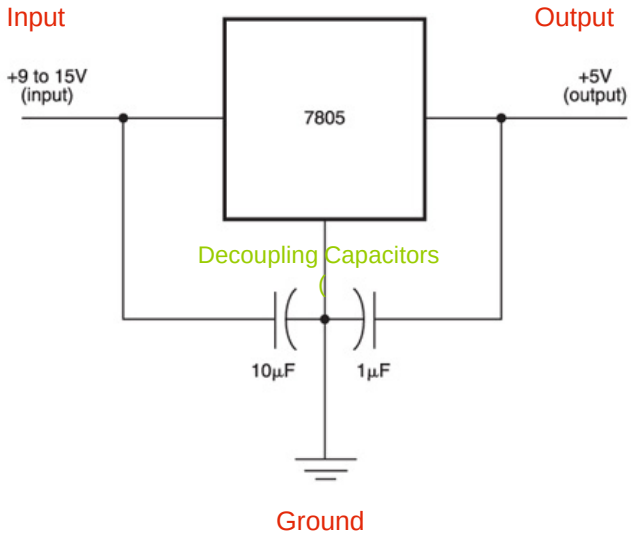
Voltage Regulators

Voltage regulators convert a higher voltage to a lower voltage.

They come in a variety of types including 3.3V, 5V and 9V.

If they get hot you probably have wired something wrong!!!!





Motors

Servo Motors

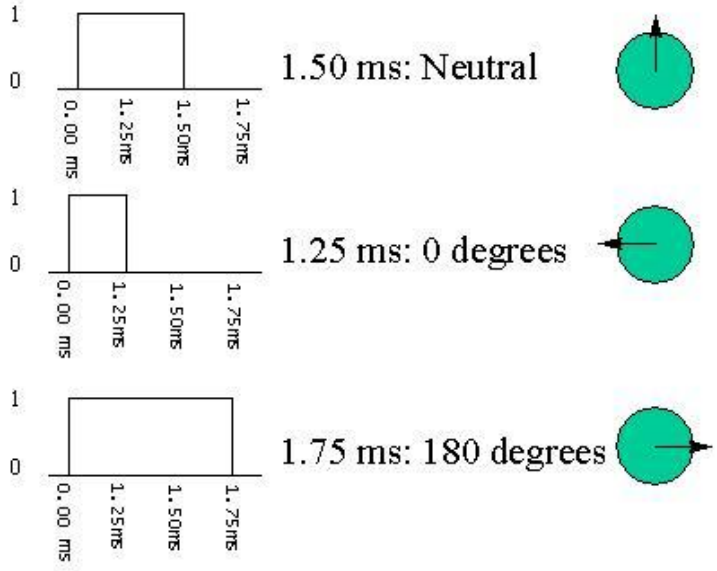
Servo Motors Use Pulse coded modulation to change a motor's position from 0 -180. When a motor receives a specific pulse it will move to the corresponding angle and maintain that position as long as it continues to receive the same signal. The signal must be resent at least every 20 ms.

Pulse Length

1.5 ms 90 Degrees (Neutral)

> 1.5 = closer to 180 degrees

< 1.5 = closer to 0 degrees.



Motors

Servo Motors

Arduino

The Arduino programming language has two methods for controlling servos.

1. Manually set the pulse lengths, min pulse, max pulse, etc.

2. Use the Software Servo library <servo.h> by Arduino Playground.

Motors

Servo Motors

Servo.h – Main Functions

attach(int)

Turn a pin into a servo driver. Calls pinMode. Returns 0 on failure.

detach()

Release a pin from servo driving.

write(int)

Set the angle of the servo in degrees, 0 to 180.

read()

return that value set with the last write().

attached()

return 1 if the servo is currently attached.

Motors

Servo Motors

Example: Control the position of a motor using a potentiometer

```
#include <Servo.h>
Servo servo1;

int potValue = 0;                          //Value of signal coming from Potentiometer

void setup() {
  pinMode(2,OUTPUT);
  servo1.attach(2);
}

void loop() {

  potValue = analogRead(0);                 // Get value of potentiomete
  potValue  = map(potValue, 0,687,0,180);   //scale the value to fit between 0-180, the range of the servo
  servo1.write(potValue);                   //Set the servo's new position
  servo1.refresh();
  delay(1);
}
```

Acitivites

Design a circuit that uses a potentiometer to control a DC Motor. You should use a 12 volt adapter as a separate power source for the motor and an NPN transistor to switch the current. You will need to use a voltage regulator to bring the voltage supplied to the motor down to 5v.

Step 1: Attach Potentiometer (and test functionality using serial window)

Step 2: Build Voltage Regulator circuit to power motor (test with multi-meter)

Step 3: Finish circuit by adding transistor and motor elements.

Design a circuit to control a servo motor. Use the potentiometer from your previous circuit to control the position of the servo. Use the AC adapter to power the Arduino board to ensure that you have enough power for the system and to make it more portable. *You can use the software servo library from Arduino Playground.*

Step 1: Attach AC adapter to Arduino Board and adjust pin for external power.

Step 2: Make sure potentiometer is working and check output range.

Step 3: Convert incoming value from the potentiometer to a value between 0 – 180 to correspond to the angle of the servo

Step 4: Attach servo