# Zen and the Art of the DC Motor

Skye Sweeney

Skye@fll-freak.com

# Categorizations of motors

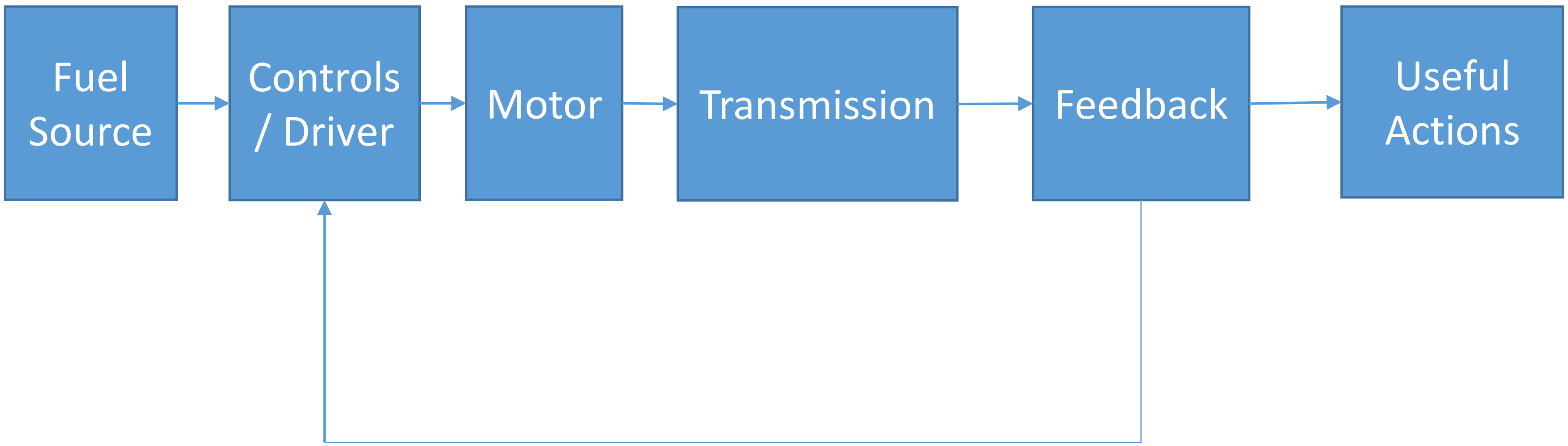
- Liquid fuel
  - Gasoline, ethanol, nitro methane, …
- Solid fuel
  - Coal, wood, gunpowder, …
- Pneumatic (Pressurized gas)
- Hydraulic (Pressurized fluid)
- Electric
  - **DC** **<- This talk**
  - AC
  - Static electricity
- Nuclear!

- Continuous Rotary
  - Spinning axle
- Linear
  - Back and forth like a piston
- Jet
  - Blast of compressed gas or fluid

# A motor is just part of a system

- Motors by themselves are not that useful
  - A car engine may need
    - Transmission
    - Control system (throttle, ECU, diagnostics, …)
    - Speedometer, odometer
  - An electric motor may need
    - Possibly a gear box
    - A motor driver
    - Encoder

# System Diagram



# Hard to separate the parts

- An RC servo has a motor, gear box, controls, and feedback in one package
- We may think of it as a motor, but it really is a full system
- So when talking about motors you may also need to talk about the other parts as well

# Motors for robots

- We tend to avoid
  - Combustion engines due to noise and fumes.
  - Jet engines
  - Hydraulic and pneumatics due to having to drag around a compressor

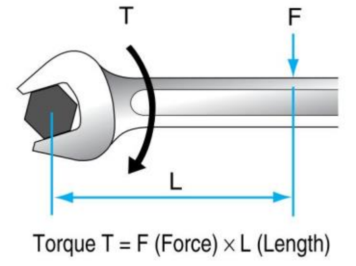
- For the most part **we like electric motors** and DC is the dominant type
  - DC is easy to source from a battery
  - But there are a **gazillion types** of DC electric motors that are used

# So how do you choose?

- You need match the type of motor to the needs of your robot
  - Voltage, power, speed, size, accuracy, cost, …
- So you need to understand the pros and cons of each type of system
- You need to find not just a motor but a system that will talk to your processor
  - May find it difficult to find a driver to work with an Arduino for that fancy-pants motor you got surplus

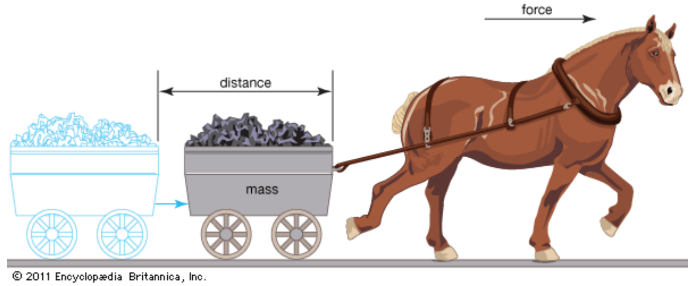
# A Few Definitions

# Torque



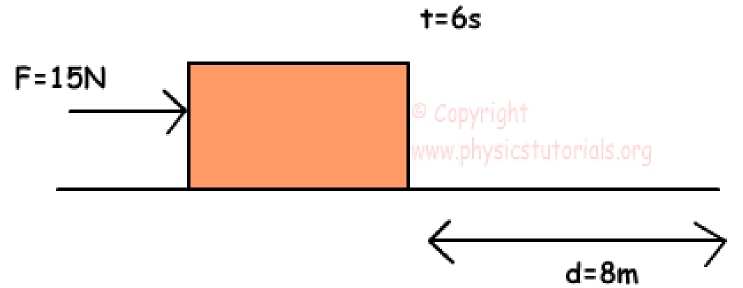

- The amount of twisting force applied/supplied by an object
- Measured in distance-force units (e.g. inch-pounds or Netwon-meters)
- When you have a tight screw you need to apply a lot of torque to remove it
- Torque **has nothing to do with speed**. You can have lots of torque at slow or at fast speeds

# Work



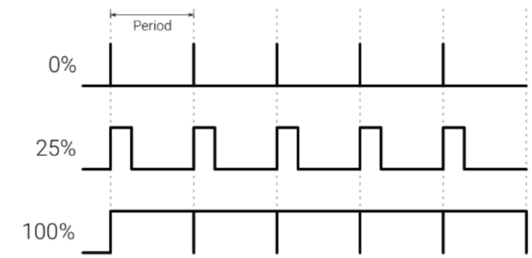

- The amount of effort needed to do a job
- In linear systems, work is the product of the force needed to move an object times the distance the object is moved
- In angular systems it is the product of torque needed to turn the item times the angle moved
- Measured in distance-force units (e.g. inch-pounds or Newton-meters)
- Moving the piano twice as **far** needs twice the work even if you take ten times as long to move it

# Power




- Rate at which work is done. (Work/time)
- I can plow an acre of land in a day with one mule or that same acre in ½ day with two mules
- In the second case I am doing the work twice as fast so I had twice the power available
  - Two mulepower rather than one
- For spinning things, Power = Torque * Rotation speed
- Units are in Watts (Netwon-meter/second)
- Power has little to do with motor speed

# Pulse Width Modulation




- A signal that turns on and off in varying ratios
- Devices receiving such a signal might see the average time on. Like dimming an LED
- Your electric oven is PWM controlled to get intermediate temperatures
  - It comes on for a while then off. The thermal mass of the oven averages out the oscillations
  - You can **sometimes** do the same thing with motors
- Not to be confused with the so called PWM of an RC servo. More on this later.

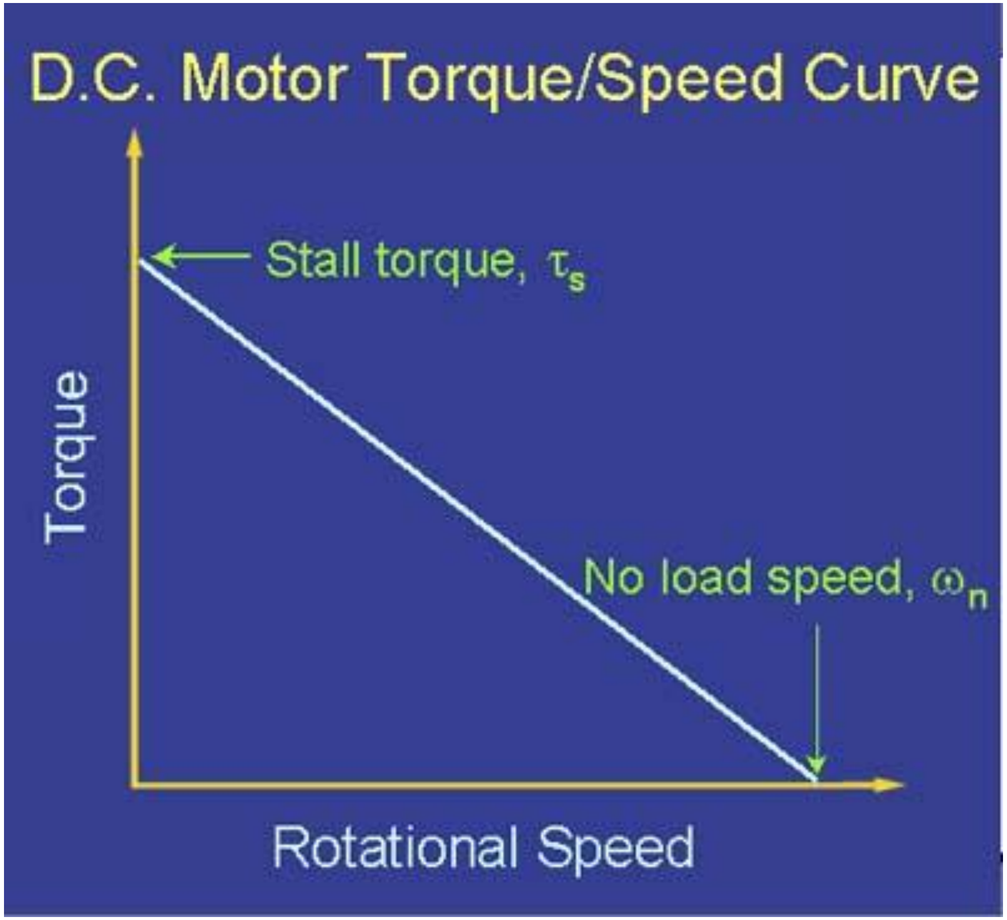
# Brushed DC motors

The “Dime a Dozen” motor

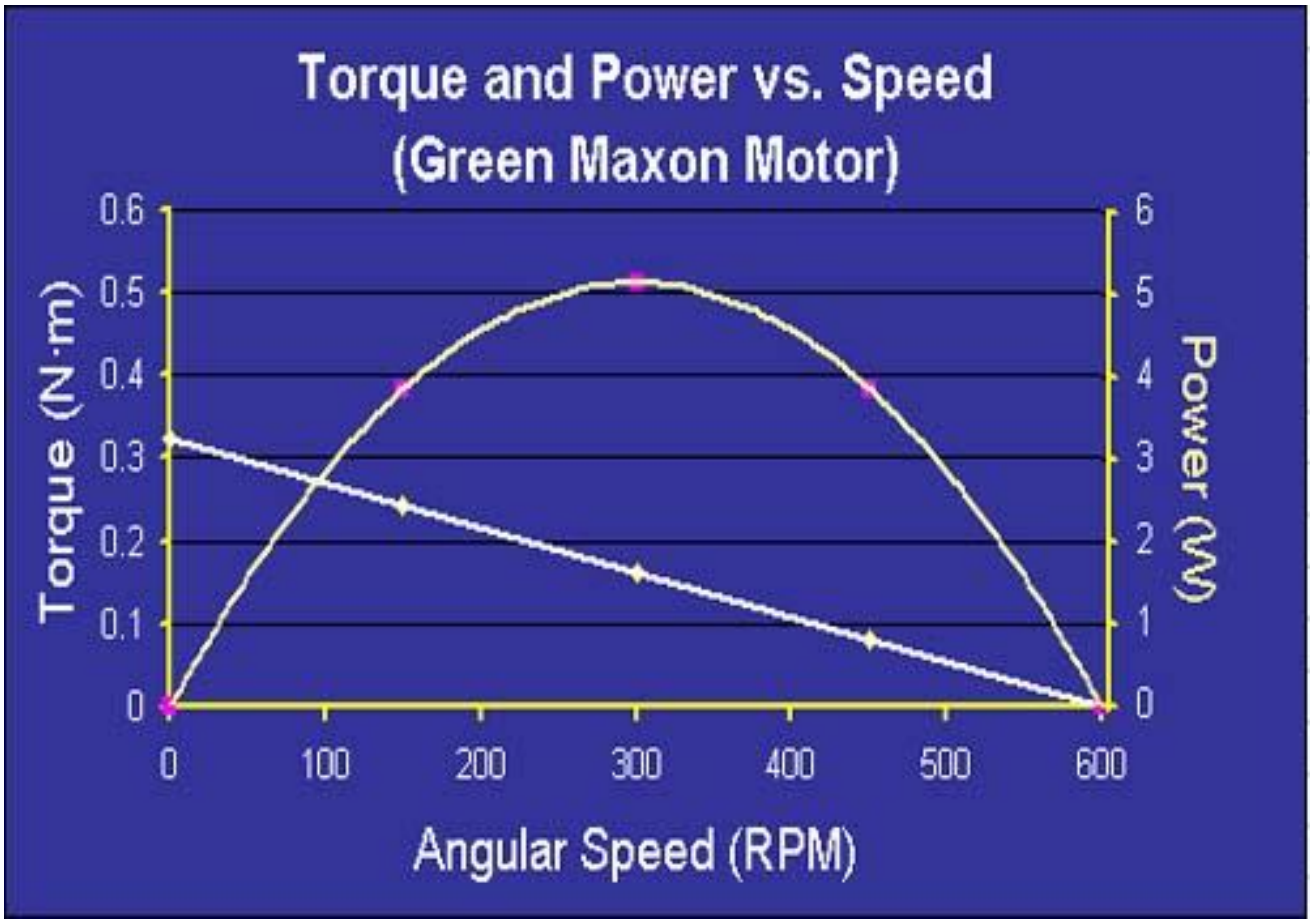
# Simple Brushed DC Motor

- Spinning caused by a rotor (the moving part) being alternately repelled and attracted to a stator (the stationary part)
- The stator is often a permanent magnet
- The rotor is often a set of electromagnets
- The alternating action is created by a commutator
  - Commutator swaps polarity on windings using brushes
- Motor can be reversed by changing polarity of power

# Torque vs Speed



# Mechanical Power



# Electrical Power

- Each motor will have a spec for its top power
- Mechanical Watts are equivalent to electrical Watts
  - Assuming a 100% efficient motor
  - Better to assume 50%
- Therefore:
  - Watts = Voltage * Current * ½ = Torque * rotation rate

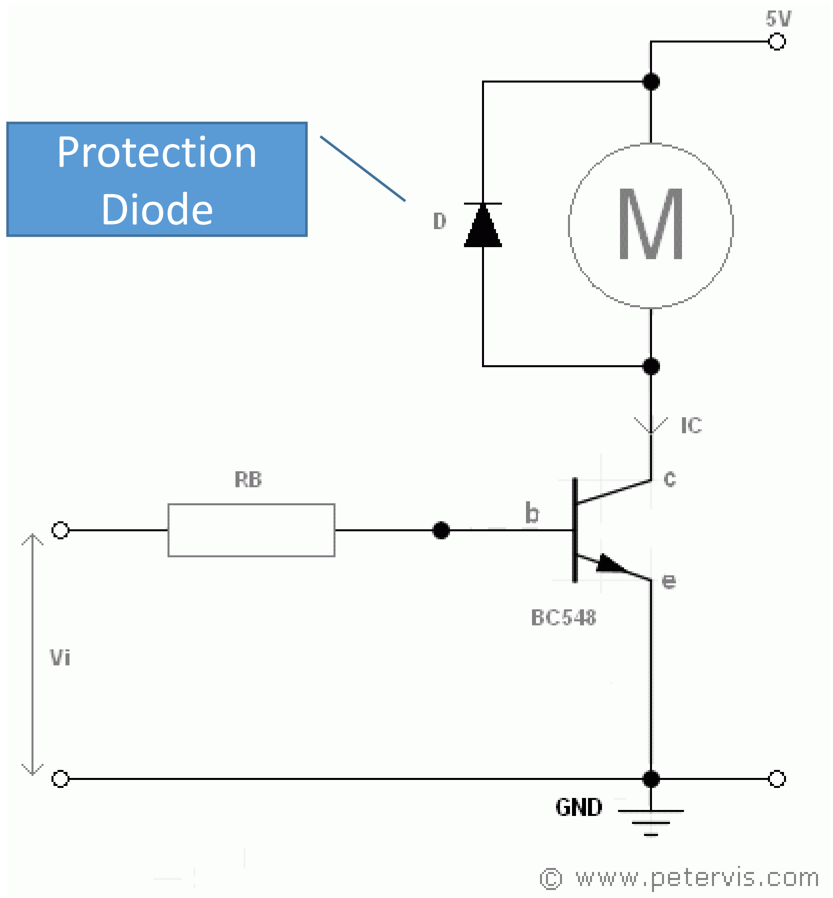
# Generators

- Many motors act as generators if you spin them
- This can become important when designing a driver. More in this later!

# Controls

- Most microprocessors will not be able to run a motor directly
  - IO pin is 5V (or 3.3V), motor needs 1.5V
  - IO pin is 5V (or 3.3V), motor needs 24V
  - An IO pin on a processor is limited to perhaps 10 ma.
  - All but the smallest motors will draw at least that without a load
  - Need a “driver” between motor and processor

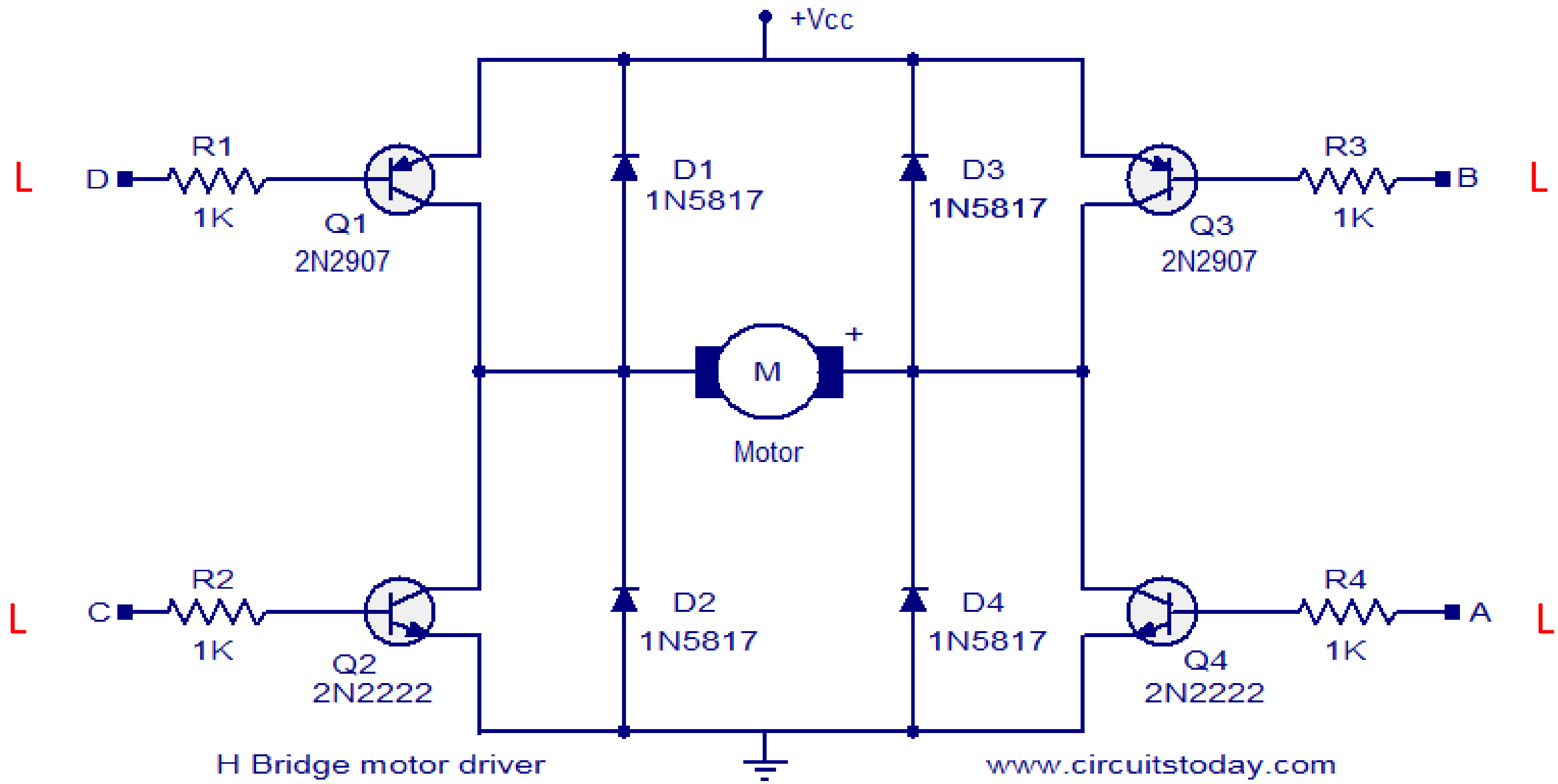
# Basic solution



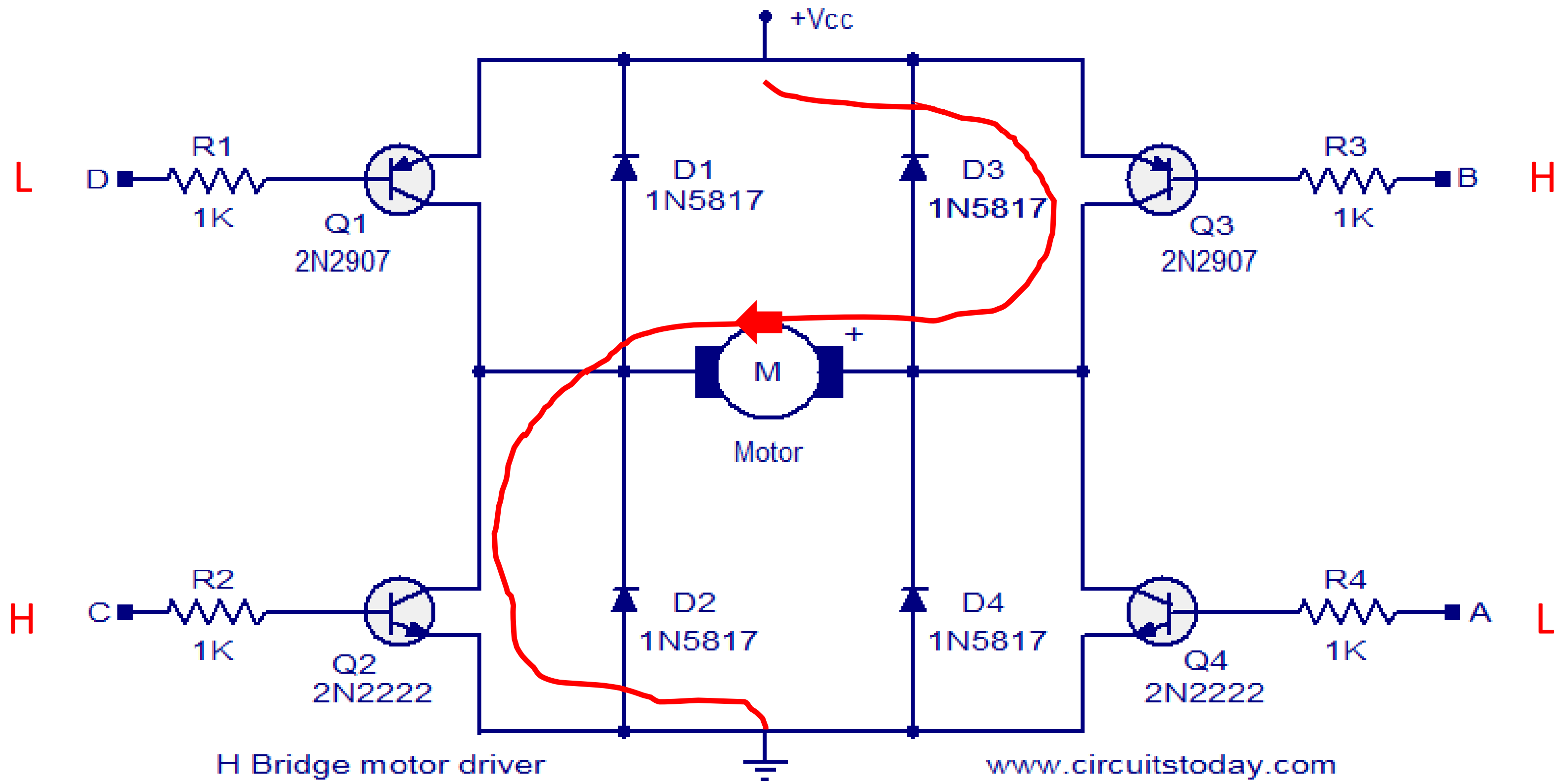
# The H Bridge

- The basic solution can’t address reversing the direction of the motor
- To solve this, the H bridge circuit was invented
- Here one signal is used to control FWD/REV and the other ON/OFF
- There are many dedicated H-bridge chips
  - Much easier than using discrete transistors as many of the likely issues will be solved for you.

# H Bridge (coast/off)

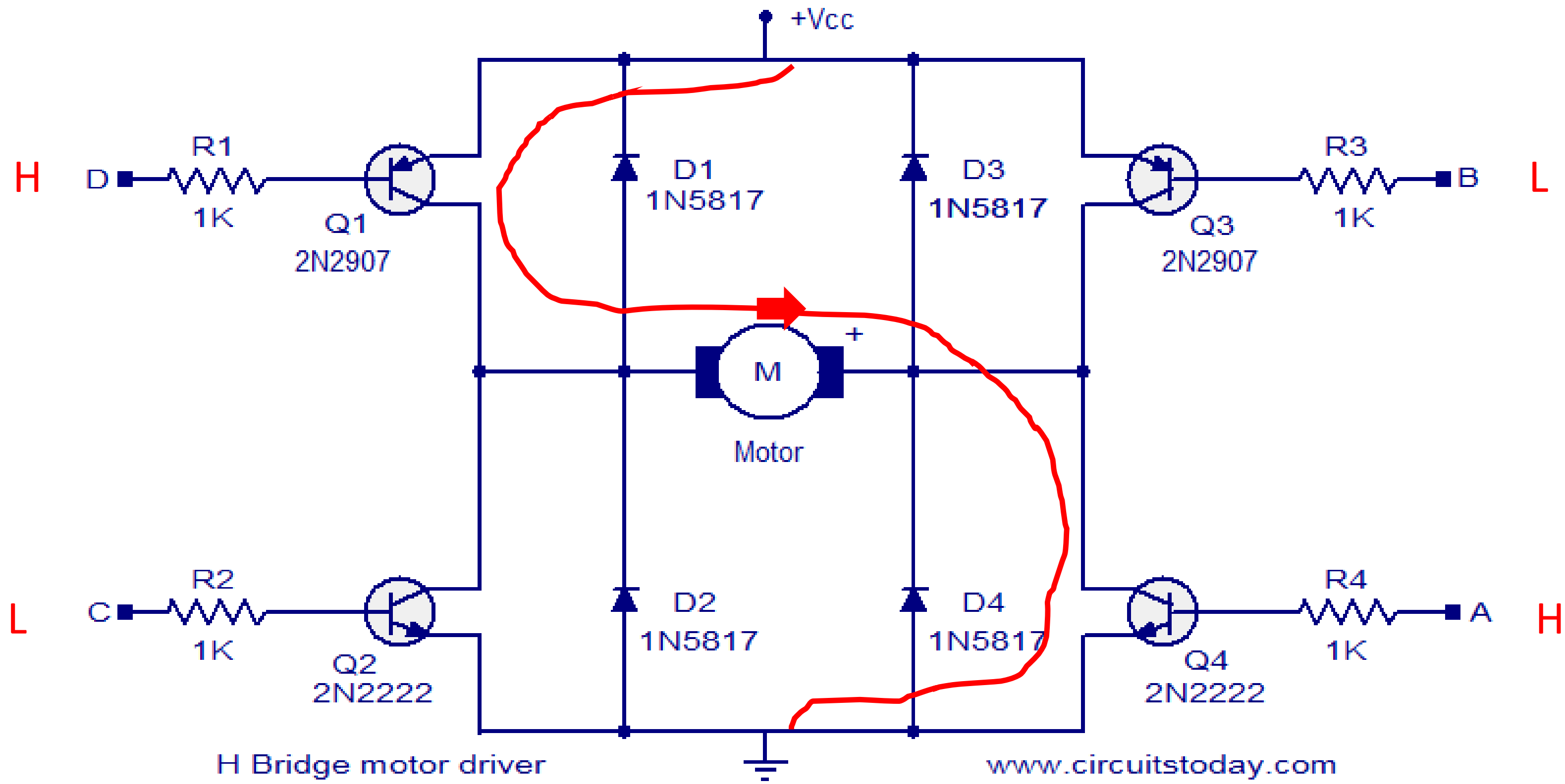


# H Bridge (Fwd)



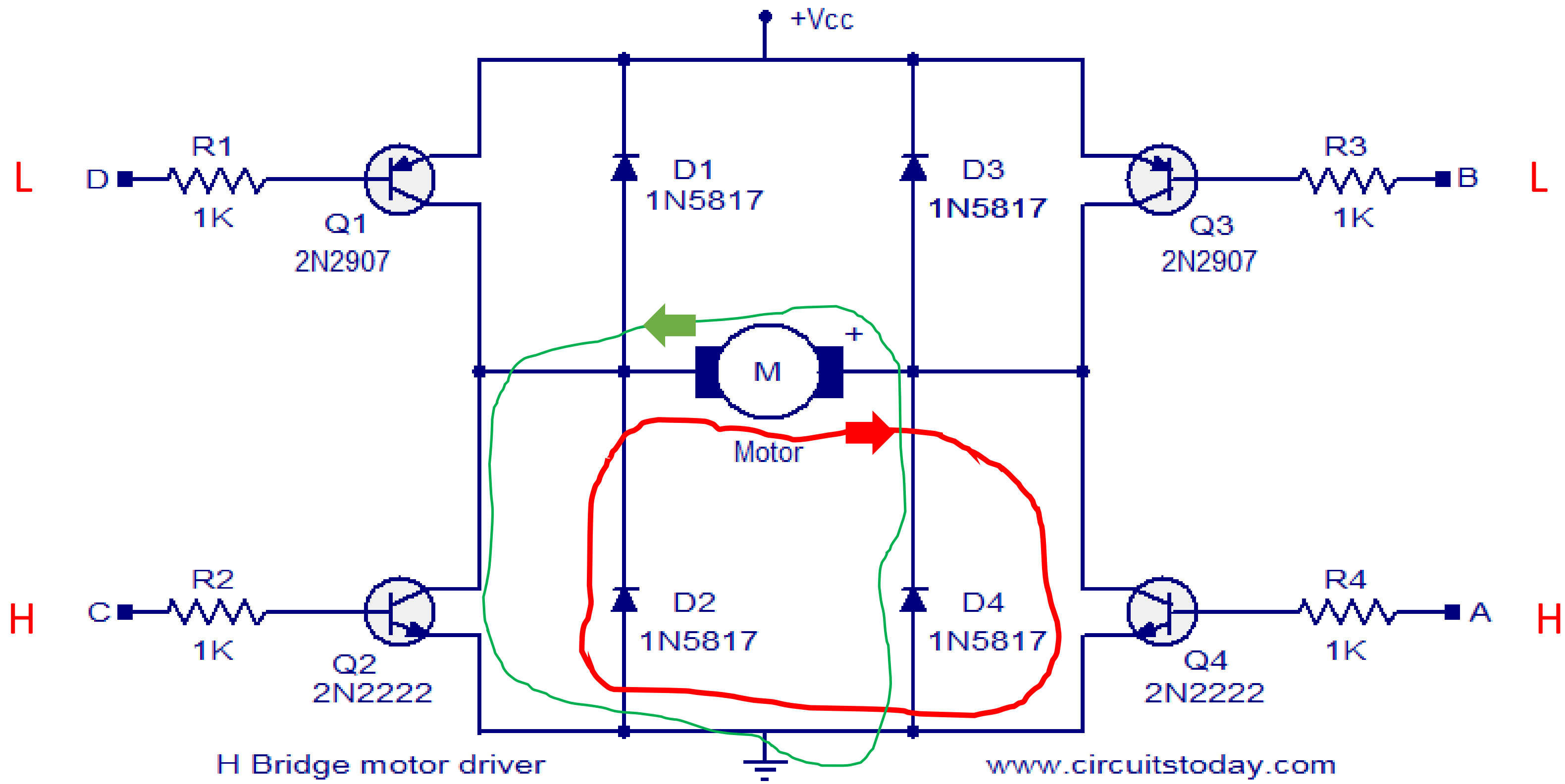
+Vcc
L
D
R1
1K
Q1
2N2907
D1
1N5817
D3
1N5817
R3
1K
Q3
2N2907
B
H
M
+
Motor
H
C
R2
1K
Q2
2N2222
D2
1N5817
D4
1N5817
R4
1K
Q4
2N2222
A
L
H Bridge motor driver
www.circuitstoday.com

# H Bridge (Rev)



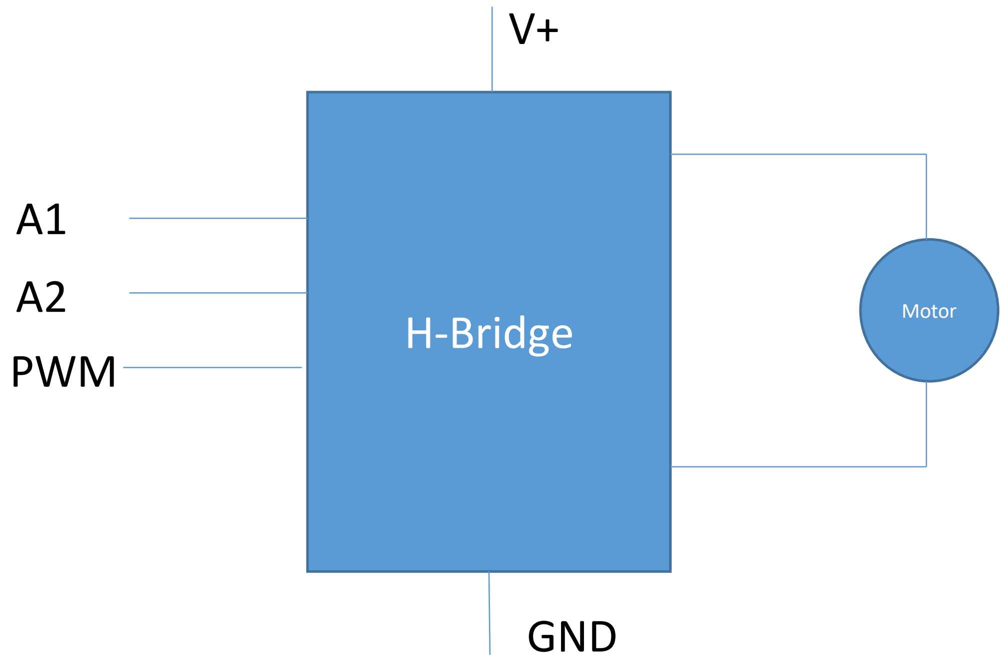
+Vcc
H
D
R1
1K
Q1
2N2907
D1
1N5817
D3
1N5817
R3
1K
B
L
Q3
2N2907
M
Motor
+
L
C
R2
1K
Q2
2N2222
D2
1N5817
D4
1N5817
R4
1K
A
H
Q4
2N2222
H Bridge motor driver
www.circuitstoday.com

# H Bridge (Brake/off)



# Simpler H-Bridge

| A1 | A2 | Action |
|---|---|---|
| 0 | 0 | Float |
| 0 | 1 | Fwd |
| 1 | 0 | Rev |
| 1 | 1 | Brake |



# H Bridge Chips and BOB

- SN754410
- L298 (Dual)
- DRV8829
- IRF7862
- L293DNE
- TB1622
- …

(Sparkfun, Adafruit, Pololu, …)

# Feedback

Supports most motor types

# Feedback

- So we know how to control a brushed DC motor
- By adjusting the voltage directly or via PWM, we adjust the power provided to the motor
- With different powers the motor will find different operating points
- Normally, with more power the motor will go faster
- But what about hills, carpets, or low batteries that also affect the speed?
- Since no two motors are the same, how do you go straight?

# Sensors

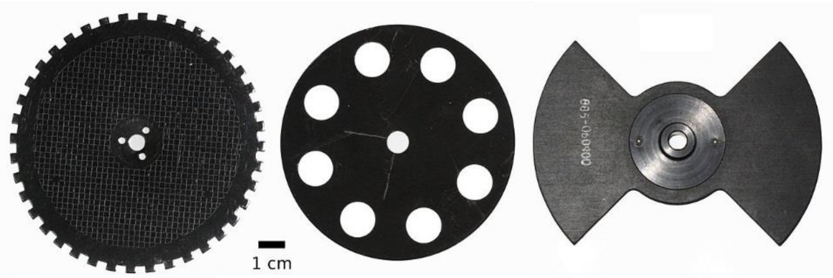
- Direct
  - Encoders (absolute or relative)
    - Optical (slit wheels, grating, …)
    - Mechanical (cam and switch, …)
    - Magnetic (Hall effect)
    - Simple/quadrature (quadrature allows for direction)
  - Back-EMF
    - Measures the voltage generated by motor when it free spins between PWM pulses.
- Indirect
  - GPS
  - Doppler radar
  - …

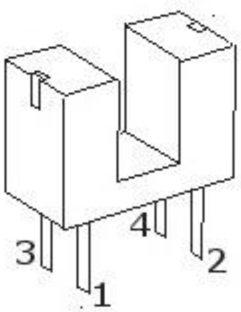
# Encoders

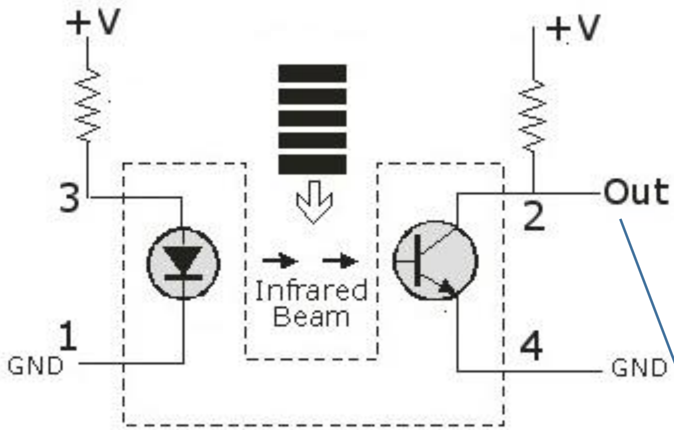
- Some motors come with integrated encoders
- There are different types of encoders
  - Single pulse train
  - Double pulse train (Quadrature)
  - Analog voltage in proportion to speed

# Simple Optical Encoder

- Chopper wheel
- IR LED
- IR photo transistor



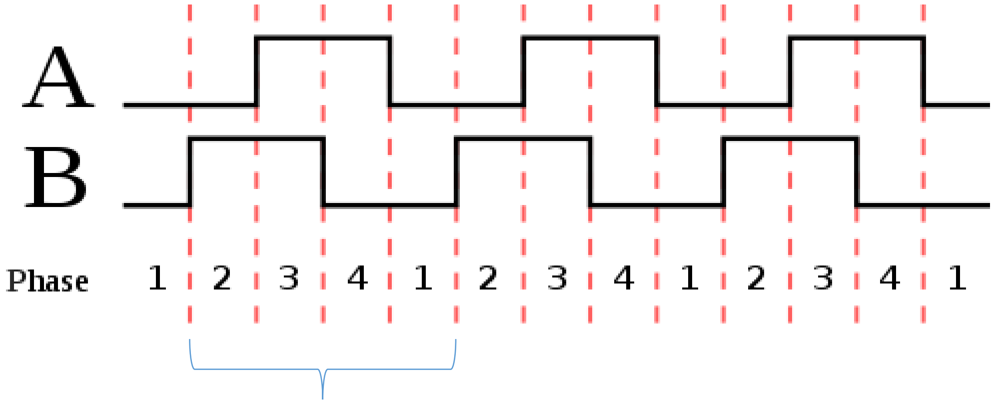





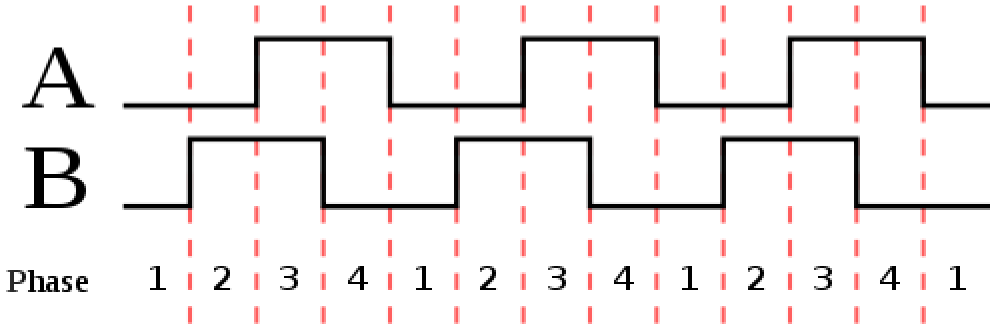
Connect to interrupt pin

# Quadrature

- Two ‘Simple’ encoders 90 degrees out of phase
- Can be used to determine
  - Speed		Rate of pulses
  - Direction		Phase between two encoders
  - Distance		Count the pulses



# Quadrature (finding direction)




Left to Right

- $B\uparrow$ and $A_l$
- $A\uparrow$ and $B_h$
- $B\downarrow$ and $A_h$
- $A\downarrow$ and $B_l$

Right to left

- $A\uparrow$ and $B_l$
- $B\uparrow$ and $A_h$
- $A\downarrow$ and $B_h$
- $B\downarrow$ and $A_l$

# Arduino Quadrature Code

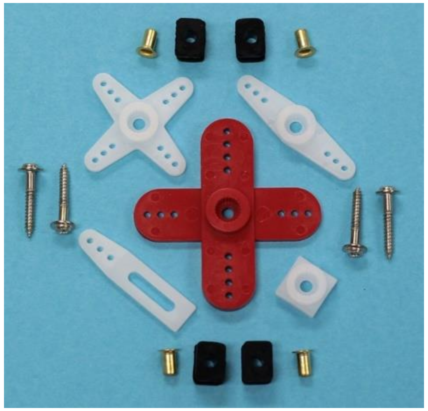
- Set A and B for rising interrupts
- ISR_A
    - Look at state of B
    - If (A_rising):
        - If (B): count++; else count --;
        - Set direction of  A interrupt to falling edge
    - Else:
        - If (B): count--; else count ++;
        - Set direction of A interrupt to rising edge
- ISR_B
    - Look at state of A
    - If (A_rising):
        - If (B): count++; else count --;
        - Set direction of B interrupt to falling edge
    - Else:
        - If (B): count--; else count ++;
        - Set direction of B interrupt to rising edge

- Simpler on processors that allow both rising and falling edge interrupts at the same time.

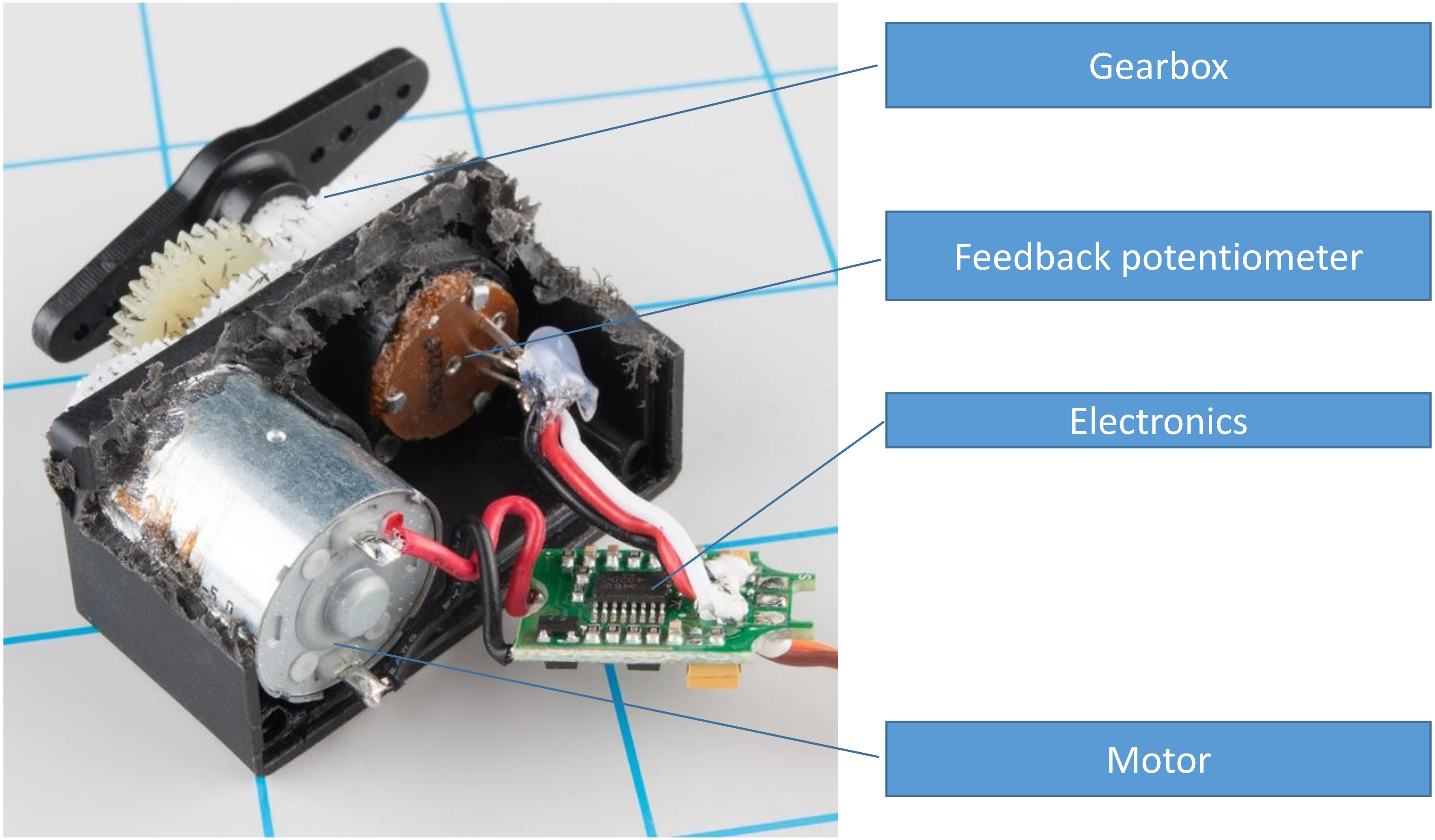
# RC Servos

# Analog RC Servos


- Readily available
- Cheap (as low as $4!)
- Lots of sizes (sugar cube and larger)
- Lots of accessories (www.servocity.com)
- Most run on 5 to 6 volts
- Use a digital pulse to command
- Highly integrated motion system.

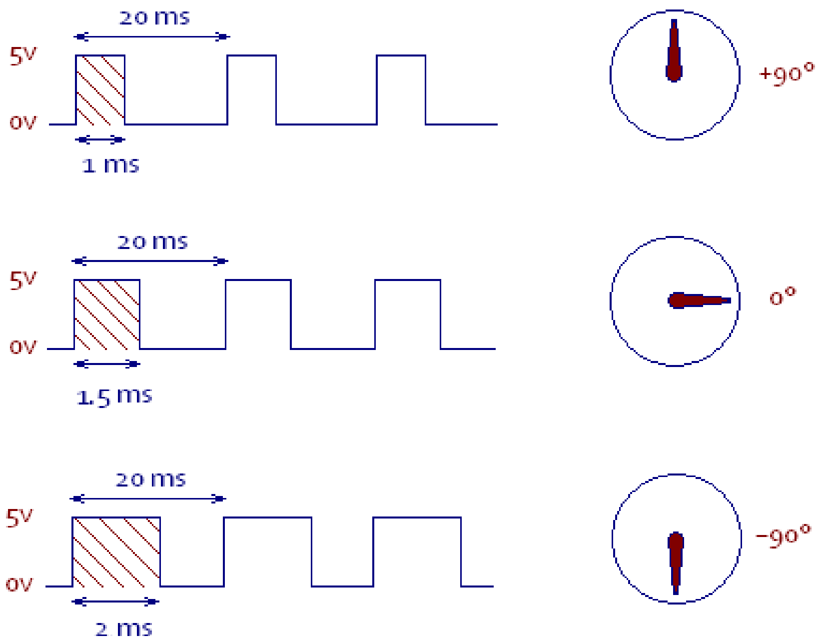


# RC Servo Internals



# Pulse Width Encoding (PWE)

- The width of the pulse dictates the position
- Pulse is repeated about 50 times a second
  - This drives the PID loop



# RC Servo Signals

- Three wires
  - Power (normally from 4 AA) 5-6V
  - Ground
  - Signal (digital pulse in the 5V0 range)
- Different pinouts based on manufacturer!

# Controlling an RC servo

- Need to generate the correct pulse train
- Easy on Arduino. Use the ‘servo’ class (not the PWM class)

```
#include <Servo.h>

Servo myservo;

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
    myservo.write(posInDeg);
}
```

- On other processors, you can often leverage a timer/counter peripheral

# RC Servo Speed Hack

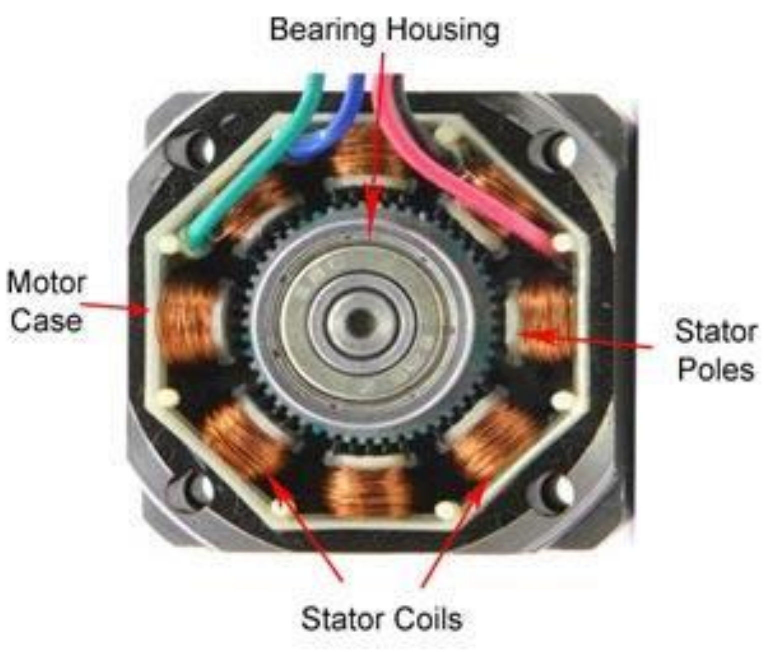
- Many servos can be modified (or purchased modified) to provide a speed feedback system
  - Pulse width dictates direction and speed of motor
  - **Very coarse control**
    - A few percent of the pulse width is max to min ☹
- Hack consists of
  - Removing the potentiometer and replacing it with two equal sized resistors
  - Cutting away the stop on the output gear that prevents 360 degree rotation
  - Google is your friend

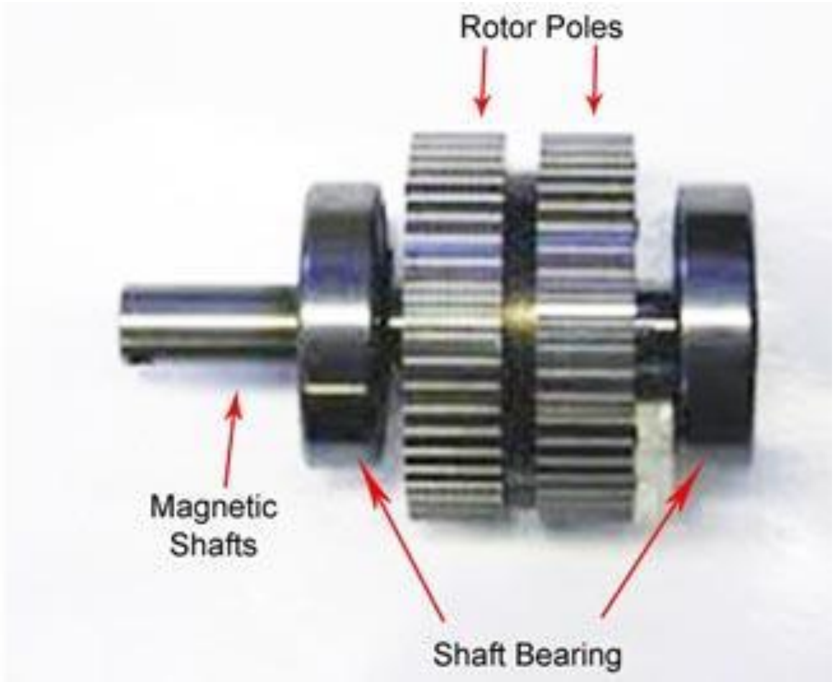
# Stepper Motors

# Stepper motors

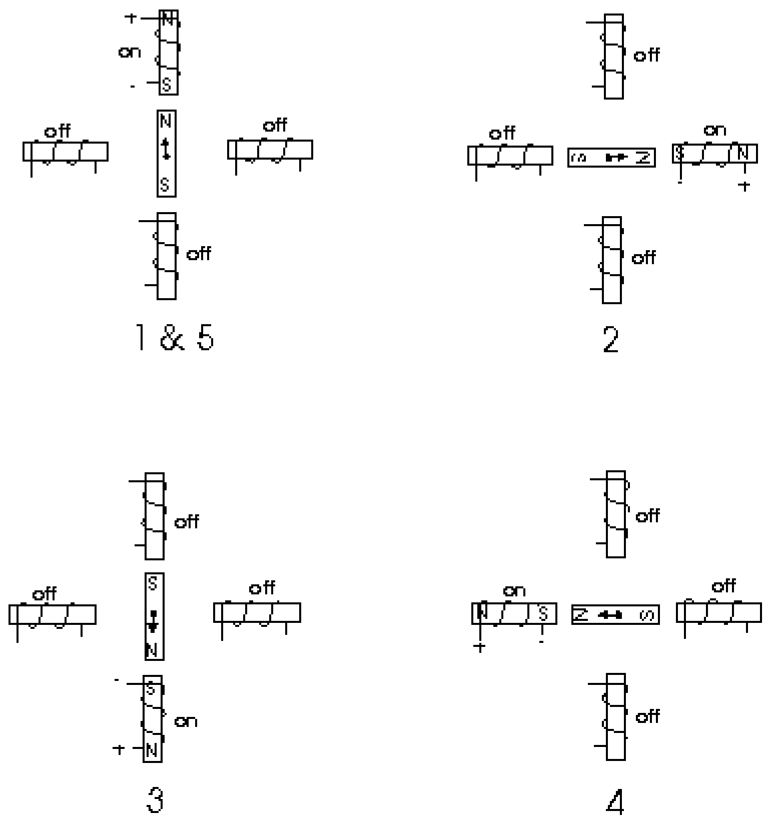
- Similar to permanent magnet brushed DC motors
  - Uses attraction and repulsion of electromagnets to turn
- Main differences
  - Rotor is a permanent magnet
  - You must provide electronic commutation
  - Shaft does not spin freely. It cogs

# Stepper Motor Internals



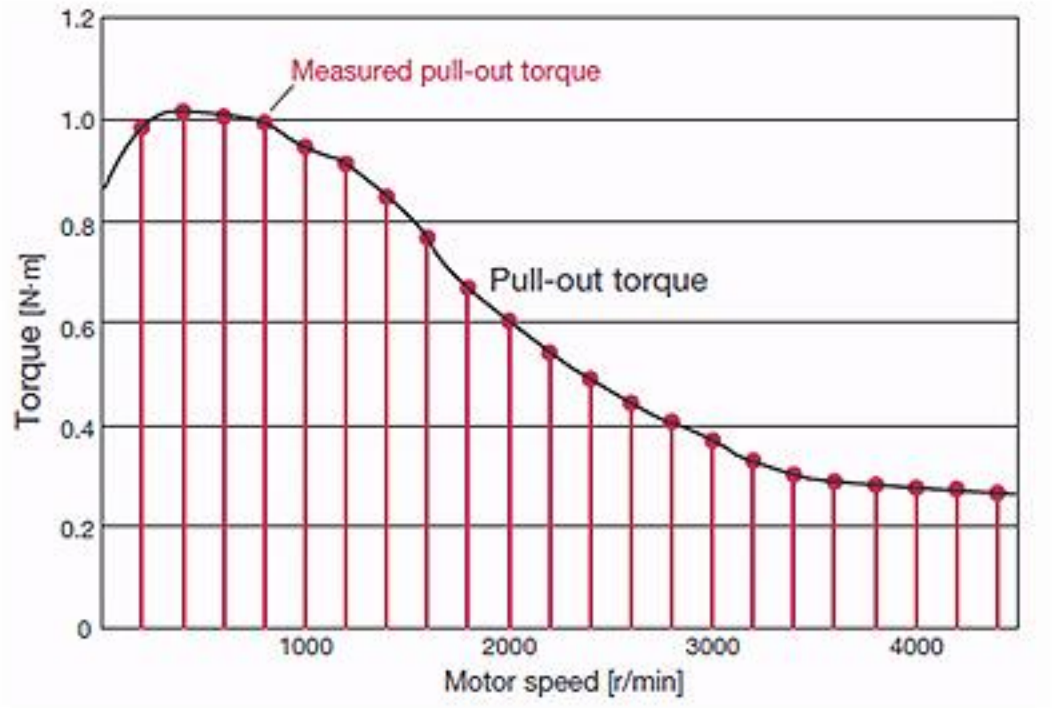
# Basic Operation



# Classification

- Physical size
- Voltage rating
- Torque and speed ratings
- Step size
- Coil arrangements
    - Bi-polar
    - Uni-polar (good for beginners)

# Torque Curve



# Stepper motor drivers

- Many cheap drivers available as chips or BOBs
- The RepRap StepStick clone drivers are $5!
- Most drivers take two digital signals
  - Step
  - Direction
- Most drivers can be configured for fractional stepping. (1/2, 1/4, 1/8,…)
  - Half stepping has two adjacent coils on at 50%. This drags to rotor to a mid point

# What are they good for?

- **Accurate** and **repeatable** but not overly fast jobs
- Floppy disk head positioning
- Printer paper handling
- 3D printers
- Sonar scanners
- Camera gimbals (although RC servos are simpler)
- Odometry competition robots

# What to watch out for

- Loss of synchronization
  - Too much torque or stepping too fast and the motor will not step when it should. You will think you have turned the motor when you have not. Without external encoders you have no way to know.
- Heat. Steppers can get very hot if you push them
- Poor coasting. The cog friction prevents smooth rotation

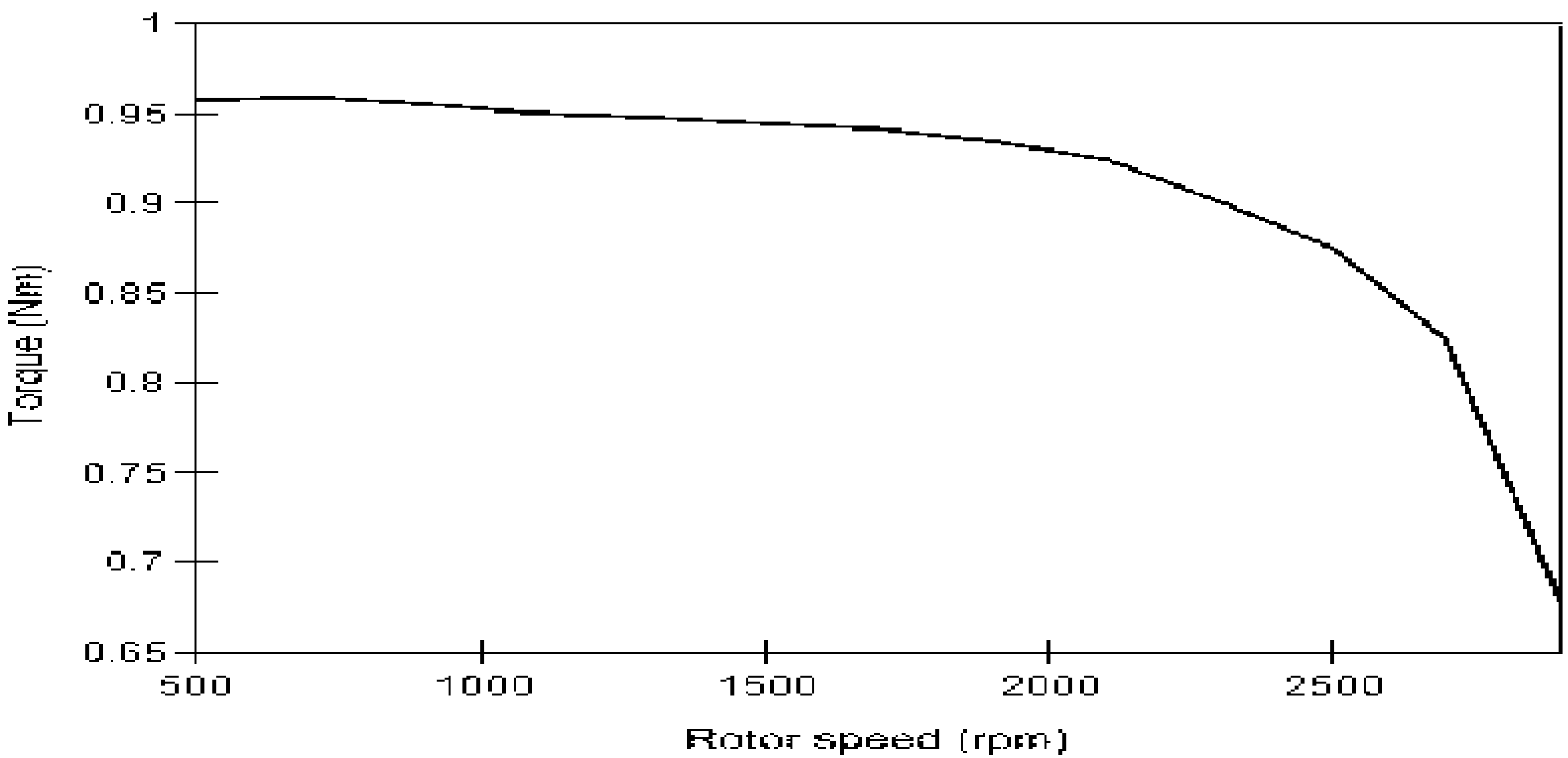
# Brushless DC

# Brushless DC motors

- These are becoming very popular due to drones and RC planes
- They are a hybrid between a stepper and a brushed motor. You provide commutation but they have no steps
- High speed, flat torque curve, low weight
- You can get a motor and a controller for $20 on eBay
- Can find surplus ones in old CD players and other equipment

# Torque Curve

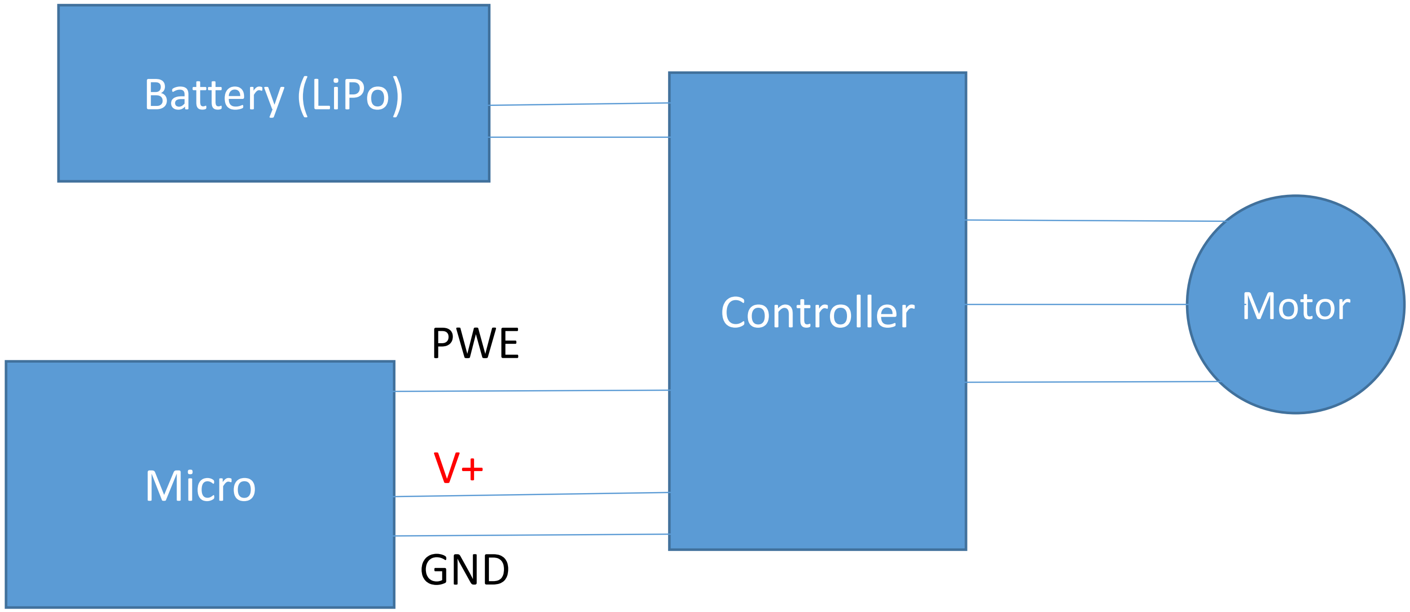
- Very flat torque curve.



# Driving a BLDC

- They require three sinusoidal voltages 120 degrees out of phase
- The faster the sinusoid the faster the motor as it drags the stator around.
- Difficult drive directly from a micro
  - Some specialized micros have direct support
- You really need a driver
- Simplest driver is an RC BLDC controller
  - It takes the same ‘PWM’ signal as an RC servo

# RC BLDC controller hookup



# Notes on using RC BLDC controllers

- These tend to implement safety interlocks to prevent the motor from coming on unexpectedly
- They insist that the PWE signal be present before power is applied to the motor
- They also insist that the PWE signal be in a preprogrammed neutral position on power on
- Most controllers also have an onboard regulator to back feed 5V to the radio receiver
- Simple solution is to snip the 5V wire running from the controller back to your micro

# Offbeat Motors

# Muscle Wire

- Wire shrinks in length by ~3% when heated
- Can heat by applying a direct current through it
- As soon as it passes a magic temperature it snaps to the shorter length
- As it cools it reverts back to its original length
  - Normally takes longer to lengthen

- Slow to operate
  - Perhaps an activation every other second
- “No Fail” linear motion
- Power hungry

# Others

- Piezo
- Voice coil/ galvanometer

# General Considerations

# Pick the type

- If you need accuracy -> stepper
- If you need less expensive -> DCPM
- If you need speed and torque -> BLDC

# Spec the Motor

- Available voltage
- Mechanical torque and power needs
- Speed needs
  - Gearbox?

# General Considerations

- Heat dissipation
  - As motors get hot the efficiency and life goes down
- Ease of attaching wheels/accessories
  - Common shaft sizes and available parts
- Mounting style
  - Body clamp, end plate screws, …
- Bearing types
  - Ball, bronze bushing, plastic bushing
- Sealed versus open frame
  - Dirt and grime can kill a motor

# Sources

- Sparkfun
- AdaFruit
- Pololu
- ServoCity
- RobotShop
- Hobby Stores (HobbyKing, …)
- eBay
- AllElectronics
- Electronic Goldmine

# Questions