Raspberry Pi Workshop

Interfacing with Sound and Motion

Nolan Lem

Columbia Computer Music Center

Nov.15, 2024

Overview

- **Keeping it simple**, for people without any programming experience
- Try to allow some space for design, ideation, and concept creation
- Talk about my own approaches in designing sound and motion

Introductions

Design Criteria

Raspberry Pi (RPi)

Basic Programming

- Using terminal
- Python
- GPIO Pins

Physical Computing with Motors

- DC Motor
- Solenoid
- Stepper

Audio Output in RPi

Other Interfaces – Sensors, haptics, etc.

Time permitting, Design?

Introductions

Who are you?

What interests do you have in working with Raspberry Pi or technology in your art practice general?

Any ideas on what you would want to make using a raspberry pi?

Some Thoughts on Design and Compositional Process

Knowing what is out there changes what you think is possible and shapes creative process

Figuring out appropriate tools to accomplish task is half of the battle!

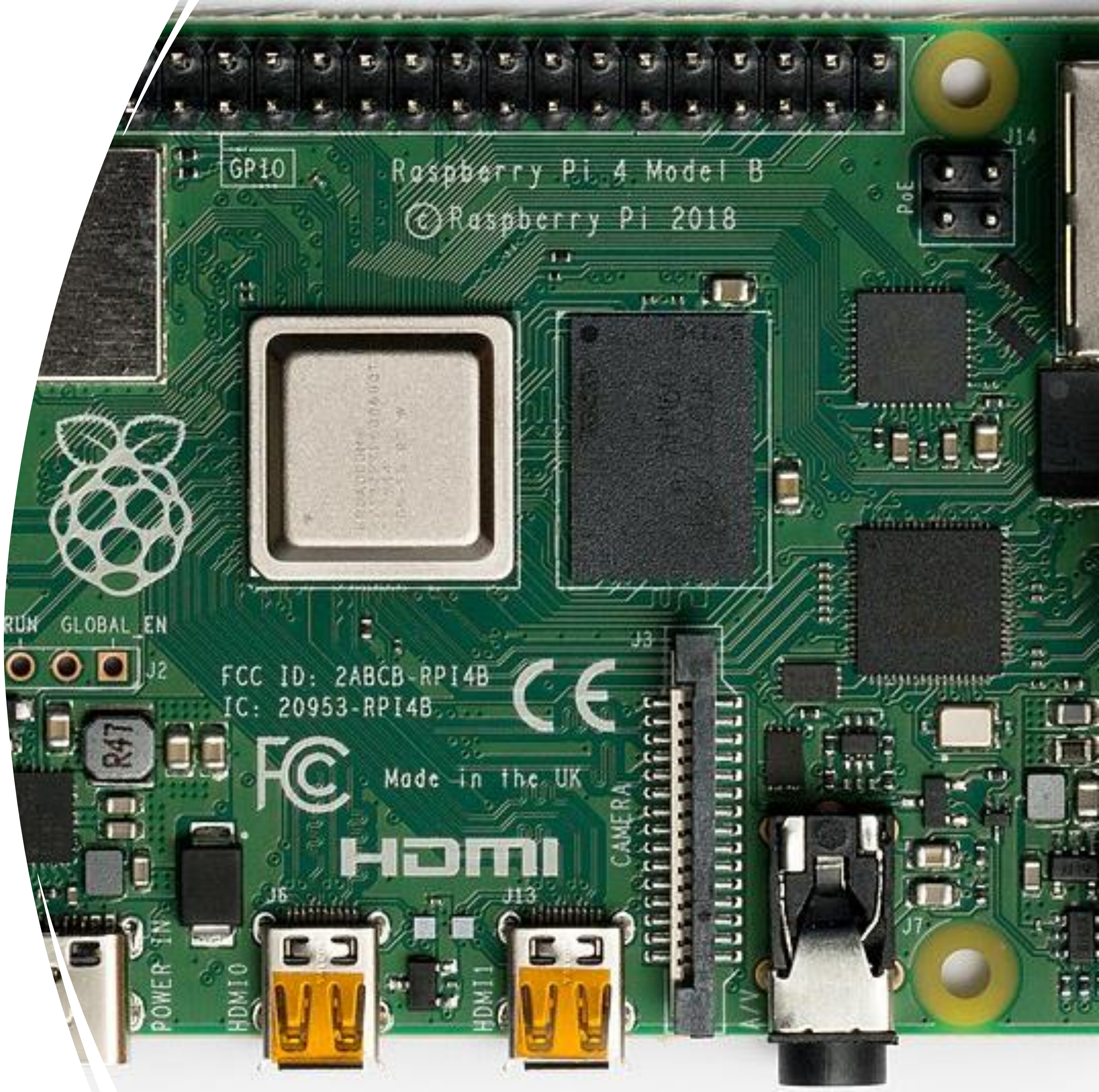
Complexity ramps up very quickly, but complex things can also be accomplished easily sometimes.

Working with Kinetic Media

- how to **produce desired sound** **AND** how to **create motion that produces sound**.
- technical complexity
- Expensive ☹
- Access to specialized tools, fabrication machinery, space, etc…
- Logistics
- Sellability?
- Can’t break!

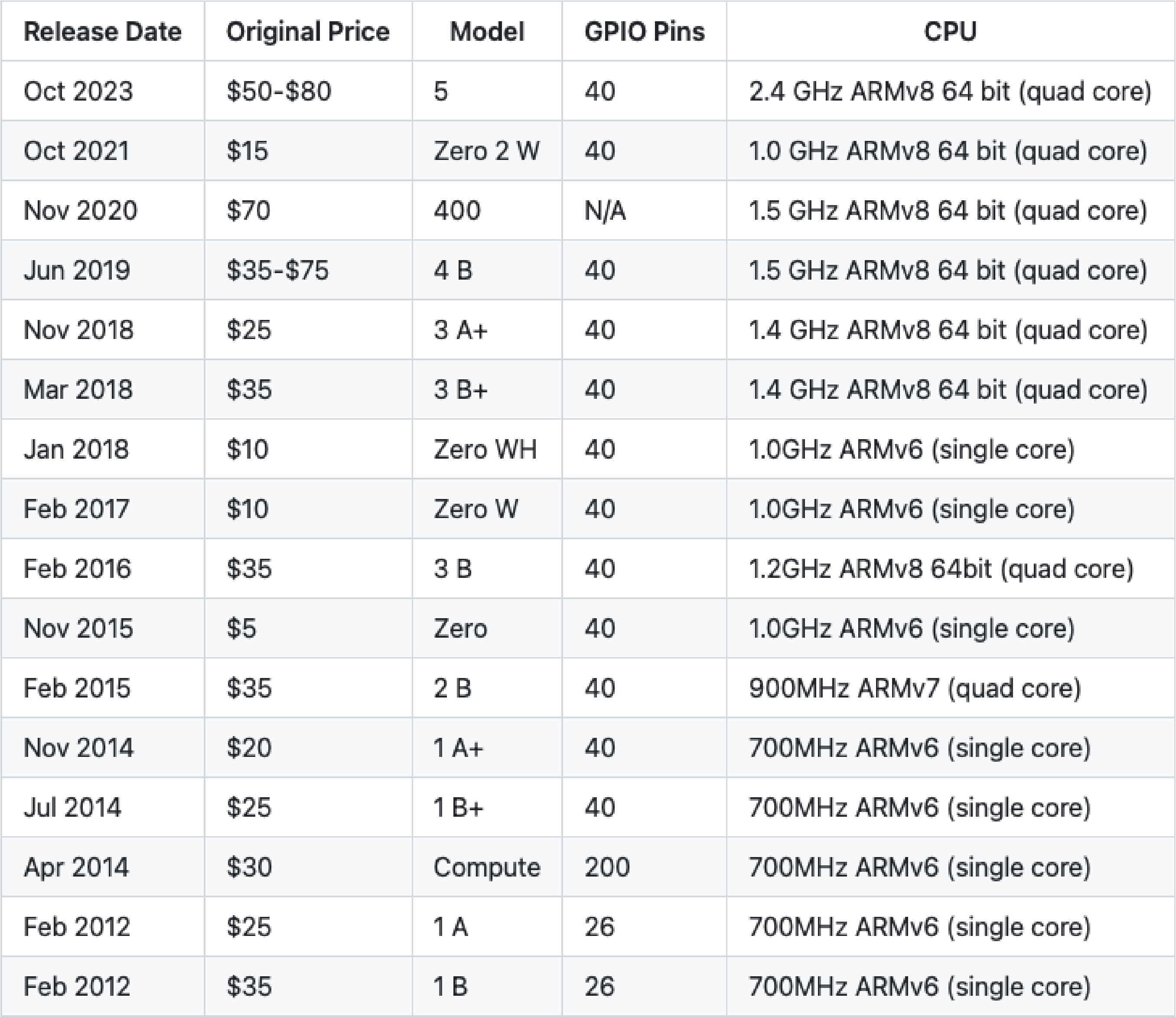
What is a Raspberry Pi?

- Small, cheap, and increasingly powerful programmable computer!



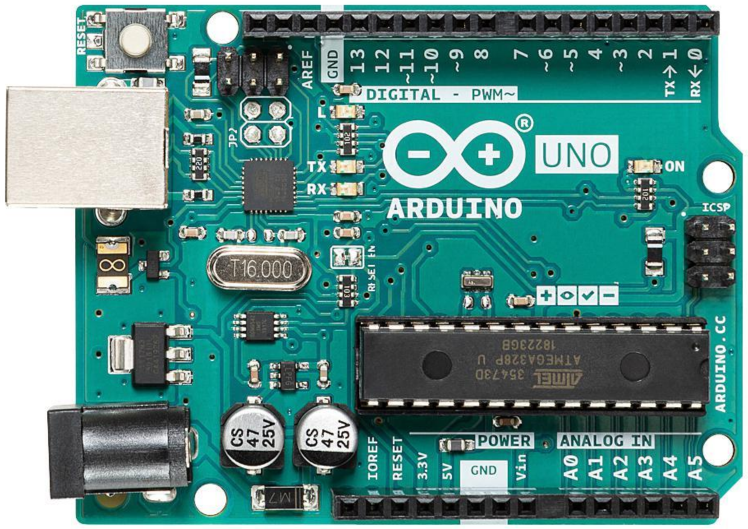
Single Board Computers

- Other SBCs: Pine64 ROCKPro64, OLinuXino, Beagleboards, OEMA68, Parallella with 18 core processors, Banana Pi, CHIP the $9 base modular computer, CubieBoards,etc.
- Desktop computer: 200 W @ 0.8 A
- Rpi: 5 W @ 0.8 A



Release Date	Original Price	Model	GPIO Pins	CPU
Oct 2023	$50-$80	5	40	2.4 GHz ARMv8 64 bit (quad core)
Oct 2021	$15	Zero 2 W	40	1.0 GHz ARMv8 64 bit (quad core)
Nov 2020	$70	400	N/A	1.5 GHz ARMv8 64 bit (quad core)
Jun 2019	$35-$75	4 B	40	1.5 GHz ARMv8 64 bit (quad core)
Nov 2018	$25	3 A+	40	1.4 GHz ARMv8 64 bit (quad core)
Mar 2018	$35	3 B+	40	1.4 GHz ARMv8 64 bit (quad core)
Jan 2018	$10	Zero WH	40	1.0GHz ARMv6 (single core)
Feb 2017	$10	Zero W	40	1.0GHz ARMv6 (single core)
Feb 2016	$35	3 B	40	1.2GHz ARMv8 64bit (quad core)
Nov 2015	$5	Zero	40	1.0GHz ARMv6 (single core)
Feb 2015	$35	2 B	40	900MHz ARMv7 (quad core)
Nov 2014	$20	1 A+	40	700MHz ARMv6 (single core)
Jul 2014	$25	1 B+	40	700MHz ARMv6 (single core)
Apr 2014	$30	Compute	200	700MHz ARMv6 (single core)
Feb 2012	$25	1 A	26	700MHz ARMv6 (single core)
Feb 2012	$35	1 B	26	700MHz ARMv6 (single core)

Comparison with Micro-controllers



Arduino	RPi
Control, repetitive tasks (reading sensor, controlling LEDs, and small simple programs)	Fully functional computer capable of running an OS and handing complex tasks such as web browsers, programming, and multimedia
No OS, runs one program continuously	Runs complete OS (Linux)
Simple, often in C/C++. Programs are uploaded directly to hardware	Support multiple programming languages
Less Power	Requires more Power
Applications: real-time control, robotics, IoT devices, automation	Applications: projects needing full computational power, media servers, machine learning, complex IoT systems

Programming

- Have to start somewhere and learn gradually!
- With RPi, the more comfortable you re with programming, the more you can do.... but you can still do a lot with very basic skills!
- Also a very good way to learn programming through projects which is the best way to learn

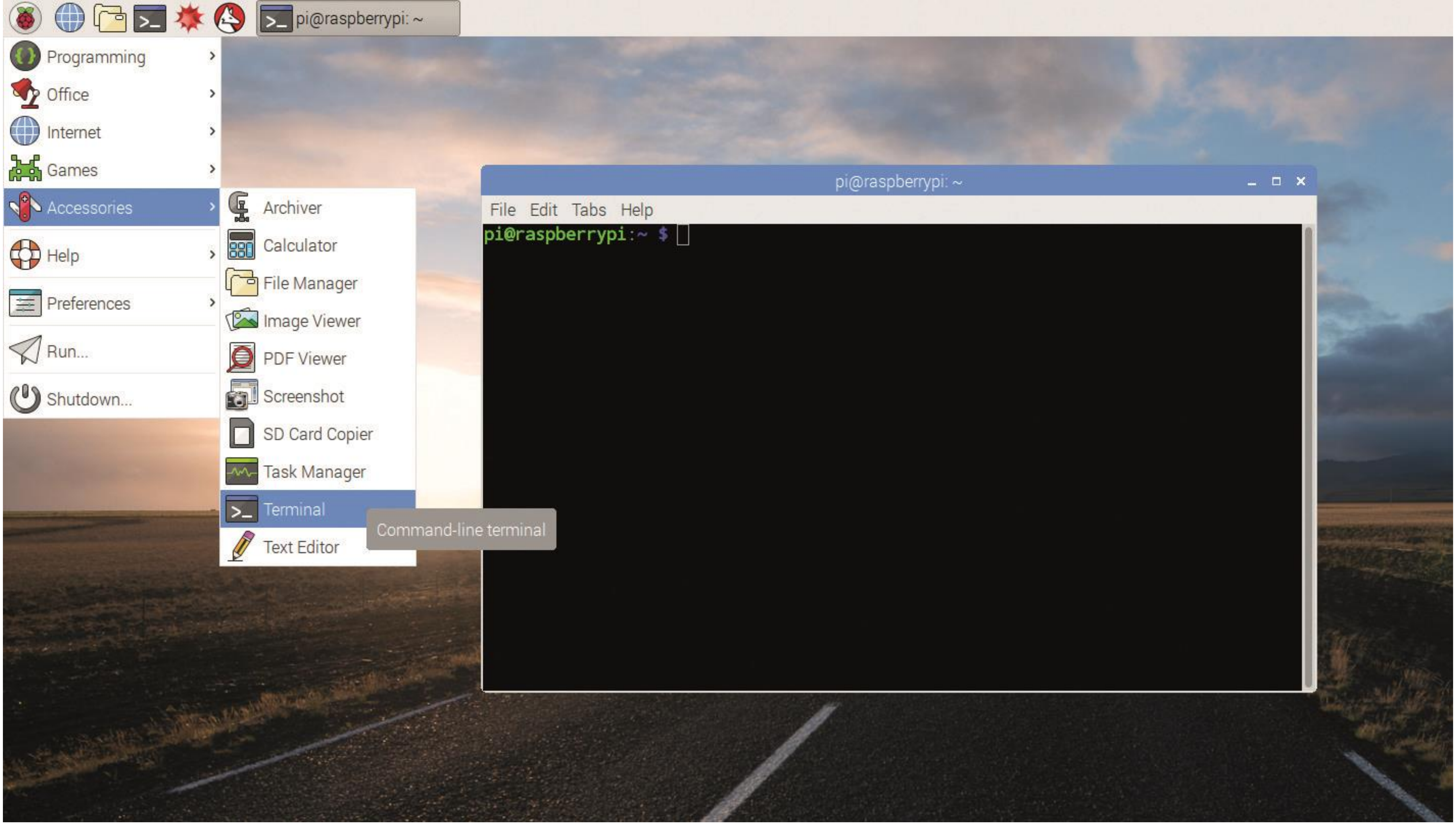
What you need

Raspberry Pis need an OS to interact with processor.

- Raspbian

Externals

- Video Monitor
- Mouse
- Keyboard



```
pi@raspberrypi:~ $ 
```

- All code for workshop is here on github: www.github.com/nolanlem/cmc-rpi-workshop

Programming: Crash course using terminal bash scripting

- Terminal: uses bash (command line shell). Click on raspberry icon in top left of screen and select terminal to open a window up.
- Terminal is a way to interact with the workings of the computer, you can pretty much do anything that you can do in the OS by just using the terminal if you wanted to.
- You type commands that the computer interprets to perform different types of processes.
- ‘ls’ = list, ‘man’ = manual, ‘mkdir’ = make directory/folder, etc…
- Computer has directory structure, which are just folders in folders etc.
- where are you currently in the file directory (’pwd’)?
- Move around file directory (‘cd’ = change directory, ‘cd ../’ = go back, ‘cd myfolder’ = move into ‘myfolder’)
- Let’s create a folder to hold our programs.
- demo (e.g. ‘mkdir myfolder’, ‘touch myprogram.py’ etc)

Programming: adding libraries

- We will be using python in a bit, but we first need to download some libraries that we will use with python.
- Rpi.GPIO and pygame (audio) should be already installed.
- Install ‘Adafruit motor shield’ library

1. Open a terminal window

2.sudo apt-get update

3.sudo apt-get –y install python-rpi.gpio

4.sudo pip3 install pygame

5.sudo pip install adafruit-motor-library

Python Programming: interactive mode vs. running program

interactive

- Type 'python' into terminal and press enter to open up interactive mode
- We will simply print a message to screen, this will come in handy for debugging purposes later on.
- print(”hello world”)
- quit()

Run program

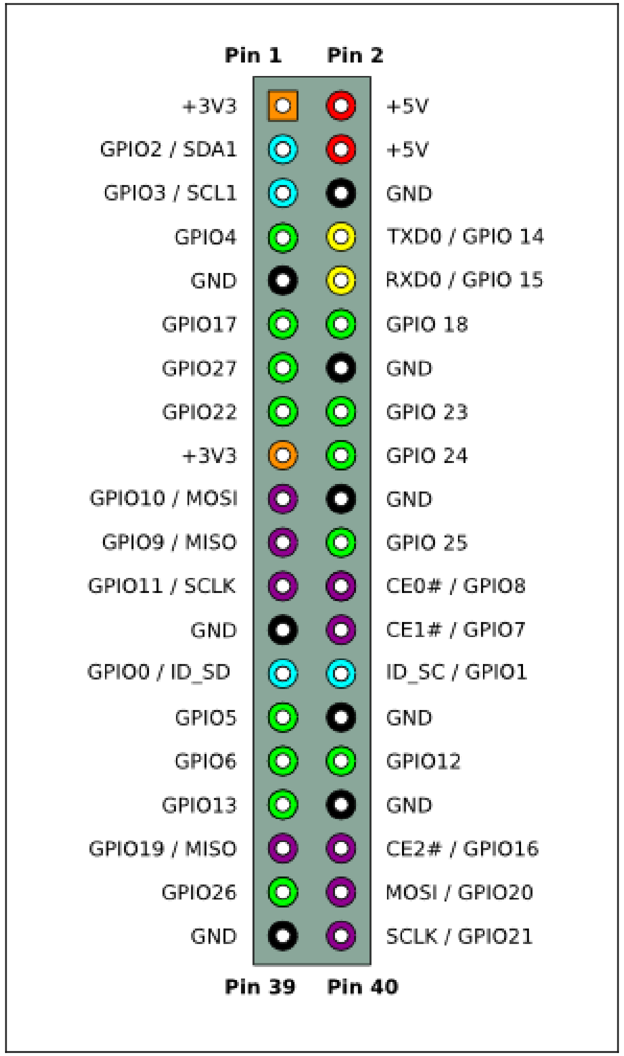
- touch myprogram.py
- nano myprogram.py
- Type:
- print(“hello, world”)
- NB: # = comment line
- Go back to terminal,
- Run program by:
- python myprogram.py enter

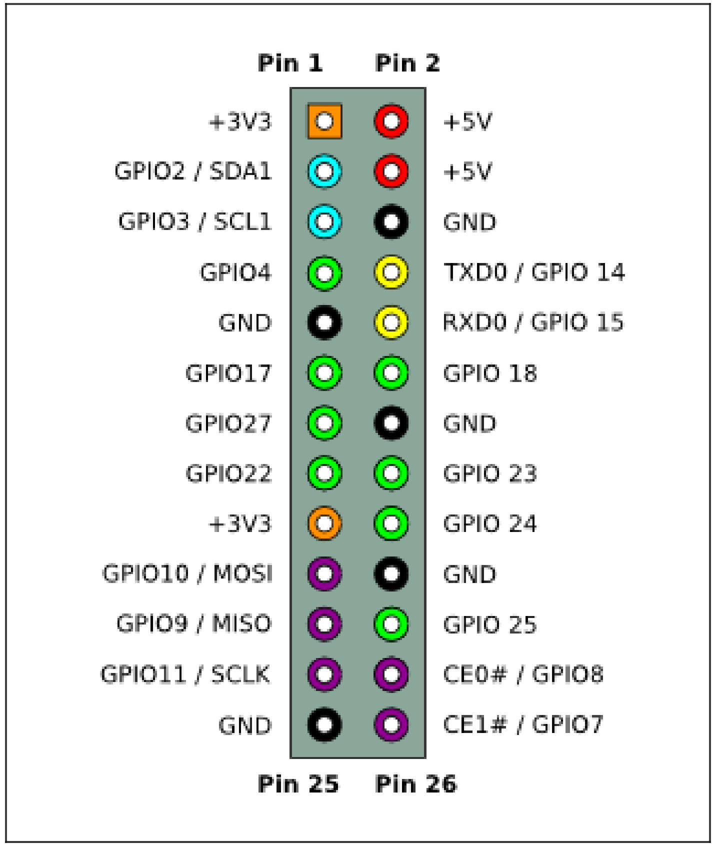
Input Output – GPIO Pins

General Purpose Input Output

Pins that you can use as control signals as we will see but also have other assigned functionality within RPi. (SPI, i2C)

also note the +5V, 3V3, and ground pins….

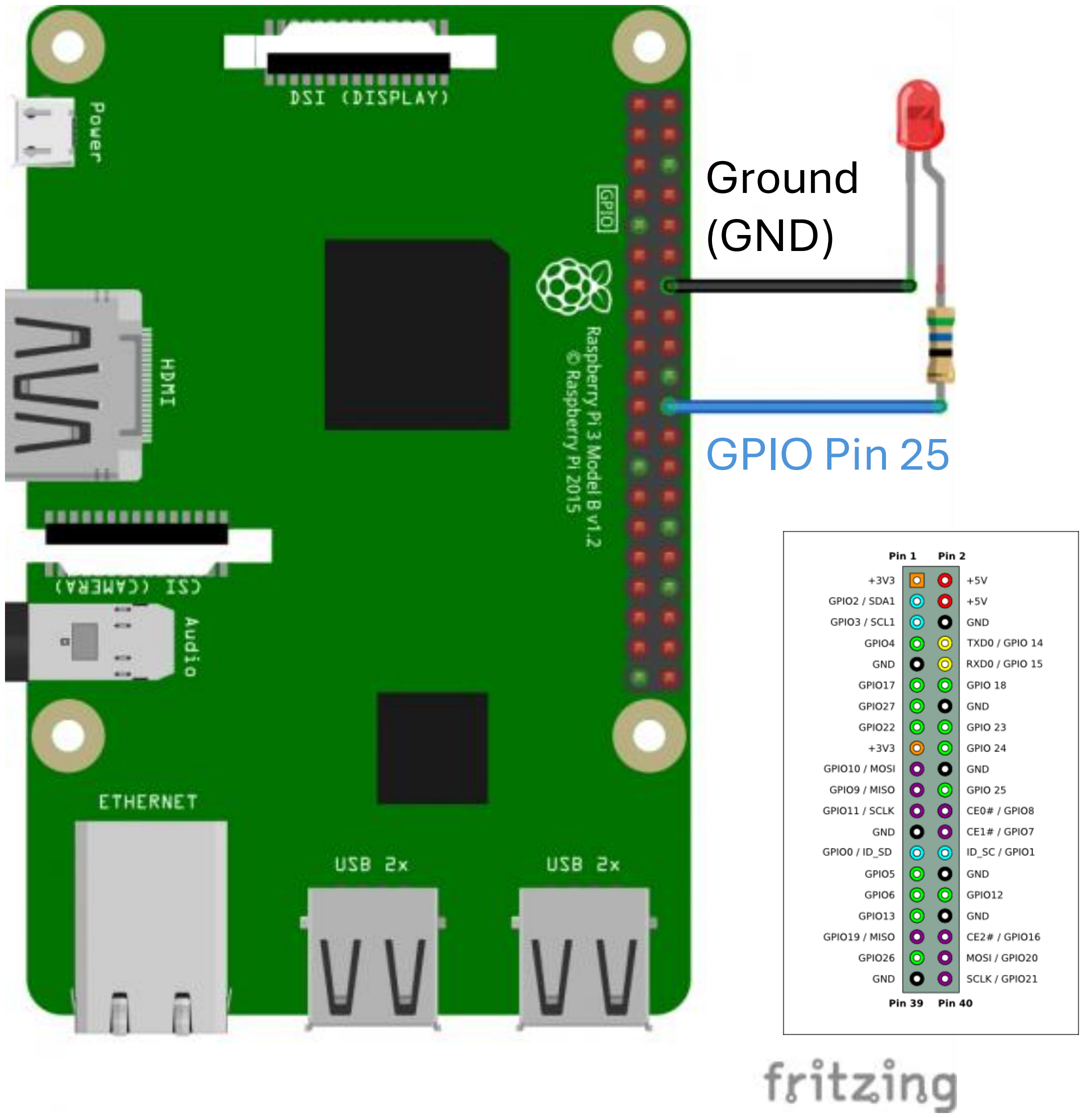




Simple demo, LED light

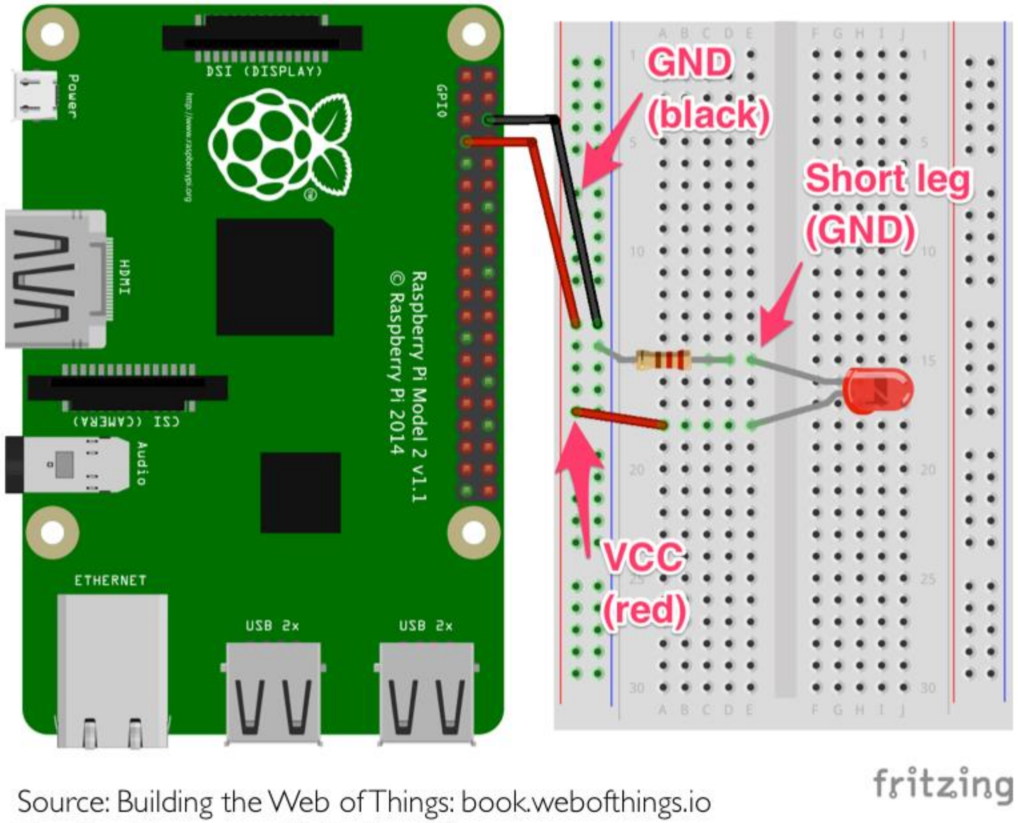
Why do this?

- GPIO pins are useful as simple triggers, that is as ‘control signals’, that are either ‘on’ or ‘off’
- ON = 3.3 V
- OFF = ‘ground’
- Demo
- On/Off control + a way to control time = programming motion/sound



	Pin 1	Pin 2	
+3V3	□	○	+5V
GPIO2 / SDA1	○	○	+5V
GPIO3 / SCL1	○	●	GND
GPIO4	○	○	TXD0 / GPIO 14
GND	●	○	RXD0 / GPIO 15
GPIO17	○	○	GPIO 18
GPIO27	○	●	GND
GPIO22	○	○	GPIO 23
+3V3	○	○	GPIO 24
GPIO10 / MOSI	○	●	GND
GPIO9 / MISO	○	○	GPIO 25
GPIO11 / SCLK	○	○	CE0# / GPIO8
GND	●	○	CE1# / GPIO7
GPIO0 / ID_SD	○	○	ID_SC / GPIO1
GPIO5	○	●	GND
GPIO6	○	○	GPIO12
GPIO13	○	●	GND
GPIO19 / MISO	○	○	CE2# / GPIO16
GPIO26	○	○	MOSI / GPIO20
GND	●	○	SCLK / GPIO21
	Pin 39	Pin 40	

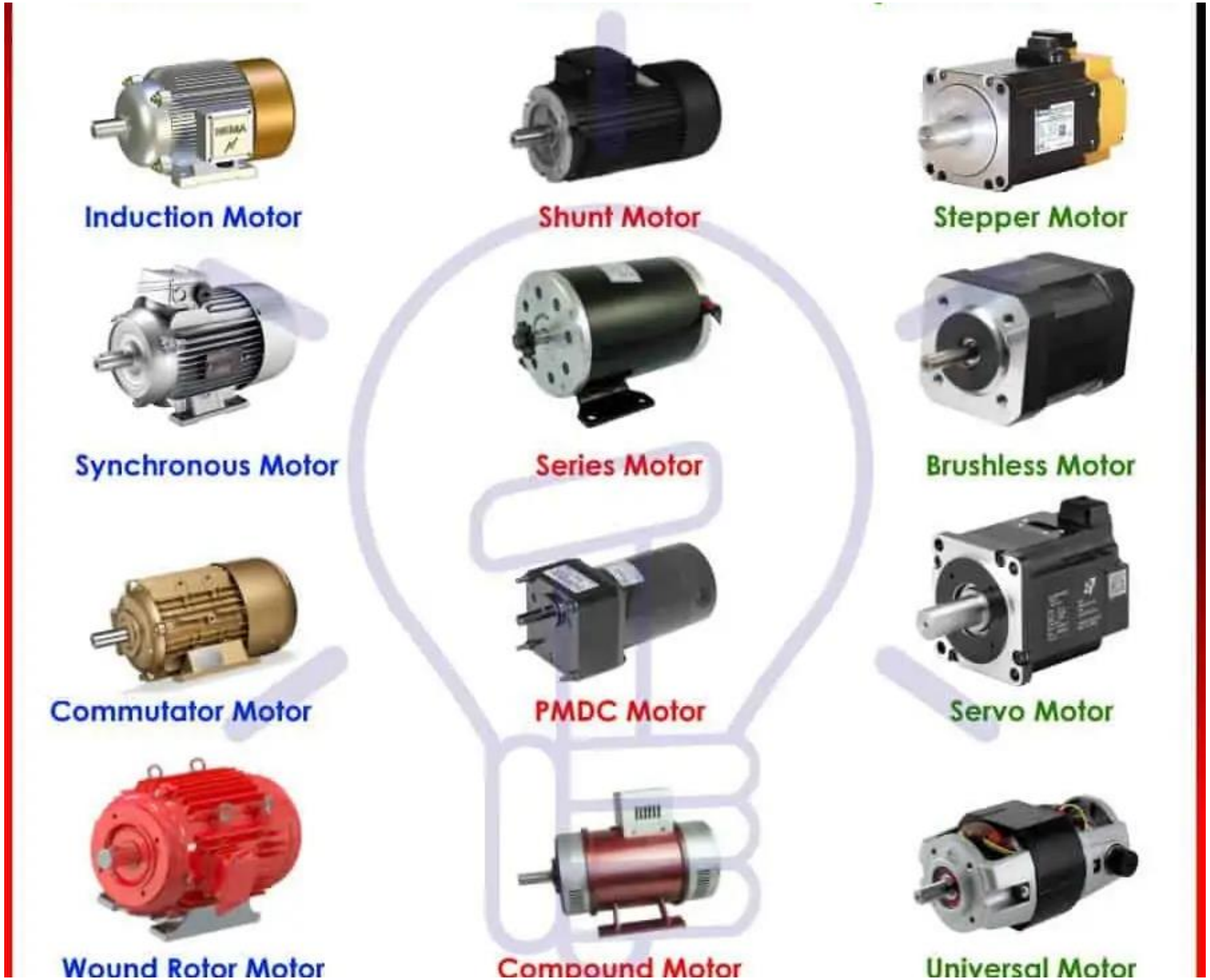
LED Demo

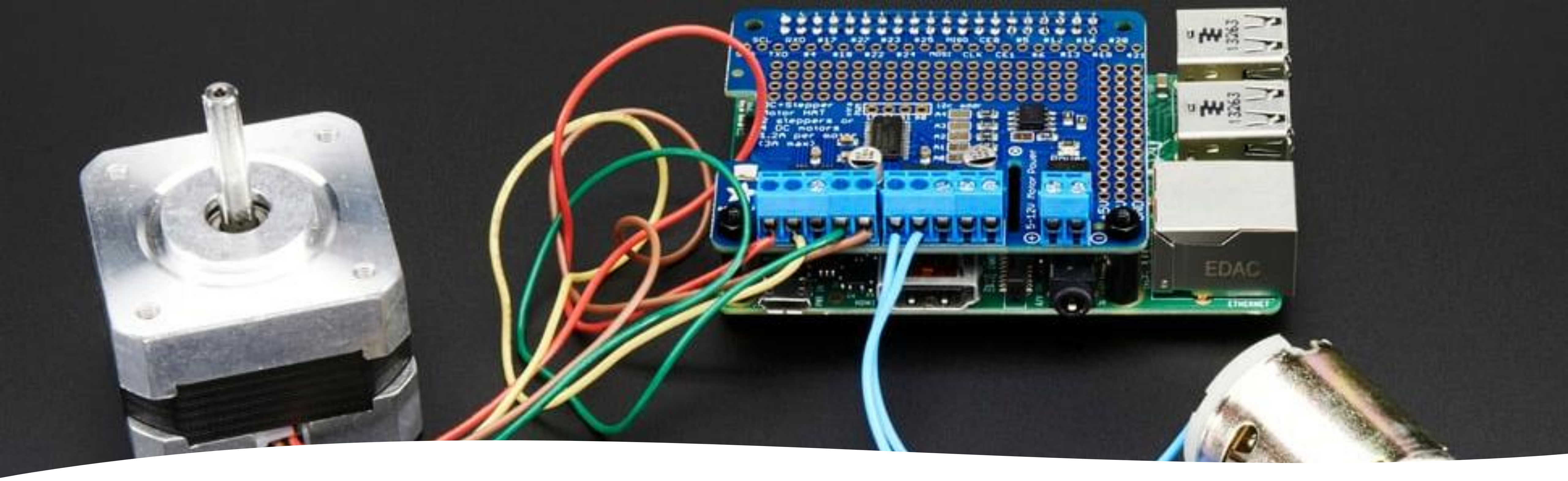


Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

```
import RPi.GPIO as GPIO
import time
# Choose which GPIO pin to use
led = 4
# set up GPIO as output
GPIO.setmode(GPIO.BCM)
GPIO.setup(led, GPIO.OUT)
# set pin GPIO7 to be "on," turning on the light
GPIO.output(led, 1)
# delays for 1 second, keeping the light on briefly
time.sleep(1)
# set pin GPIO7 to be "off," turning off the light
GPIO.output(led, 0)
# delays for 1 second, keeping the light off briefly
time.sleep(1)
```

Motors


Induction Motor
Shunt Motor
Stepper Motor
Synchronous Motor
Series Motor
Brushless Motor
Commutator Motor
PMDC Motor
Servo Motor
Wound Rotor Motor
Compound Motor
Universal Motor

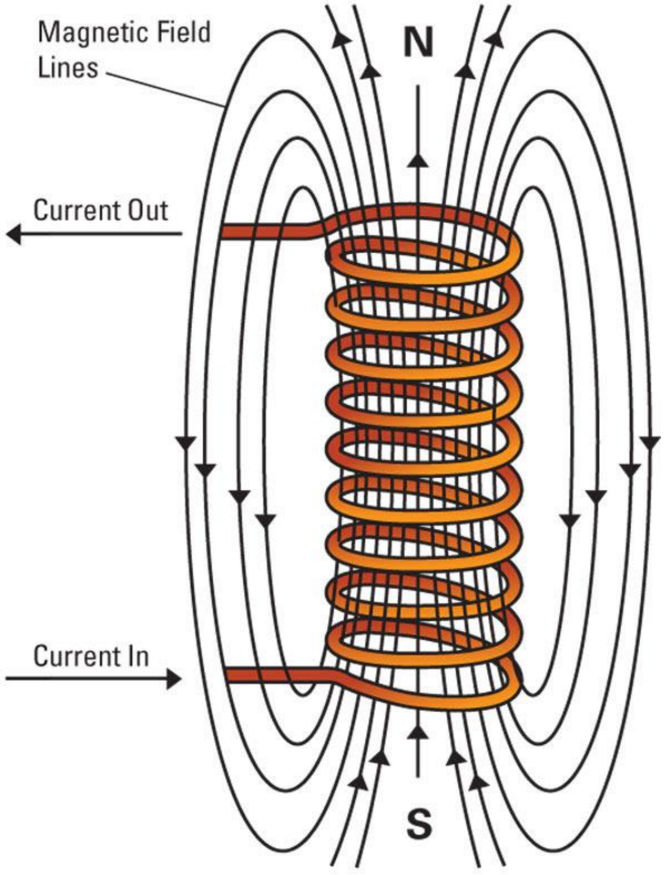


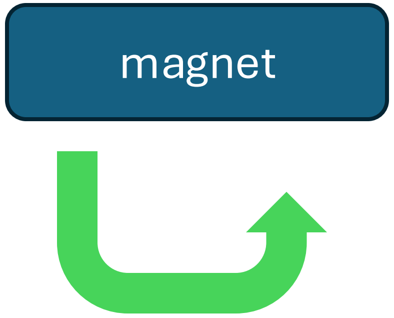
Motor Shields

- What is a ‘motor shield’?
- External circuit printed circuit board that is form fitted for specific raspberry pi or microcontroller that provides power and functionality for controlling motors and other elements.
- Motors need electrical power in the form of Voltage, Current in order to run.
- Avoids complexity of designing high power analog circuits, sometimes built in circuitry that scales to accommodate larger motor arrays, often libraries or software that is written to accommodate specific shields

Basic DC Electromagnetics

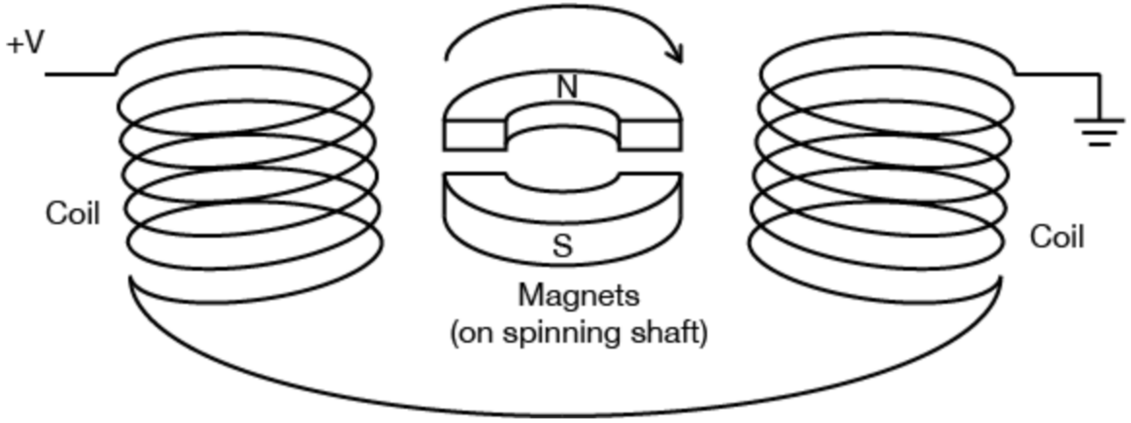
Battery
Voltage Source





magnet

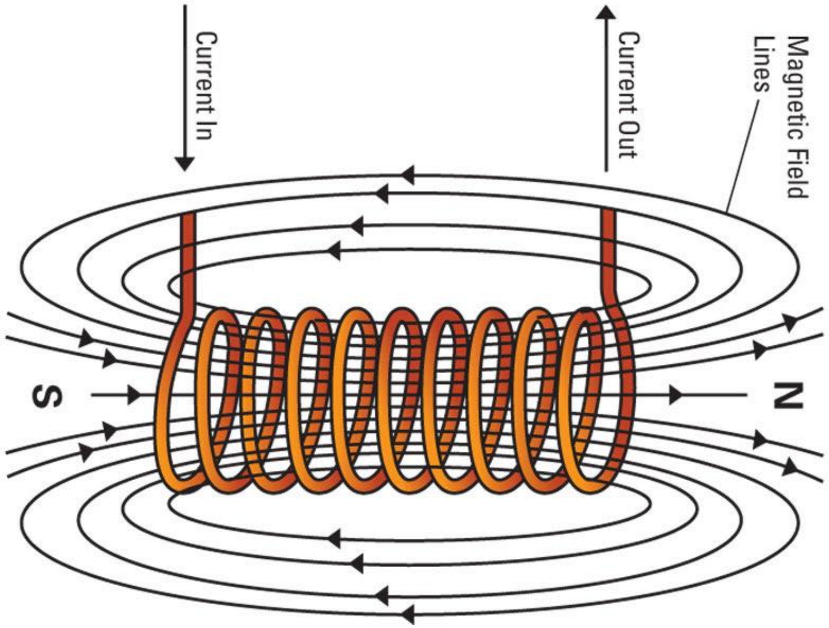
'wants to move'

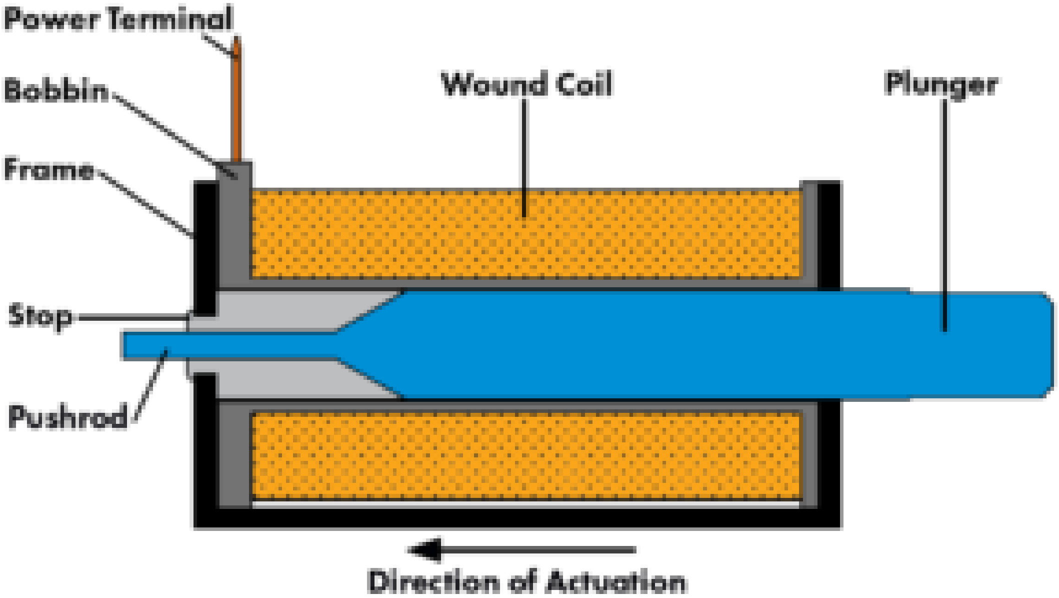


direction of current determines polarity of magnetic field

Higher current = greater magnetic field = greater force of attraction/repulsion

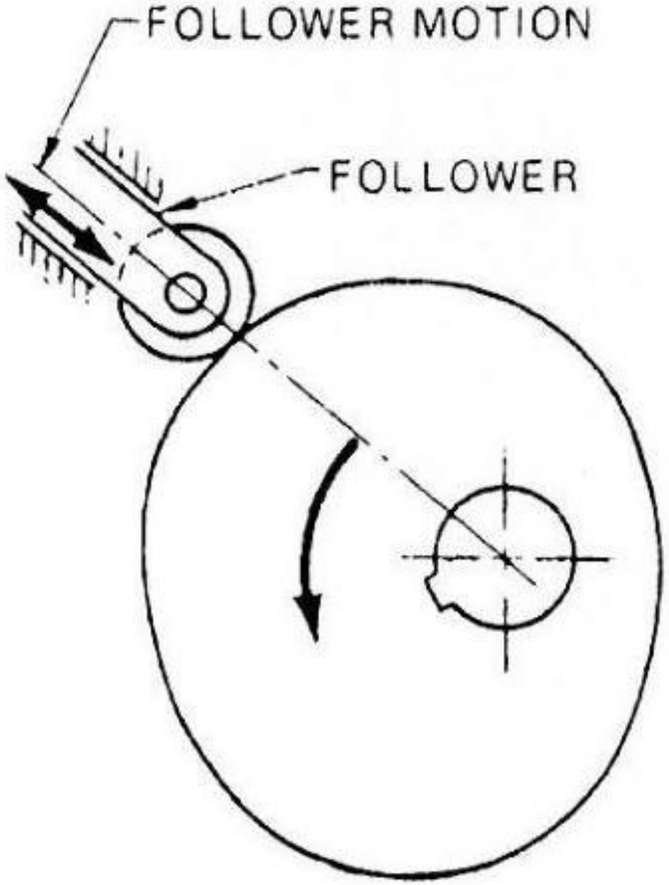
Push-pull Solenoid Demo





Generating "Pushing” / ”Striking” motion

- Mechanical device used to transmit motion to a follower by a direct contact
- Cam – driver: follower driven
- In a cam – follower pair, the cam normally rotates while the follower may translate or oscillate
- Basic idea: Rotary motion into reciprocating motion
- Simple, but very powerful

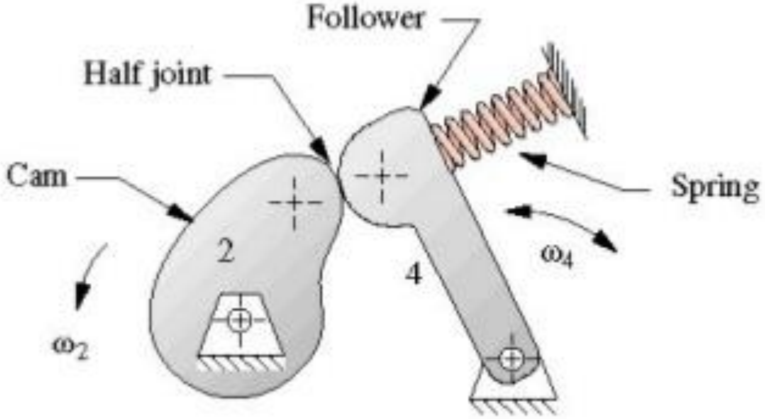


(A) RADIAL

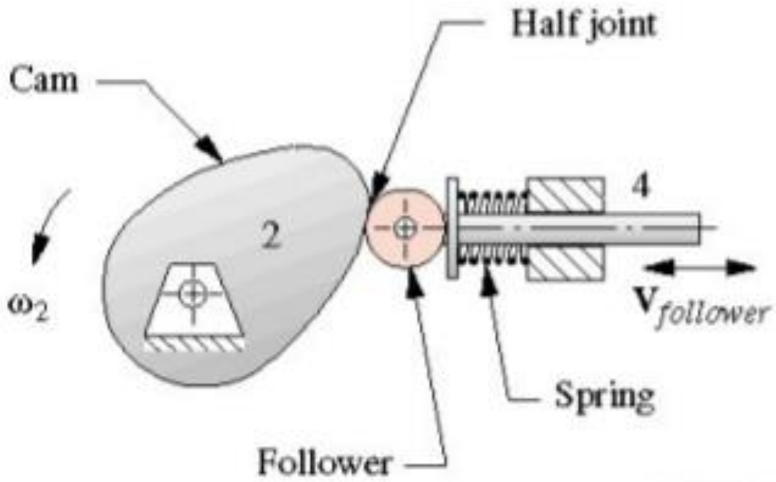
Generating "Pushing” / ”Striking” motion

Type of Follower Motion

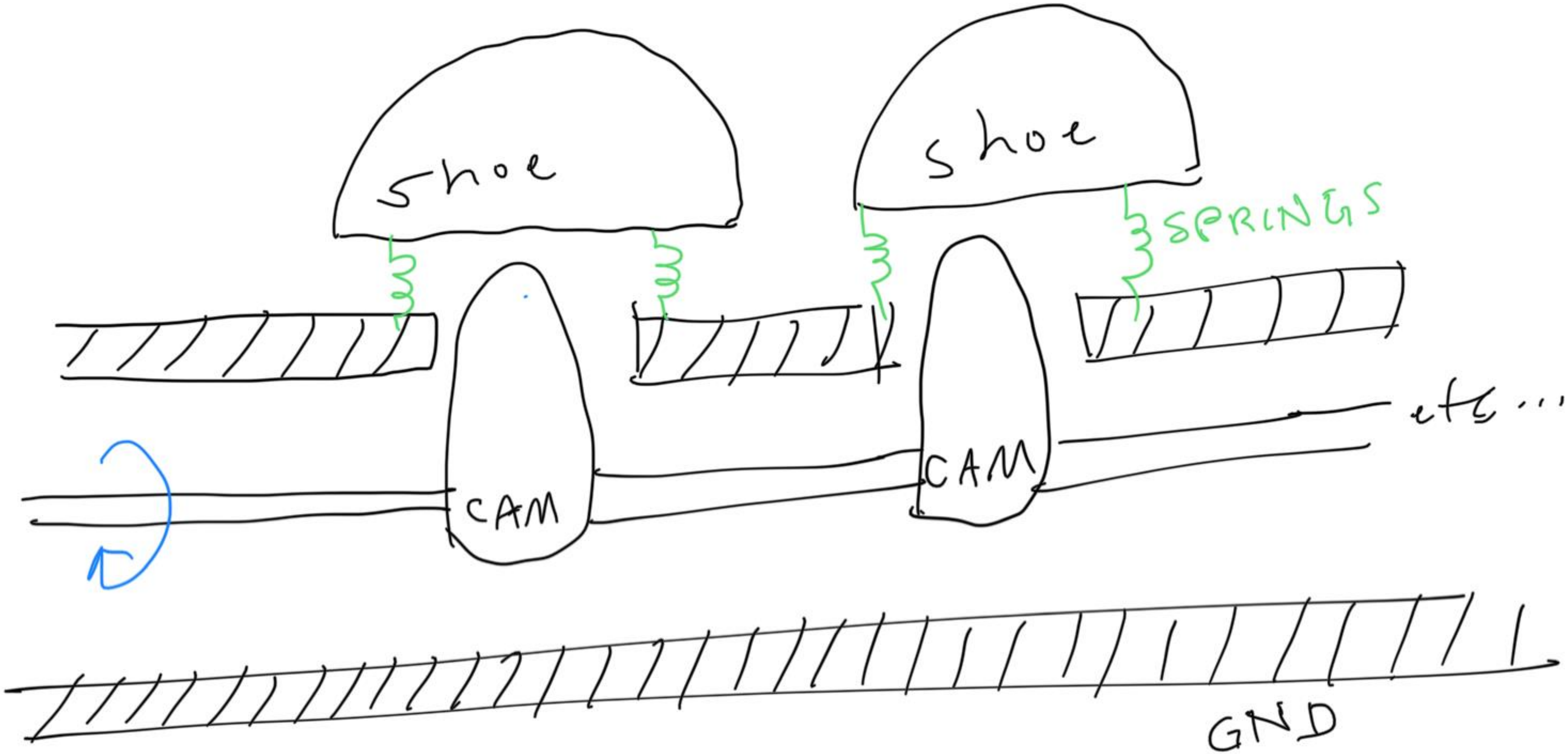
Oscillating follower



Translating follower



Example of Using Cams


shoe
shoe
SPRINGS
CAM
CAM
etc...
GND

DC motors: Voltage Specifications

- The rated voltage of a motor is the voltage at which it operates at peak efficiency.
- can be operated somewhat above or below their range, but it’s best to plan to operate them at their rated voltage.
- Dropping below rated voltage reduces the motor’s power, and operating above the rated voltage may burn the motor out.
- Plan on the motor’s top speed being at rated voltage, and slowest speed at no more than 50% less than the rated voltage.

Example Motor Spec

Description:
DC 12V 300RPM
Specification:
Input Voltage (V):12
No Load Speed (RPM):300
No load current (mA): <1100
Rated Torque (kg.cm): 16kg
Rated Speed (RPM):195
Rated current (A): < 6.0
Unloading Current(A):20

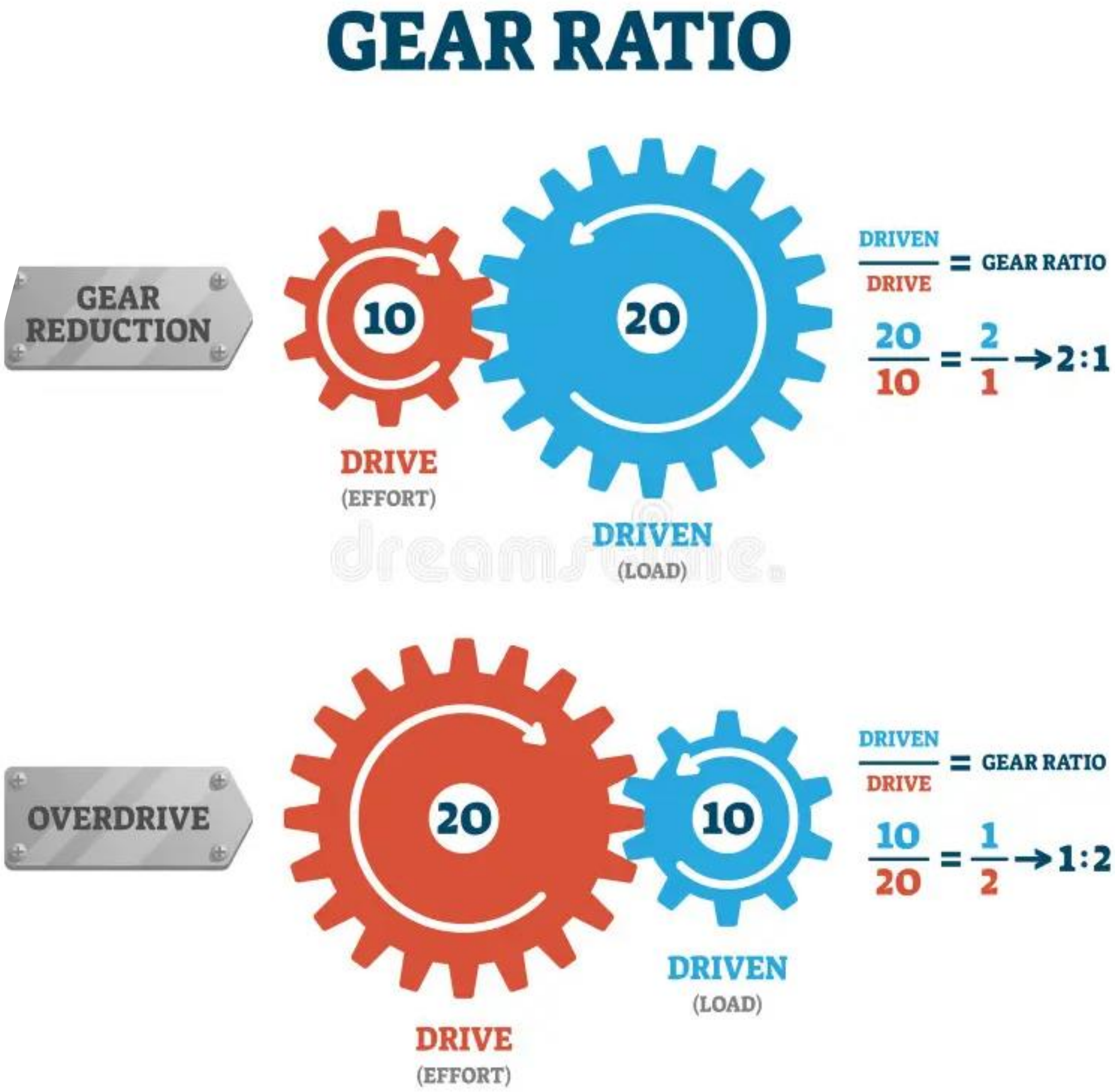
Interpreting motor specs

Torque = Rotational force

Proportionate to Revolutions per Minute (RPM

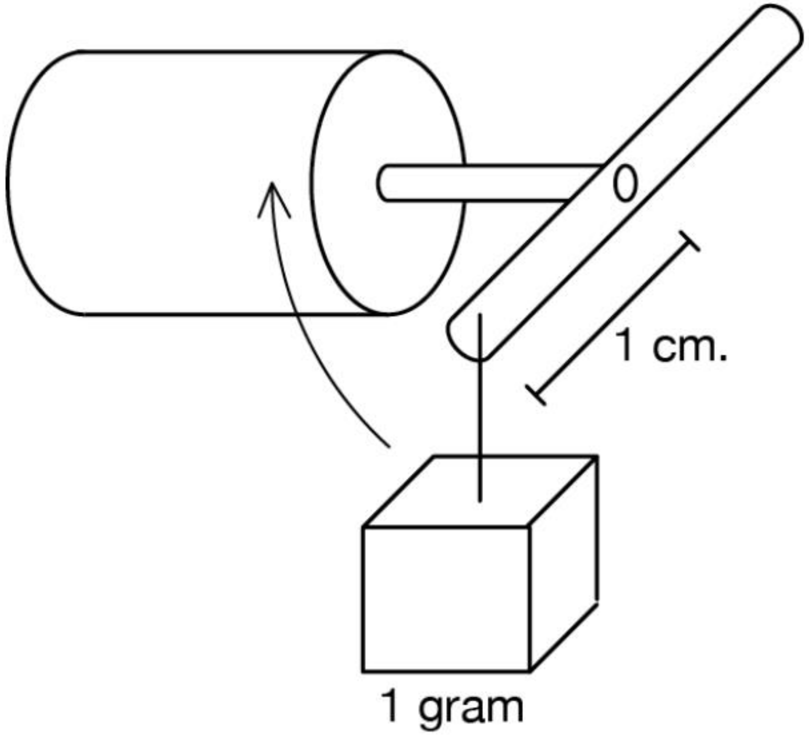
Different motors have different torques that depend on type of motor

Typically, drive shaft is small and high RPM, you use gear(boxes) to step up Torque to create higher force but lower RPM



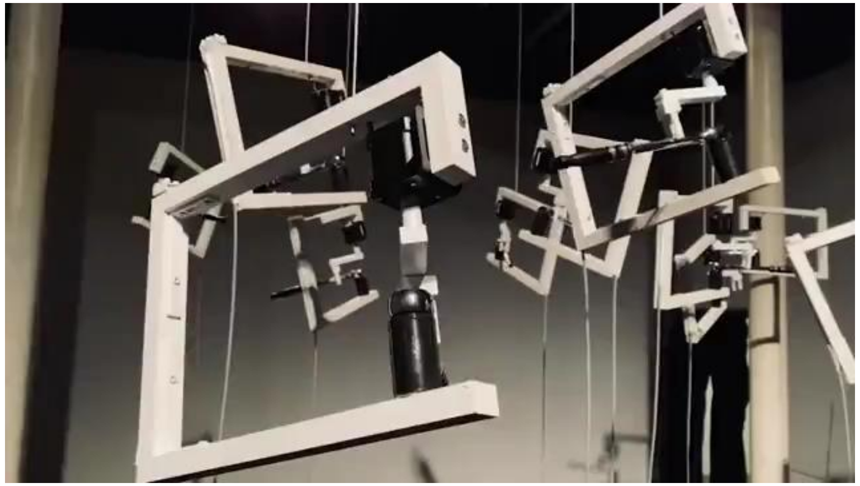
Interpreting motor specs

Rotating a Small Door Handle: ~5–10 kg·cm
Turning a Bike Pedal: ~50–150 kg·cm
Operating a Power Drill: ~150–300 kg·cm
Rotating a Small Office Chair with Someone Sitting: ~200–500 kg·cm (depending on weight)
Turning the Wheels of a Small RC Car: ~500–1000 kg·cm
Electric Scooter Wheel Torque: ~200–800 kg·cm
Rotating a Car Steering Wheel: ~300–800 kg·cm
Lifting a Small Garage Door: ~1000–3000 kg·cm
Opening a Refrigerator Door: ~10–50 kg·cm
Rotating a Large Wind Turbine Blade: Thousands to tens of thousands of kg·cm

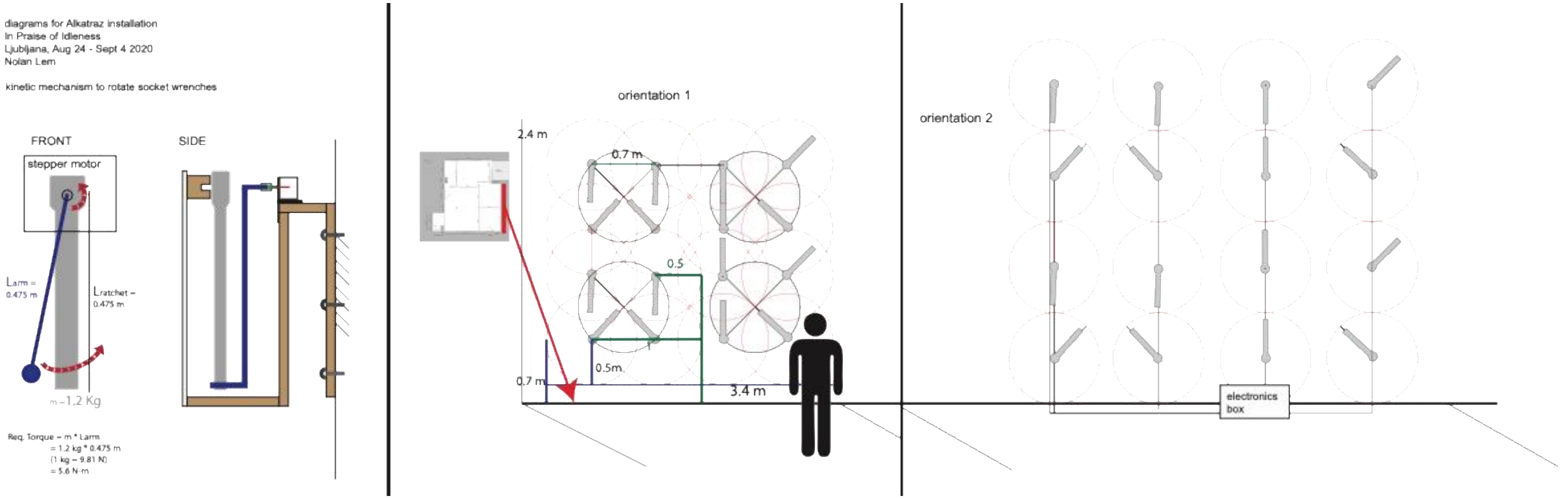


https://www.omnicalculator.com/physics/torque#

Torques in Action: examples



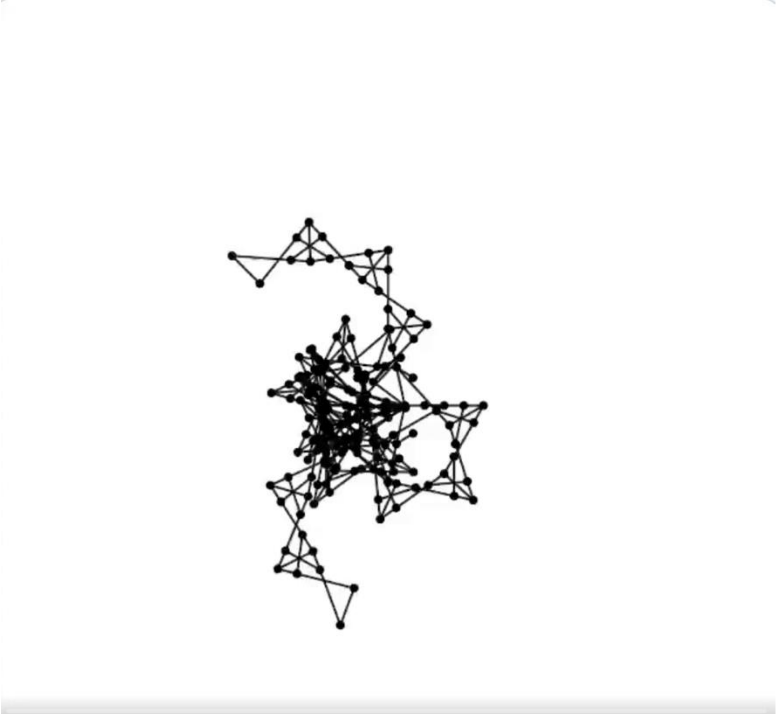
"In praise of idleness"

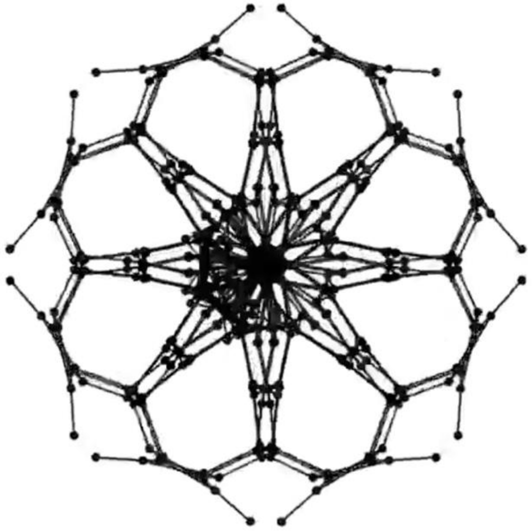


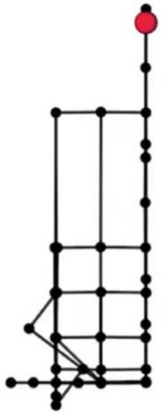
"rocks in roll"
https://vimeo.com/242010661

On that note….

Nearly all kinetic motion can be thought of as translating circular rotation into different types of movement







Keishiro Ueki @chocolinkage

Adafruit DC & Stepper Motor HAT for RPi

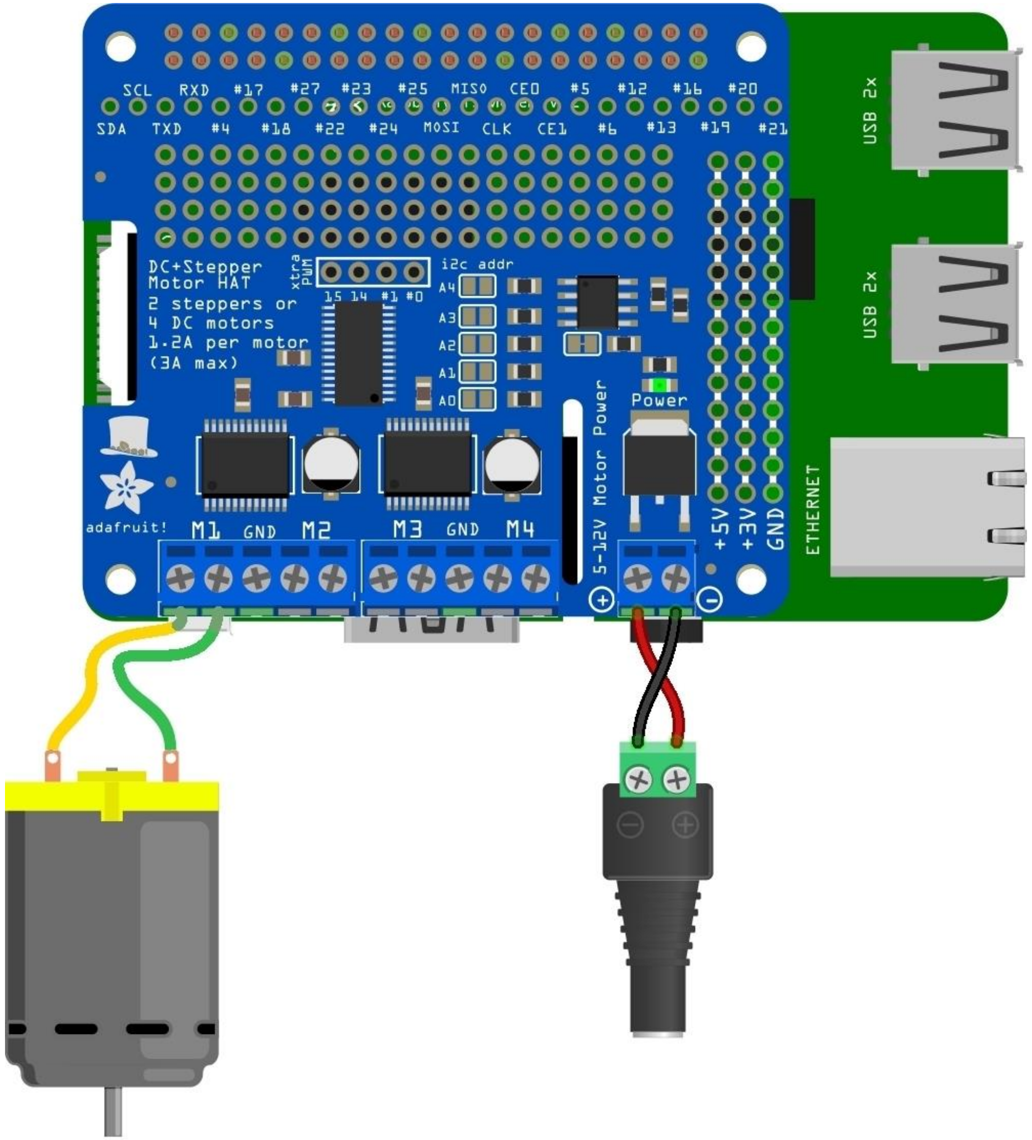
1. Open terminal
2. sudo pip3 install adafruit-circuitpython-motorkit

Solder on header pins ?

Turn off raspberry pi

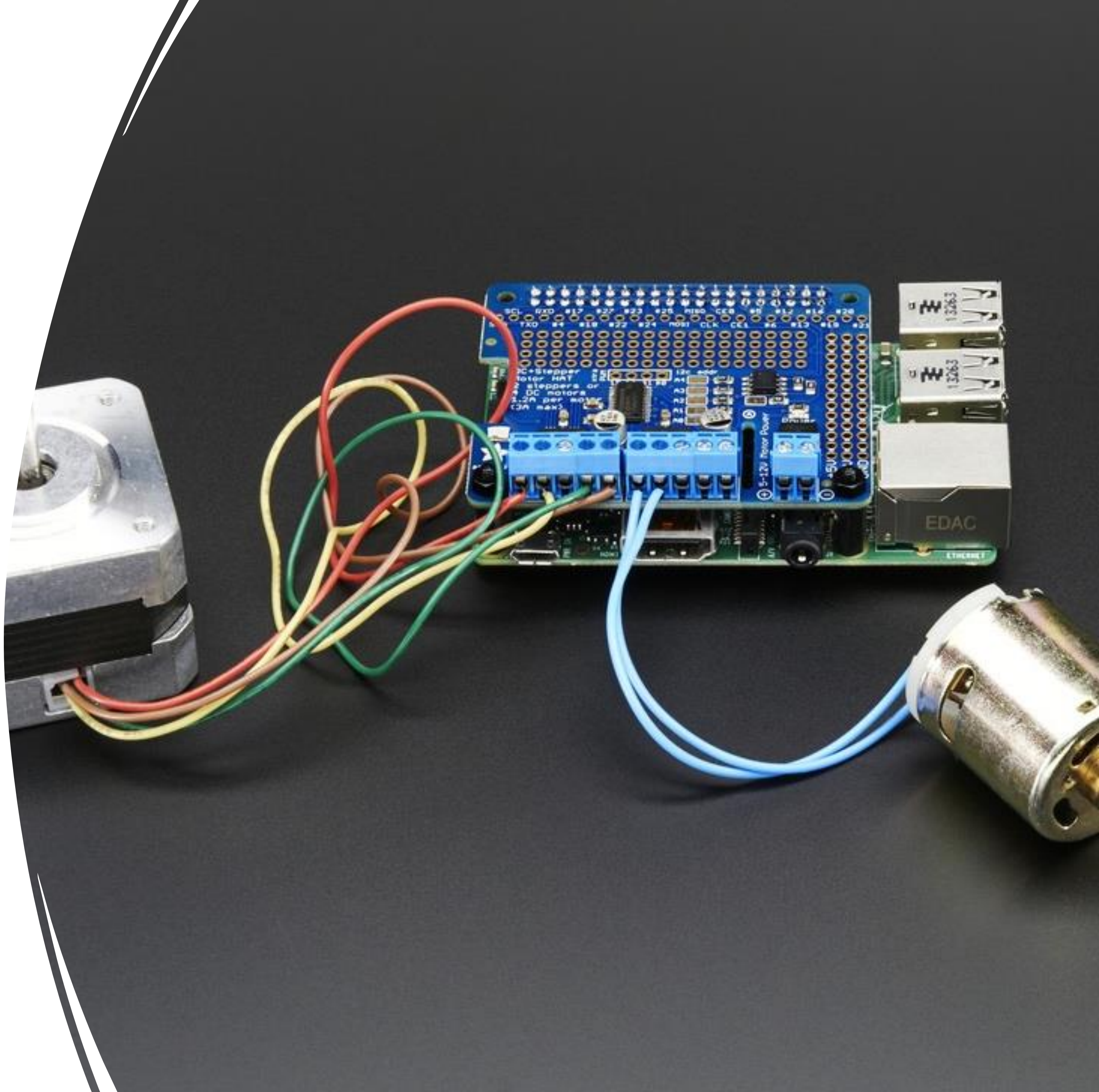
Put shield onto raspberry pi

Setting up Motor Shield



Adafruit Motor Shield

- Uses i2c (Inter-Integrated Circuit), which is “synchronous, multi-master, multi-slave communication protocol”…. standardized way to communicate with multiple devices! Also SPI
- 1.2 A max per channel
- 4 bi-directional DC motors
- 2 stepper motors
- Stackable, up to 128 boards!... = 512 DC motors….
- Flyback diodes across outputs …



Programming the Motor Shield in python

We will use the libraries that we downloaded

Lets do this in ***interactive*** mode:

- open terminal

```
>> python
>> from adafruit_motorkit import MotorKit
>> kit = MotorKit()
>> kit = MotorKit(i2c=board.I2C()) # i2c address… this is how you control N boards
>> kit.motor1.throttle = 1.0 # motor should begin running at full speed
>> time.sleep(2) # wait two seconds
>> kit motor1.throttle = 0 # stop motor
>> time.sleep(2) # wait 2 seconds
>> kit motor1.throttle = 0.5 # run at half speed
>> kit.motor1.throttle = None # let motor coast to zero
>> quit() # quit interactive mode to return to bash window
```

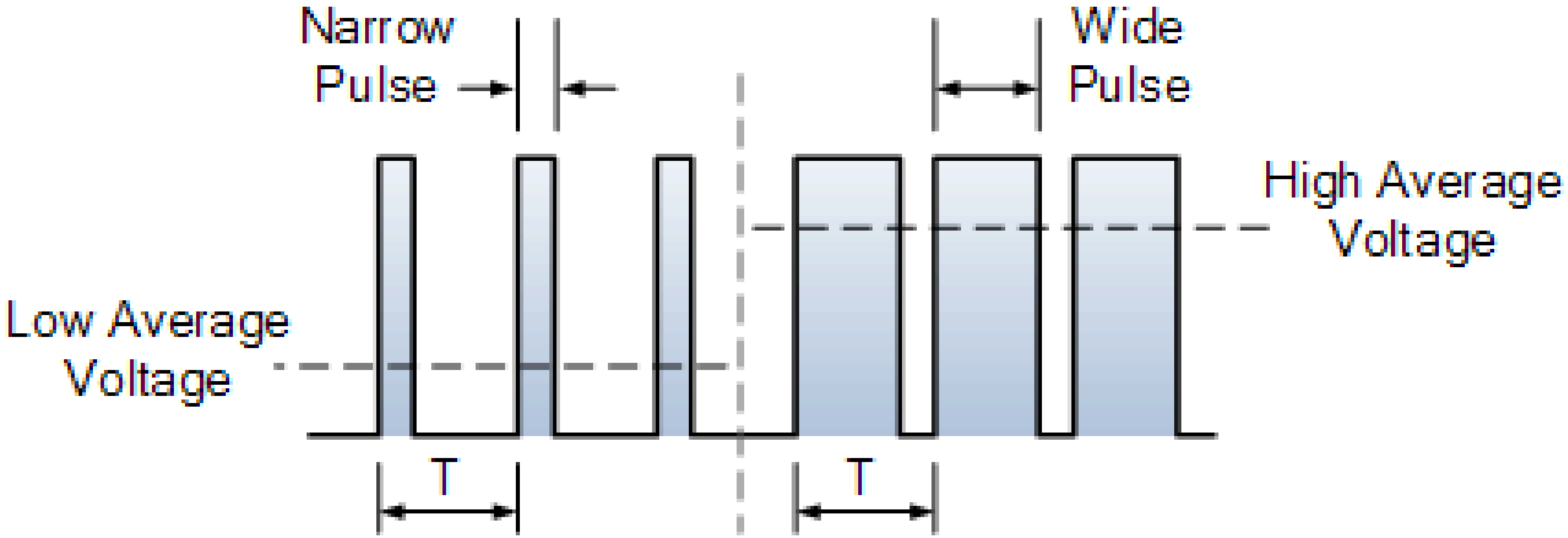
Motor Control: Now as a ‘program’

- Open terminal
- Touch motor.py
- Open motor.py # will open in rapsbian default code editor
- How can we program time into our python program?
- Python has library ‘time’

```
import time
import board from adafruit_motorkit import

MotorKit kit = MotorKit(i2c=board.I2C())
kit.motor1.throttle = 0.5 # run motor at half speed
time.sleep(2.0) # wait 2 seconds
kit.motor1.throttle = 0 # turn off
```

SAVE!
Now go back to terminal window
Type ‘python motor.py’



Motor Control: How does it change speeds?

- Certain RPi pins: **Pulse Width Modulation**
- Internally, in the RPi, certain pins have the ability to provide control signals that are pulse width modulated.

NB: This is because wound coils are 'inductors' which will 'store energy' in the form of current. When applied voltage is turned off, it will discharge current which essentially means that it cannot respond the input signal right away when it changes.

Motor Control: Solenoid

- Let’s hookup our push-pull solenoids and see if we can program something to control them in the same fashion.
- 5 min Can try on your own!
- Remember, solenoids need to be turned **on** and **off**.

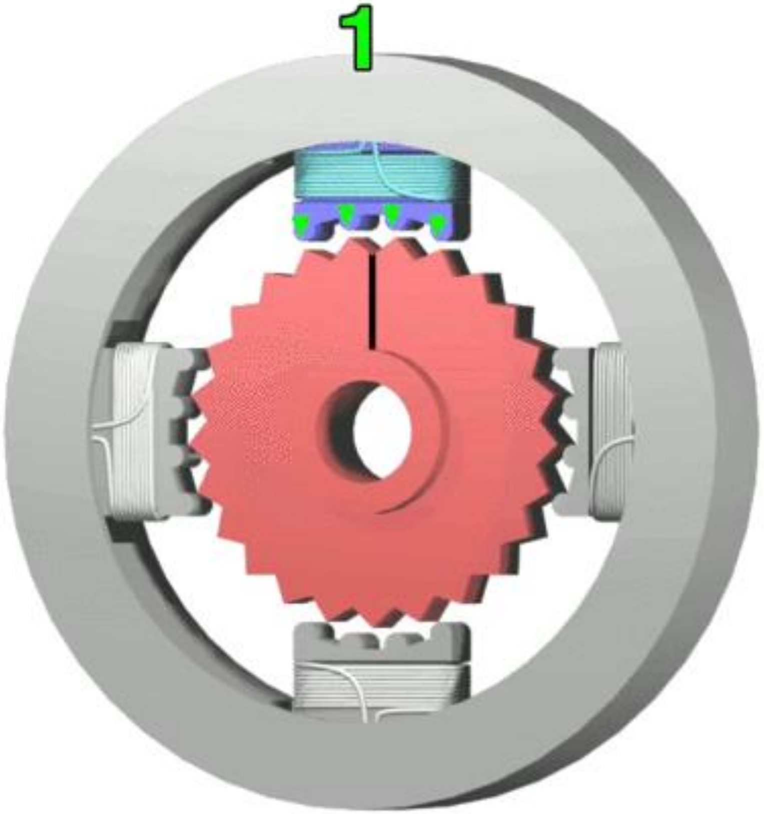
Motor Control: Stepper Motor

Absolute Positioning

Control applied signal to four coils to induce magnetic field.

Stepping is achieved by varying the coils that are triggered. (single, double, microstepping)

But thankfully.....
Our motor shield handles the applying the proper power and timing



RPI: Audio Output

- Depending on model, Rpi has stereo audio out of HDMI, USB, or headphone (3.5 mm output). All essentially “line” level.
- On board Digital-Audio-Converter is famously bad.. .best to use. Audio interface as an output device for high quality audio. Or by a RPI audio HAT for high-fi audio.
- Different ways to play back, write, and record
- Supercollider, Pure Data

Let’s test the sound in terminal

```
Check audio output on Rpi
(top left corner, sound->
audio output etc)

Open terminal.

sudo raspi-config

Playback sound in terminal
Open terminal

aplay applause-1.wav
```

RPI: Supercollider and Pure Data, Processing

- Audio Programming Languages much more powerful and dynamic!
- Typical Setups, host Supercollider Server on RPi on port: can send interact with client server via Python using various libraries to send OSC messages.
- Pure Data
- Can talk between programs using Serial port
- RPI can run Processing

Many projects

- Computer Vision l(live video processing using camera)
- Haptics
- Sensors
  - Force Sensing Resistors
  - Photo cells (detect light)
  - Hall effect sensors (detect magnets)
  - Piezo elements
  - Textile Fabric Sensors via pressure

Typical setups for installation

- Can program RPi to begin program on boot. Or use switches to trigger programs to run.
- Pesky CPU threading! Unlike microcontrollers, OS is constantly switching or running tasks in parallel, this can affect timing…
- Workarounds

Thank you!

- Questions?
- nlem@ccrma.stanford.edu