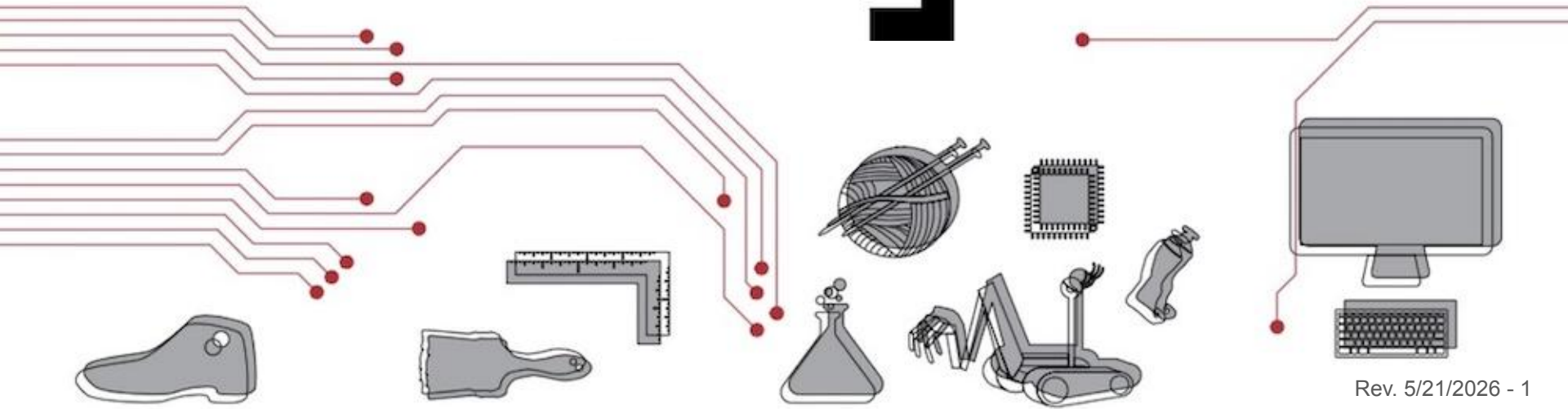
Introduction to Electronics & Microcontrollers

Paul Sakion

MAKE
santa fe



Welcome

- What projects do you want to explore?
- Experience with programming?
- Future areas of interest?
- Me: EE degree specializing in the measurement of sound and vibration
- Interests: Microcontrollers, sensors, motors, batteries, internet of things, robotics, 3D printers, etc.
- Introduction: The self balancing robot and the grunt-bot

Class Overview

- Modern electronics is more about using components: microcontrollers, smart sensors, motors, LEDs, displays, radios (WiFi, Bluetooth)
- Even knowing what to search for can be difficult: ESP32, ESP8266, Arduino, Raspberry Pi, NeoPixels, LoRa, MicroPython
- This class is an introduction to microcontrollers, inputs (sensors), and outputs
- What motor should I use? DC, AC, stepper, servos?
- Which battery should I use? Size, type, charge/discharge speed
- Richard Feynman: Just learn what is needed to understand/solve problems

Basic Safety

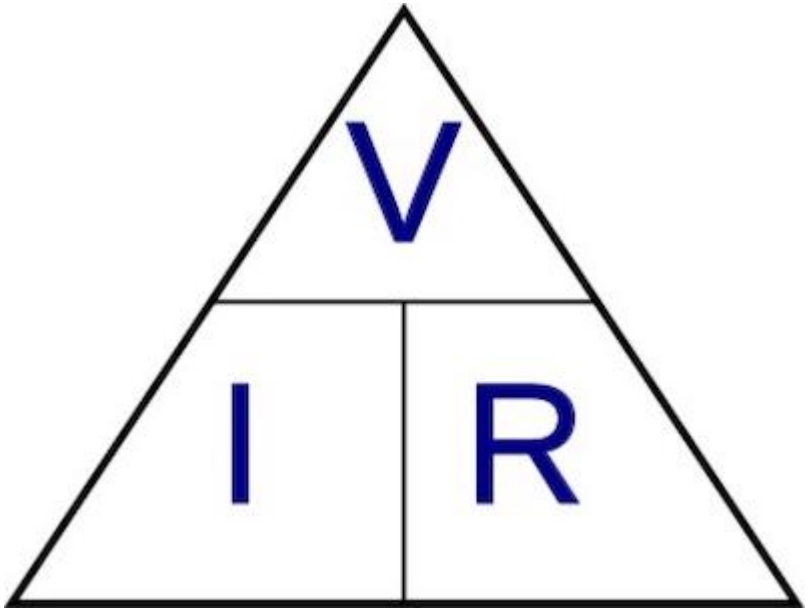
- One hand rule, buddy rule, capacitors, hot components
- Static grounding / Power ratings / Choosing wire size / Batteries

Voltage

- Electrical equivalent of pressure (think garden hose)
- Measured between points (like water pressure or distance)
- Two different types: Direct and Alternating
- Alternating Current: Power outlet, AC power adapter
- Direct Current: Battery, Plug-in adapter (AC to DC), USB ports, DC supply

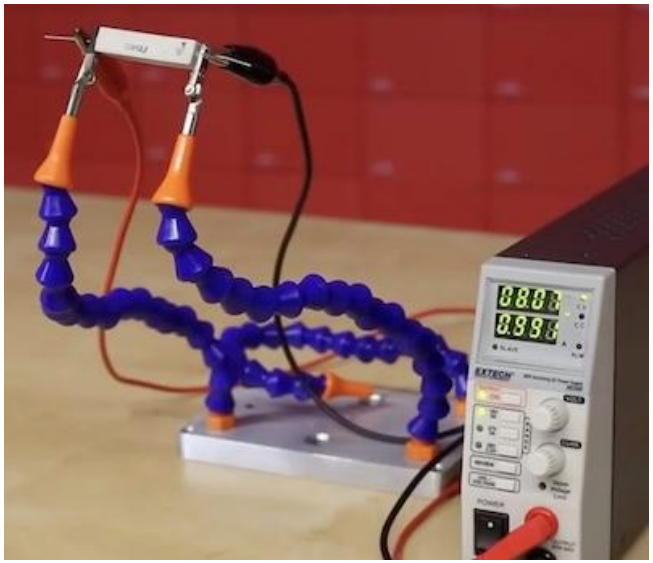
Voltage, Current, and Resistance

- Voltage = Current x Resistance
- Voltage is electrical pressure
- Current is the flow in amps (I)
- Resistance is the friction in ohms
- 5 volts = 1 amp x 5 ohms
- (Used when calculating LED current)



Power: Watts (P = V x I)

- Work done by the flow of electrons (amps) by pressure (voltage)
- 100W incandescent vs. 10W LED
- 1500W hair dryer (120V / 15A circuit)
- Power rating of components
- An excellent explanation video

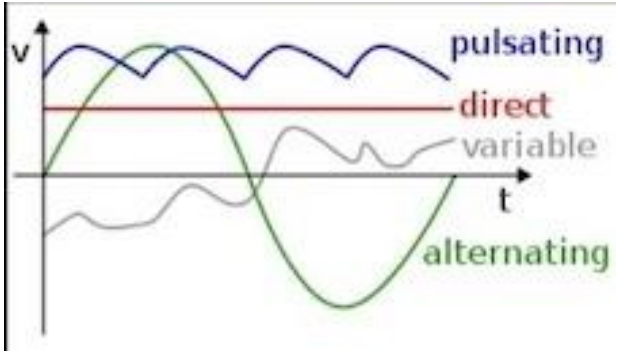


Spark Fun Video

Watt Meter

Frequency

- When voltage changes over time: Hertz or cycles/second
- Our wall sockets are 60 Hz / 120v (thanks to Tesla/Edison)
- Computer microprocessor clock ~ 4.5 GHz (Mac M4)
- Electromagnetic: 2.4 GHz (microwaves) / 5 GHz / 300 GHz
- Wavelength - 10E8 (c) / frequency (mmwaves = 300 GHz)

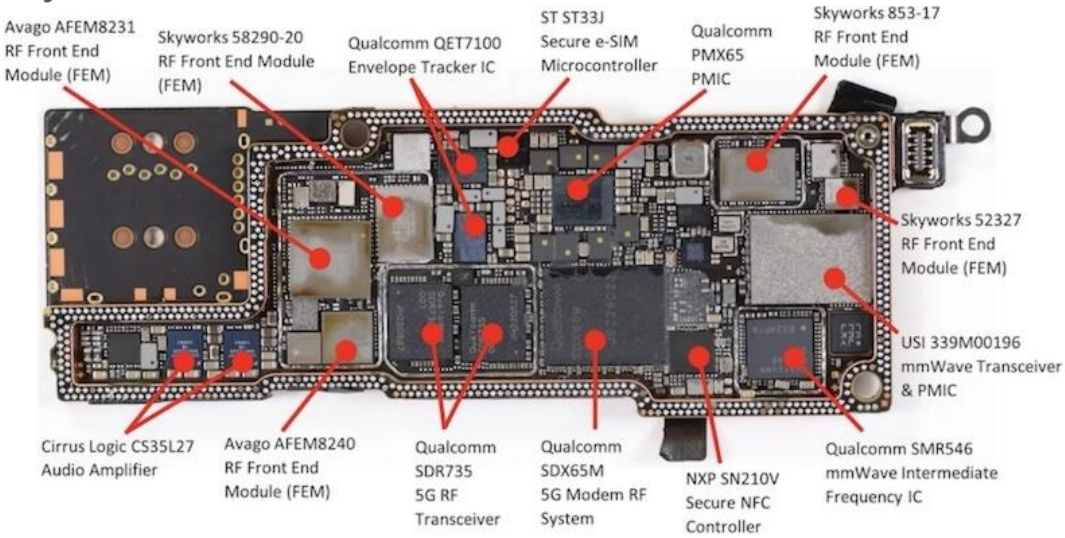


Modern Electronics

- Microcontrollers, breadboards, LEDs, sensors
- Motors (DC, AC, stepper, servos), displays, WiFi, Bluetooth
- Simulation software like CircuitLab, TinkerCad
- Demo of Multimeter and a great video
- Demo of Power Supply and a tutorial
- Explanation of oscilloscope and another video
- The SparkFun Youtube channel is awesome

Circuit Boards

- All those little specks are electronic parts
- Resistors, capacitors, inductors, integrated circuits
- Connected by wires embedded in the circuit board



Microcontrollers

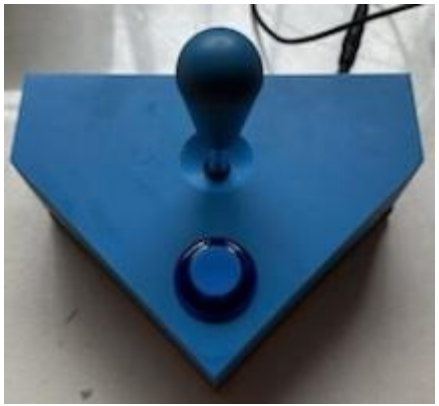
- Arduino / ESP32 - Open-source hardware / software electronics platforms intended for anyone making interactive projects (Arduino cookbook)
- Raspberry Pi - A credit card sized computer that can be used to learn coding and to build electronics projects, also works like a desktop PC (beginner's kit / cookbook / Sparkfun guide)
- Micro:bit - An open-source hardware designed for education. It has a processor, several sensors, Bluetooth / USB, 25 LEDs, two buttons, and can be powered by either USB or an external battery pack
- Video explaining many of the common types

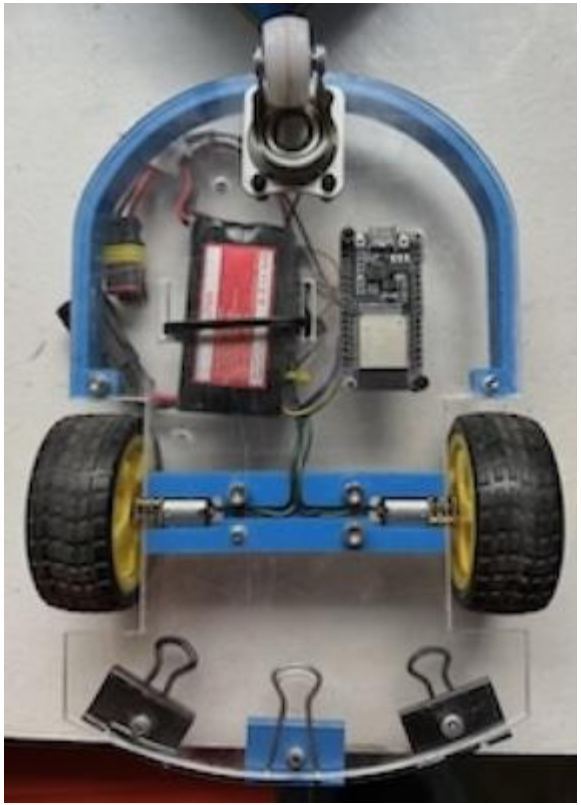
Inputs and Outputs

- Sensors - Thousands of types including temperature, pressure, humidity, light, microphones, cameras, movement, speed, distance, switches
- Indicators - LEDs, buzzers, speakers, displays
- DC Motors - Servos (airplane flaps / car steering), Stepper (precise movement), Brushed DC (requires motor controller called an ESC), Brushless DC (drones, requires an ESC Electronic Speed Controller)
- ESC is an Electronic Speed Controller, a specialized board controlled by a PWM signal (Pulse Width Modulation)

Andrew’s Car

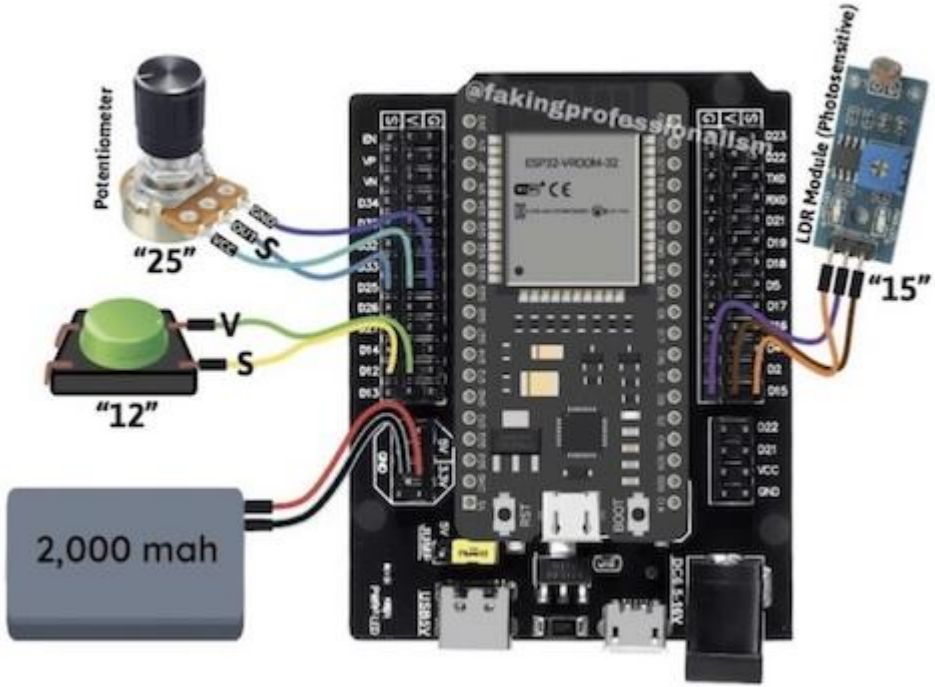
- A microcontroller, motors, motor driver, battery, voltage boost converter (3.7 volts to 12 volts)
- 3D printed and laser cut body





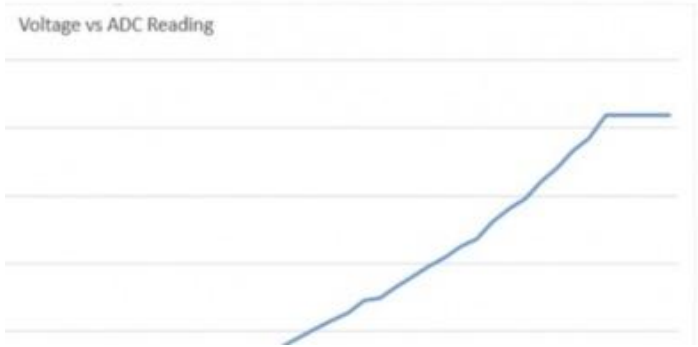
Inputs

- Button (digital)
- Battery (analog)
- Light level sensor (analog)
- Potentiometer or “pot” (analog variable resistor)



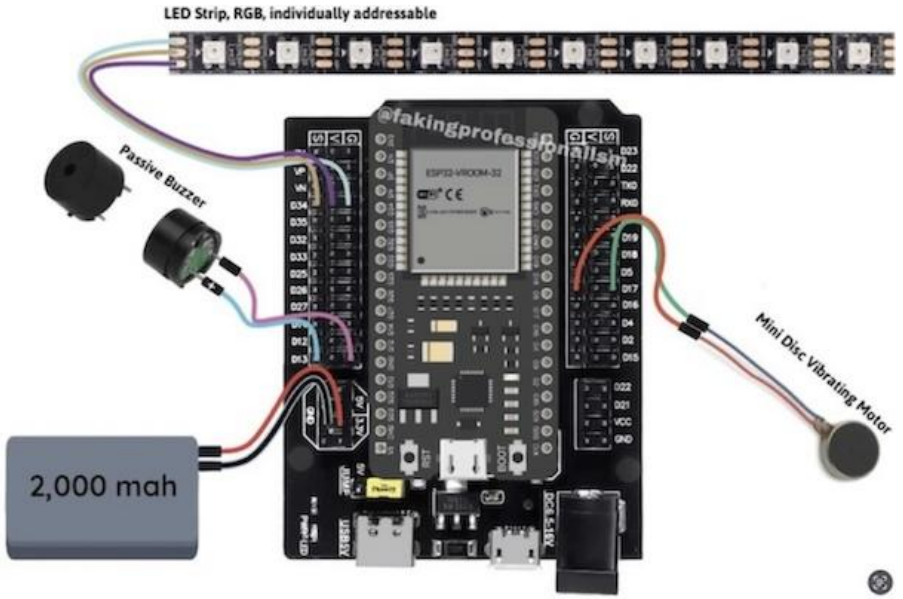
More About Analog Inputs

- Pushbuttons - Pull up and pull down resistors
- Voltage inputs - Analog to digital converters (ADC), battery voltages (3.3v)
- Higher voltages - Require voltage dividers
- Current input - Current shunt (a precision, low-resistance resistor)
- Potentiometers or “Pot” (type of voltage divider)
- ESP32 issue with reading 3.3v vs 3.2v
- Some devices have better ADCs



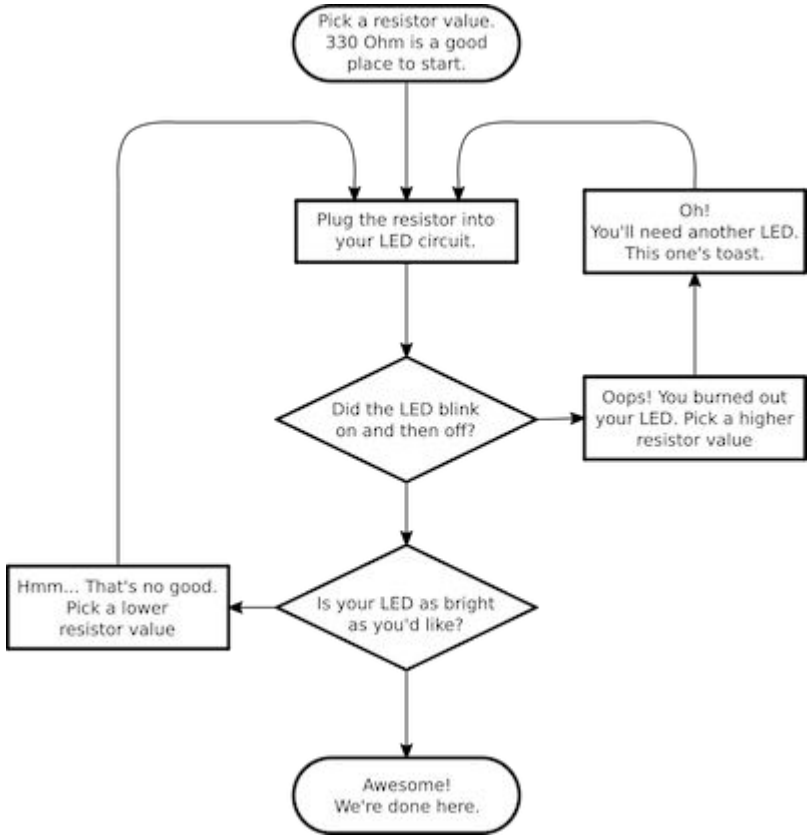
Outputs

- Passive buzzer (analog speaker)
- Vibrating motor (PWM)
- NeoPixel (WS2812) LED strip (digital, one output for all)
- Battery (but these pins can be voltage output as well)



More About Analog Outputs

- LED - One color for each LED and needs a current limiting resistor
- Beeper or Buzzer - Can also use this as a speaker
- PWM - Pulse Width Modulation to control LED brightness, servos for precise repeatable positioning, and control speed in motor controllers

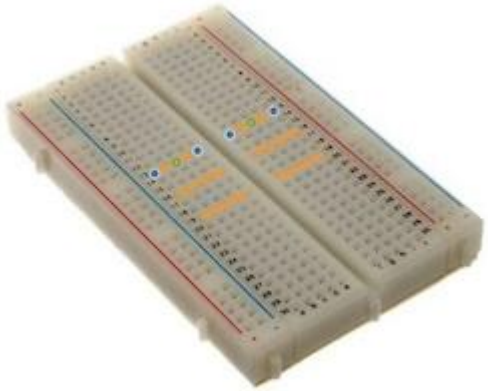


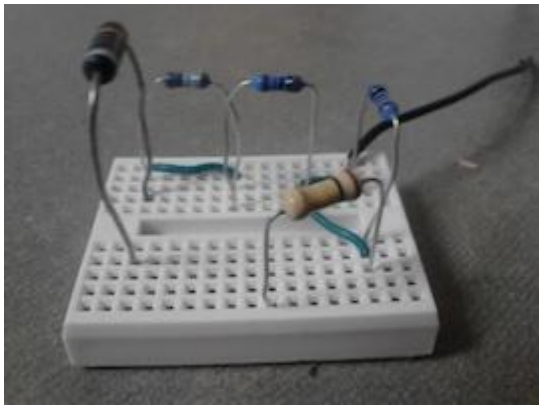
Digital Inputs and Outputs - Outputs

- “Busses” - I2C (inter-integrated circuit bus) and SPI (serial peripheral interface bus)
- Temperature / Humidity - DHT22
- Distance - HC-SR04 “sonar” (time of flight) sensor
- Neopixel LEDs - Adafruit guide and a technical guide
- LCD Displays - Two line display guide (I2C)
- LED Matrix Displays - MAX7919 guide
- OLED Displays - SSD1306 guide

Building Circuits

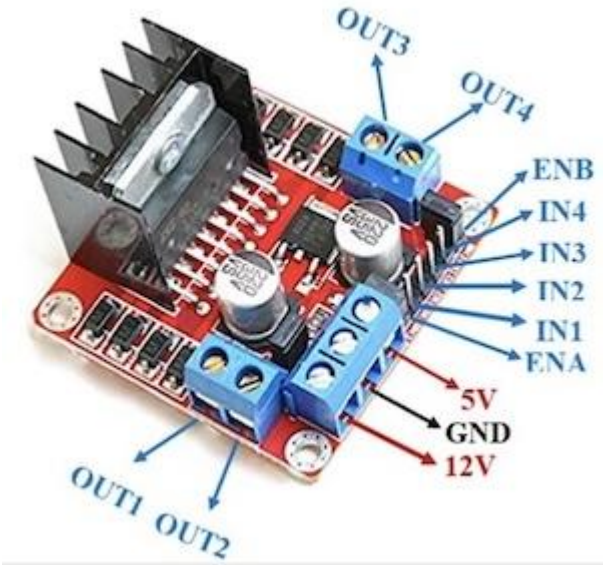
- Breadboards (no soldering)
- Breakout boards - easier access to each pin
- Printed Circuit Boards (PCBs) - As low as $2 for 5 but requires design skills (KiCad) and soldering skills
- An expert level example video





Troubleshooting

- Requires a systematic approach
- Break down and simplify the problem by isolating parts
- Review the spec sheet, confirm inputs, confirm outputs
- Test power to motor then test motor then test connections
- There’s a whole book on it



MicroBlocks Programming

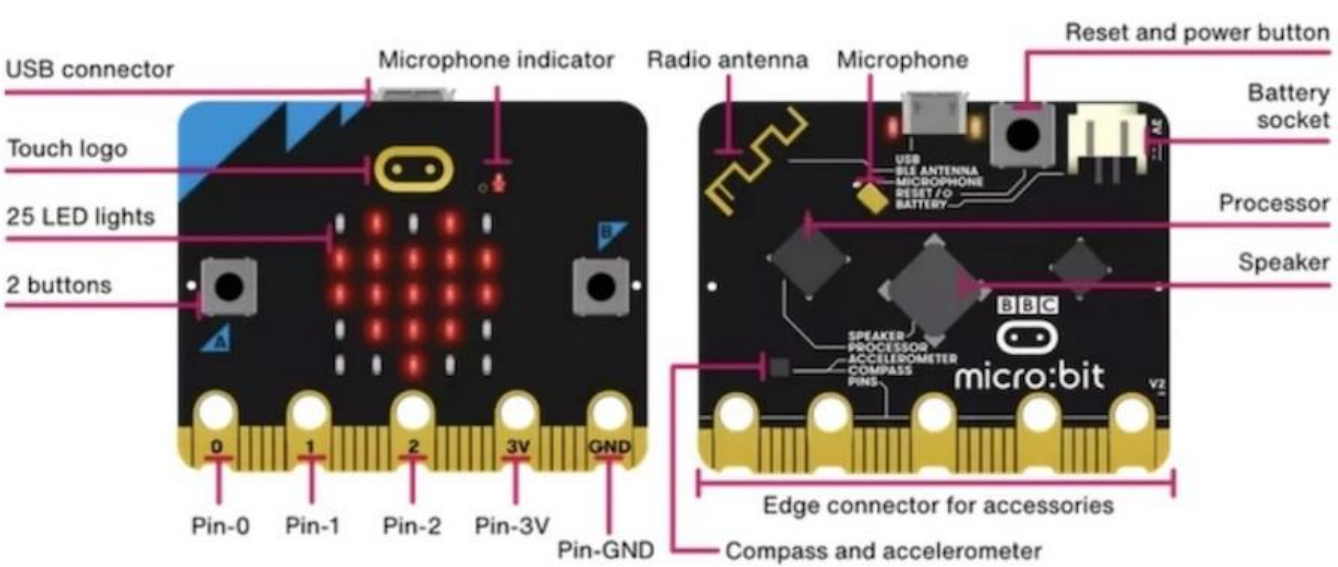
- Easy way to get started with real-time programming
- Update firmware first with detailed instructions on the Make Wiki
- Command, reporter, “C,” “hat” blocks and other toolboxes
- Using Libraries and the Help System
- The “Hello World” of microcontrollers - Blink!
- Learn about reporter blocks with the tilt demo

MicroBlocks Programming

- Load LED DISPLAY library
- Display a letter/number on LED
- Display longer text and numbers on LED
- Display icon pictures selected from the menu
- Program button A / B: display text, picture, animation
- Discover the ENCODED information - everything is a number

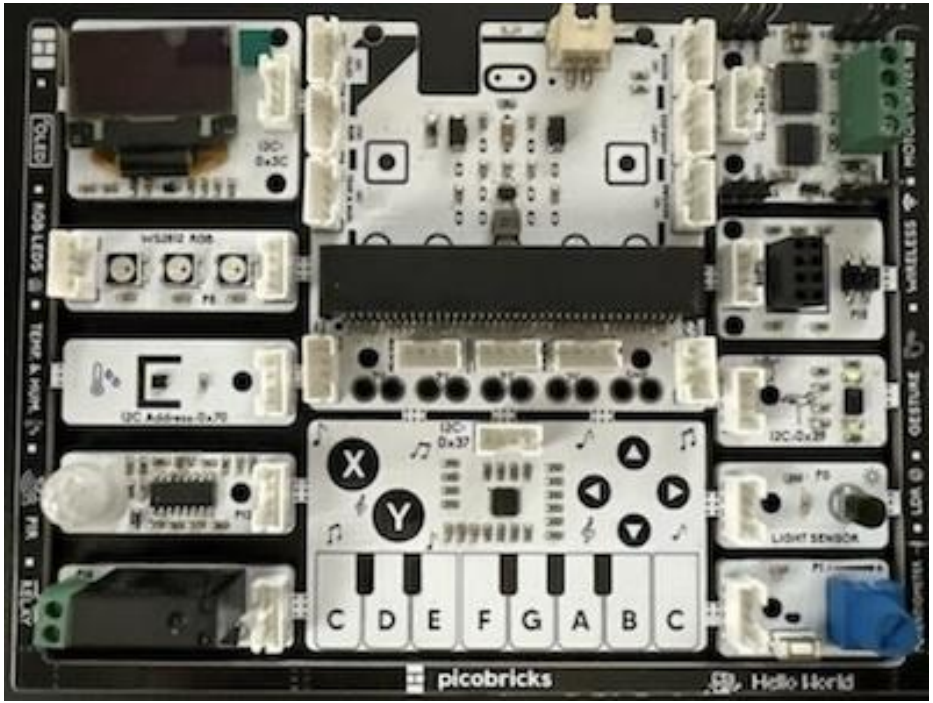
Micro:bit Microcontroller

- Micro:bit Overview - Open-source hardware with a processor, sensors, Bluetooth/USB, 25 LEDs, two buttons, microphone, USB or battery power



PicoBricks

- An electronic development board that includes several sensor modules
- OLED display, Neopixels, temperature, presence, relay, keyboard, motor/servo drivers, communications, gesture, light, and a potentiometer



PicoBricks Exploration

- Explore hardware features: Display, RGB LEDs, Temp/Humidity sensor, Passive IR (PIR) sensor, Relay, Motor module, Gesture sensor, LDR Light sensor, Potentiometer, Touch key sensor
- Learn about basic operational principles and use of the modules
- Display: bitmap images, print information and graphics
- RGB LEDS: how colors are made
- Temp/humidity sensor: monitor the environment, activate AC/Heating/Watering (greenhouse / home automation applications)

PicoBricks Activities

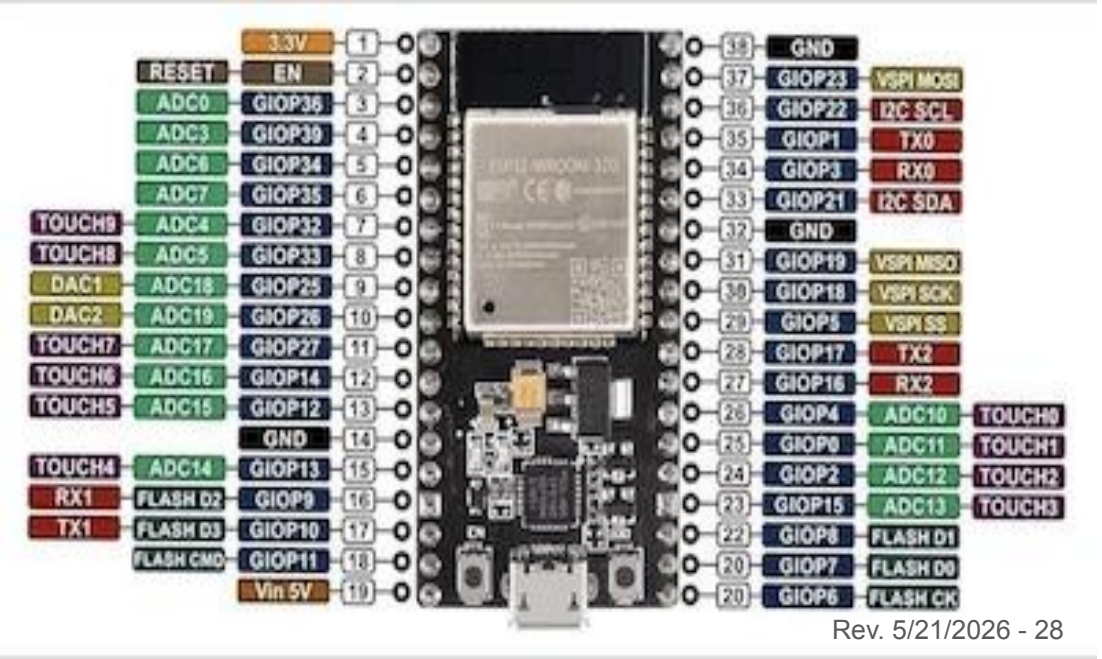
- PIR sensor: detect motion in area, use to build alarm systems, monitor activity
- Relay: turn power on/off, use to control devices, motors, lights, appliances
- Motor module: DC / Servos, cars, robots, open/close entrances
- Gesture sensor: detect hand motions nearby, use to control activities remotely
- LDR Light sensor: detect light levels, use to control lights or window shades
- Potentiometer: turn to change, use for input for sound, light, temp, etc control
- Touch Key sensor: activates when touch is detected, use for control of anything you can think of or play music/games

Advanced Project with PicoBricks

- Let’s make a letter and display it on the screen: first with micro:bit, then on OLED display
- OLED display: text in various sizes, geometric shapes
- What’s with all the blinking going on the OLED display, how do we get rid of it?
- Program the pot to control OLED brightness using backlight
- RGB colors (0-255): primary colors and combination colors
- Program NEOPixels for desired patterns, colors, movements
- Program to display temp/humidity on the OLED screen
- Explore displaying values in multiple lines, in various sizes
- Program the PIR sensor to alert for motion and sound an alarm
- Motor module: make motors and servos move
- Program hand movements to control micro:bit or other modules, display pictures
- Program LDR light sensor to dim/brighten OLED display
- Program to plot potentiometer values, rescale values to change (for example): size, volume, brightness, etc.
- Use a touch key sensor to play musical notes and display the note names on the OLED Display. Make the note play duration controlled by the user (HINT: need sensor input)
- Display multiple sensor values on the OLED and update them in real-time Eg: temp, potentiometer, Light sensor, etc.
- Design and implement a system with sensor detection and corresponding action. Eg: Alarm system, Appliance use automation, turn on/off motors, etc.
- Design a mini broadcast system based on Radio that transmits messages, pictures, sensor values over different channels. Users will tune into the channel and receive the information.
- Use joystick to remotely control a servo over Wifi (requires two micro:bit controllers)

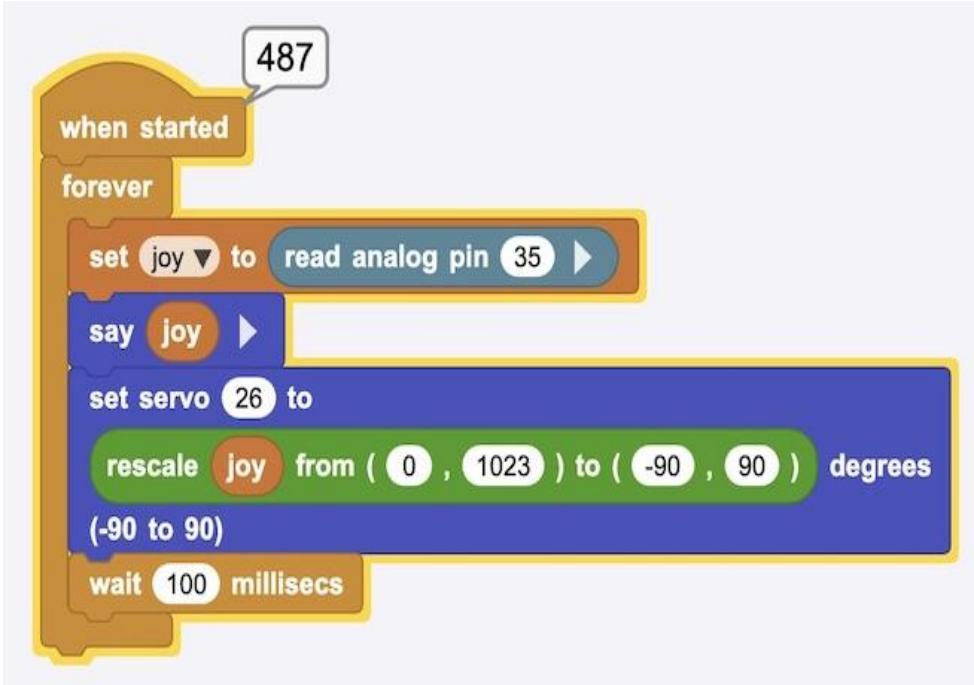
ESP32 Tips and Tricks

- Avoid the forbidden pins by referencing this tutorial
- Safe pins for input AND output: 13, 18, 19, 21, 22, 23, 25, 26, 27, 32, 33
- More ESP32s / YouTube video
- Pinout changes by model
- ADC Pins: 25, 26, 27, 32, 33
- BUT 25, 26, 27 are for WiFi!
- BUT 34, 35, 36, 39 are inputs
- SO 32-36 & 39 are ok for analog
- So much for General Purpose I/O
- Make Wiki Page



Joystick and Servo Demo

- Can connect small servos to the ESP32 directly (needs 5v power)
- Connect joystick to pin 35: 3.3v to “+5v” pin, GND, VrX or VrY
- Output of Say (Analog Pin 36) should be around 500
- Connect servo to pin 26: Red (5v), Brown (Gnd), Orange (Signal)
- Hands-on is the [best way to learn]



Cutebot Pro Car Kit

- A programmable robot with a 4-way infrared line-following sensor, an encoder motor, the LED rainbow light, an ultrasonic sensor, and other devices

CuteBot Overview

- Hardware features: Chassis, power switch, battery, encoder motors, Micro:bit slot, distance sensor, IR sensor, headlights (LEDs), NEOPixel Lights, line tracking sensors, calibration button, Expansion ports
- Basics of Robotics 3C's: communication, command, and control.
- Communication: IR Remote control, Radio Control
- Command: manipulating the car using library commands that are preprogrammed, getting information from the sensors
- Control: maneuvering the car in various ways: by motor power %, by distance, by wheel rotation, programming responses to sensor data, acting on external events.
- What is a Control LOOP? What is programmed versus continuous control?
- What is a Line tracking sensor, how does it work?
- What is an Encoder and why do we need it?
- What are minimal car controls that we need to successfully maneuver the car?
- What is an IR Remote Controller, how does it work?

Cutebot Projects

- Program car movements: forward, backward, turns in place
- Program turns by rotation angles forward and backwards
- Program car movements using sensor values: distance, speed, obstacles
- Program a script to detect lines
- Program a control script using IR Remote for car control
- Program displays for telemetry information display
- Program obstacle avoidance using the distance sensor
- Program the car to follow the track
- Program a control script using radio for car control
- Enhance car movements by using the headlights, NEOPixels, and sounds

Other Programming Options: MicroPython

- Python designed to work under limitations like those found in microcontrollers
- Includes modules to access low-level hardware like GPIO
- Several steps required before it can be used with a microcontroller (upload base firmware, download and use the Thonny IDE, etc.)
- Tutorials are available and AI can be used to write code

Other Programming Options: Arduino IDE

- Web based Arduino code editor (IDE) where you can write code, access libraries, and upload to board (ESP32 and other boards as well)
- Tutorials are available and a language reference
- The Arduino Cookbook
- Many kits are available: For less than $30 get ESP32, temperature/humidity, distance, joystick, infrared, and other misc parts
- More kits are available on the Make Electronics Wiki page

Analog Components

- Capacitor - Stores energy in an electric field (like a tiny tank). Two conductive plates separated by an insulator.
- Inductor - Stores energy in a magnetic field (like a tiny water wheel connected to a mass). A coil of wire.
- Resistor - Resists the flow of electrons like a thinner piece of pipe. A wire with resistive material (carbon, etc.)
- Lots of videos online

Other Components and Circuits

- Diode - Allows current to flow only in one direction
- Transformer - Two inductors coupled by a magnetic field
- Transistor - Semiconductor used to amplify or switch
- Integrated Circuits (ICs) - Combines transistors, resistors, and capacitors
- Series and parallel circuits
- Analog Circuits - Current or voltage vary continuously with time
- Digital Circuit - Voltages are assigned to discrete values (ex. 0=0v / 1=3.3 or 5v) to represent logical and numeric values

USB-C Power

- Unlike USB-A the voltage can vary and be negotiated with a trigger board
- Standard voltage steps are 5 V, 9 V, 15 V, 20 V, and 28V
- Formal name is USB-C Power Delivery
- Qualcomm has a proprietary standard called QC
- Experimental boards: HUSB238 and ZY12PDN
- A more complete guide and a video

Solar Powered Systems

- Several factors - wattage, temperature, type, orientation...
- Mono-crystalline - higher efficiency / cost than poly
- 400-watt panel can produce about 1,600 watts / day
- 400-700 watts - can power a range of small appliances
- 1400 watts - Running a small air conditioner for 12 hours
- Semi DIY $450 - 200 Watts 12 Volt/24 Volt Mono-crystalline Solar Panels with 30A PWM Charge Controller and 100aH LiFePO4 Battery
- Solar "Generator" System $700 - 200W Solar Panel, 1070 Wh LiFePO4 Battery,1500W AC/100W USB-C Output

Batteries

- Lithium Ion (Li-ion) - Good general rechargeable, usually cylindrical. Found in phones, laptops, flashlights, etc.
- Lithium Polymer (LiPo) - Higher discharge rate, usually rectangular or pouch shaped
- Lithium Iron Phosphate (LiFePO4) - Heavier, 0%-100% every day, 2000+ charge cycles (vs 500), good for solar battery systems

Basis	Lithium Ion	Lithium Polymer
Ageing	Loses actual charging capacity over time	Retains charging capacity better than Lithium =ion
Energy Density	High Energy density	Low as compared to lithium ion
Conversion Rate	The capacity to convert battery into actual power 85-95%	75- 85%
Safety	More Volatile as compared to lithium polymer	More safety. Less chance of explosion
Cost	Cheaper	Slightly Expensive(+30%)
Weight	Heavier	Light Weight
Charging duration	Longer Charge	Comparatively Shorter

Light Emitting Diodes (LEDs)

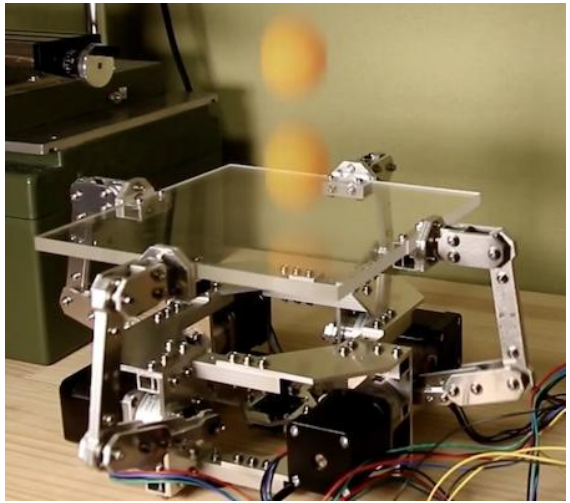
- LED - One color for each LED and needs a current limiting resistor
- Analog LEDs are very inexpensive (100 for $5) - Guide
- WS2812B / “NeoPixels” - Digital so one output for many (hundreds!)
- Requires microcontroller with 5 volt logic or a “level shifter”
- Many colors, shapes, and sizes - including wearable
- Super guide / Technical guide
- Adafruit wearable kits (and here and here)

LED Control and Other Microcontrollers

- Dedicated system to use an ESP32 via a web server/ smartphone app to control NeoPixels (aka WS2812) - WiFi only (no Bluetooth)
- WLED project website. Installation website
- Circuit Playground “Ring” with sensors, LEDs, BLE (guide)
- LCD Displays - Two line display guide (I2C)
- LED Displays - MAX7919 guide
- OLED Displays - SSD1306 guide

Miscellaneous Interesting Devices / Projects

- Noods - Flexible LEDs “Noodles”
- Special Purpose Controllers - Proffieboard for lightsabers
- Robot kit or maybe a 3D printed self balancing robot
- Retro gaming system / Musical instrument
- DIY weather station
- Computer vision / the “bouncer”



Where to Go From Here

- Sparkfun Engineering Essentials
- Hackspace free Maker Books
- Adafruit general guide and IoT guide
- Arduino project hub (over 5,500 projects)
- PiMyLifeUp (Arduino, Linux, Electronics)
- Circuit simulation
- Digikey Maker projects
- YouTube multi-hour electronics playlist
- Links from class on the Make Electronics Wiki