

# Crash Course to Microcontrollers

Presented by tom

And colin

# What is a microcontroller

- A tiny programmable computer on a single chip
- It can read inputs (from buttons, sensors, whatever)
- It can control outputs (like LEDs)

**50 x Xiao RPN2040**  
**25 x Xiao ESP32C6**  
**25 x Xiao ESP32S3 Sense**  
**25 x Xiao nR52840**  
**5 x Grove Creator Kits**



# MakeMIT relevant reference

- > The Arduino Nano is a 5V chip. Feed it 5V please.
  - > The ESP32 is a 3.3V chip. Feed it 5V only through the VIN/BAT pin (as there is a converter on it) and do not put 5V on any of the other pins.
  - >
  - >
  - >
- 
- These servos can be directly hooked up to an Arduino. Alongside power and ground, for other servos, hook up power and ground separately, and make sure to tie grounds together.

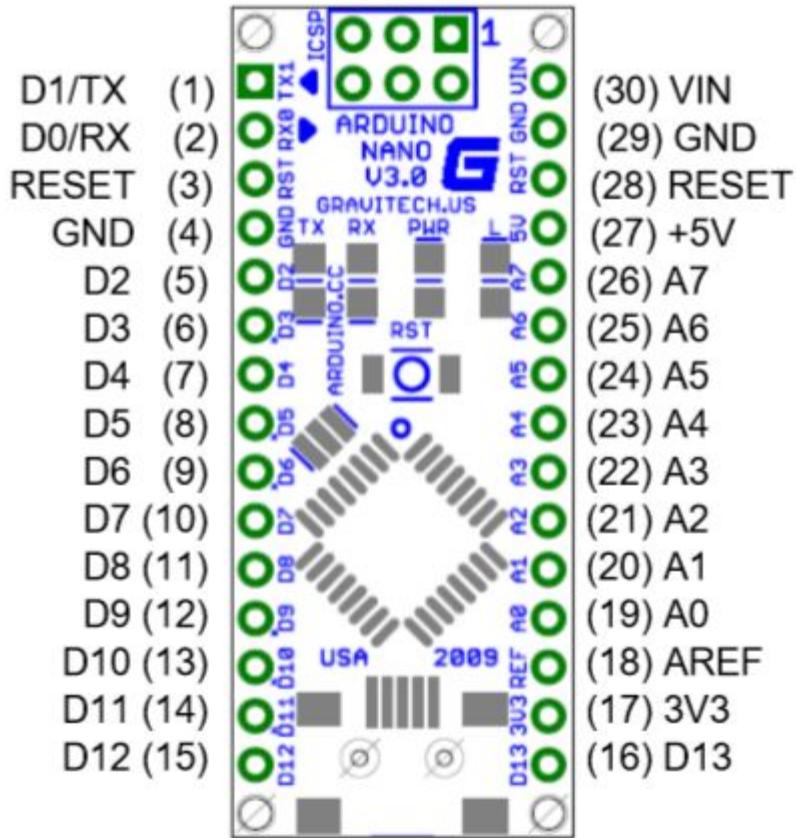
# MakeMIT relevant reference 2



For stepper motors you need to use a motor driver. We will have plenty of the [A4988 stepper motor driver.](#)

For DC motors we will have the [DRV8833 motor driver.](#)

For brushless motors we will have hobby grade ESC. You can provide them a PWM signal as if they were a servo.



Pin No	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A0-A7	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

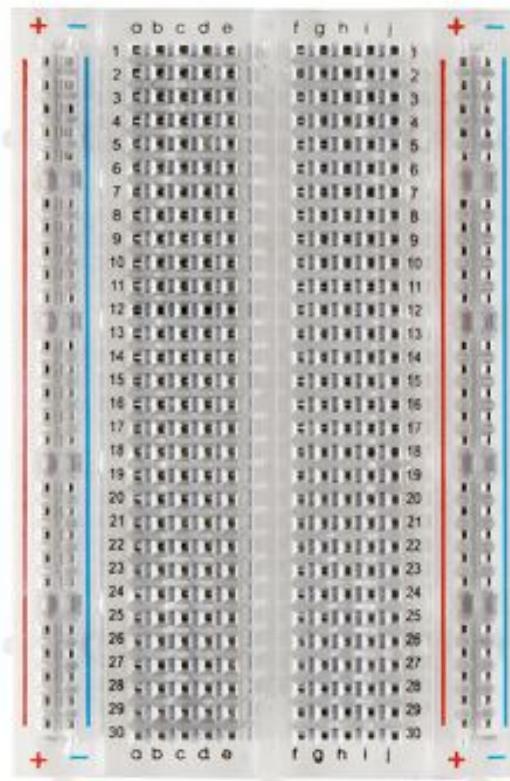
Tom will yap about Arduino IDE and how to connect to ur arduino now

You may have to install CH340 drivers

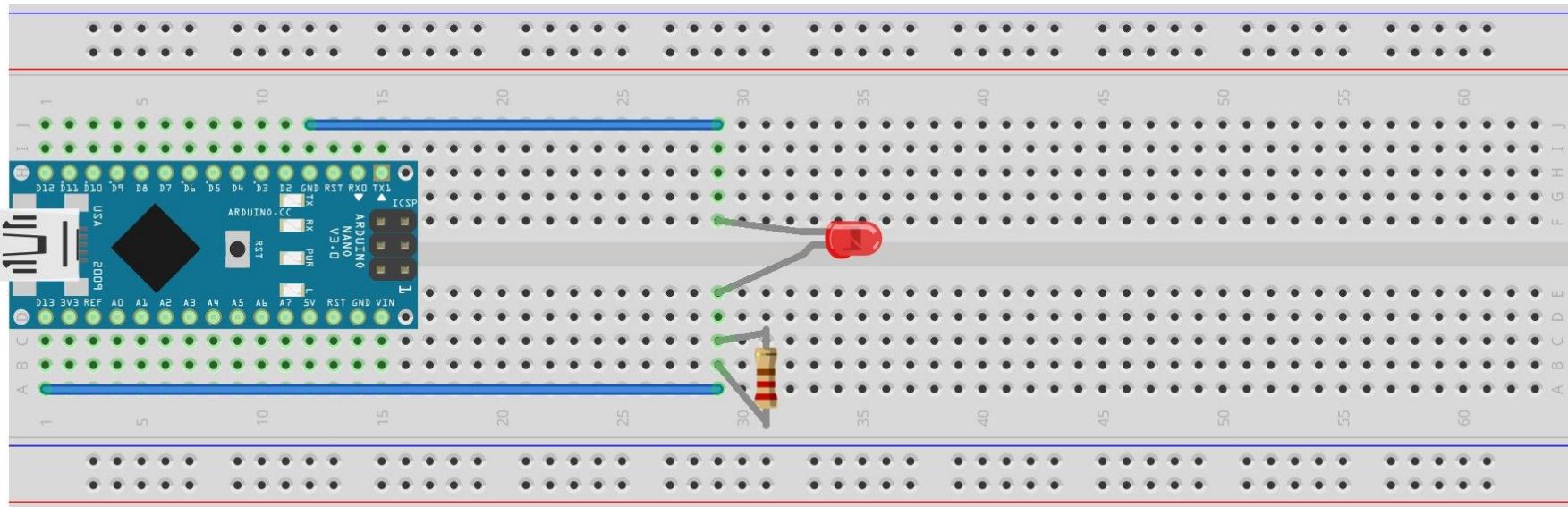
[How to Install CH340 Drivers - SparkFun Learn](#)

<https://tinyurl.com/rhxa3s57>

# Bread board :3



# Breadboarding (do this rn)

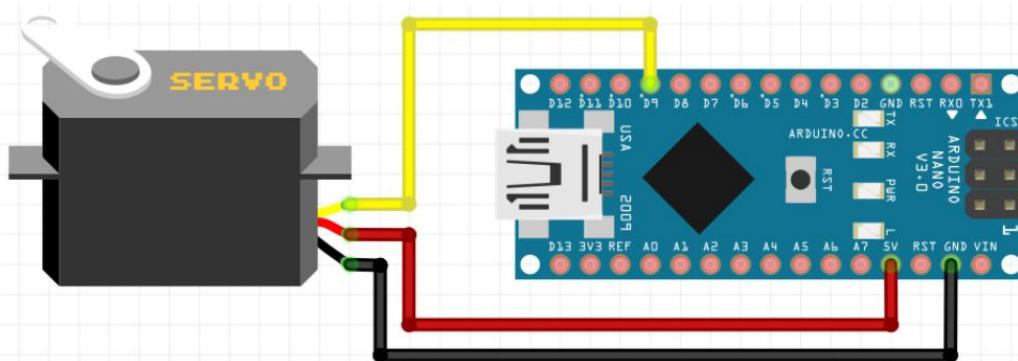


fritzing

```
int redPin = 13;
void setup() {
    // initialize Leds
    pinMode(redPin, OUTPUT);
}
void loop() {
    digitalWrite(redPin, HIGH);
    delay(1000);
    digitalWrite(redPin, LOW);
    delay(1000);
}
```



# Try to do a servo



Yellow wire is signal D9  
Red is positive 5V  
Black is negative GND

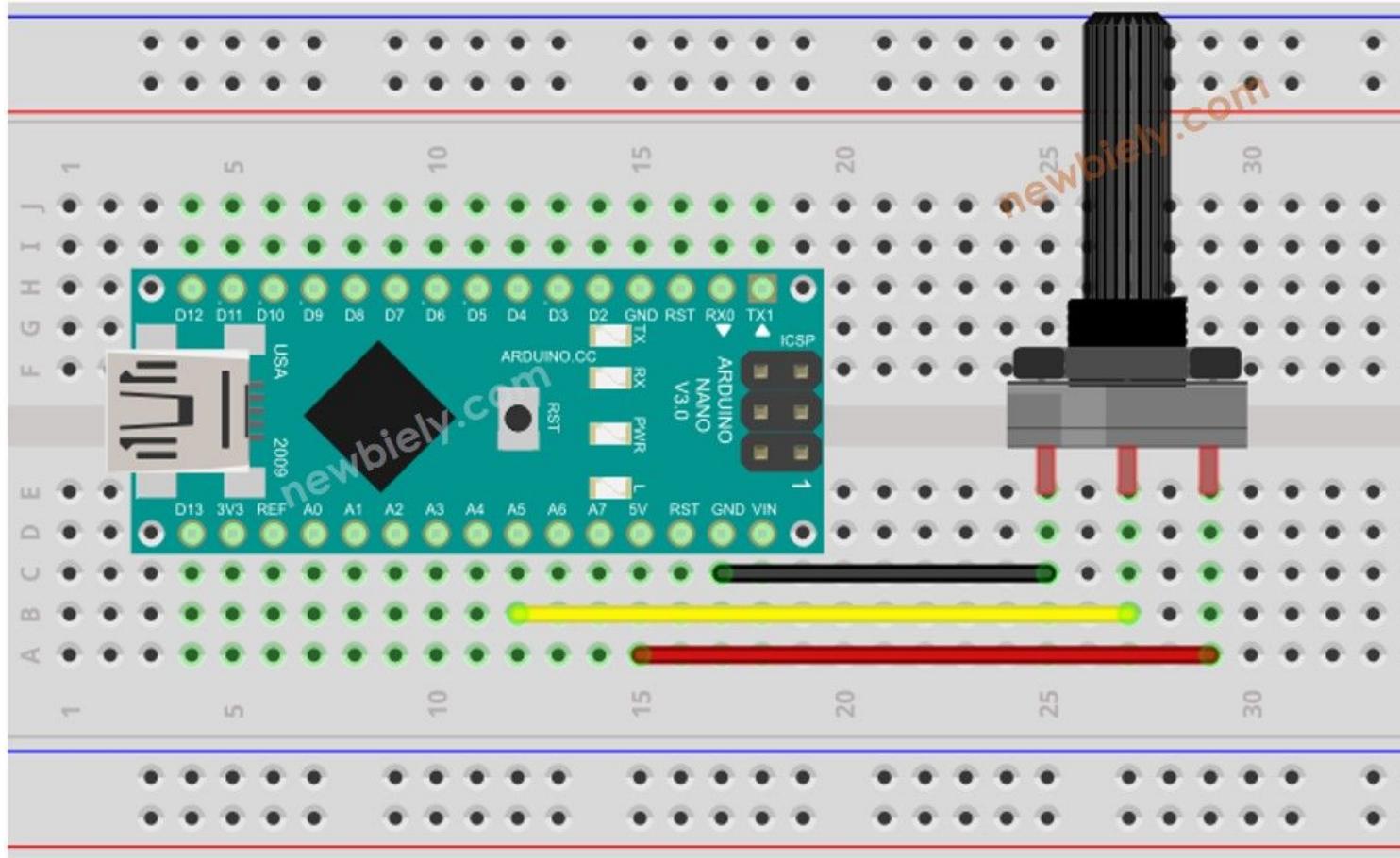
itzing

## Important Note:

For any servos bigger than the tiny blue servos, the Power (Red) and GND (Black) must come from a separate 5V source. It cannot come from the Arduino.

You will also need to tie the grounds together between the servo and Arduino (ask ChatGPT what the means if you end up using a bigger servo during the hackathon)

```
1 #include <Servo.h>
2
3 Servo pogchamp;
4
5 void setup() {
6     pogchamp.attach(9);
7 }
8
9 void loop() {
10    pogchamp.write(0);
11    delay(1000);
12    pogchamp.write(180);
13    delay(1000);
14 }
```



[newbiely.com](http://newbiely.com)

```
1  #include <Servo.h>
2
3  Servo pogchamp;
4  int variablePot;
5  int angle;
6
7  void setup() {
8      pogchamp.attach(9);
9      pinMode(A5, INPUT);
10 }
11
12 void loop() {
13     variablePot = analogRead(A5);
14     angle = map(variablePot, 0, 1023, 0, 180);
15     pogchamp.write(angle);
16     delay(15);
17 }
```

# ESP32 Section

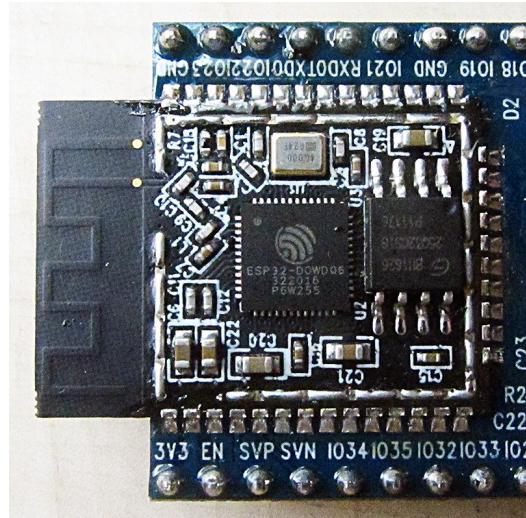
“Much Wow”



# Why use ESP32?

Bluetooth

WiFi



16Mb Flash

10-240Mhz

Can technically  
run Minecraft

# ESP32 Project Ideas



IoT Devices

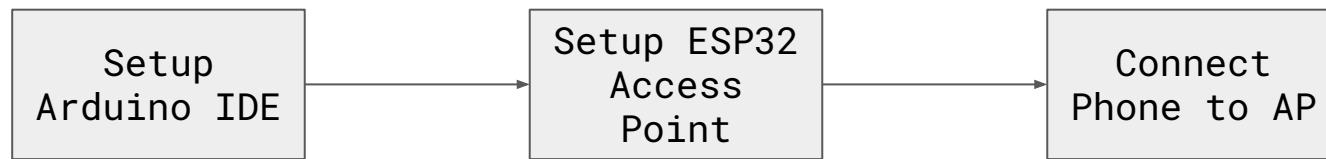


Wifi Snooping



Ethical Flying Machines

# Today's Project



# Setting Up ESP32 in Arduino IDE

File -> Preferences -> Additional Board Manager URLs -> [https://bit.ly/esp32\\_index](https://bit.ly/esp32_index)

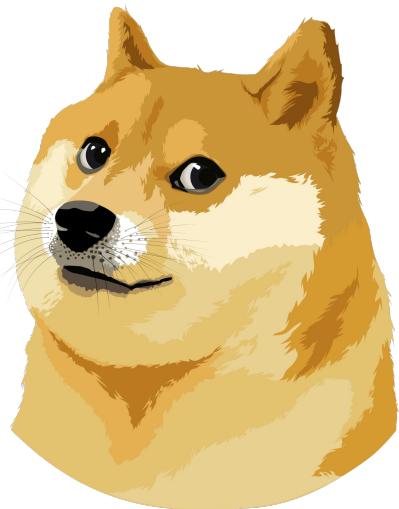
Boards -> ESP32 (by Espressif)

# Setup ESP32 Access Point

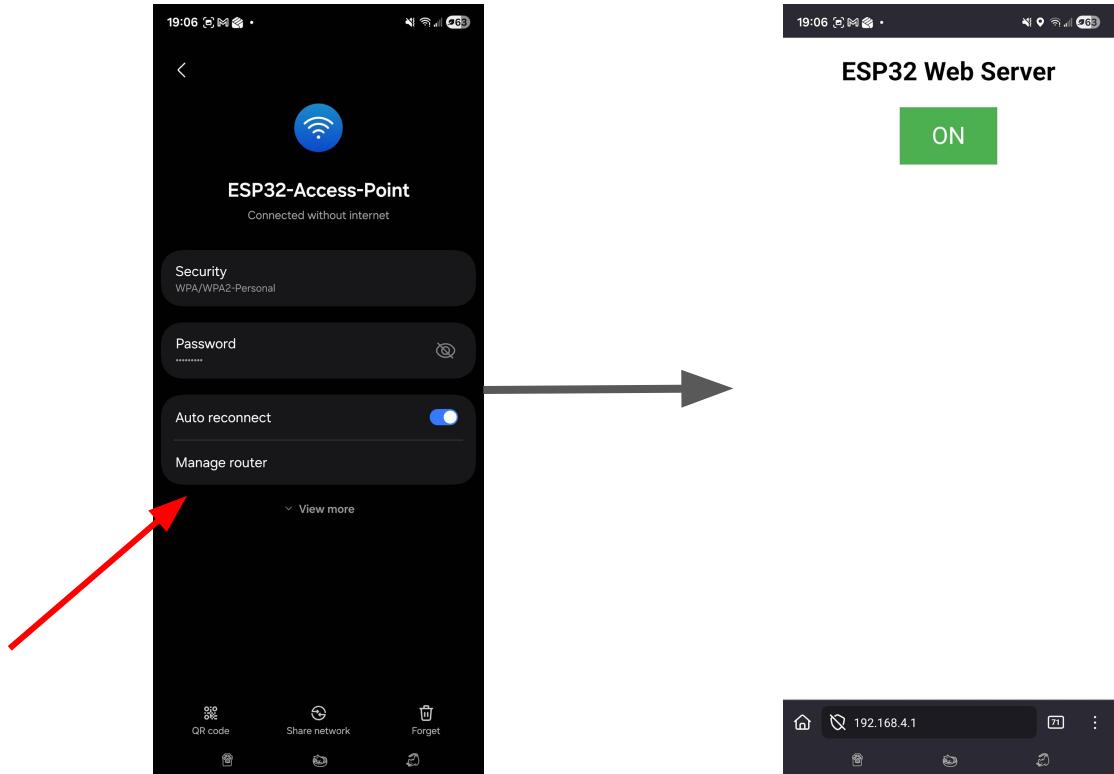


[github.com/SuperC03/  
makemit-slides](https://github.com/SuperC03/makemit-slides)

uc -> access\_point.c



# Connect to AP



Probably 192.168.4.2