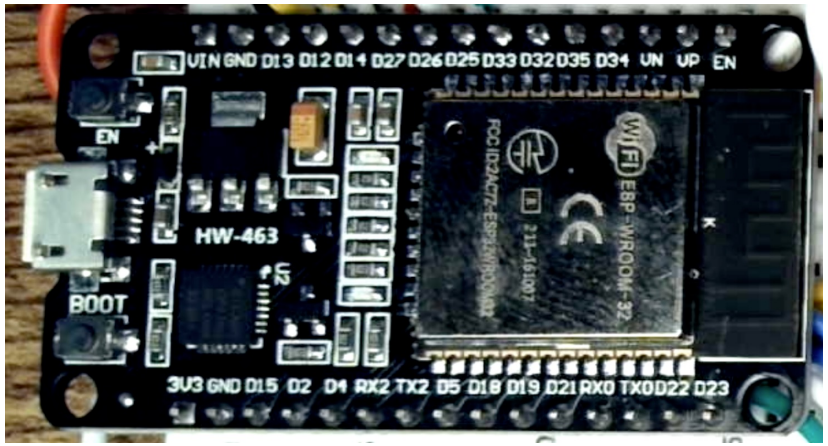
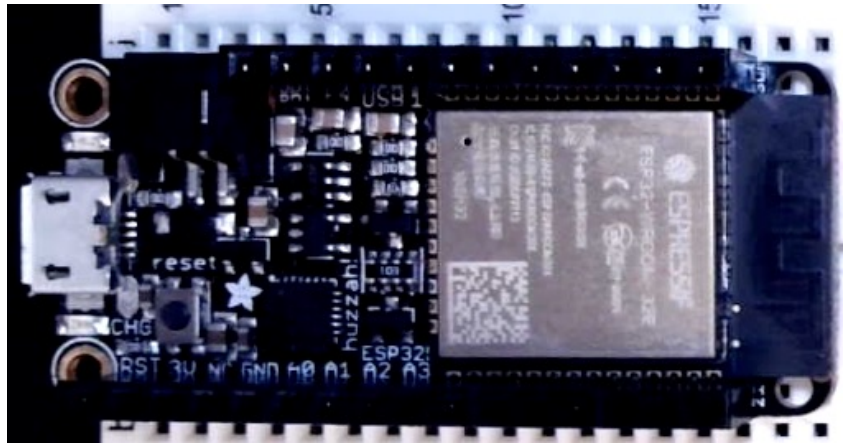
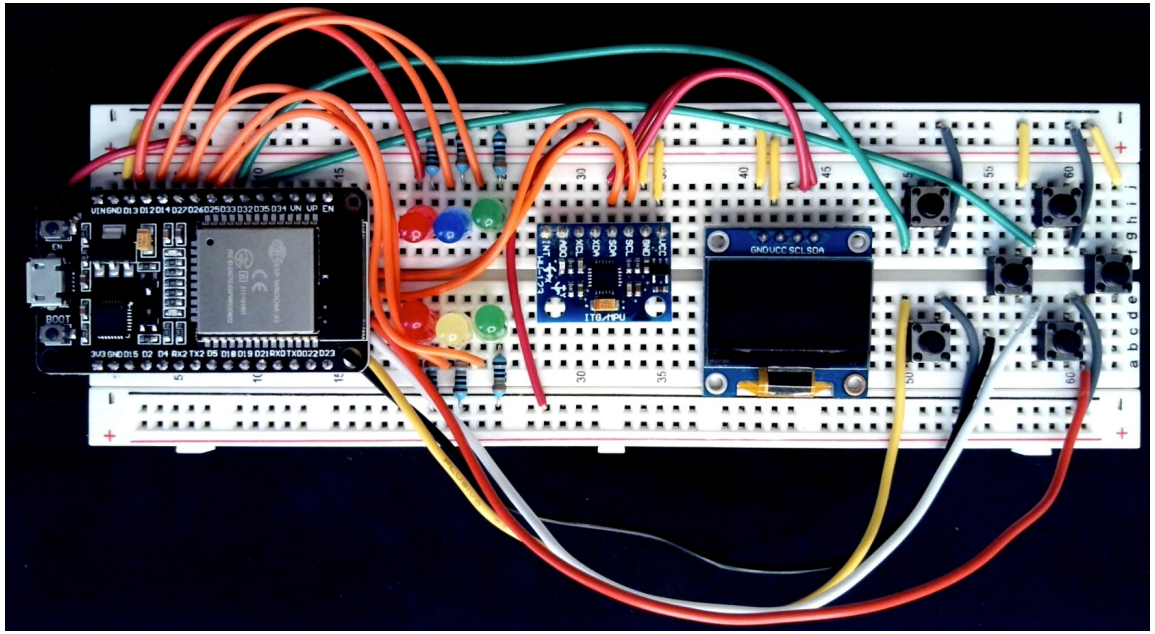
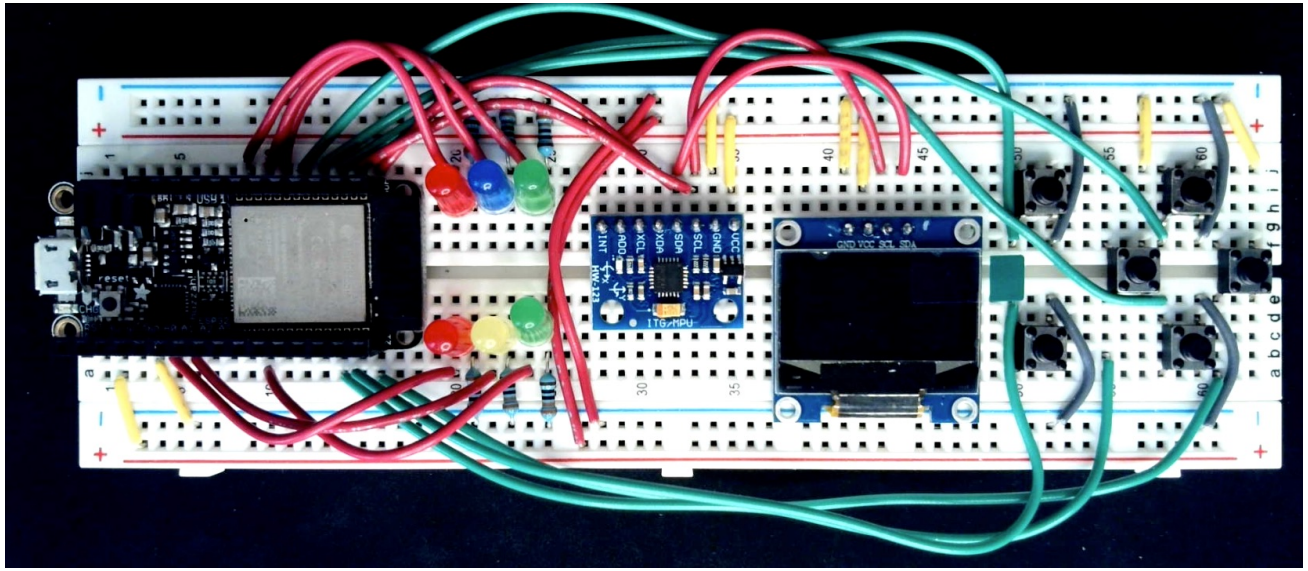



# ESP32 Educational Kit: EEK

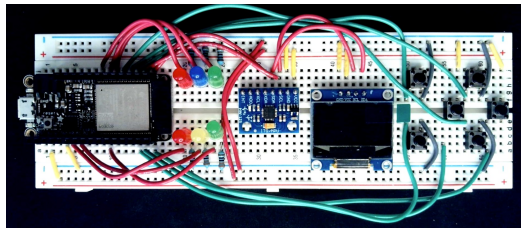
## Part 1: Assembling and Exploring the Assembled EEK





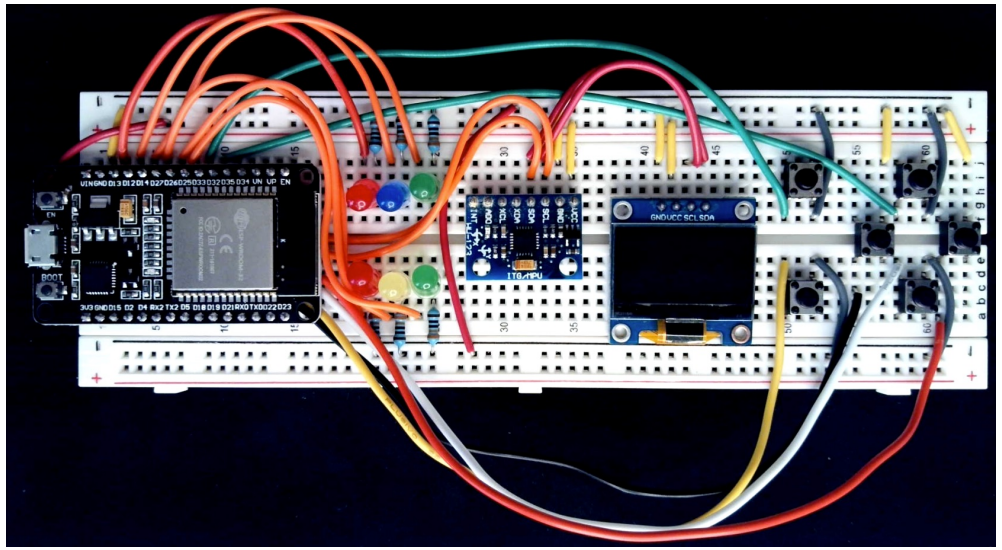
# ESP32 Education Kit: EEK

About this course and its organization



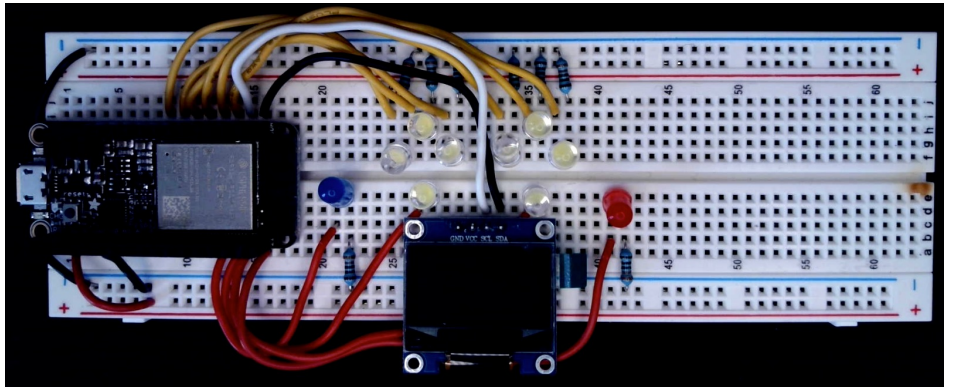
# EEK course concept and design

- The course is designed to include "just enough" hardware elements that, when combined, implement and demonstrate interesting applications
- The course assumes minimal programming experience and no Arduino background
- The course covers and requires no other MCU platforms besides the ESP32

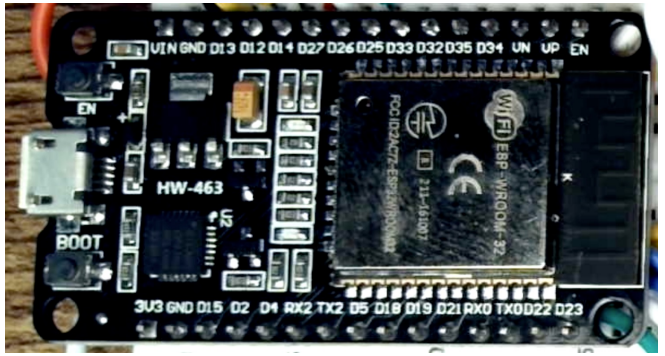


# Building on EEK course Part 1

- Cyber-Physical context is central: controls and effects leading eventually to controlled drone flight
- Part 2 of the course will feature another physical MCU and/or a drone like a Tello to expand the Cyber-Physical context even more
- Learn to connect other sensors and actuators to EEK including gamepads and environment monitoring devices
- Explore other IoT simulations such as home environment control scenarios and fall detection



# Outline for this Course: EEK Part 1




1. EEK Introduction
2. Arduino IDE setup for EEK and Course
3. Survey of ESP32 boards and their compatibility with EEK
   - Two boards are fully supported
4. EEK Breadboard Assembly
5. EEK Basics: Hardware components and examples
   - Most lessons end with a Mini-Project
   - Capstone Mini-Project brings together most of the material covered
6. EEK WiFi
7. EEK Bluetooth and Gamepad integration

# Needed Software and Host Computer Support

- Windows or Mac Computer
  - Windows 10 (Windows 11 has not been tested) or Mac OS Ventura
    - Testing with Mac OS Sonoma expected soon
  - Linux computer may work since Arduino IDE is supported there (not tested)
- Latest version of Arduino IDE installed (covered in course section 2)
  - Version 2.2.1 was used primarily during course development
- USB serial cable compatible with your ESP32 board
  - Included in EEK kit; Android data-certified phone cable should work
- Updated drivers for USB serial cable support may be needed on Windows
- PacketSender app for UDP network testing (usage covered in course)
  - Browser is sufficient for TCP/IP network testing; needed only for section 7

# EEK Hardware

Breadboard

USB Serial Cable

ESP32 board

- LiPo Battery (Optional)

6 push button switches

6 LEDs

- LED leads trimmed with wire cutter

6 330-ohm resistors

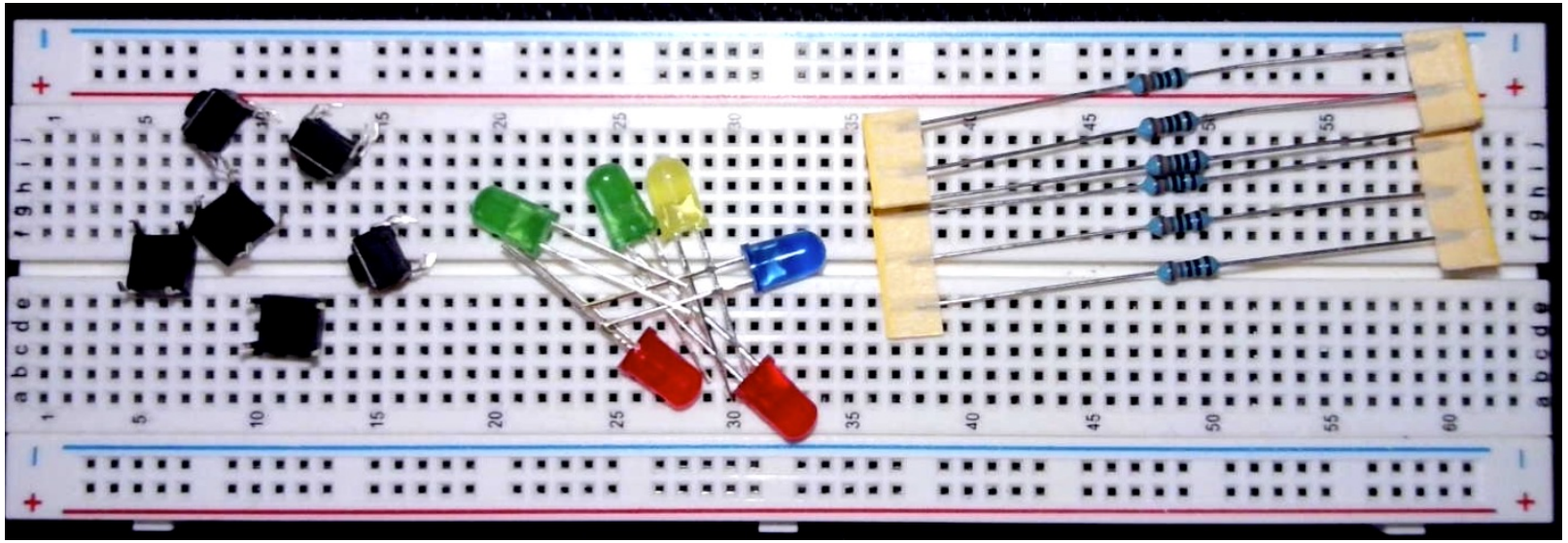
- Resistor wires trimmed with wire cutter

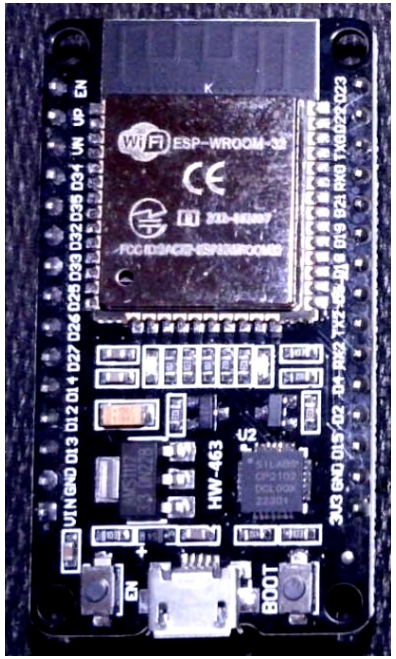
OLED display (SSD1306 compatible)

MPU 6050: accelerometer and gyroscope

Connecting wires sized to useful lengths

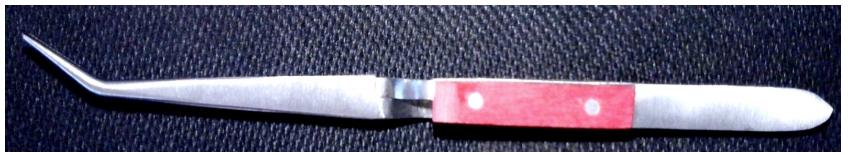
- You can use a spool of wire and a good wire stripper
- You can also get pre-cut wire
- Jumper wires often used with breadboard projects are **NOT** recommended

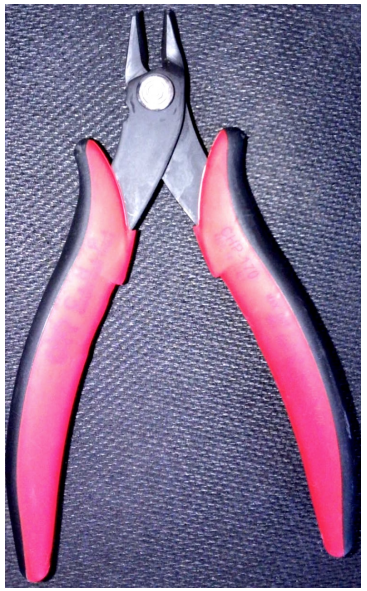


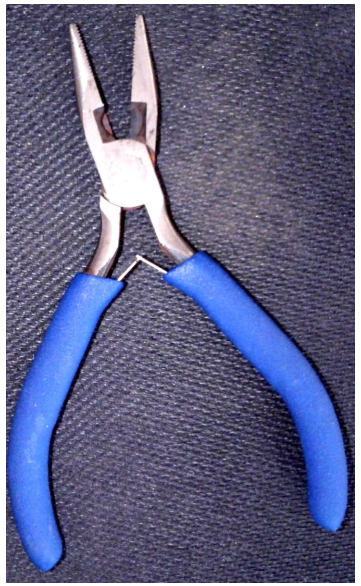


# Useful tools for assembly

- Tweezers
- Spudger/pry tool (small)
  - Less likely to bend ESP32 pins when removing from breadboard
- Magnifying glass
- Wire stripper
  - Cheap stripper usable but not recommended
- Wire cutter/snipper
- Needle nose pliers
  - Smaller is better



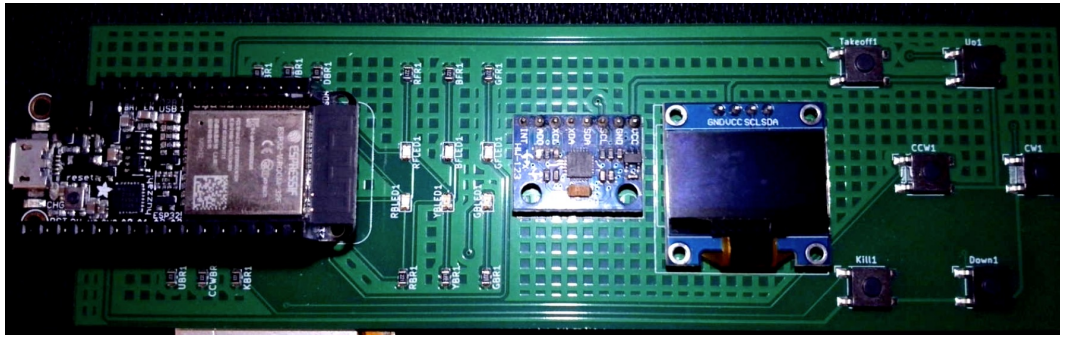




# How to acquire EEK components

- All EEK hardware components are available off the shelf
  - Original prototyping of EEK was done mostly with parts acquired from Amazon US
    - Amazon US List URL: https://www.amazon.com/hz/wishlist/ls/3MC9TVGZPJ79V?ref_=wl_share
  - Parts descriptions and individual links are supplied as a resource for this course
  - Parts can be acquired from many sources besides Amazon
- EEK Kits with all parts including pre-cut wires are available at the storefront:
  - https://shop.iotinurhand.store/
  - Instructor will personally fulfill orders made through this store
  - Both breadboard-based kits and PCB EEK versions are available
  - Exact product offerings and prices will change as EEK designs evolve
  - Prices do not include shipping
  - Buying an EEK kit is NOT required

EEK PCB alternative



# Setting up Arduino IDE for EEK

ESP32 Support in Arduino IDE

- Tech Explorations website has good information on Arduino and ESP32
  - The discussion and screen illustrations refer to the legacy Arduino IDE (latest version 1.8.19)
  - Most of the information still applies and Windows and Mac are both covered
  - A recent and brief YouTube video shows a Windows treatment with the latest IDE
- Latest Arduino IDE is version 2.2.1 and that has latest ESP32 board support
- You need to make two URL entries in the IDE Settings/Preferences and use Board Manager to install the needed libraries and examples

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

https://raw.githubusercontent.com/ricardoquesada/esp32-arduino-lib-builder/master/bluepad32_files/package_esp32_bluepad32_index.json
```

- After installing, you can check installation by connecting your ESP32 board to your host through the IDE and run some examples
  - You don't need to assemble the full EEK: just connect the ESP32 board directly

# ESP 32 boards

AdaFruit ESP32 Feather Huzzah
- Fully Supported and Recommended
- Higher cost: $28 with battery and headers
- Has LiPo battery support

ESP32 Dev Board Clone (30 pin version)
- Fully Supported
- Inexpensive: Under $10
- USB connector seems fragile

Arduino ESP32 Nano
- Bluetooth Bluepad32 library not functional
- Rest of EEK examples should work

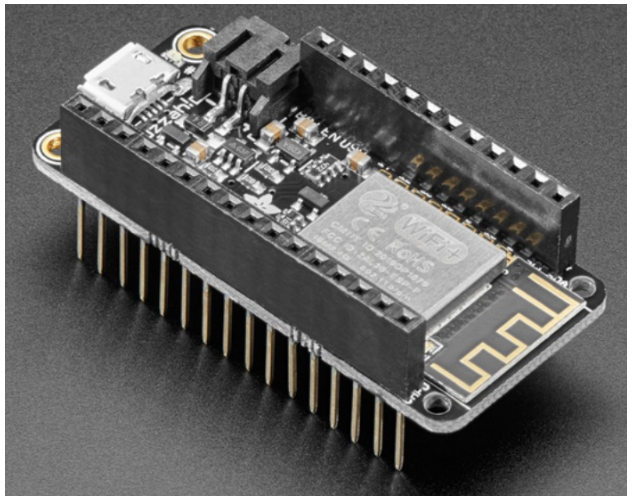
AdaFruit ESP32 Feather S3
- Arduino IDE has some limitations
- Has LiPo battery support
- Does not come with headers

ESP32 Dev Board 38 pin version
- Should work but EEK breadboard will be cramped with all components added

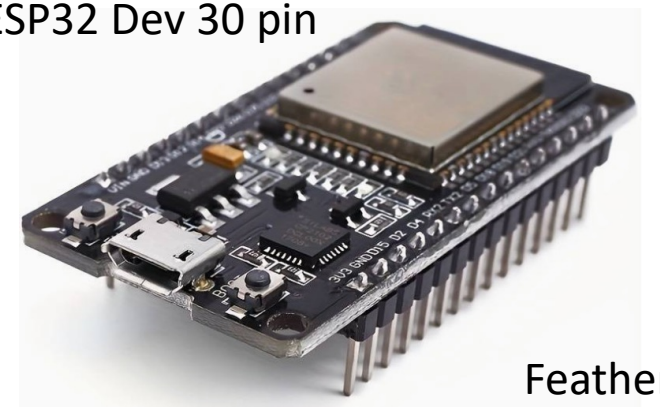
Other ESP32 boards may work but have not been tested

Some breadboard designs using these boards for similar applications are shown in subsequent slides

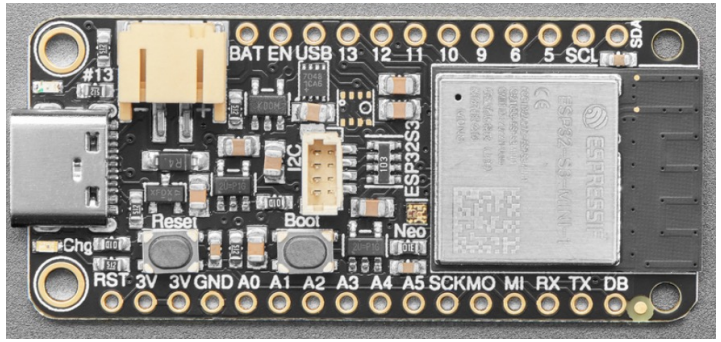
Feather Huzzah



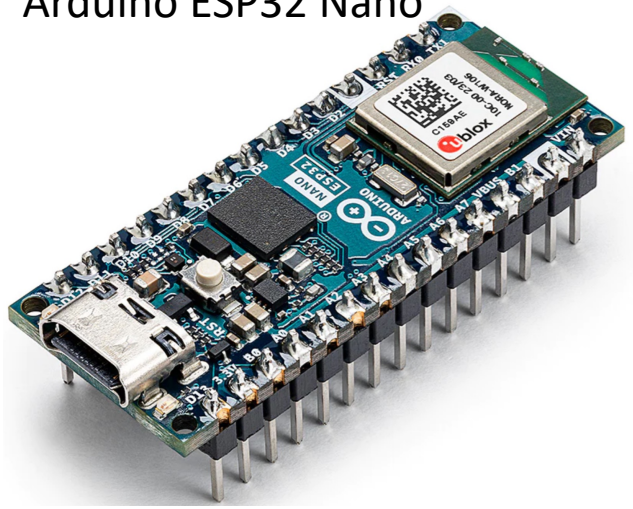
ESP32 Dev 30 pin



Feather S3



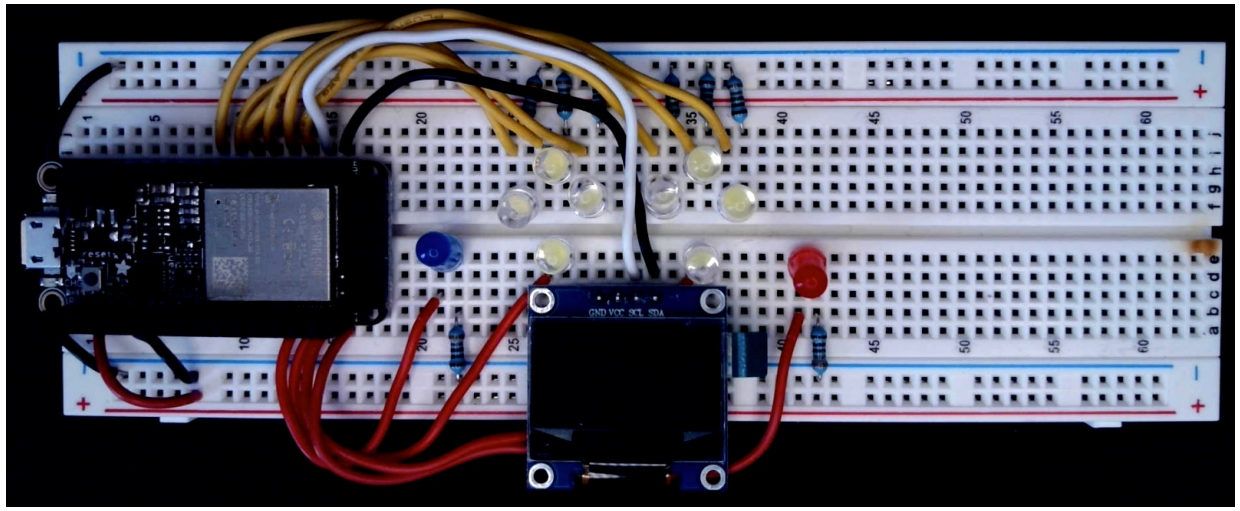

Arduino ESP32 Nano

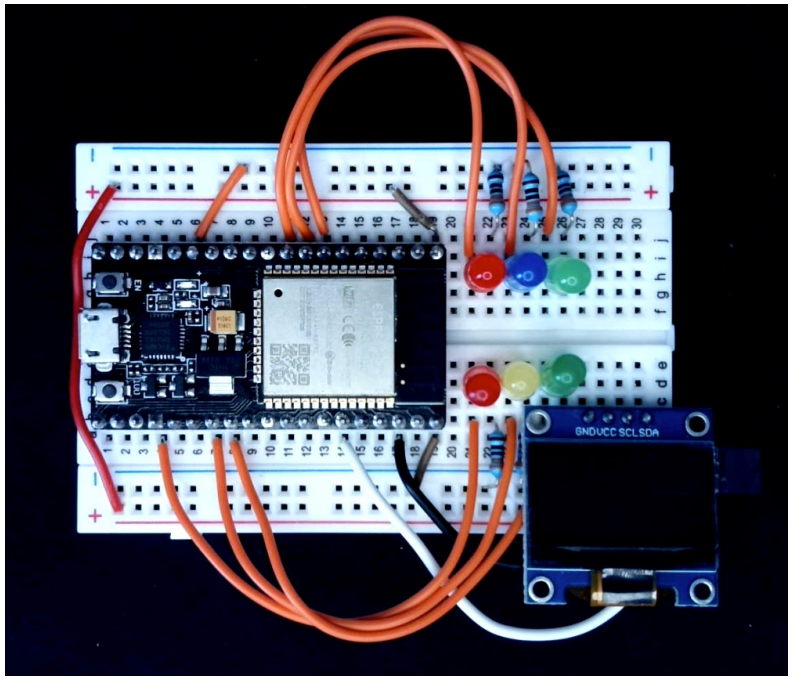


ESP32 Dev 38 pin

EEK: ESP32 Educational Kit

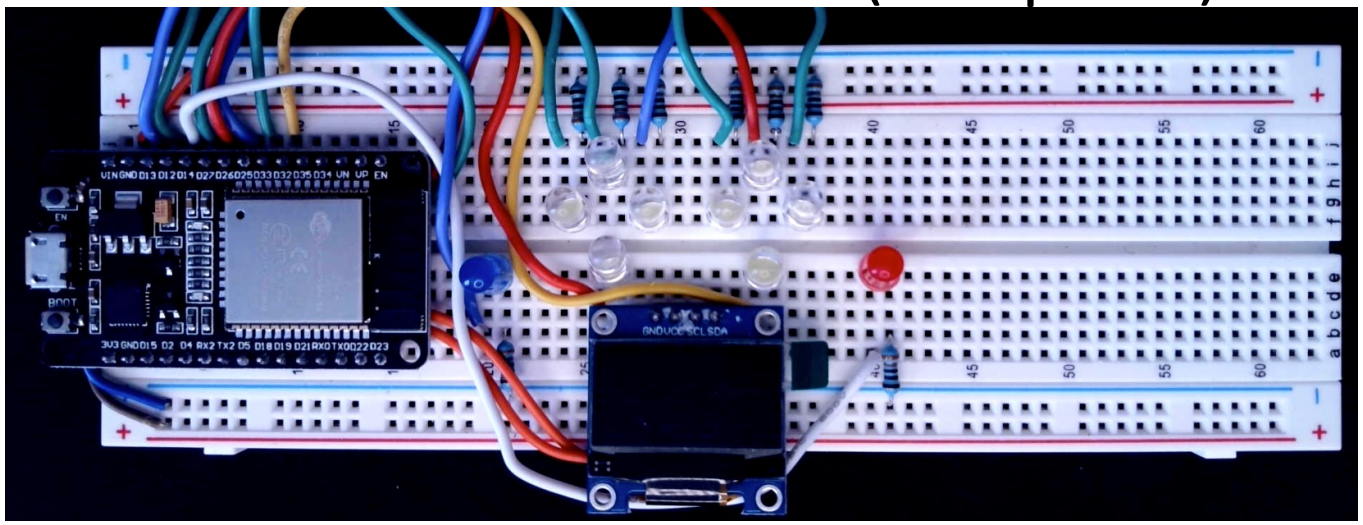
# AdaFruit ESP32 Feather Huzzah

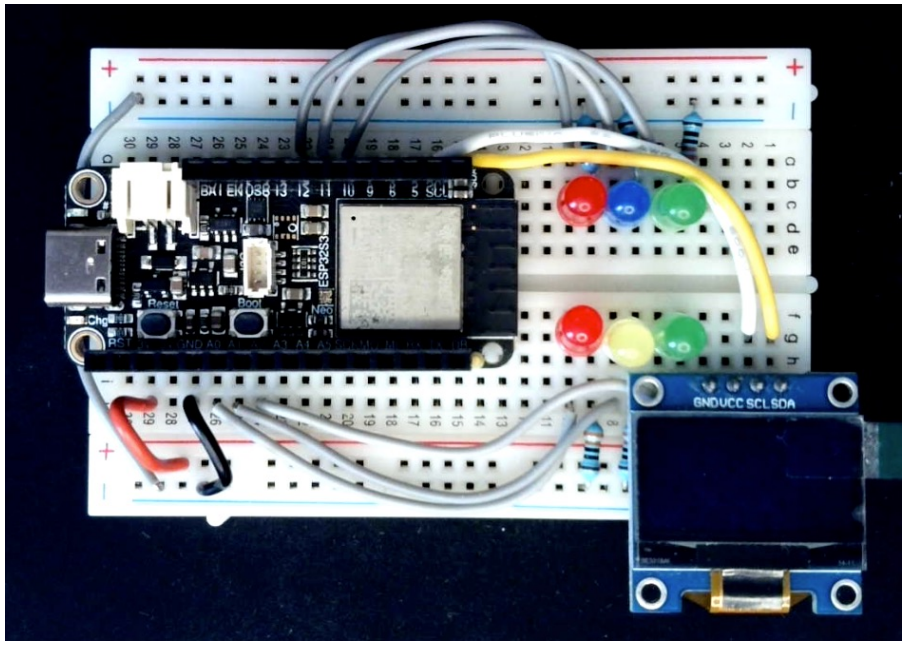





- Huzzah is a fully supported EEK board
- Breadboard shown here on the left is the Huzzah configured to act as a drone simulator of sorts
- We explore this configuration as one of the main sections of Part 2 of the EEK course
- We can use another ESP32 (38 pin ESP32 DevKit shown on the right) connected to a Gamepad and then connect via WiFi to the simulator
  - Or connect to a real Tello and fly it using Gamepad controls: buttons, D-Pad and Joysticks
  - Gamepad connects to the EEK via Bluetooth

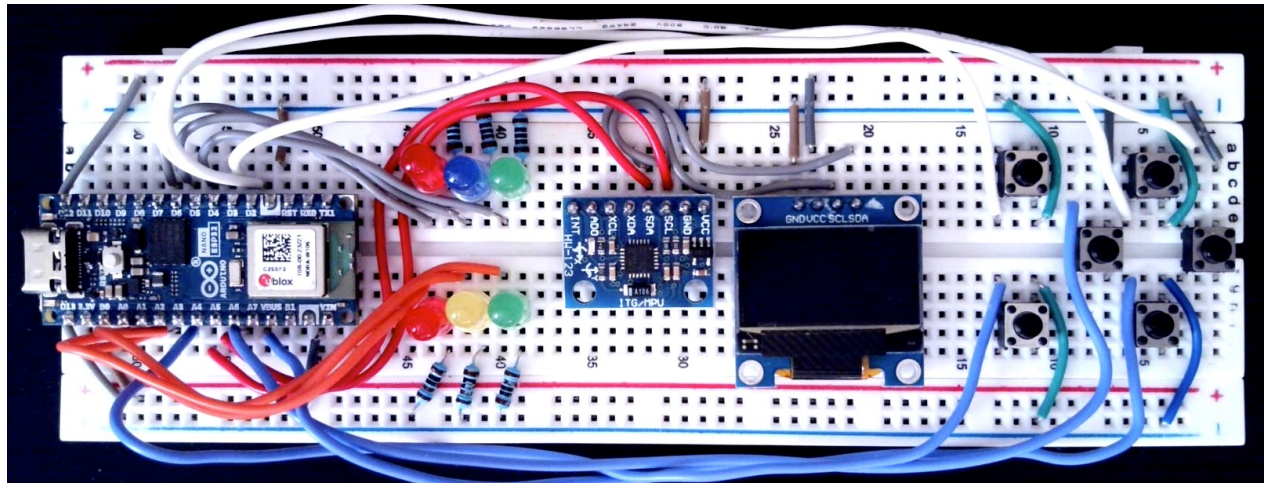
# ESP32 DevKit Clone (30 pins)





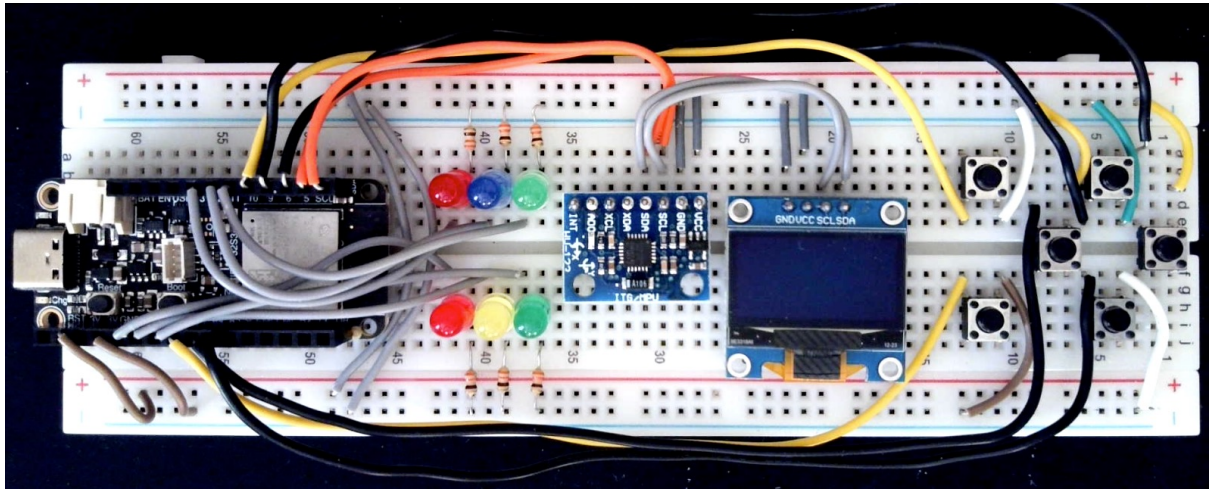
- 30 pin ESP32 Devkit Clone is a fully supported EEK board
- Breadboard shown here on the left is the DevKit clone configured to act as a drone simulator of sorts
- We explore this configuration as one of the main sections of Part 2 of the EEK course
- We can use another ESP32 (AdaFruit Feather S3 shown on the right) connected to a Gamepad and then connect via WiFi to the simulator
  - Or connect to a real Tello and fly it using Gamepad controls: buttons, D-Pad and Joysticks
  - Gamepad connects to the EEK via Bluetooth

# Arduino ESP32 Nano




- As shown in the picture, the EEK breadboard can be configured with the Nano
  - The Nano is smaller than many DevKit boards
- The Bluepad32 library and board extension is not working with the Nano but the EEK as shown can be used to fly the Tello and the buttons and gestures, based on the MPU6050 work
  - Connecting a gamepad is not an option currently for the Nano
- Almost all of the other EEK Course Part 1 sketches will work with customized pin assignments

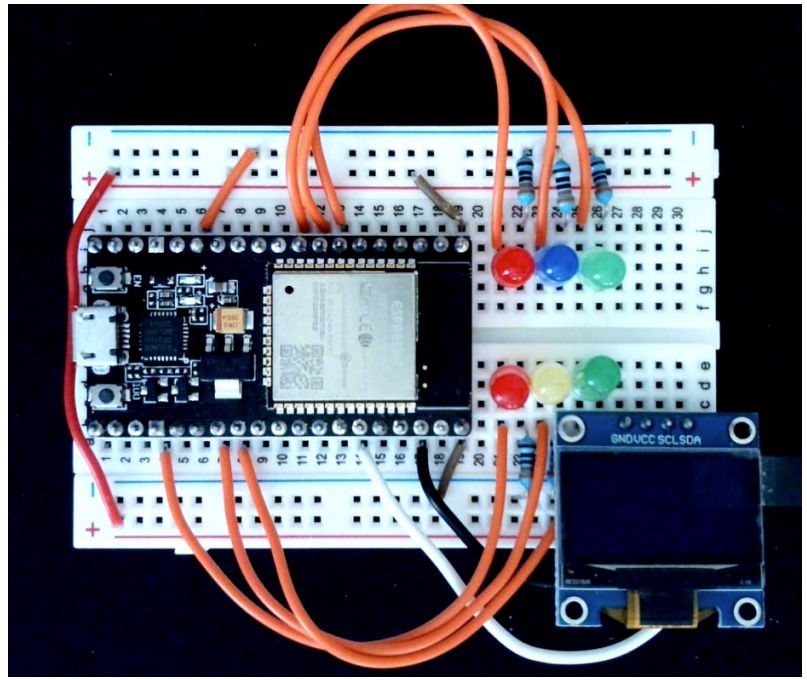
# AdaFruit Feather S3

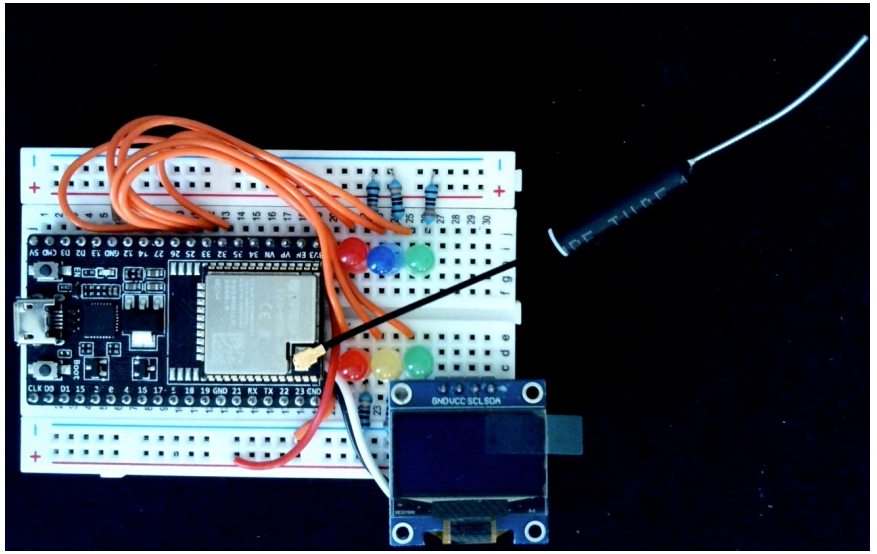



- As shown in the picture, the EEK breadboard can be configured with the AdaFruit Feather S3
  - GPIO pin assignments are different than the Huzzah
- The S3 in the name comes from the use of the ESP32S3 variant
- The Bluepad32 library and board extension does work with the Feather S3
  - The Arduino IDE has some special configuration steps
  - Gamepads should include support for Bluetooth Low Energy (BLE)
  - One of the compatible Arduino game system pads is shown on the right
- All the EEK Course Part 1 sketches should work with customized pin assignments appropriate for the Feather S3

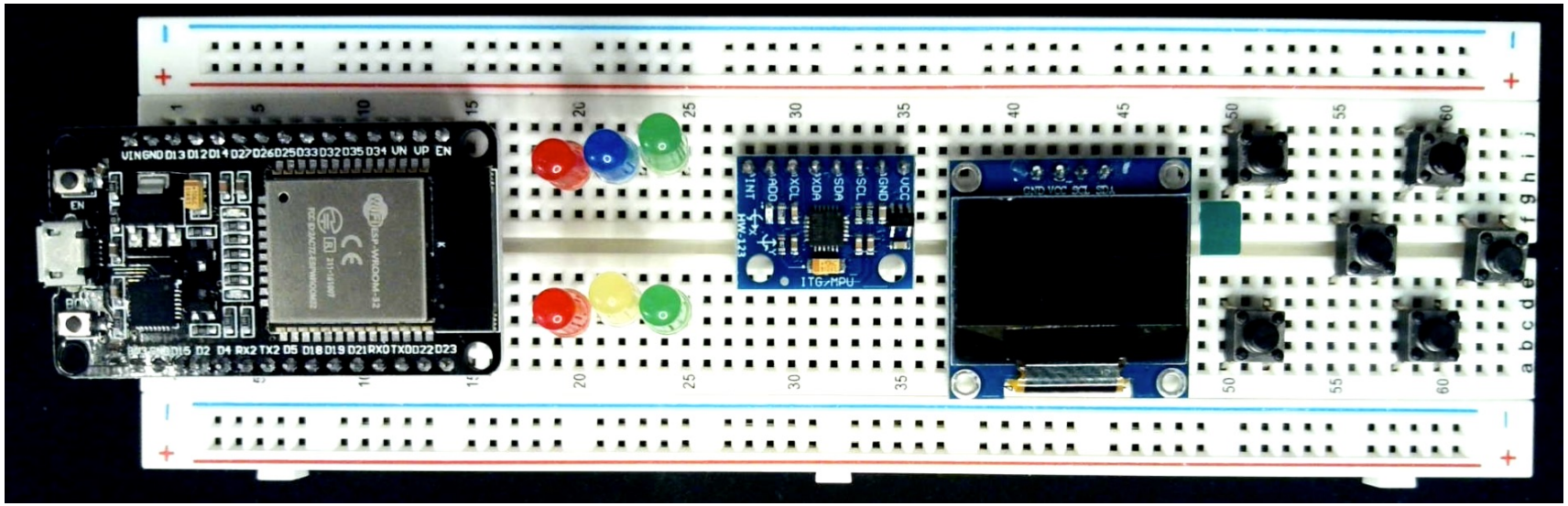
# ESP32 Development Board (38 pins)

- Two 38 pin ESP32 Development boards are shown
  - The bottom one has a connection for an external antenna allowing for better WiFi range but is as wide as the DevKit clone
  - The top one is not as wide as the DevKit clone board so breadboard pin rows on the top and bottom are available for connections
- Gamepad connection is possible with the Bluepad32 library so even without the MPU and switches, Tello flight control is supported
  - Simulator interaction too



# Placing EEK components on breadboard



Breadboard column numbers go from 1 to 63

ESP32 board will be mounted flush to the left USB side LAST!

LEDs are placed starting at column 20: 3 on top and 3 on bottom, occupying 2 columns each, placed in middle row

Anode side of LED should be on the left and Cathode should be on the right; ESP32 pins connect to Anode, resistors to Cathode

MPU has 8 pins (only 4 will be connected) and plugged into columns 28 to 35

Use fourth row of pins on top of breadboard for MPU

OLED display has 4 pins and should be inserted so that GND is in column 41 and top edge aligns with MPU

Left column of switches spans columns 51 to 53

Middle 2 switches are in columns 55 to 57 and 61 to 63 and straddle the depression in the breadboard center

Right column of switches between the middle switches span columns 58 to 60

Use care when inserting the switches so as not to deform the claw pins that should be aligned vertically

Switches should span 4 rows or the center depression on the breadboard

# EEK Assembly Overview

- Take inventory of parts and investigate pin numbering for your board
- For the DevKit Board clone, some wires will need to be placed on the breadboard underneath where the ESP32 board will be attached: 3.3v, SDA, SCL and 4 outbound switch wires
- It is helpful to use the breadboard column numbers 1 through 15 for the clone and 1 through 16 for the AdaFruit Huzzah for orientation during the wiring process outbound from the ESP32
  - See tables on next slide
- Trim LED leads and place the LEDs next to where the ESP32 board will be inserted (start at column 20, anode to left)
- Trim resistor leads and insert to connect cathode pin of LED to nearest GND (-) row of breadboard (3 on top and 3 on bottom)
- Place the MPU 6050 and OLED display next and connect SDA and SCL wires from ESP32 to MPU
  - Wire SDA of MPU to SDA of OLED and SCL of MPU to SCL of OLED
  - Use short wires to connect VCC and GND for both MPU and OLED to the top (-) and (+) rows of the breadboard
- Insert 6 button switches at right side of breadboard
  - Use short/medium wires to connect right side of switch to GND (-) row: top and middle rows of switches to top GND and bottom row of switches to bottom GND; when necessary, bend wires to get around switch body
- After cutting wires to appropriate lengths, finish outbound wire connections from the ESP32 pin positions
  - There will be a 3.3V connection, a GND connection, SDA and SCL connections, 6 connections to button switches and 6 connections to LEDs
  - Suggest doing Rail bridge connections first, then SDA/SCL connections, then switch connections next, and then LED connections; Huzzah and DevKit board clones have different pin connections
  - Connect GND of ESP32 to each GND (-) horizontal row of the breadboard; use separate wire to connect both GND (-) rows for Huzzah
  - Connect 3.3V to top power (+) row of breadboard

# Breadboard Column to Pin Destination mappings

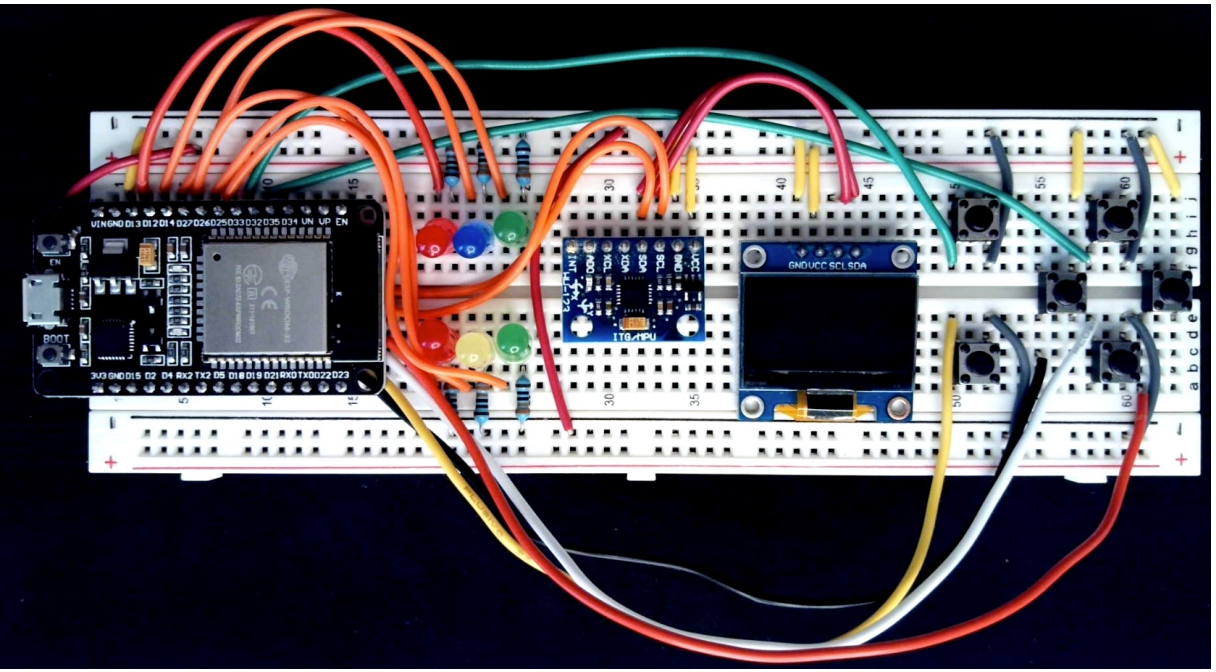
## ESP32 DevKit clone

| Breadboard Column | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Top Pin Destination | | Top GND rail | Top Red LED | Blue LED | Top Green LED | Bottom Red LED | Yellow LED | Bottom Green LED | Top Left Switch | Top Right Switch | | | | | |
| Wire Length | | 0.5 in | 3 inch | 3 inch | 3 inch | 3 inch | 3 inch | 3 inch | 5 inch | 5 inch | | | | | |
| Bottom Pin Destination | Top 3.3V rail | Bottom GND | Middle Left Switch | | Bottom Left Switch | | | Middle Right Switch | Bottom Right Switch | | MPU SDA | | | MPU SCL | |
| Wire Length | 2 inch | | 7 inch | | 7 inch | | | 7 inch | 7 inch | | 3 inch | | | 3 inch | |

## ESP32 AdaFruit Huzzah

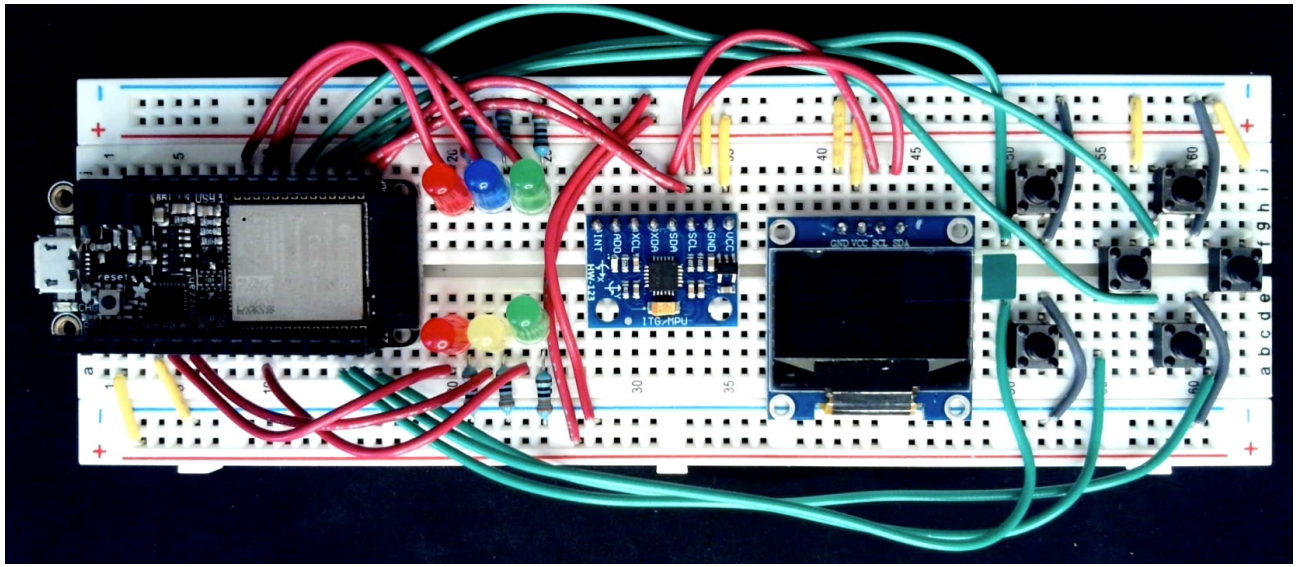
| Breadboard Column | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Top Pin Destination | No Pin | No Pin | No Pin | No Pin | | | | | Top Red LED | Blue LED | Top Green LED | Top Left Switch | Top Right Switch | Bottom Right Switch | MPU SCL | MPU SDA |
| Wire Length | | | | | | | | | 2 inch | 2 inch | 2 inch | 5 inch | 5 inch | 5 inch | 3 inch | 3 inch |
| Bottom Pin Destination | | Bottom 3.3V rail | | Bottom GND rail | Bottom Red LED | Yellow LED | | | | Bottom Green LED | | | | Middle Left Switch | Bottom Left Switch | Middle Right Switch |
| Wire Length | | 0.5 in | | 0.5 in | 2 inch | 2 inch | | | | 2 inch | | | | 5 inch | 5 inch | 5 inch |

# Assembled EEK boards



**ESP32 DevKit Clone**

**ESP32 AdaFruit Feather Huzzah**



# Assembled EEK breadboard Connection Tests

- 4 sketches are provided to check that LEDs, OLED Display, Switches and MPU are all functioning as expected
  - Look for the AssemblyTest folder in the code archive for this course.
- You will not need to understand details about these for now to use them for testing purposes
  - EEK Basics section will explore all the EEK board component groupings in some detail
- If the tests fail, most likely there is a loose connection or a miss-wired one
  - Re-check the steps you performed during assembly
  - Getting the polarity wrong on an LED is common: just extract and flip it around and re-test
  - Switch failure may be due to the wrong numbered pin on the ESP32 being used: re-check the pins used; also, re-seat the switches into the breadboard if they appear to be loose
  - It is unlikely that there is a hardware failure but if a replacement component is available try to replace the one that is not working properly

# Breadboard Column to ESP32 GPIO mappings

## ESP32 DevKit clone

| Breadboard Column | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Top Pin Destination | | Top GND rail | Top Red LED | Blue LED | Top Green LED | Bottom Red LED | Yellow LED | Bottom Green LED | Top Left Switch | Top Right Switch | | | | | |
| GPIO Number | | GND | 13 | 12 | 14 | 27 | 26 | 25 | 33 | 32 | | | | | |
| Bottom Pin Destination | Top 3.3V rail | Bottom GND | Middle Left Switch | | Bottom Left Switch | | | Middle Right Switch | Bottom Right Switch | | MPU SDA | | | MPU SCL | |
| GPIO Number | 3.3V | GND | 15 | | 4 | | | 5 | 18 | | SDA 21 | | | SCL 22 | |

## ESP32 AdaFruit Huzzah

| Breadboard Column | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Top Pin Destination | No Pin | No Pin | No Pin | No Pin | | | | | Top Red LED | Blue LED | Top Green LED | Top Left Switch | Top Right Switch | Bottom Right Switch | MPU SCL | MPU SDA |
| GPIO Number | | | | | | | | | 12 | 27 | 33 | 15 | 32 | 14 | SCL 22 | SDA 23 |
| Bottom Pin Destination | | Top 3.3V rail | | Bottom GND rail | Bottom Red LED | Yellow LED | | | | Bottom Green LED | | | | Middle Left Switch | Bottom Left Switch | Middle Right Switch |
| GPIO Number | | 3.3V | | GND | 26 | 25 | | | | 4 | | | | 16 | 17 | 21 |

# PCB Versions of EEK

Limited numbers of the EEK PCB boards are available

- AdaFruit Feather Huzzah is used

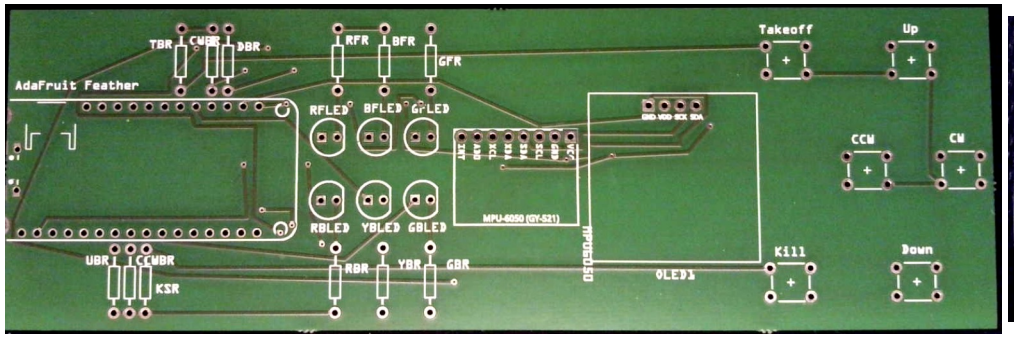
One version uses the same hardware as the Breadboard

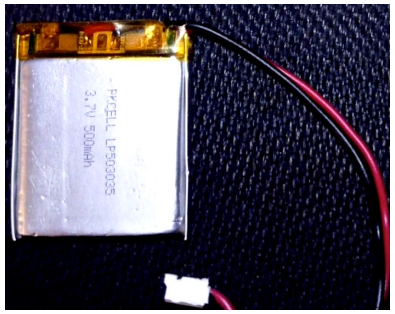
- You can solder the EEK Kit components onto the PCB
- 6 additional 10K Ohm resistors are provided since the switches were wired as INPUTs rather than INPUT_PULLUPs

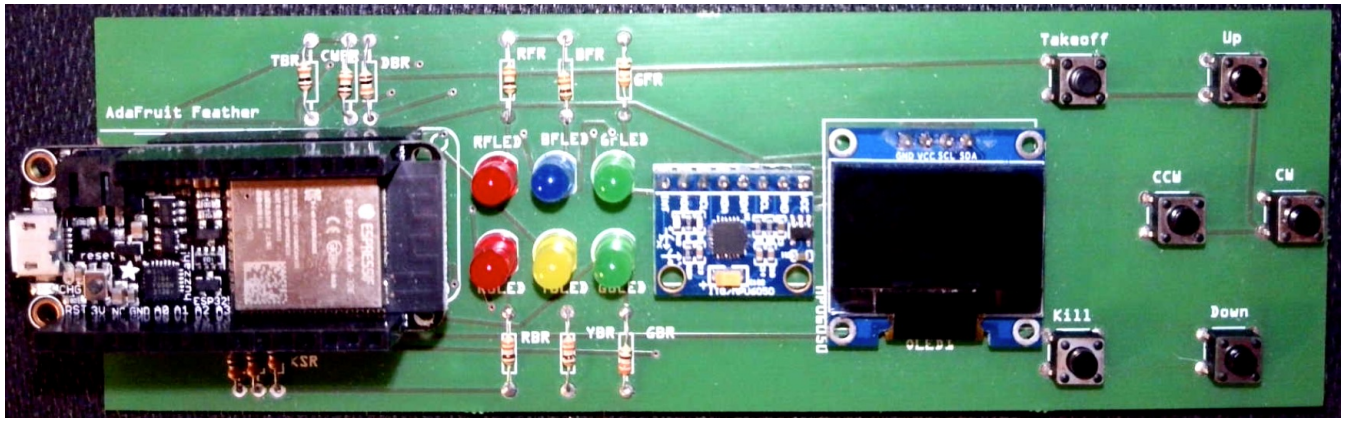
Second version uses surface mount components and comes completely assembled

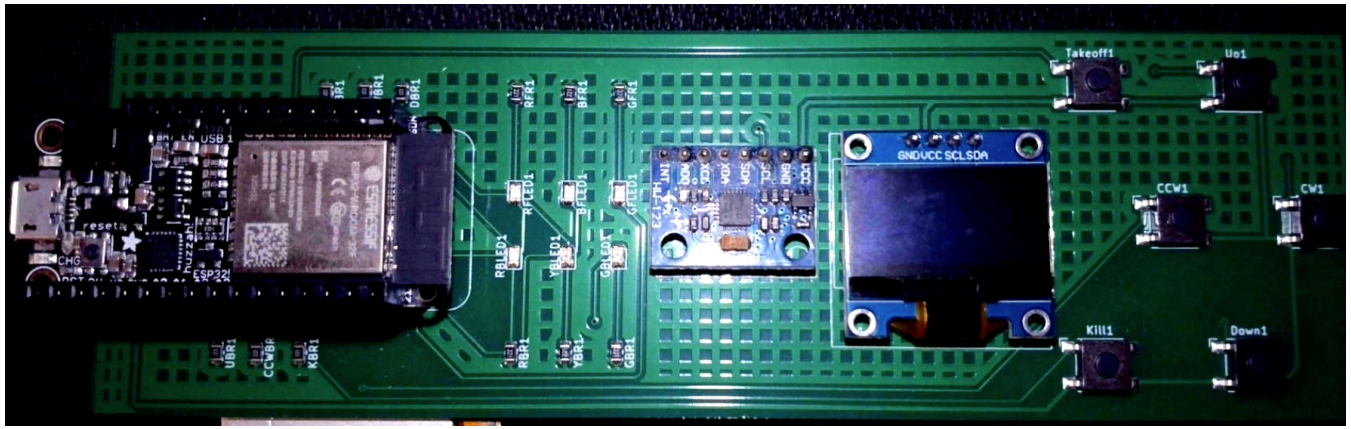
- A redesigned EEK PCB version may be available soon using the Dev Kit version of the ESP32 and a simplified circuit design that does not include a battery
- Redesigned version may be cheaper

Each Huzzah PCB will include LiPo battery








https://shop.iotinurhand.store/

# EEK Basics Introduction

- What is this section about
  - After assembly you verified that the components and their connections were operational by running some pre-made Arduino code from the Arduino IDE
  - Two different ESP32 boards are available as EEK kits or you may have provided your own board
  - Here we will break down the assembled EEK into its main constituent parts and learn some of the logical underpinnings of each component group and how to program them
- What you will learn by completing this section
  - You will understand how the ESP32 is able to interact with all the main EEK logical and hardware components:
    - LEDs, OLED Display, Push-Button switches, and MPU Sensor board
  - You will be able to modify portions of the code that was originally used to verify correct assembly of the EEK

# LEDs using GPIO Pin assignments

6 LEDs provide status information about the control functions implemented on the EEK (used when flying a Tello drone or interacting with a Simulator)

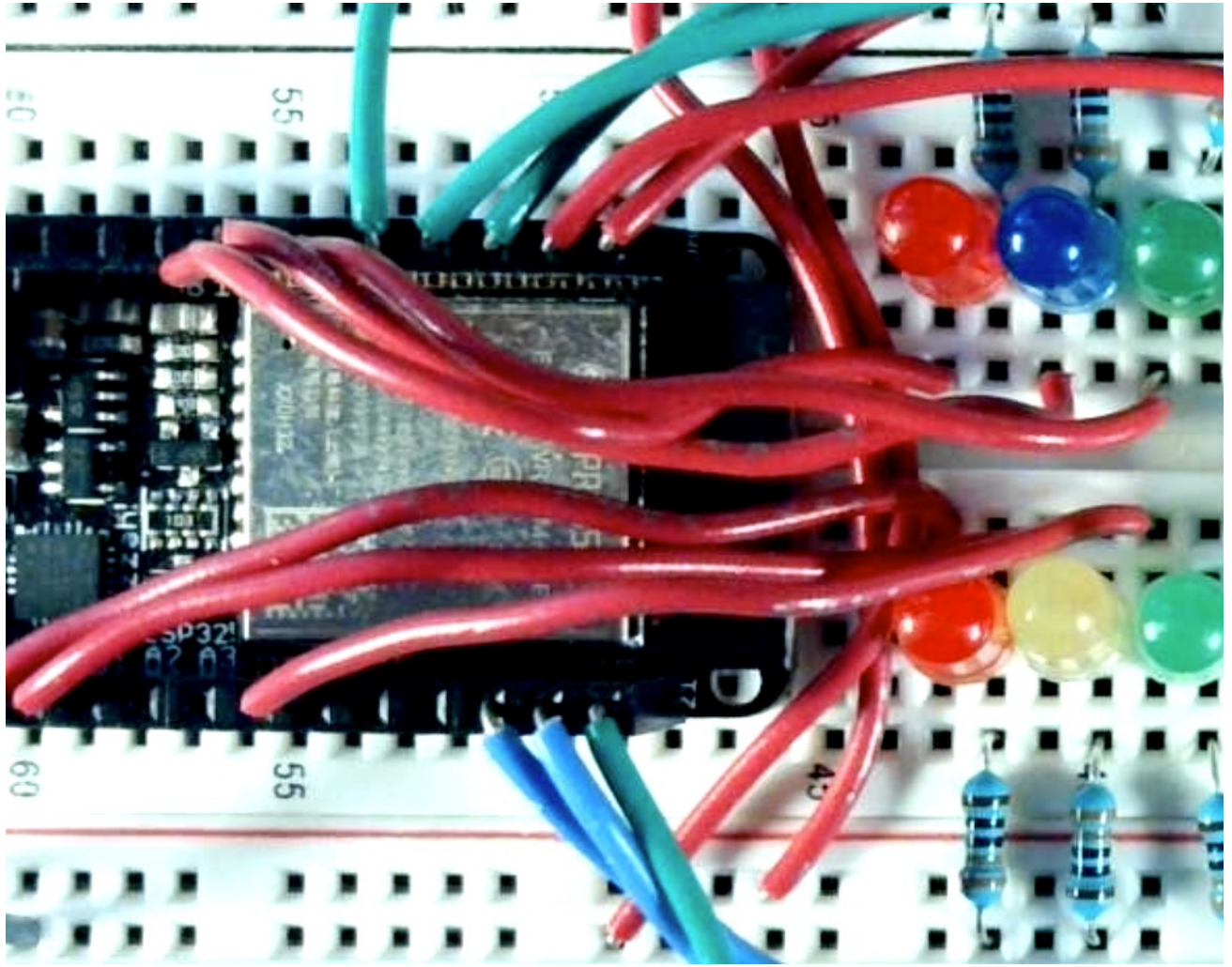
Red Wires connect to the positive side of each LED (the Anode); resistors connect to the negative side of the LED (cathode), which are then connected to GND

The positive wires are connected to GPIO pins on the ESP32

Depending on which ESP32 board you use, the Pin numbers will be different

Arduino source code can be included to handle pin number variability for different ESP32 boards

This section ends with mini-project where each LED blinks in sequence



# GPIO Pins used for EEK board LEDs

**AdaFruit Huzzah**

- TOP_RED: 12
- BLUE: 27
- TOP_GREEN: 33
- BOTTOM_RED: 26
- YELLOW: 25
- BOTTOM_GREEN 4

**Dev Board Clone**

- TOP_RED: 13
- BLUE: 12
- TOP_GREEN: 14
- BOTTOM_RED: 27
- YELLOW: 26
- BOTTOM_GREEN 25

# EEK LEDs Mini-Project

- For an assembled EEK board, create a sketch that blinks each of the 6 LEDs individually for 2 seconds each
- Also write a message through the Serial Monitor whenever the colored LED turns ON and OFF
- If you want to support multiple ESP32 board types, use the #define and #ifdef/#endif macro calls to do so
  - If you are sticking with a single ESP32 board type, use #define matching LED names to GPIO numbers for your chosen board
- After completing this project, you will have duplicated one of the smoke test sketches you ran after assembling the EEK board

# OLED Display based on SSD1306 Library

The OLED display (in frame) is capable of a 128x32 character display along with simple graphics

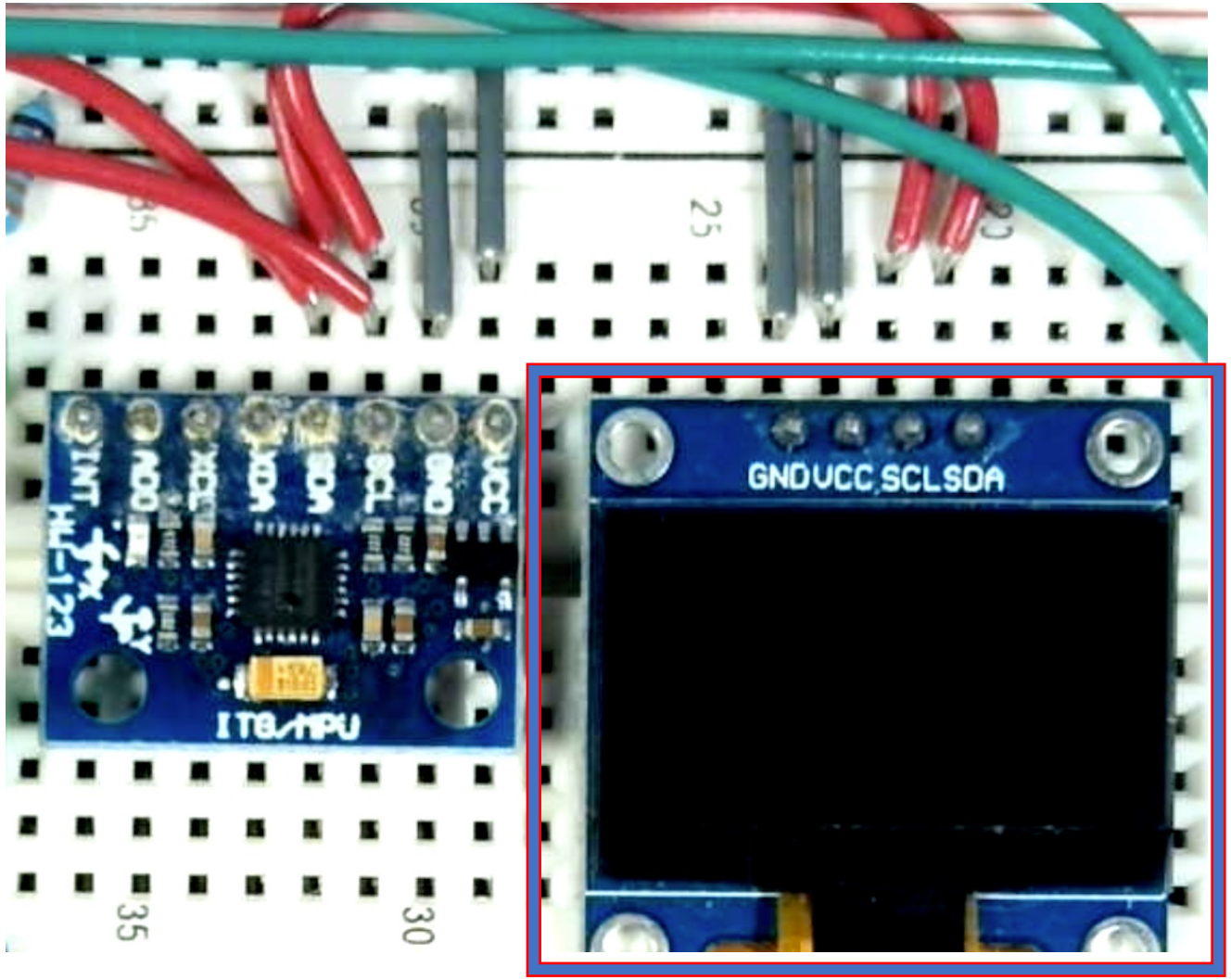
It uses the $I^2C$ communication protocol supported by the ESP32 and the SDA and SCL pins available on the ESP32

Connections to +3.3V (VCC) and GND must also be made

The actual SDA and SCL pin numbers will vary by the type of ESP32 board used

A sample sketch provided by AdaFruit is used to make sure the right pins are connected

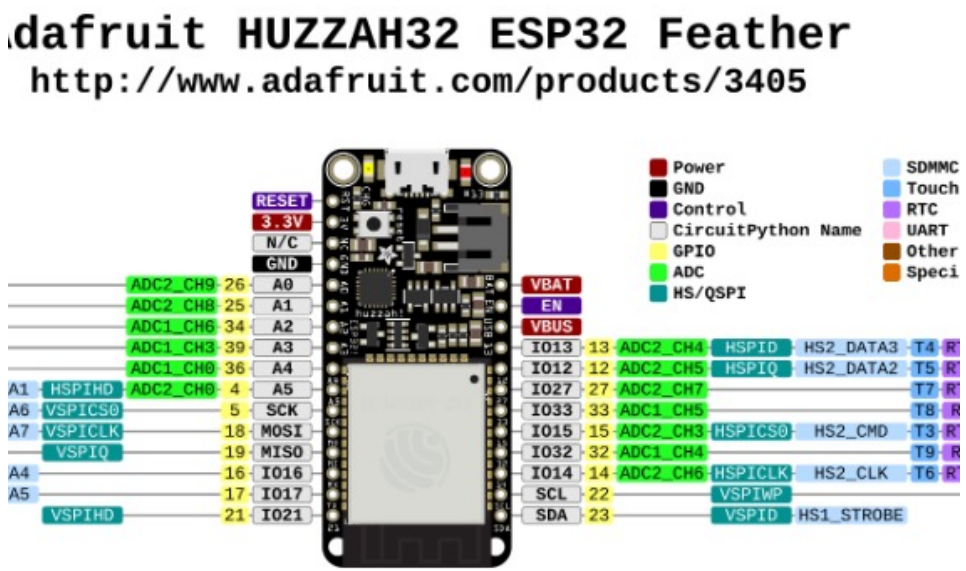
A mini-project for the EEK board is presented that echoes strings typed into the Arduino Serial Monitor to the OLED display



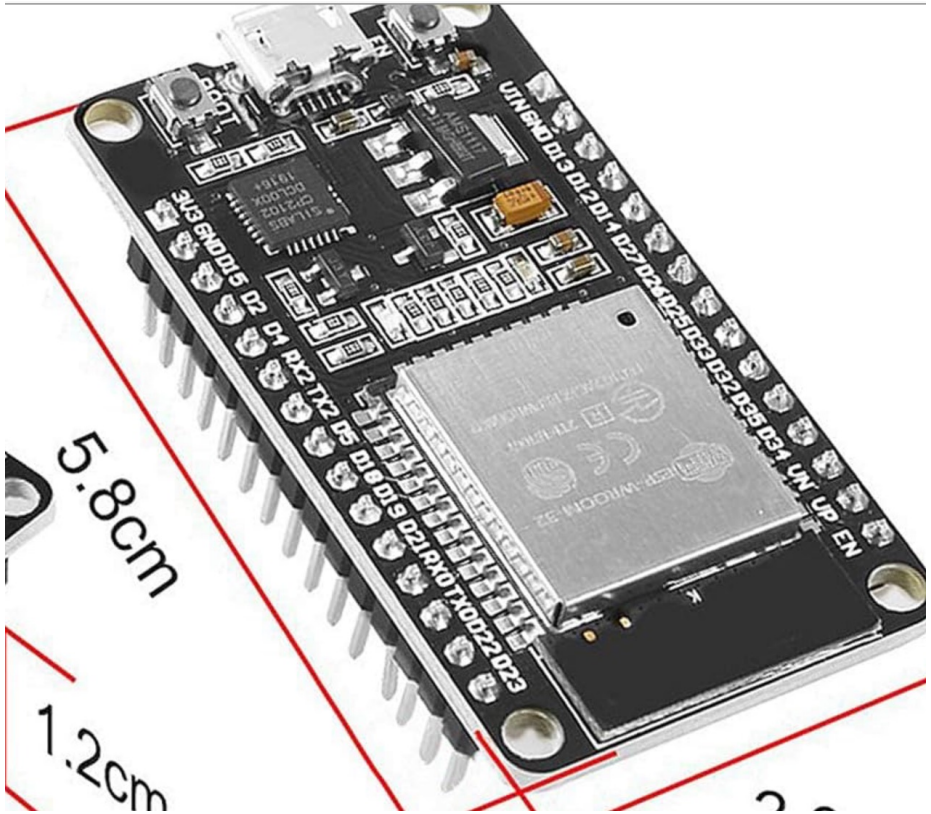
# ESP32 Pins

SCL is GPIO 22 for both boards
SDA is GPIO 23 for the Huzzah Board
SDA is GPIO 21 for the Dev Board clone

## AdaFruit Feather (GPIO# in Yellow)




## ESP32 Dev Board Clone (DGPIO#)



# Switches and Easy Button Library

Just like LEDs can indicate status conditions, simple switches can act as actuators to cause certain actions to happen on the EEK

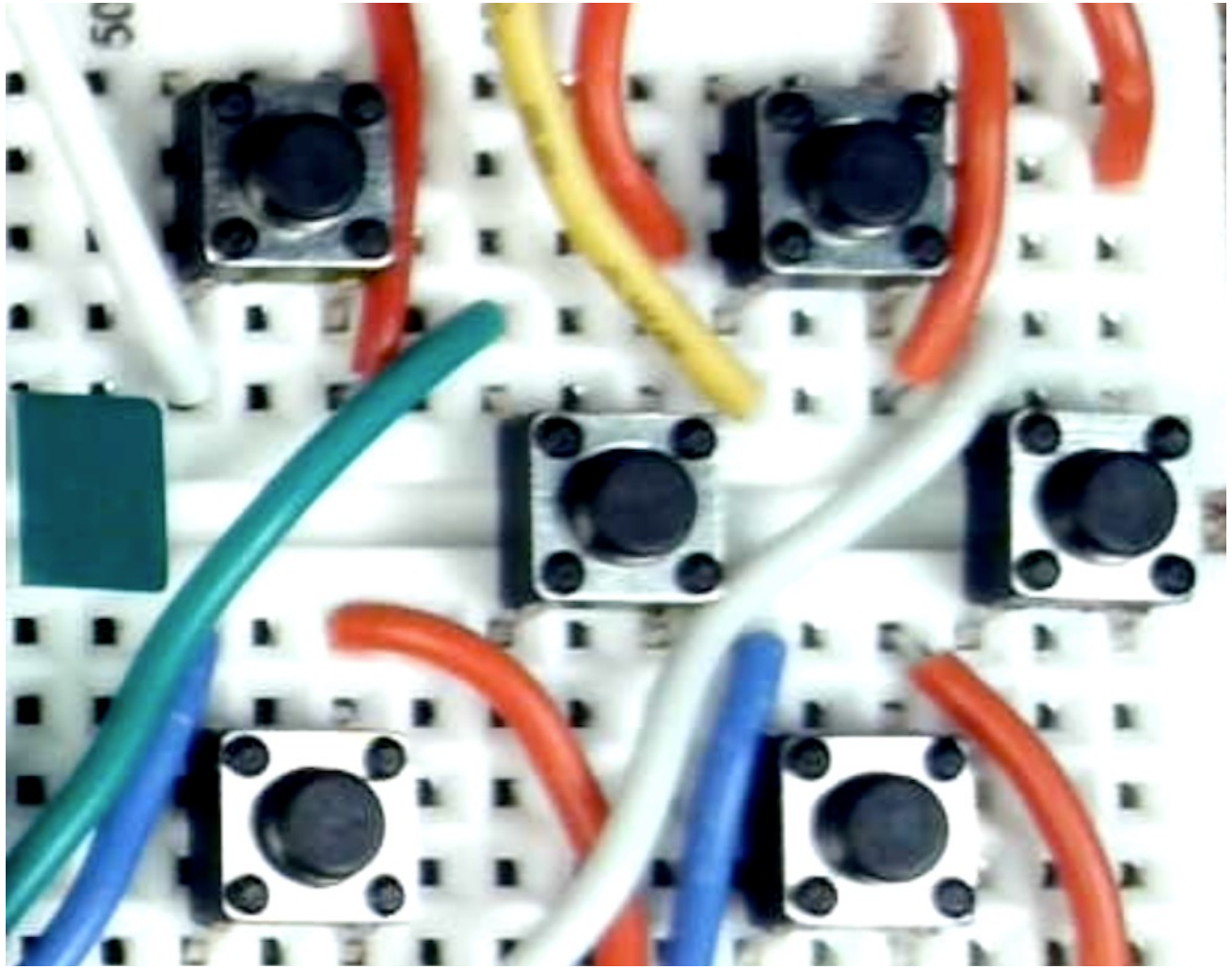
Some simple button examples will be shown

It turns out that getting reliably sensed switch pushes and releases can be problematic (due to so-called de-bouncing effects)

The EasyButton Arduino library helps with this sensing and provides an easy-to-use event programming model to take actions based on switch button pushes

A mini-project is defined where the EasyButton library is used to implement EEK LED display behaviors

Another mini-project will show how to take advantage of PWM (Pulse Width Modulation) to have additional control over LEDs beyond simply being ON or OFF. EasyButton events are used to vary the brightness of LEDs that are connected



# GPIO Pins used for EEK Switch Connections

**AdaFruit Huzzah**

- TOP_RIGHT 14
- TOP_LEFT 15
- MIDDLE_RIGHT 32
- MIDDLE_LEFT 17
- BOTTOM_RIGHT 16
- BOTTOM_LEFT 21

**Dev Board Clone**

- TOP_RIGHT 32
- TOP_LEFT 33
- MIDDLE_RIGHT 5
- MIDDLE_LEFT 18
- BOTTOM_RIGHT 4
- BOTTOM_LEFT 15

# ESP32 PWM

https://techexplorations.com/guides/esp32/begin/pwm/

- PWM stands for Pulse Width Modulation
- Using PWM we can affect outputs whose values are not fixed, like the brightness of an LED
- In the EEK LED section we were concerned about setting them ON or OFF
- ESP32 has a different mechanism than regular Arduino MCUs to vary brightness
  - LED pins are bound to a channel that has been initialized with configuration properties to control the frequency and magnitude of the variability of the pulse
  - Methods `ledcSetup()` and `ledcAttachPin()` are used to configure the channel and pin
- A simple example, from the URL above, will show how to proceed
- Buttons can be used to achieve different affects on different LEDs

# MPU6050 Sensor platform and Libraries

The MPU6050 (in frame) is a well-known auxiliary board including accelerometer and gyroscope processing elements

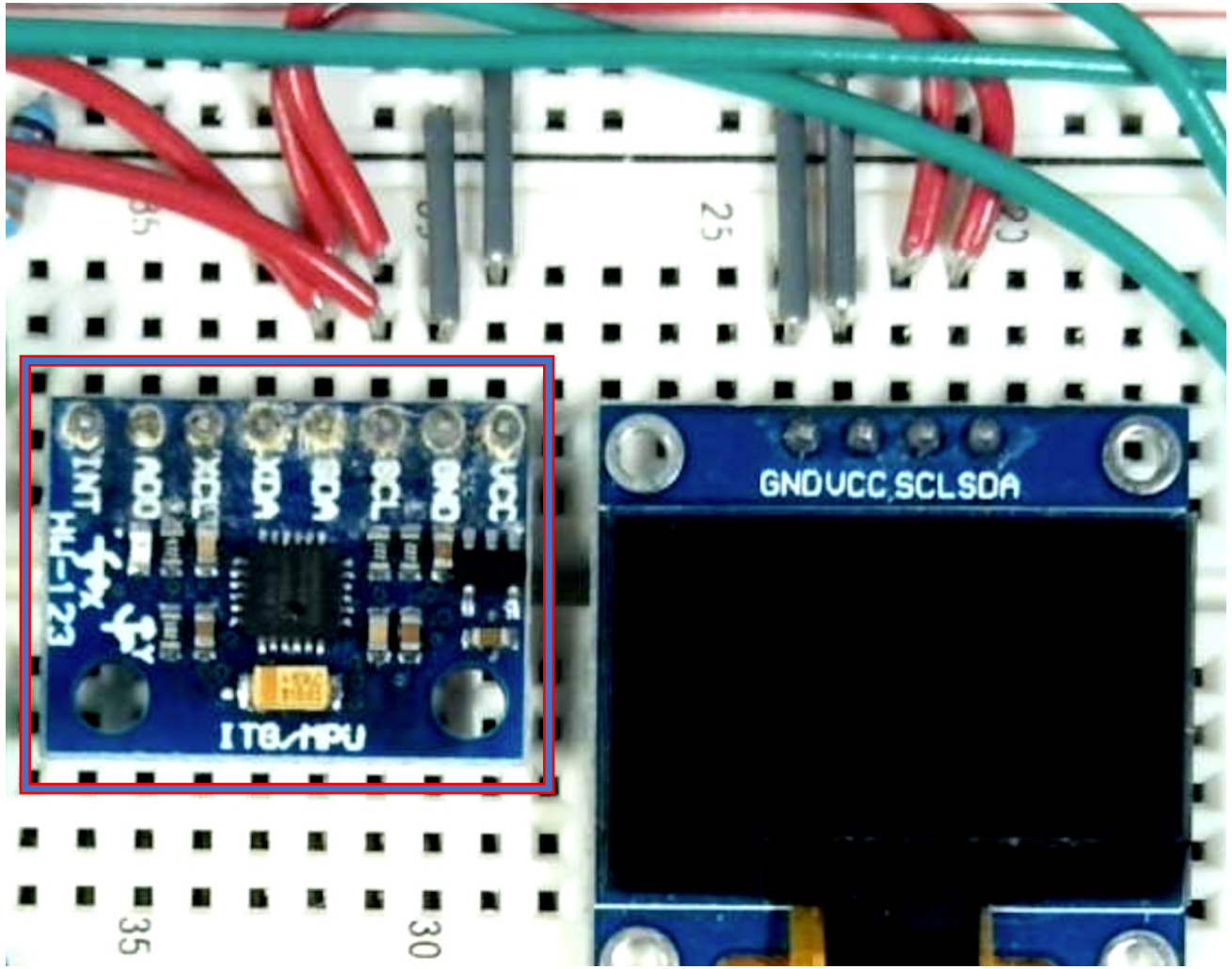
Several Arduino libraries are available to make sense of, and interpret, the raw data produced by the MPU6050

This board is used to sense gestures when the EEK is held in your hand and can be used for gesture-based flying of a Tello

It also uses $I^2C$ and requires connections to SDA, SCL, GND and VCC (+3.3v) like the OLED

Some simple examples will be used to see the raw data available from the board

A mini-project will be provided to set LEDs ON and OFF based on board orientation (gestures) and some baseline data will be shown on the OLED display



# EEK Capstone Mini-Project

- Combination of LED Blink, OLED display, EasyButton switch control, PWM and MPU6050 behaviors
- Builds on logic found in the previous mini-projects
- Partial implementation of Capstone sketch provided
  - Added challenge: ignore the partial implementation and just use previous mini-project code patterns and start with an empty sketch
- Changes needed to partial implementation include:
  1. Fade bottom green LED and toggle bottom RED LED (reverse fade and toggle button behaviors) for lower RIGHT and LEFT button presses
  2. Change OLED displayed messages according to the button logic changes
  3. Change to PWM based effects for the Roll LEDs based on MPU angle gestures: Top RED and Top Green on the EEK
  4. Revert to the ON and OFF LED effects for the Pitch LEDs: Blue and Yellow based on MPU angle gestures
- Test your modified version after each step is completed.

# GPIO Pins used for EEK board LEDs

**AdaFruit Huzzah**

- TOP_RED: 12
- BLUE: 27
- TOP_GREEN: 33
- BOTTOM_RED: 26
- YELLOW: 25
- BOTTOM_GREEN 4

**Dev Board Clone**

- TOP_RED: 13
- BLUE: 12
- TOP_GREEN: 14
- BOTTOM_RED: 27
- YELLOW: 26
- BOTTOM_GREEN 25

# GPIO Pins used for EEK Switch Connections

**AdaFruit Huzzah**

- TOP_RIGHT 14
- TOP_LEFT 15
- MIDDLE_RIGHT 32
- MIDDLE_LEFT 17
- BOTTOM_RIGHT 16
- BOTTOM_LEFT 21

**Dev Board Clone**

- TOP_RIGHT 32
- TOP_LEFT 33
- MIDDLE_RIGHT 5
- MIDDLE_LEFT 18
- BOTTOM_RIGHT 4
- BOTTOM_LEFT 15

# EEK WiFi

https://dronebotworkshop.com/wifimanager/

https://packetsender.com/download

- There are several WiFi modes available including Station and Soft Access Point
- EEK Assembly section used the ESP32 WiFi scanner example to verify that WiFi is functional; we will revisit along with a simple NTP example
- Simple web server demo gets IP address from your local WiFi connection
  - Local network credentials must be embedded in the sketch source code in this case
- We want to achieve some EEK effects depending on the received http message; we will use a web browser to send the message
- WiFi-Manager library can be used to set up a captive portal app to enter credentials for the local network (portal usually at address 192 . 168 . 4 . 1)
  - Some previous examples will be modified to use WiFi-Manager
  - Tello SSID can be connected to the EDC flight controller app with WiFi-Manager support
- Simple UDP server example is similar to web server, and we want an EEK effect based on udp packet received; we will use PacketSender to send udp messages
  - Tello Simulator in EEK part 2 will be an extension of this server; Tello drone is a UDP server at its core

# EEK and Bluetooth

https://gitlab.com/ricardoquesada/bluepad32

Cheap EEK Compatible Gamepad

- Video game controllers (gamepads) work with EEK and Bluetooth
  - Drone flight control with an EEK and gamepad combo will be covered in Part 2 of the EEK course
- You may already have a gamepad
- The bluepad32 board extension and library are used and an example sketch is provided with bluepad32
  - Standard Arduino ESP32 Bluetooth functions and library are not used currently
- The photos show some of the supported gamepads: Xbox and Playstation 4
  - Upper left photo shows a gamepad that may be available for under $10 from sites like AliExpress; Amazon US link above
- Various sketch examples are available
- Mini-projects using EEK OLED, LEDs and PWM actuation are presented



# Pairing Gamepads with EEK

- Android Clone
  - Press and hold X button and then press and hold Home
- PS4
  - Press Share first and then home: PS4 button
- Xbox
  - Press X button to turn on and then hold connect button (small button next to USB connector)
- 8BitDo
  - Use Keyboard mode (R button on upper right shoulder + Start) to turn on
  - Then press and hold Select for 3 seconds to get in pairing mode)
- Sometimes a restart of the ESP32, after pairing, is needed for the pairing to be effective (e.g. Xbox and 8bitdo)
- After pairing, using the Home key alone will usually work to re-establish connection
- Pressing and holding the Home key for several seconds will often turn off the gamepad



# Expanded View of Dev Board Pinout

Previous Dev Board photo shows a mixture of GPIO pin numbers and logical pin designations

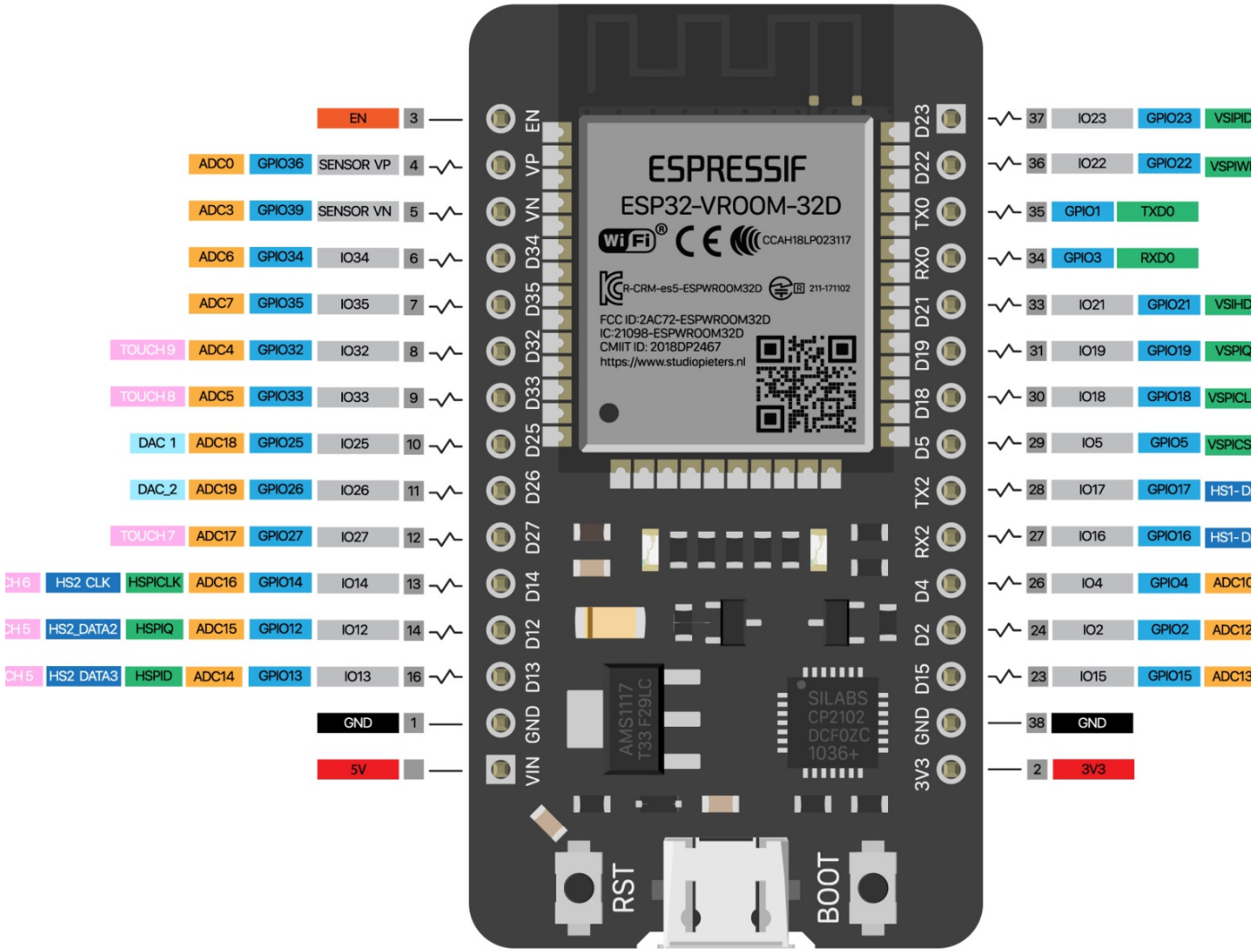
Like: RX0, TX0, RX2 and TX2

These pins also can be identified by their GPIO numbers

Like: GPI01 and GPI03

Some of these pins are used to connect to EEK switches and the connecting wires are attached to the breadboard under the ESP32 board

Pin number choices are documented in the assembly lectures and supporting documentation




https://www.studiopieters.nl/esp32-pinout/