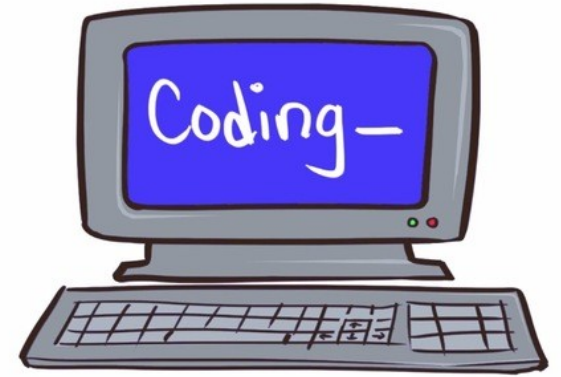


# Class 37

## Talking to your robot



- **WiFi** – hey the ESP32 ALWAYS wants wifi running!
- **Blynk** – sophisticated widgets on your smartphone to control your ESP32

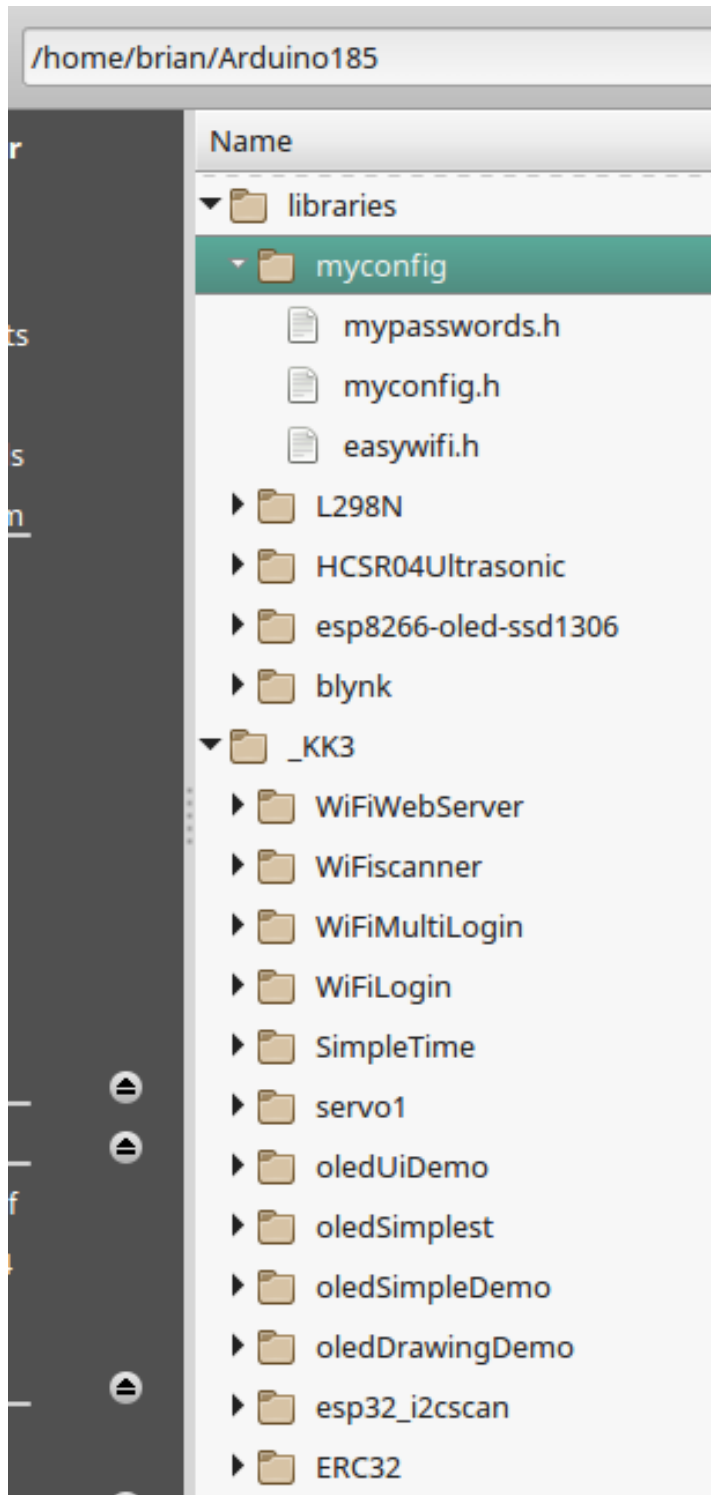
# Sketches this week

- WiFiLogin
- WiFiScanner
- **WiFiMultiLogin**
- WiFiWebServer
  
- **BlynkGpioSimple**
- BlynkVpinIn
- BlynkVpinOut

## Libraries this week:

- (easywifi)
- Blynk

# Reorganised “myconfig”



# Wifi - barebones

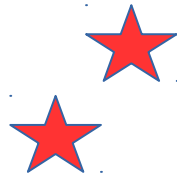
```
#include <WiFi.h>
#include "myconfig.h"

void setup()
{
  WiFi.begin(WIFI_SSID, WIFI_PW);
  .....
```

# easywifi – our preferred

- 1) Supports up to 4 SSID credentials
- 2) Chooses strongest signal
- 3) Recovers smoothly after a wifi fail

```
1  #include "myconfig.h"
2  #include "easywifi.h"
3
4  void setup()
5  {
6
7      wifiInit();
8  }
9
10 void loop()
11 {
12     wifiWatch();
13 }
14
15
```



## Multi Login using “easywifi” libr

passwords.h

```
// primary entry:
#define WIFI_SSID "theBeach"
#define WIFI_PW "theBestSpot"

// With "easywifi", supports up to 3 more SSID
#define WIFI_SSID2 "blackrat"
#define WIFI_PW2 "brian123"

#define WIFI_SSID3 "KinkyKomputing"
#define WIFI_PW3 "RaspberryPi"

// & WIFI_SSID4
```

# Primitive webserver

WiFiWebServer.ino

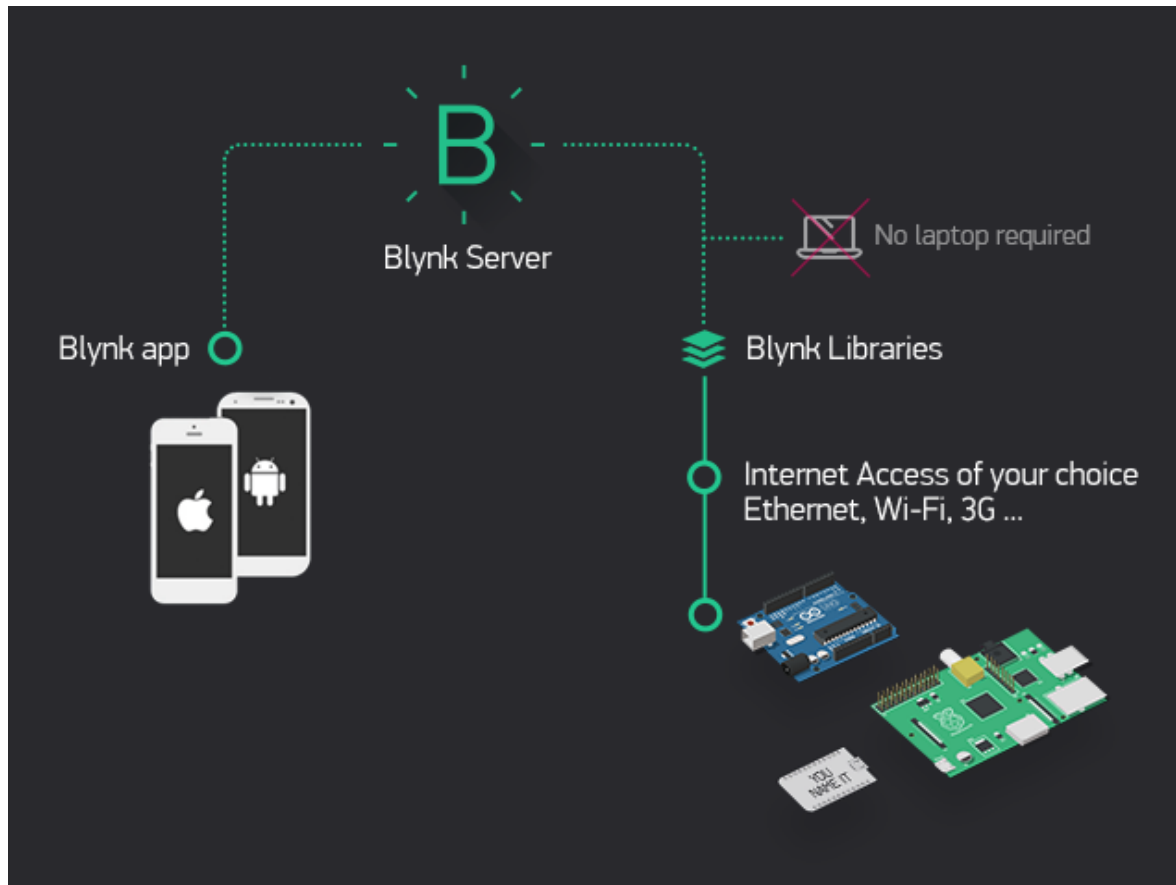
Use ESP's IP number and address your browser there.

Mouse click to turn LED on/off.





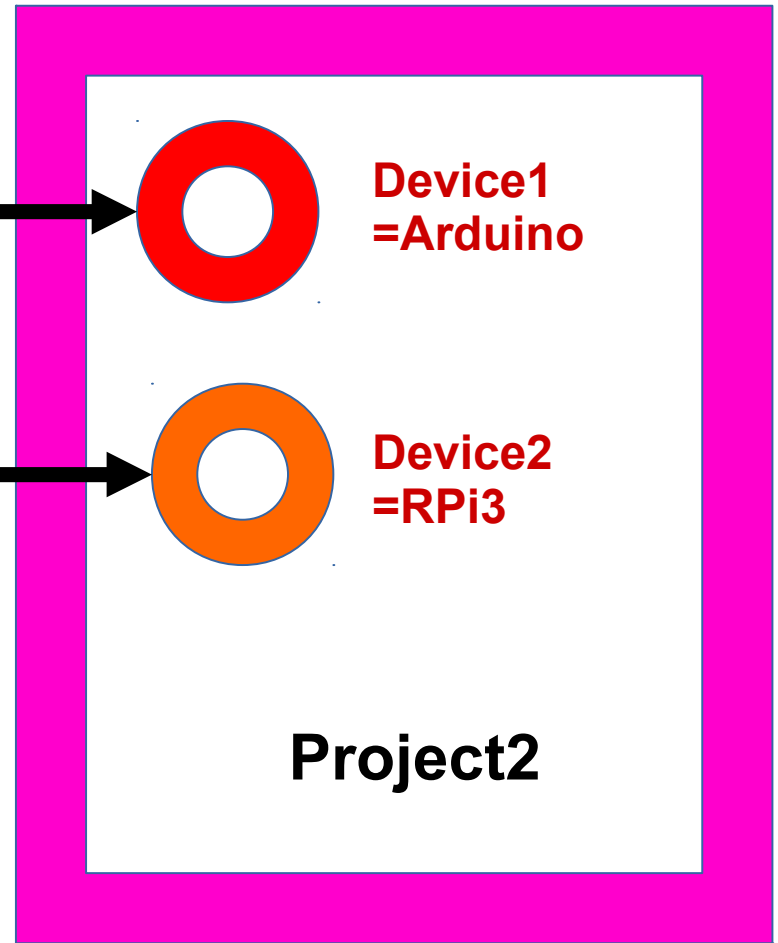
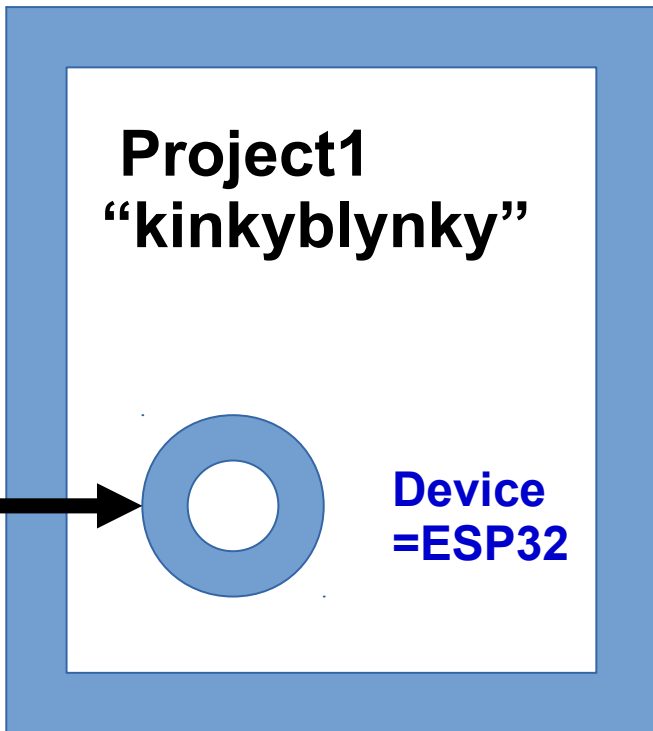
# Blynk





Your account,

one or more projects



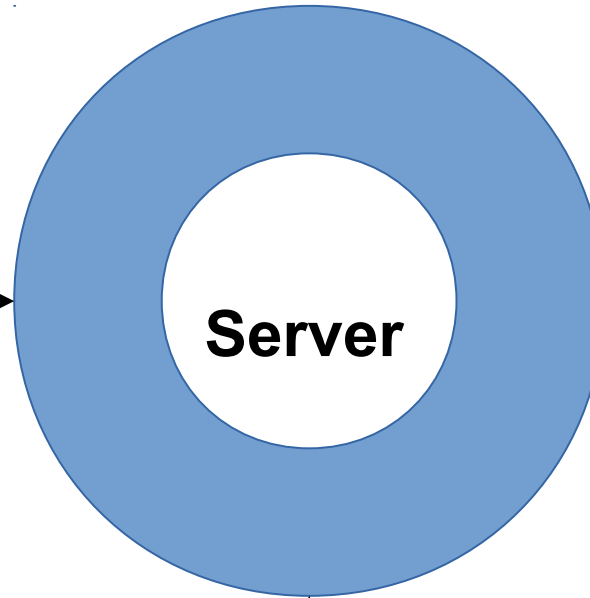
# Projects & Devices

AUTH TOKENS belong to each device

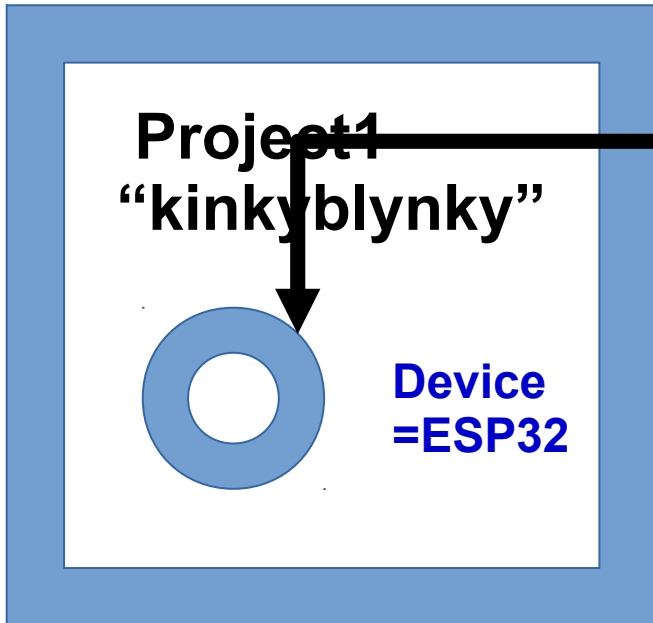


*Plain TCP  
or SSL*

Actually,  
everything  
connects via  
Server

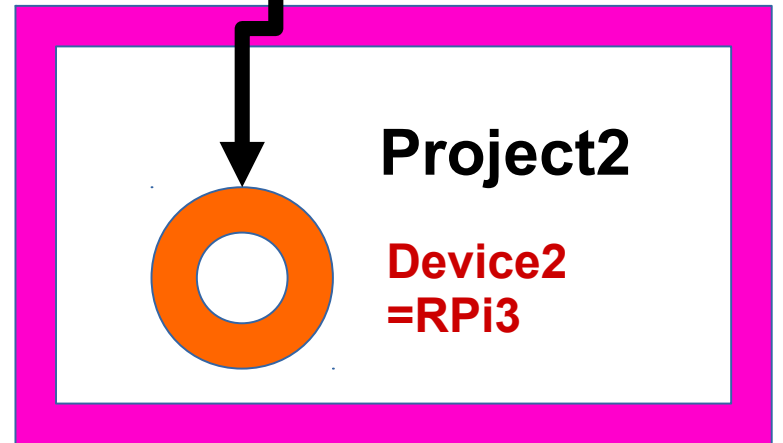


**Server**



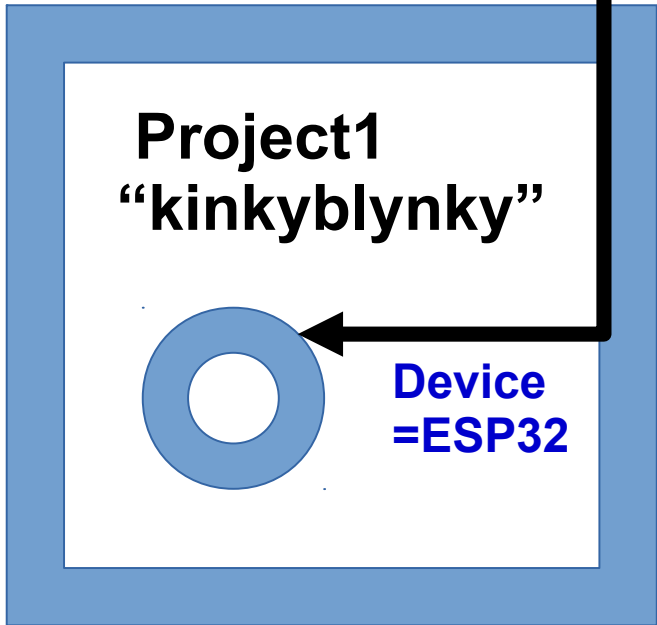
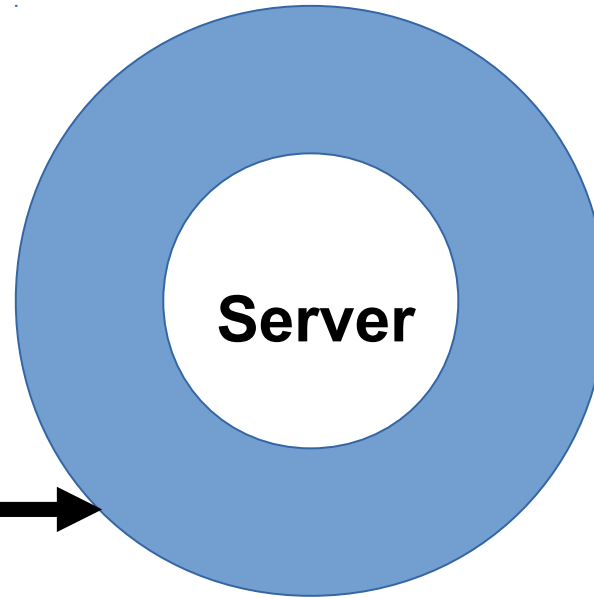
**Project1**  
"kinkyblynky"

**Device  
=ESP32**

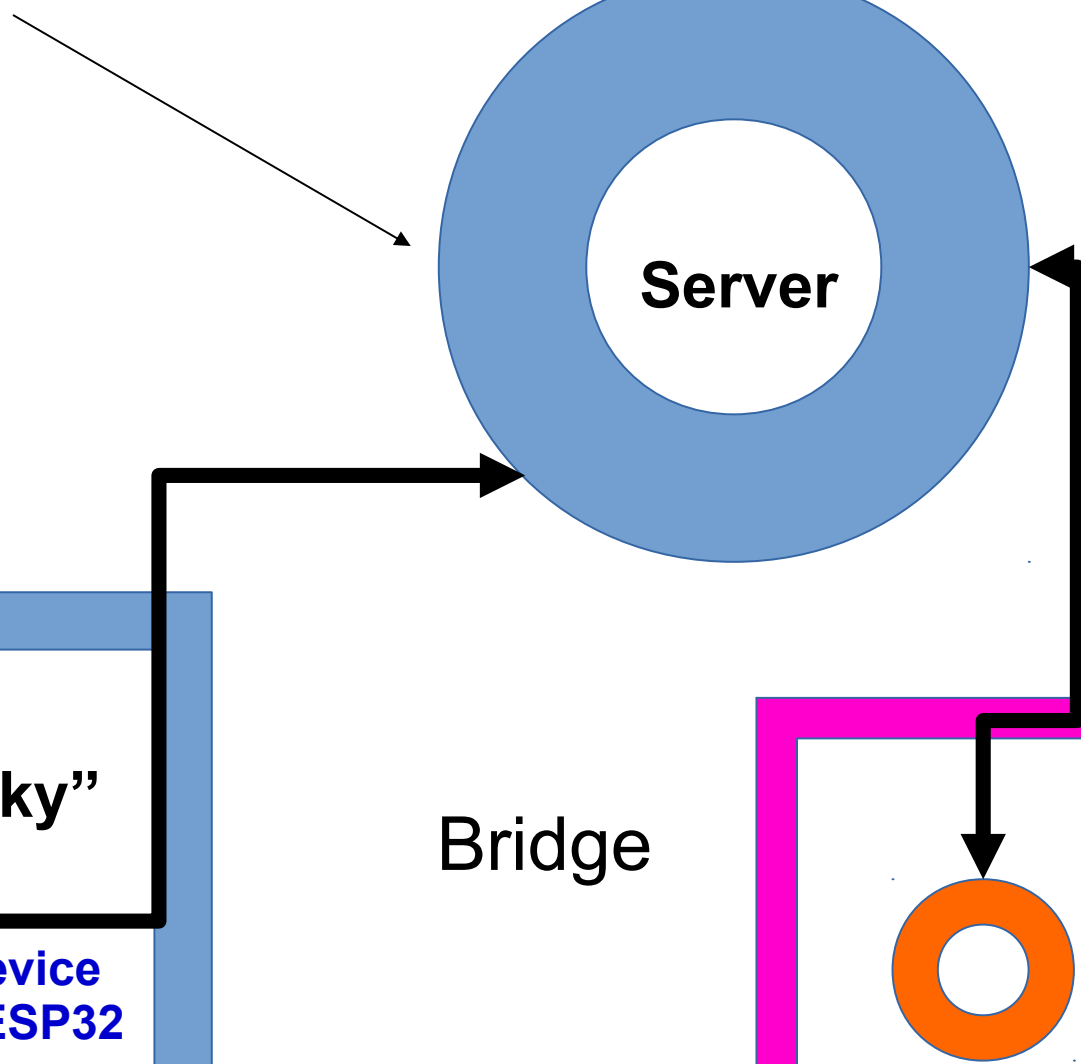
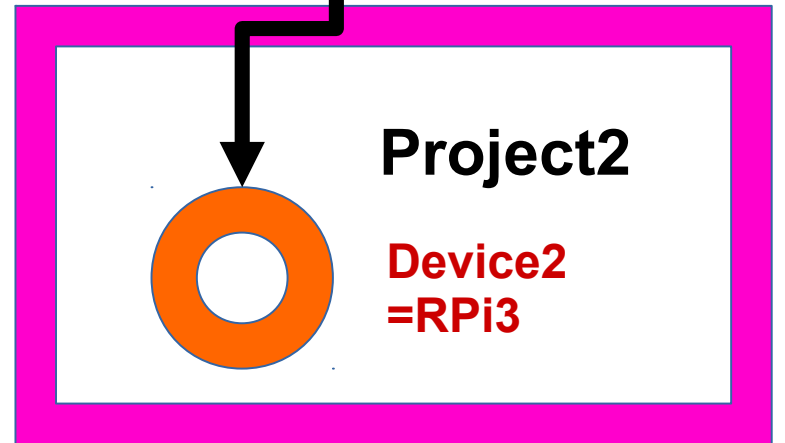


**Project2**

**Device2  
=RPi3**



Bridge



# How?

## **On smartphone:**

Install Blynk APP

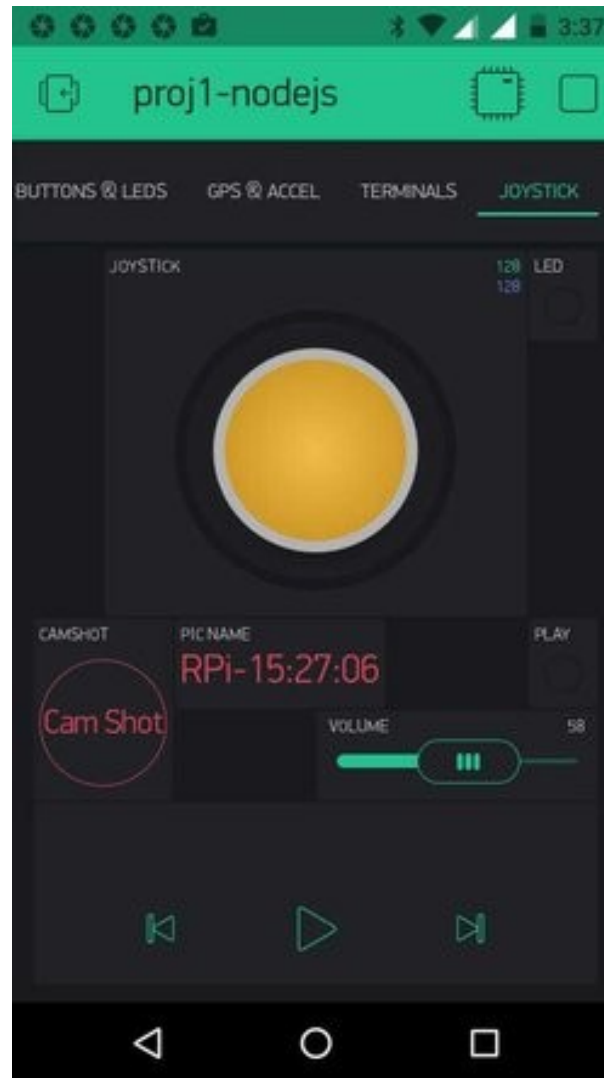
Sign up for blynk account *keep login/pw (emailed)*

Create new project *keep auth token (emailed)*

## **On ESP sketch:**

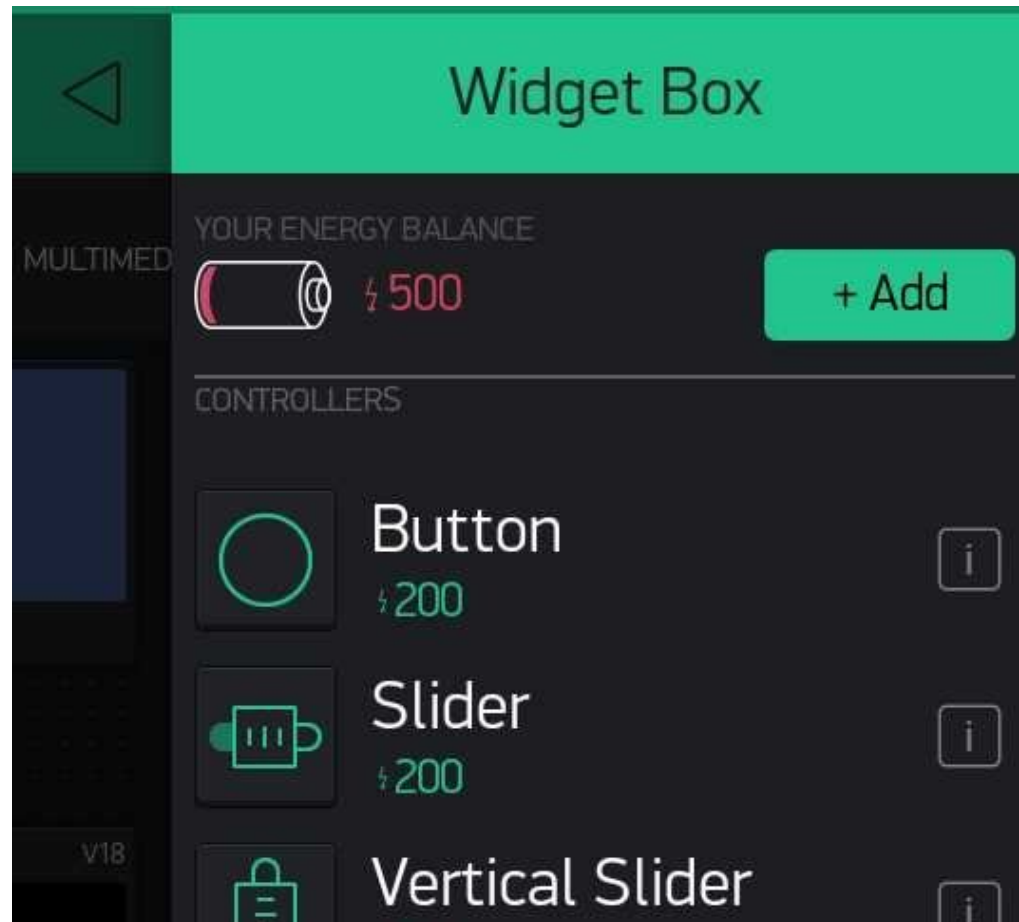
Edit your “auth token” into mypasswords.h

# Lots of APP Widgets



Buttons    LEDs    Joystick    GPS    Accelerometer    Gauge  
Slider    Email    Popup-notification    Terminal    LCD-display    ...

# Widget Palette



# Terms

## **“widget”:**

- 1) communicating icon in APP
- 2) function in controller to talk to APP's widget

## **“write”:**

(=PUSH) = send data. So always think “which end is doing the writing?”

## **“read”:**

(=POLL) = send request. Other end is expected to write data back.

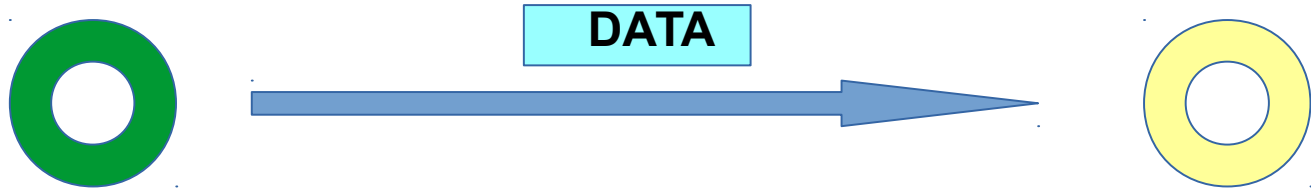


# Configure widgets at APP

Some poll for data (set the timing)

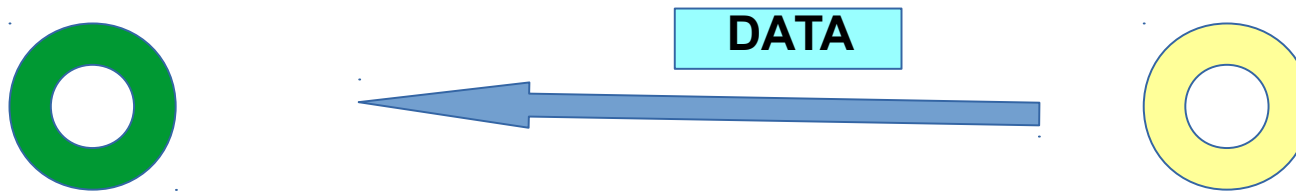
Some just wait for data

Some push data

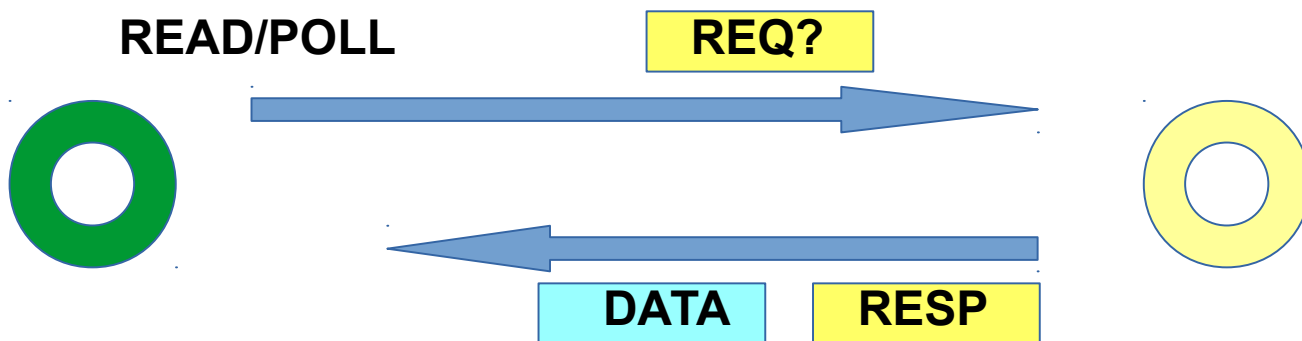


**PUSH/WRITE**

(other end)



**LISTEN/WAIT**



**READ/POLL**

# “PINS”

- **Digital / GPIO pins** – direct control of GPIO  
payload = on/off (0/1)
- **Analog pins**  
payload = some number (eg 0-1023 for arduino)
- **Virtual pins**  
payload = any data, even array of multiple values

Exc for simplest GPIO pin functions, usually a few lines of code needed at device end to receive/send pin messages & act on them.

# Data payload

Simple data packet rather like:

**gpio 21 out 1**

e.g. turn a gpio on

**vpin 13 157**

e.g. LED pwm value

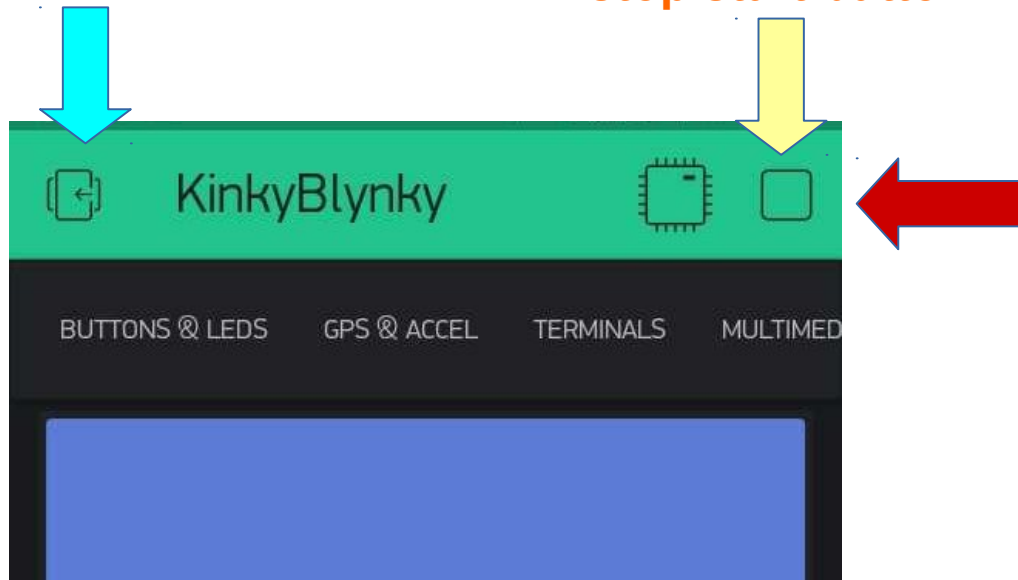
**vpin 23 [ 0.1 , -0.05 , -9.85 ]**

e.g. accelerometer x/y/z readings

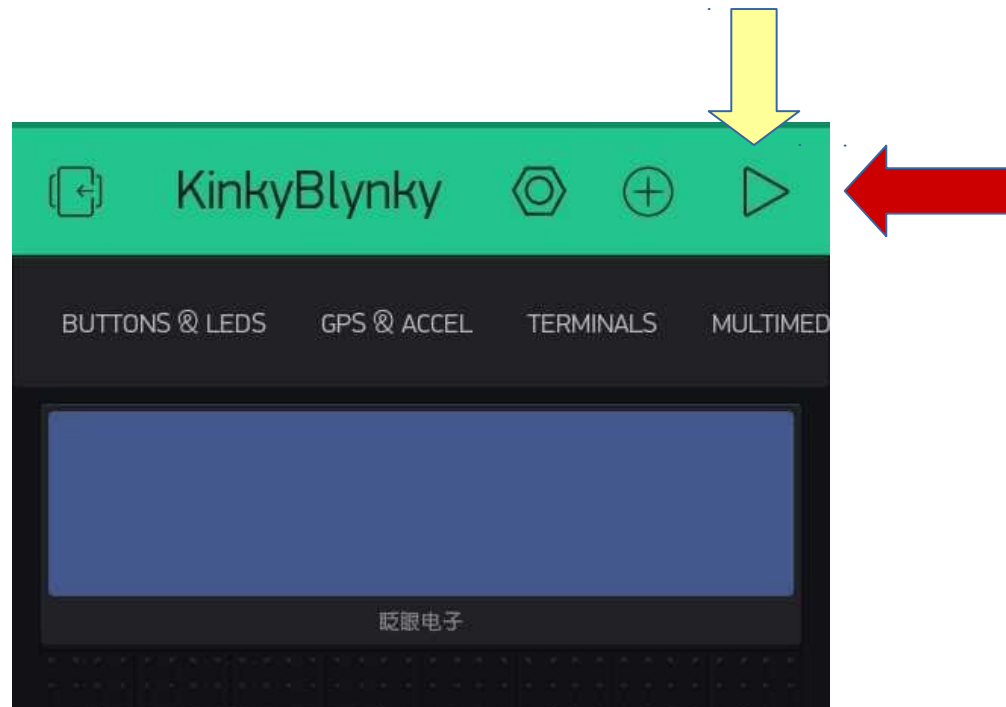
Exit to project chooser

stop/start button

# Controls

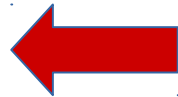
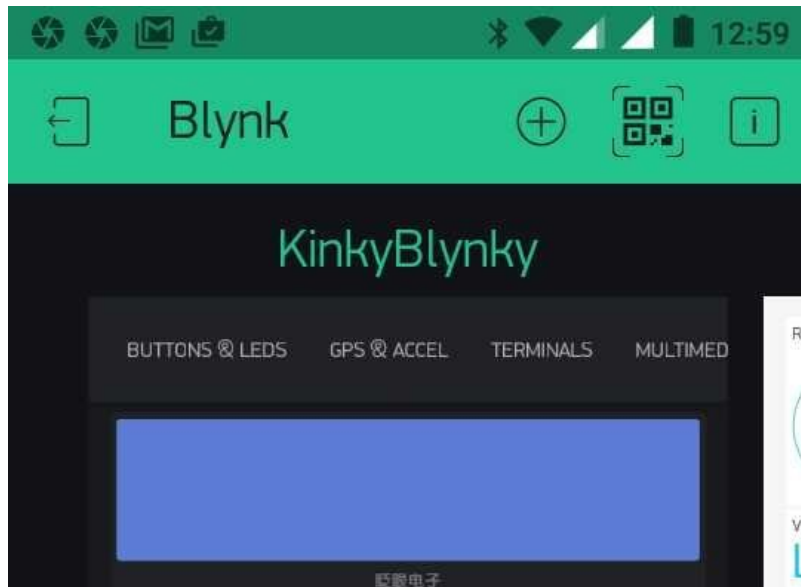


These = "RUNNING"



These = NOT RUNNING  
(edit mode)

# Project control



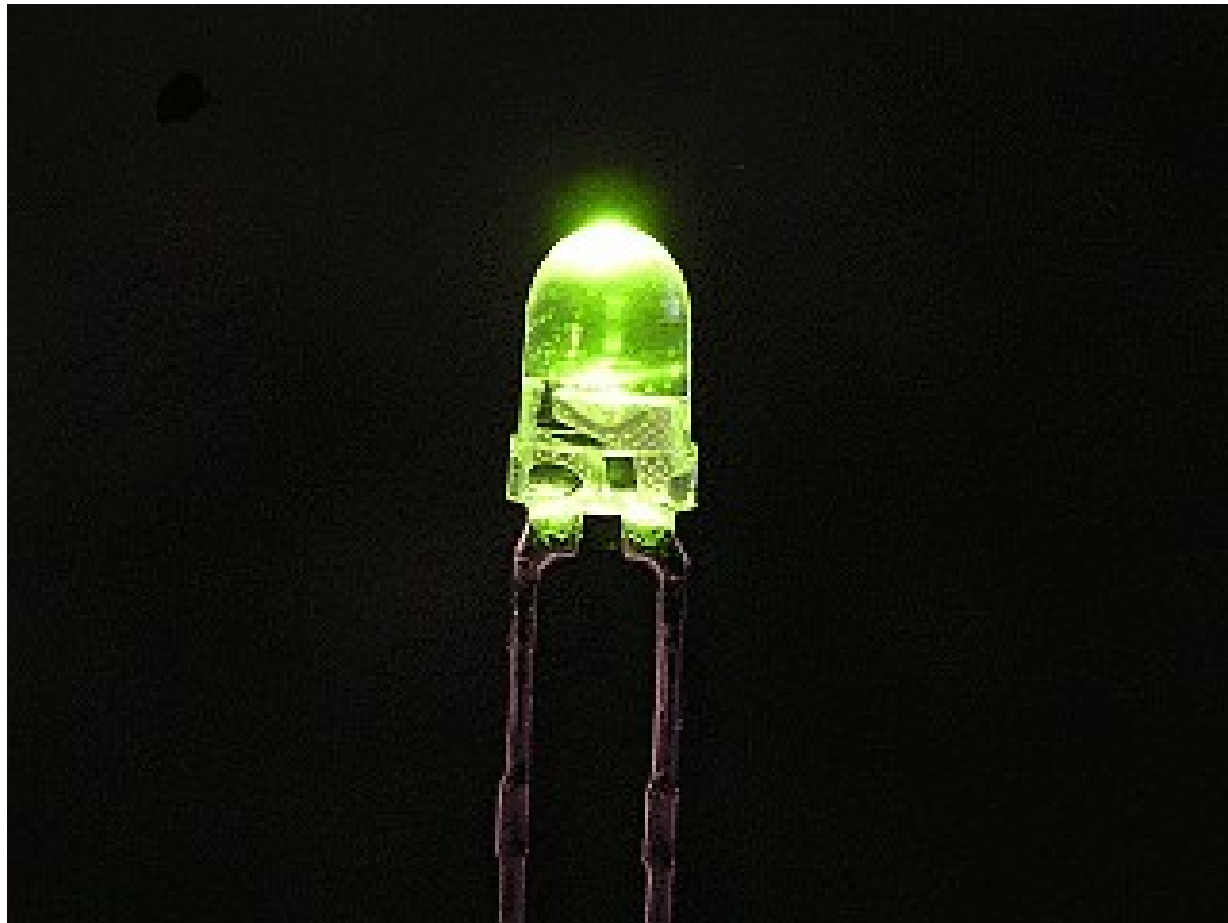
These = the project chooser



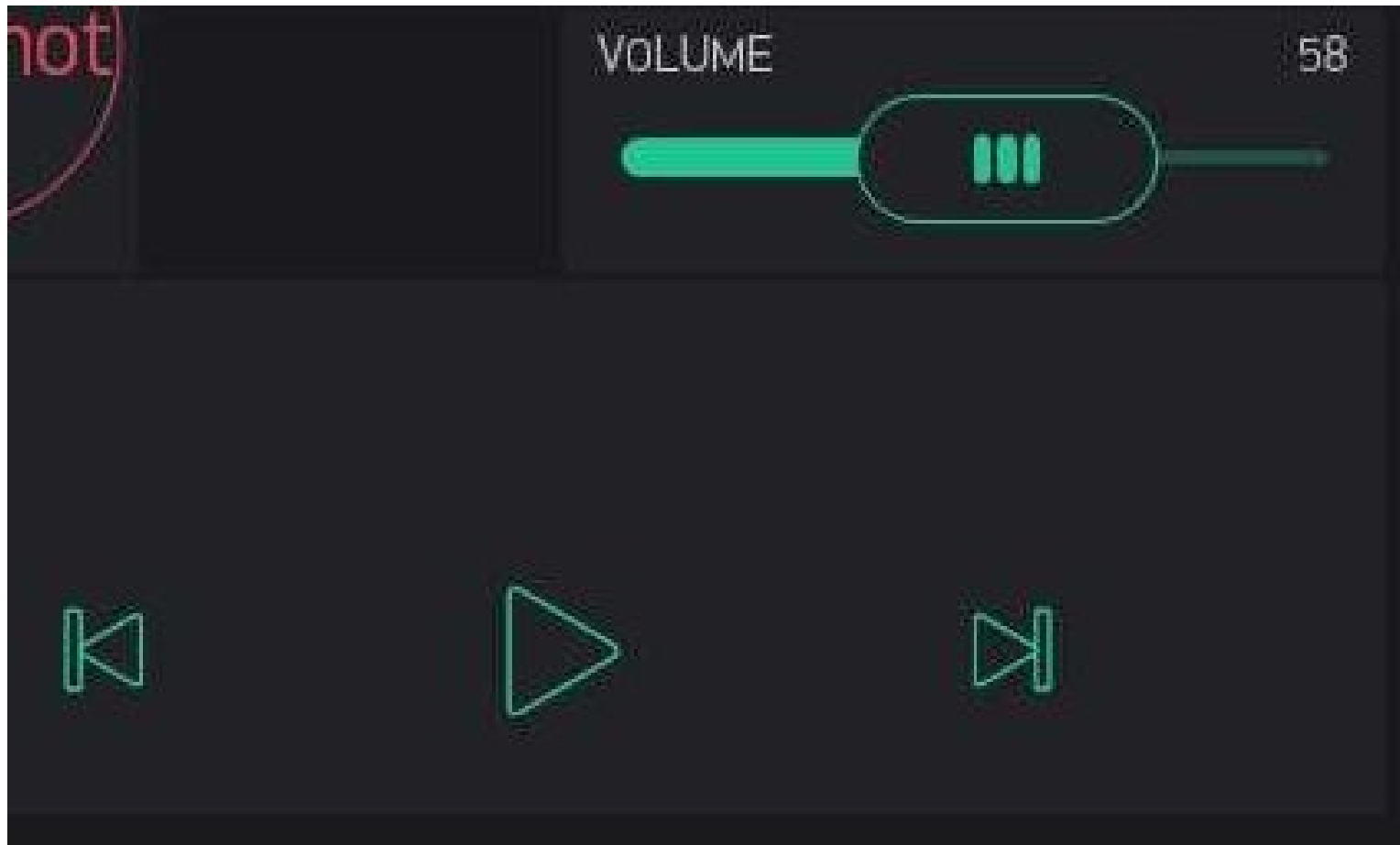
Swipe across: Proj1, Proj2, "new project"



# GPIO – leds & buttons at ESP



# Media buttons





# Starting blynk

sketch = blynksimple

*Plain vanilla GPIO in/out now useable from APP*

*Try the inbuilt LED (5)*

# BlynkSimple.ino

This first sketch only has support for **automated** connection to

- input GPIOs (ie buttons) and
- output GPIOs (ie LEDs)

At APP, set 2 widgets:

- a “button” addressed to the LED on ESP (05)
- a “labelled value” reading from a button on ESP (00?)  
(You could use a wire gpio00 – GND to simulate a button)  
(gpios 00 has onboard pullup, good for using button to GND)

# BlynkSimple

```
1
2 #include "myconfig.h"
3 #include "easywifi.h"
4
5 #define BLYNK_PRINT Serial
6 #include <BlynkSimpleEsp32.h>
7
8 void setup()
9 {
10   Serial.begin(SERIAL_BAUD);
11   wifiInit();
12
13   Blynk.config(BLYNK_TOKEN);
14   Blynk.connect();
15 }
16
17 void loop()
18 {
19   wifiWatch();
20   if(WiFi.isConnected())
21     Blynk.run();
22 }
```



# More comms: Later

Blynk controlling fancier devices

Infrared TV Remote Reader

Bluetooth from phone?