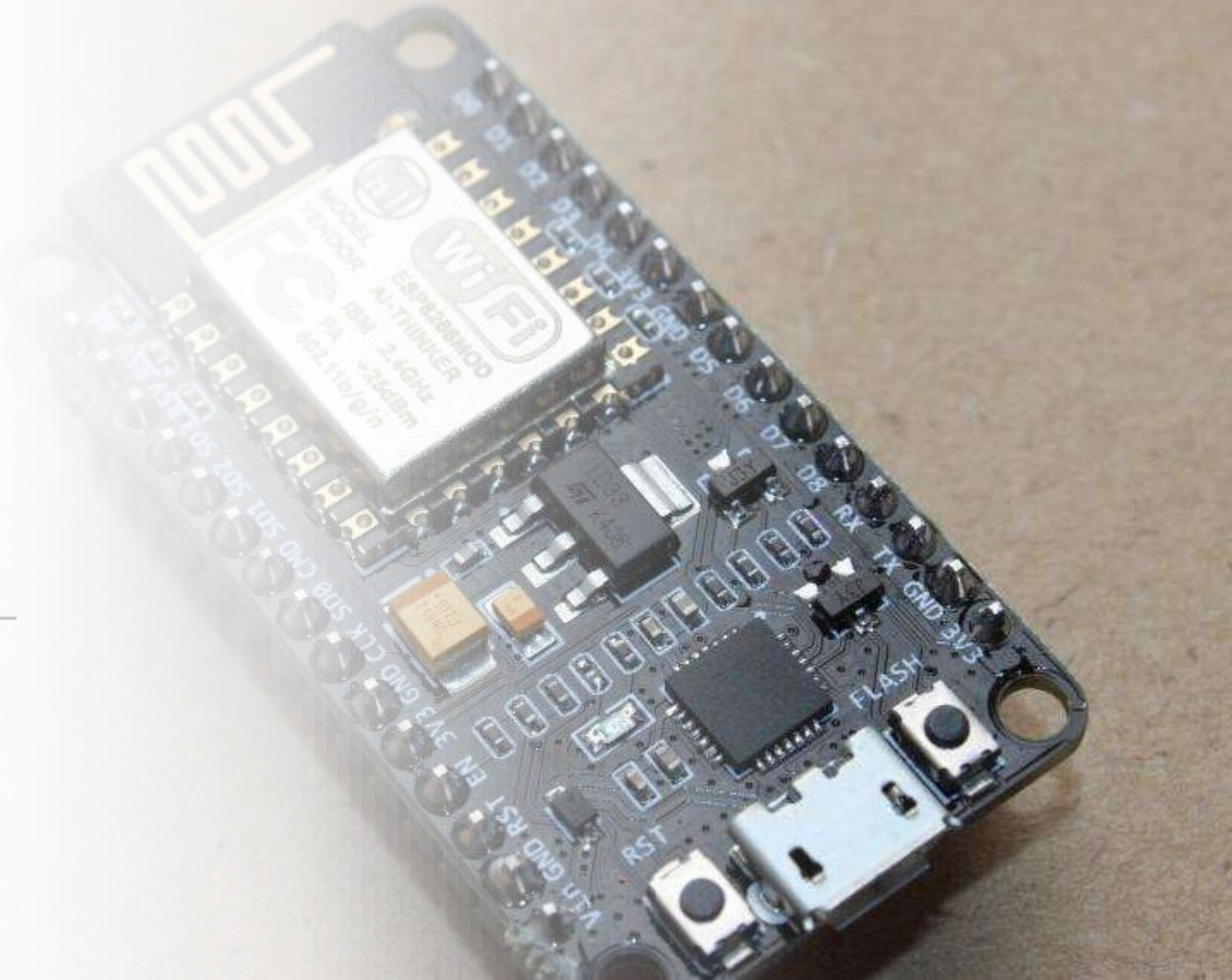
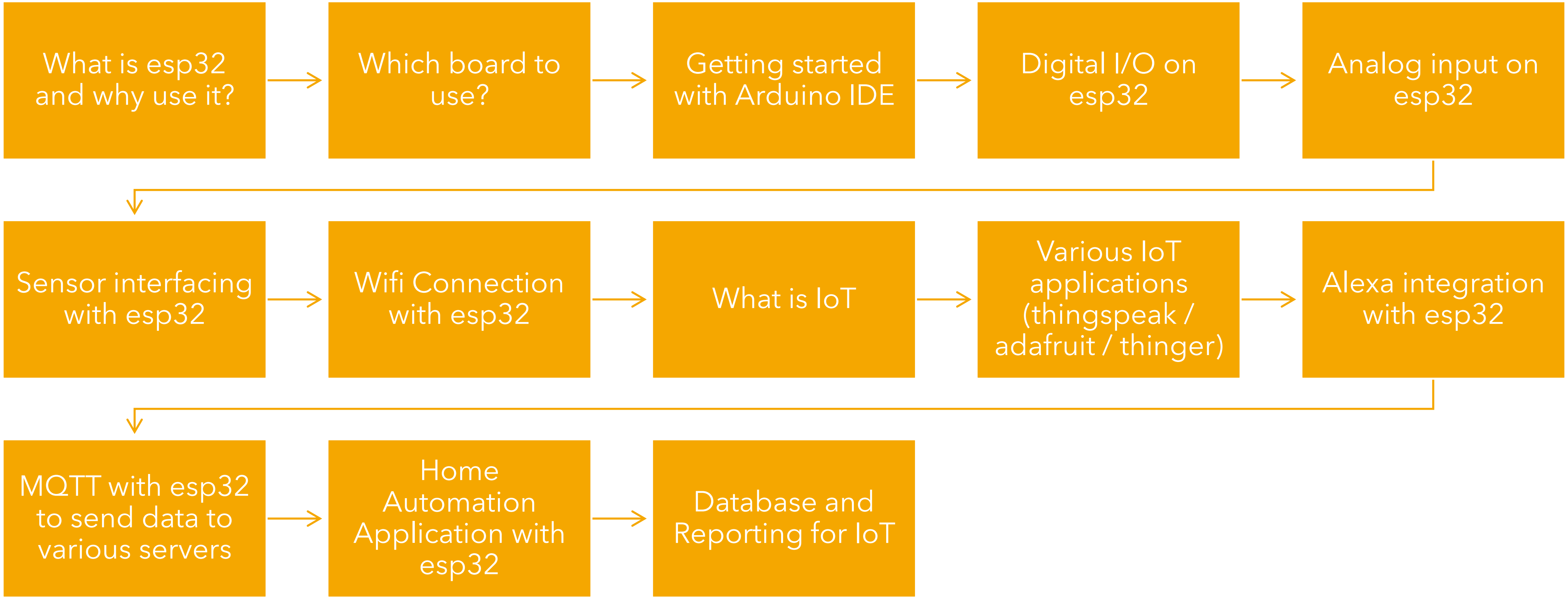
ESP32 and IoT

Introduction



Course Introduction



What is esp32 and why use it?

Which board to use?

Getting started with Arduino IDE

Digital I/O on esp32

Analog input on esp32

Sensor interfacing with esp32

Wifi Connection with esp32

What is IoT

Various IoT applications (thingspeak / adafruit / thinger)

Alexa integration with esp32

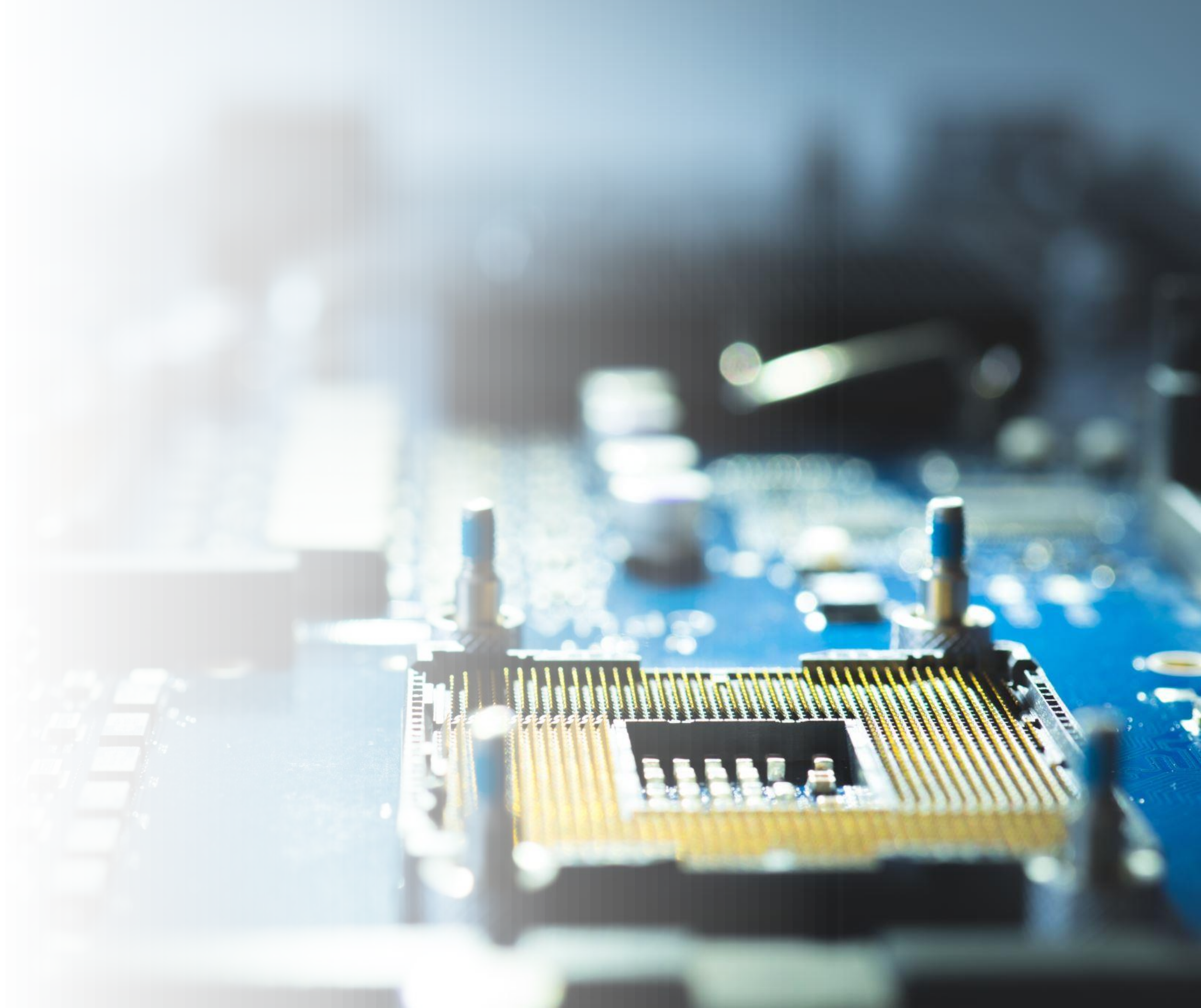
MQTT with esp32 to send data to various servers

Home Automation Application with esp32

Database and Reporting for IoT

What is ESP32?

- Microcontroller?
- SoC System on Chip
- Far complex than general purpose microcontroller
- Meant for IoT Applications
- Developed by expressif systems



Applications?

Home Automation

Smart Agriculture

Smart Building / lightening

Healthcare

Toys / wearable electronics

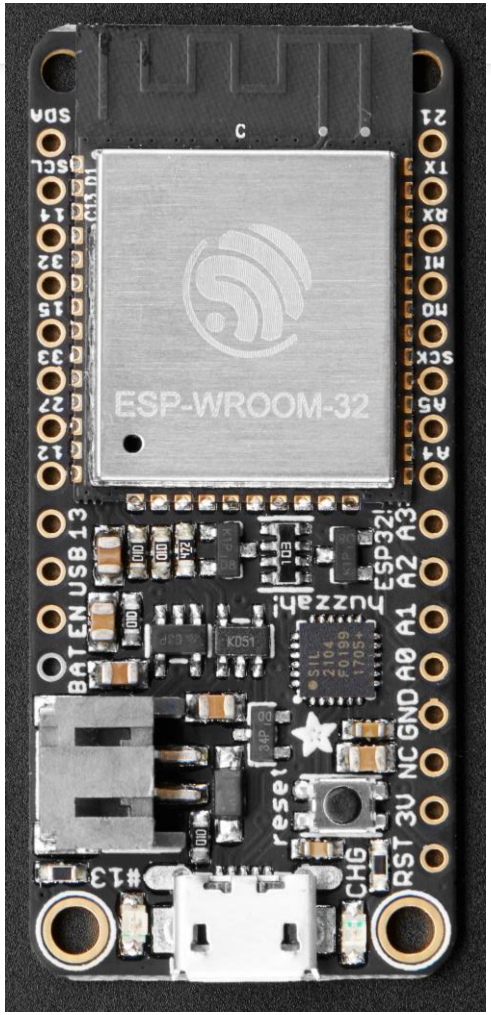
ESP32 Chip and Modules

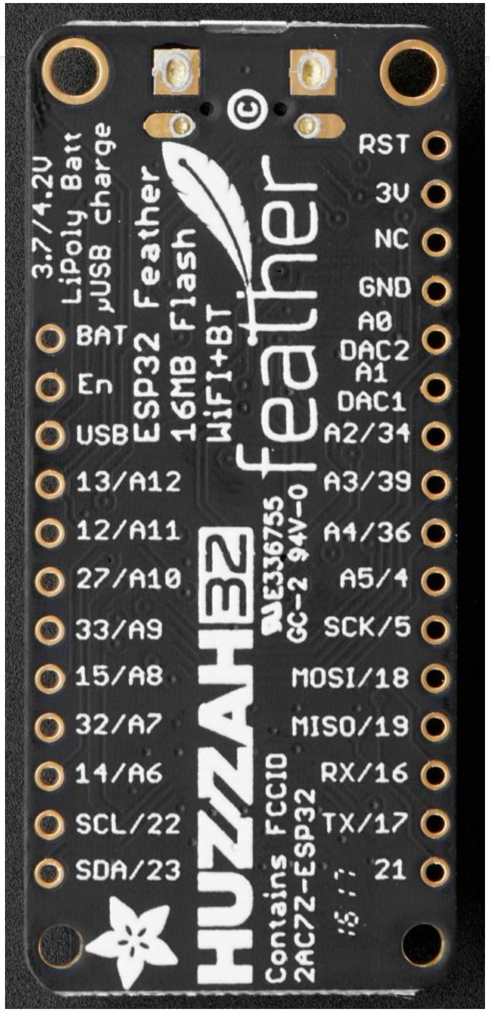
- Various modules makes it easier to use esp32 chip

Which Modules to use?

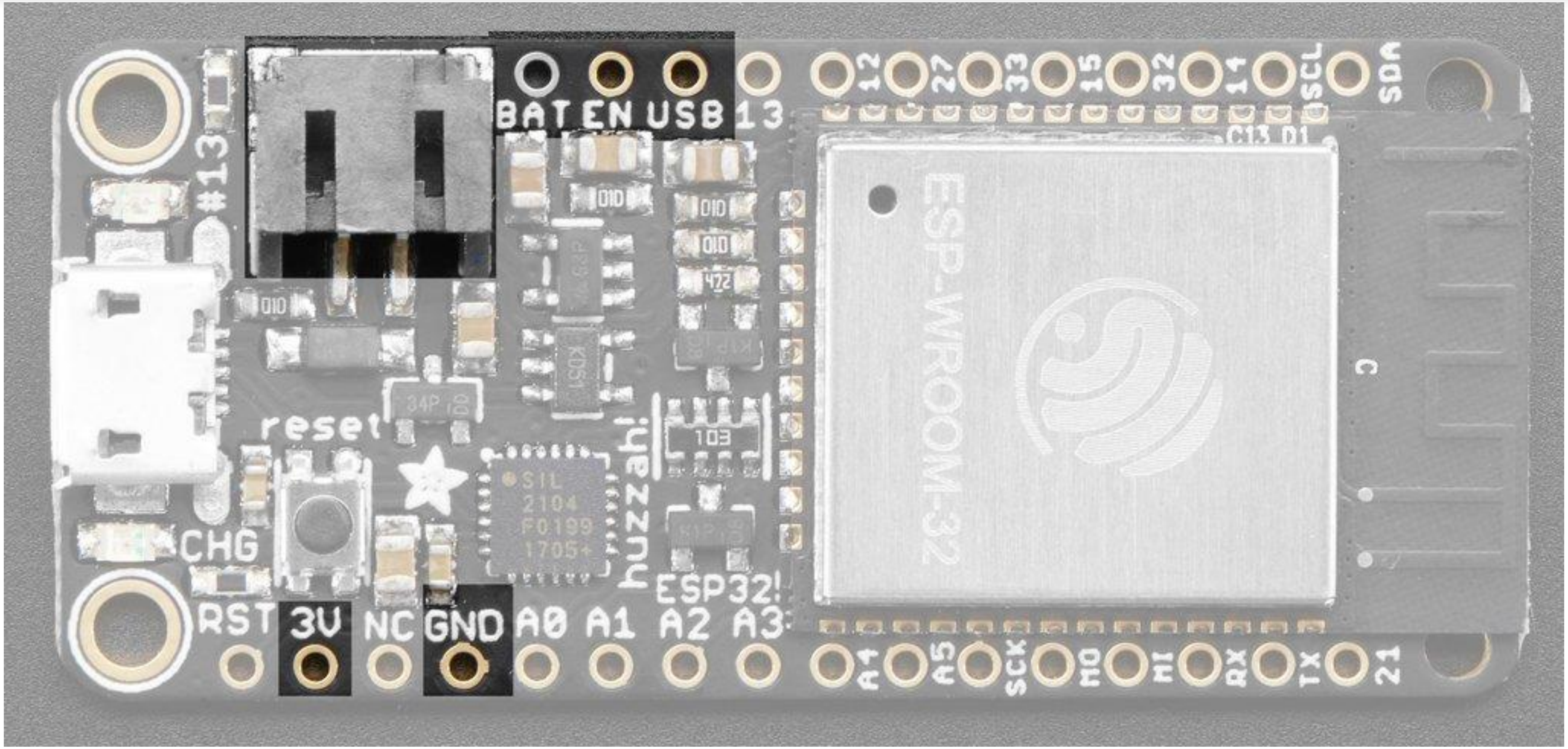
- Lots of different modules available
- Different manufacturers, different pinouts
- Issue in learning is common ground
- We’re going to use Adafruit Huzzah32 Feather Module
- Universal, available across globe

ESP32 Huzzah

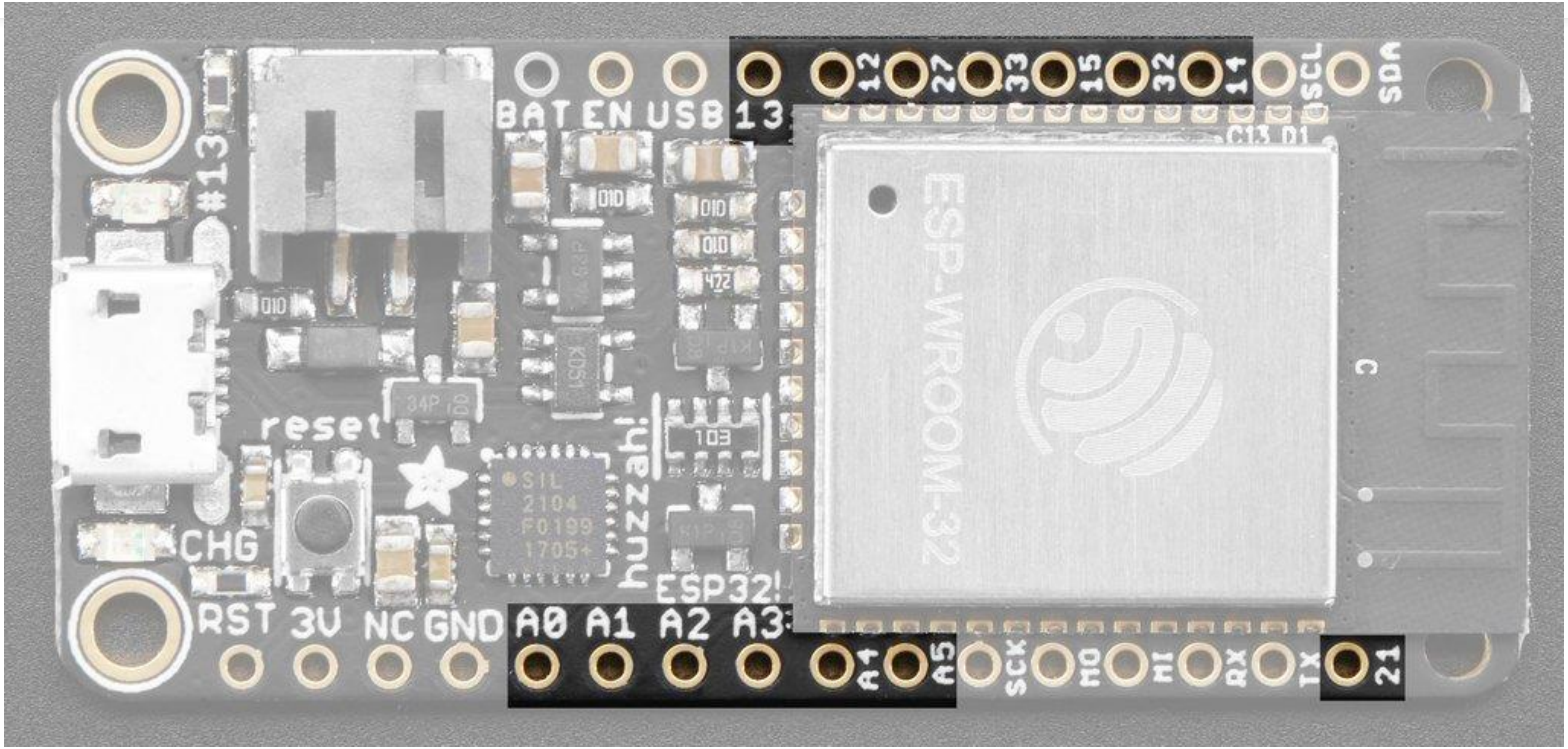




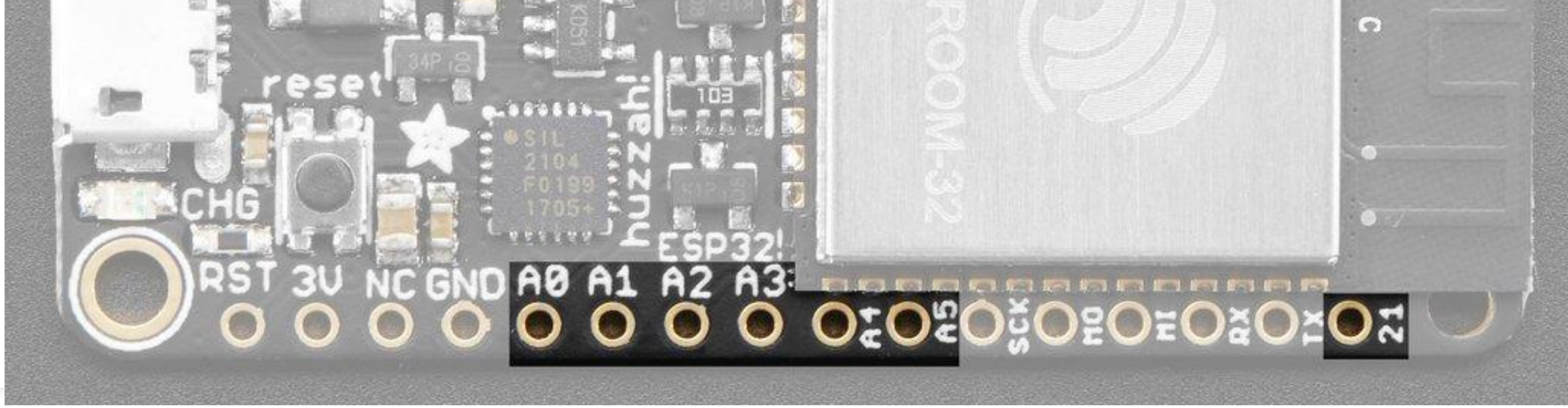
Pinout of Adafruit Huzzah32 → Power Pins



Pinout of Adafruit Huzzah32 → GPIO

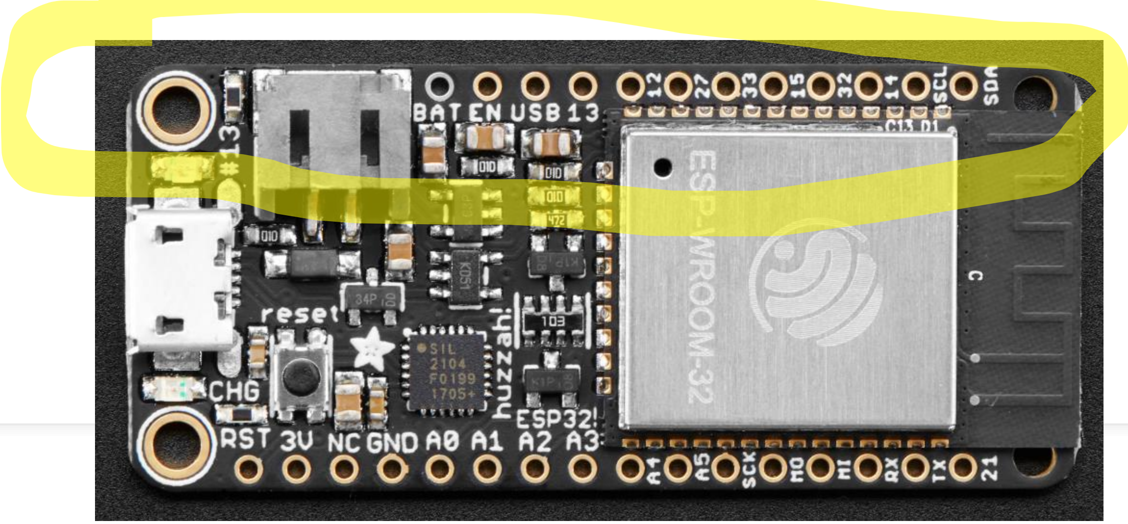


Bottom Row



- **RST , 3v, GND**
- **A0** - this is an analog input A0 and also an analog output DAC2. It can also be used as a GPIO #26. It uses ADC #2
- **A1** - this is an analog input A1 and also an analog output DAC1. It can also be used as a GPIO #25. It uses ADC #2
- **A2** - this is an analog input A2 and also GPI #34**. Note it is *not* an output-capable pin! It uses ADC #1**
- **A3** - this is an analog input A3 and also GPI #39. **Note it is *not* an output-capable pin! It uses ADC #1**
- **A4** - this is an analog input A4 and also GPI #36. **Note it is *not* an output-capable pin! It uses ADC #1**
- **A5** - this is an analog input A5 and also GPIO #4. It uses ADC #2
- **SCK, MO, MI** – SPI Pins
- **RX, Tx** – Uart Pins
- **21** - General purpose IO pin #21

Top Row



13 - This is GPIO #13 and also an analog input A12 on ADC #2. It's also connected to the red LED next to the USB port

12 - This is GPIO #12 and also an analog input A11 on ADC #2. This pin has a pull-down resistor built into it, we **recommend using it as an output only,** or making sure that the pull-down is not affected during boot.

27 - This is GPIO #27 and also an analog input A10 on ADC #2

33 - This is GPIO #33 and also an analog input A9 on ADC #1. It can also be used to connect a 32 KHz crystal.

15 - This is GPIO #15 and also an analog input A8 on ADC #2

32 - This is GPIO #32 and also an analog input A7 on ADC #1. It can also be used to connect a 32 KHz crystal.

14 - This is GPIO #14 and also an analog input A6 on ADC #2

SCL, SDA – I2C Pins

Important Note

- Note you can only read analog inputs on **ADC #2** once WiFi has started as it is shared with the WiFi module.

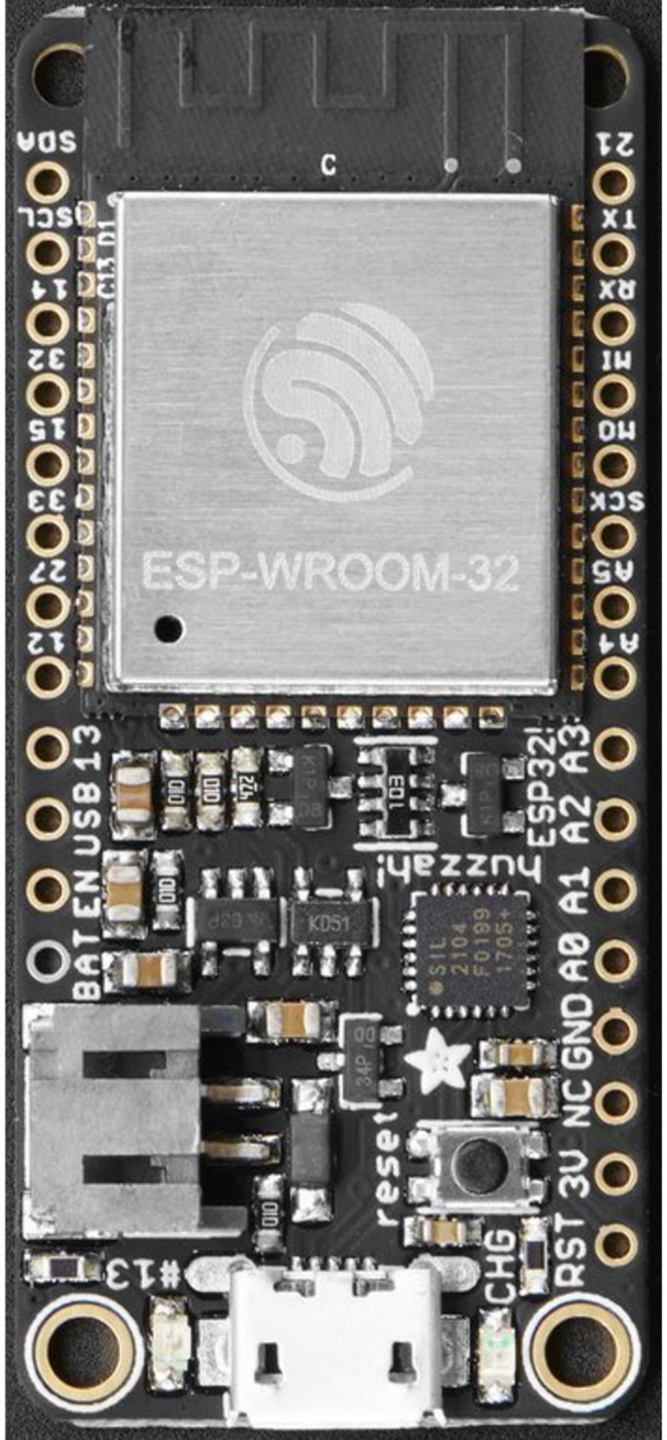
ADAFRUIT HUZZAH32 PIN DIAGRAM

A13 not exposed. It's used for measuring the voltage on the battery. The voltage is divided by 2 so multiply the analogRead by 2.

GPIO#12 Used for booting up. Adafruit suggests not using it or only using for output.

ADC#2 does not work when WiFi is activated. The ESP32 internally uses ADC#2 for WiFi

PWM is possible on every GPIO pin



I2C SDA | GPIO #23

I2C SCL | GPIO #22

TOUCH6 | A6 ADC#2 | GPIO #14

TOUCH9 | A7 ADC#1 | GPIO #32

TOUCH3 | A8 ADC#2 | GPIO #15

TOUCH8 | A9 ADC#1 | GPIO #33

TOUCH7 | A10 ADC#2 | GPIO #27

TOUCH5 | A11 ADC#2 | GPIO #12

TOUCH4 | A12 ADC#2 | GPIO #13

Positive voltage from USB jack, if connected. ~5V

Drive LOW to disable 3.3V regulator

Positive voltage from LiPoly battery, if connected. ~3.7V

GPIO #21

GPIO #17

GPIO #16

GPIO #19 | SPI MISO

GPIO #18 | SPI MOSI

GPIO #5 | SPI SCK

GPIO #4 | A5 ADC#2 | TOUCH0

GPI #36 | A4 ADC#1

GPI #39 | A3 ADC#1

GPI #34 | A2 ADC#1

GPIO #25 | A1 ADC#2 | DAC1

GPIO #26 | A0 ADC#2 | DAC2

GND. Common ground for power and logic

No connect pin (not used)

3.3V output from on-board regulator. Can supply up to 500mA.

Drive LOW to reset (or press Reset button)

By @jonfroehlich

Peripherals Needed for This Course

- Micro USB Cable x 1
- Adafruit Huzzah32 Feather Board x 1
- Bread board x 1
- Few LEDs
- Tactile button / switches
- Relay Module
- DHT22 Temperature and Humidity Sensor
- HC-SR04 Ultrasonic Distance Sensor
- A digital Multi meter
- Connecting Wires

How to add Adafruit Huzzah32 to Arduino

- https://learn.adafruit.com/adafruit-huzzah32-esp32-feather/using-with-arduino-ide
- Install Arduino from Arduino.cc
- https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html

Add below link to preferences of Arduino

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

Search for Board

- Open Boards Manager from Tools > Board menu and install esp32 platform (and do not forget to select your ESP32 board from Tools > Board menu after installation).

Serial Communication

Between esp32 and Computer

Why?

- Easiest output device
- Can print text of any length
- Gives a quick way to print data
- Simple to use

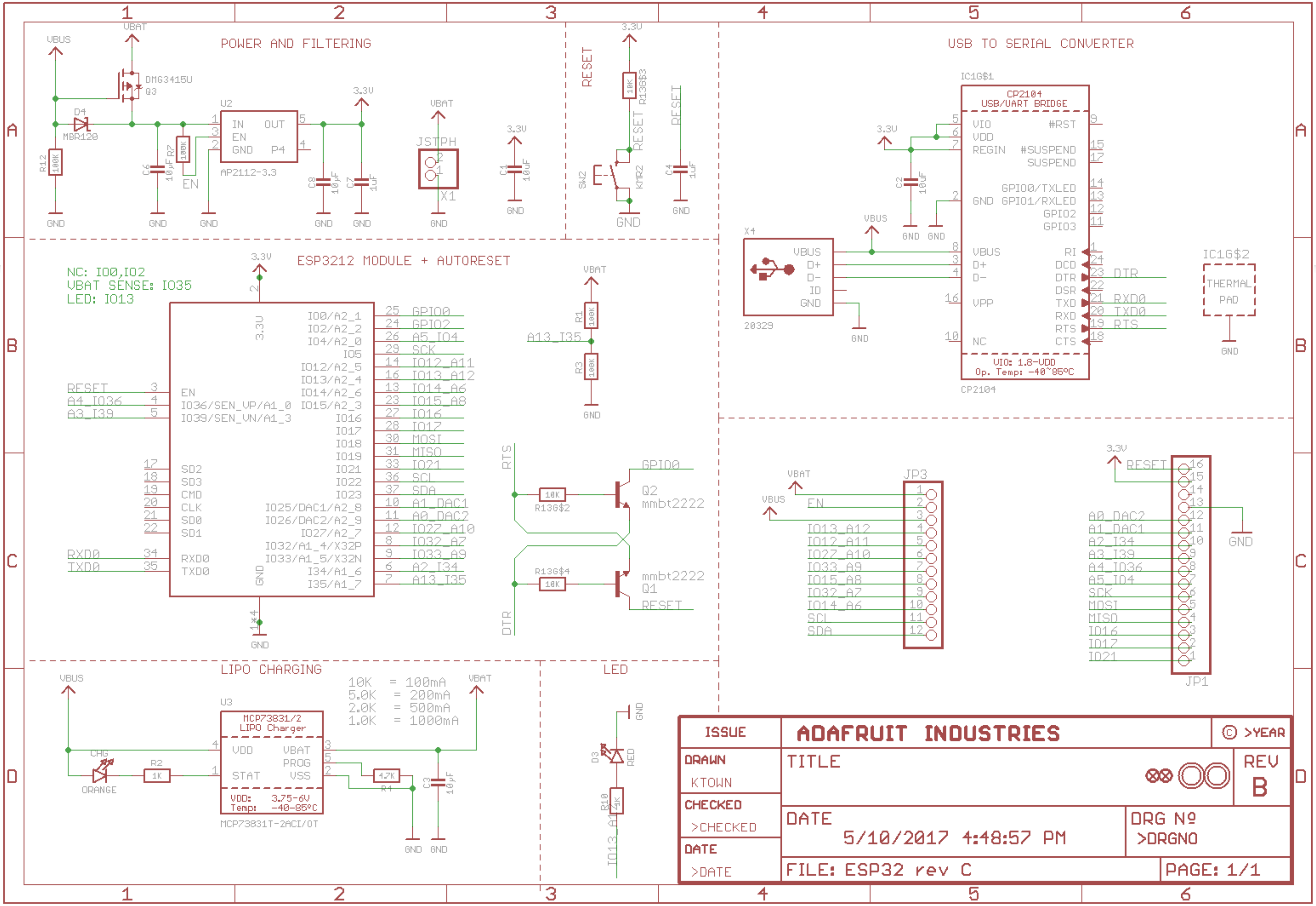
Basics of Serial Communication

Serial Communication Library in Arduino

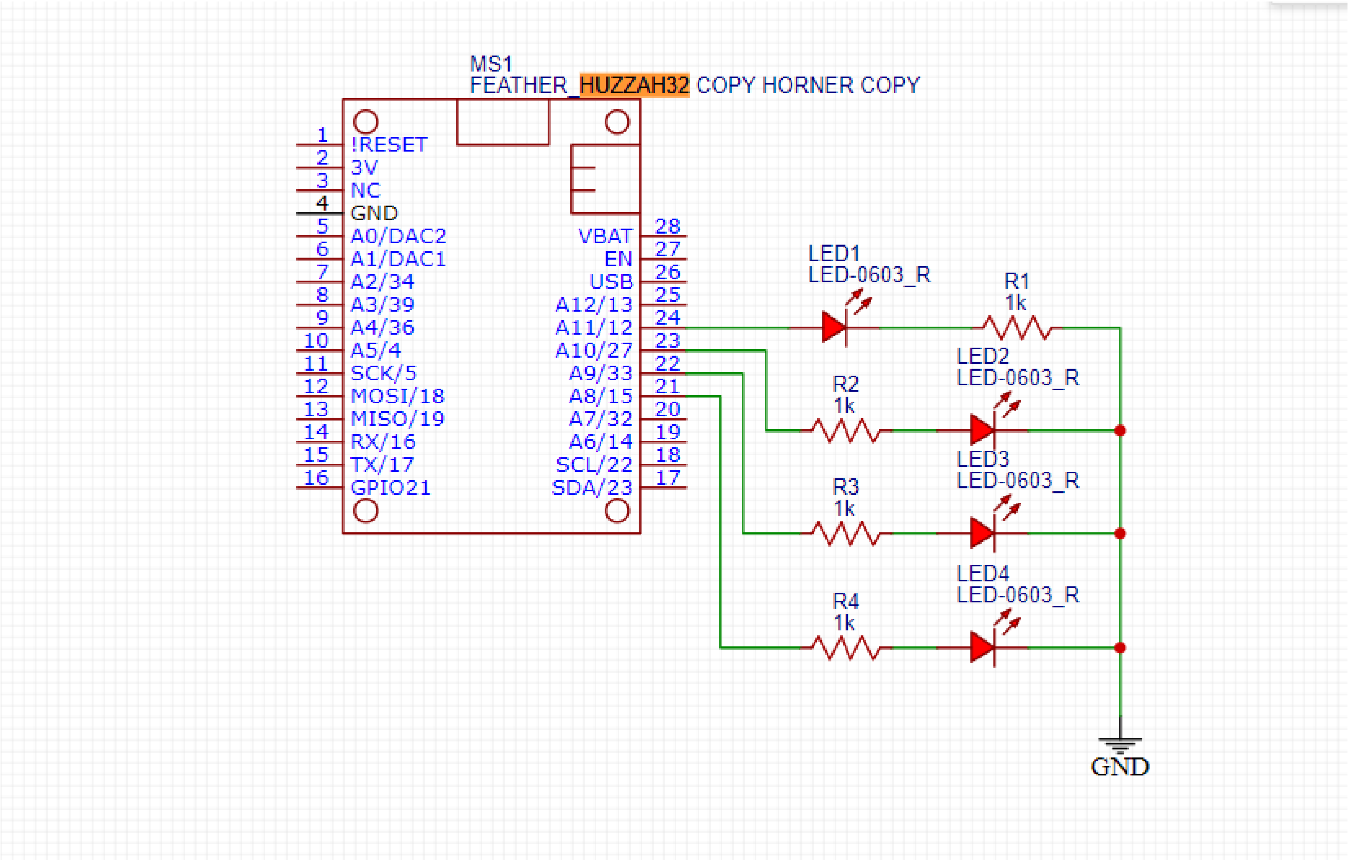
- begin ()
- write ()
- print ()
- println ()
- available()
- read()

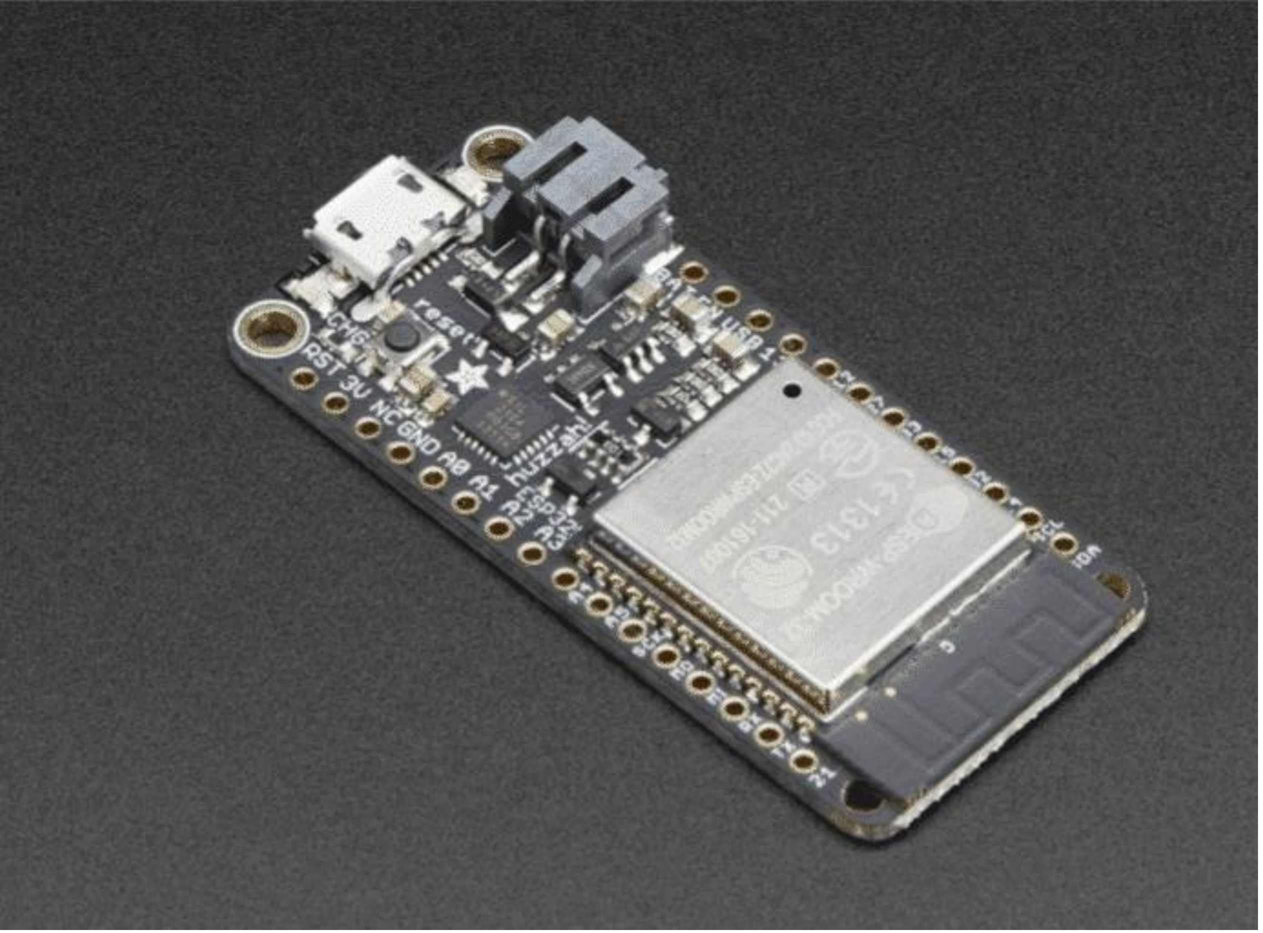
Blink Code

- pinMode
- digitalWrite


POWER AND FILTERING
VBUS
VBAT
DMG3415U
Q3
D4
MBR120
R12
100K
C6
10µF
R7
100K
EN
U2
IN
OUT
EN
GND
P4
AP2112-3.3
3.3V
C8
10µF
C7
1uF
JSTPH
X1
C1
10uF
GND
RESET
10K
R13G$3
SW2
KMR2
C4
1uF
USB TO SERIAL CONVERTER
IC1G$1
CP2104
USB/UART BRIDGE
VIO
VDD
REGIN
#RST
#SUSPEND
SUSPEND
GPIO0/TXLED
GPIO1/RXLED
GPIO2
GPIO3
VBUS
D+
D-
VPP
NC
RI
DCD
DTR
DSR
TXD
RXD
RTS
CTS
VIO: 1.8-VDD
Op. Temp: -40~85ºC
C2
10uF
X4
ID
20329
DTR
RXD0
TXD0
RTS
IC1G$2
THERMAL PAD
ESP3212 MODULE + AUTORESET
NC: IO0,IO2
VBAT SENSE: IO35
LED: IO13
IO0/A2_1
IO2/A2_2
IO4/A2_0
IO5
IO12/A2_5
IO13/A2_4
IO14/A2_6
IO15/A2_3
IO16
IO17
IO18
IO19
IO21
IO22
IO23
IO25/DAC1/A2_8
IO26/DAC2/A2_9
IO27/A2_7
IO32/A1_4/X32P
IO33/A1_5/X32N
I34/A1_6
I35/A1_7
IO36/SEN_VP/A1_0
IO39/SEN_VN/A1_3
SD2
SD3
CMD
CLK
SD0
SD1
RXD0
TXD0
GPIO0
GPIO2
A5_IO4
SCK
IO12_A11
IO13_A12
IO14_A6
IO15_A8
IO16
IO17
MOSI
MISO
IO21
SCL
SDA
A1_DAC1
A0_DAC2
IO27_A10
IO32_A7
IO33_A9
A2_I34
A13_I35
RESET
A4_IO36
A3_I39
A13_I35
R1
R3
100K
RTS
DTR
R13G$2
R13G$4
10K
Q2
Q1
mmbt2222
JP3
JP1
LIPO CHARGING
10K = 100mA
5.0K = 200mA
2.0K = 500mA
1.0K = 1000mA
U3
MCP73831/2
LIPO Charger
VDD
VBAT
PROG
STAT
VSS
VDD: 3.75-6V
Temp: -40-85ºC
MCP73831T-2ACI/OT
CHG
ORANGE
R2
1K
4.7K
R4
C3
10µF
LED
D3
RED
R10
1K
IO13_A12
ISSUE
ADAFRUIT INDUSTRIES
© >YEAR
DRAWN
KTOWN
CHECKED
>CHECKED
DATE
>DATE
TITLE
REV
B
DATE
5/10/2017 4:48:57 PM
DRG Nº
>DRGNO
FILE: ESP32 rev C
PAGE: 1/1

Interfacing of 4 x LEDs with ESP32 Huzzah Boad

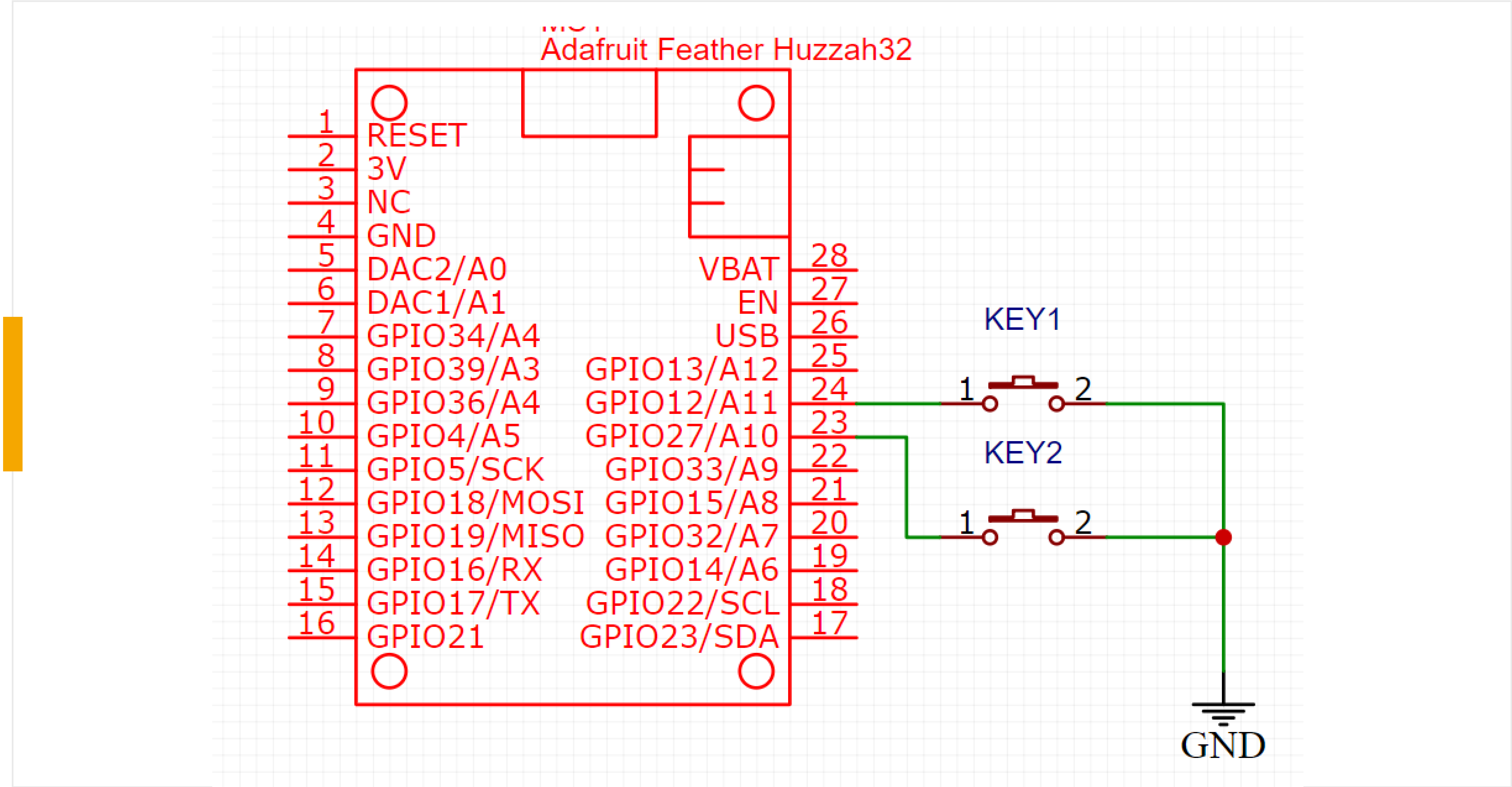




Accepting digital Input on ESP32

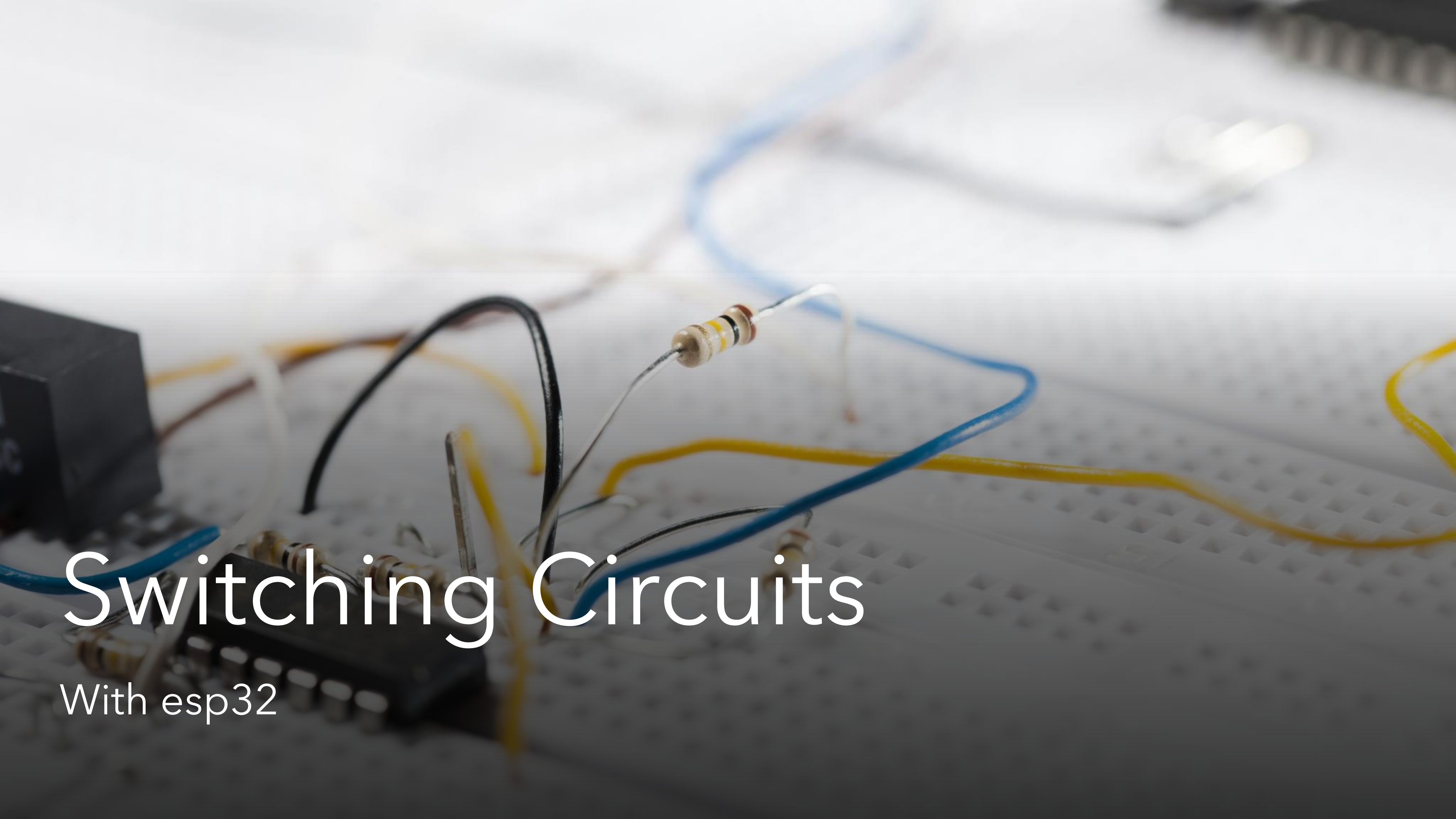
Pull Up Resistor

Pull Down Resistor


Adafruit Feather Huzzah32
1 RESET
2 3V
3 NC
4 GND
5 DAC2/A0
6 DAC1/A1
7 GPIO34/A4
8 GPIO39/A3
9 GPIO36/A4
10 GPIO4/A5
11 GPIO5/SCK
12 GPIO18/MOSI
13 GPIO19/MISO
14 GPIO16/RX
15 GPIO17/TX
16 GPIO21
VBAT 28
EN 27
USB 26
GPIO13/A12 25
GPIO12/A11 24
GPIO27/A10 23
GPIO33/A9 22
GPIO15/A8 21
GPIO32/A7 20
GPIO14/A6 19
GPIO22/SCL 18
GPIO23/SDA 17
KEY1
1
2
KEY2
1
2
GND

Tasks Based on Switch

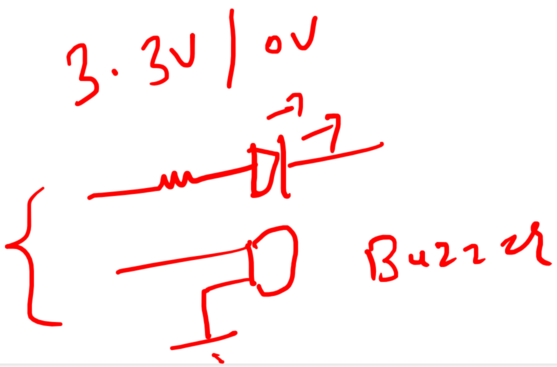
- Up Down Counter using switch
- LED Control using switch
- 2 LEDs and 2 switches. When one switch is pressed, both ON and when other switch is pressed both OFF
- 2 LEDs and 2 switches. When one switch is pressed, LED2 OFF and LED1 ON when another switch is pressed LED1 OFF and LED2 ON



Switching Circuits

With esp32

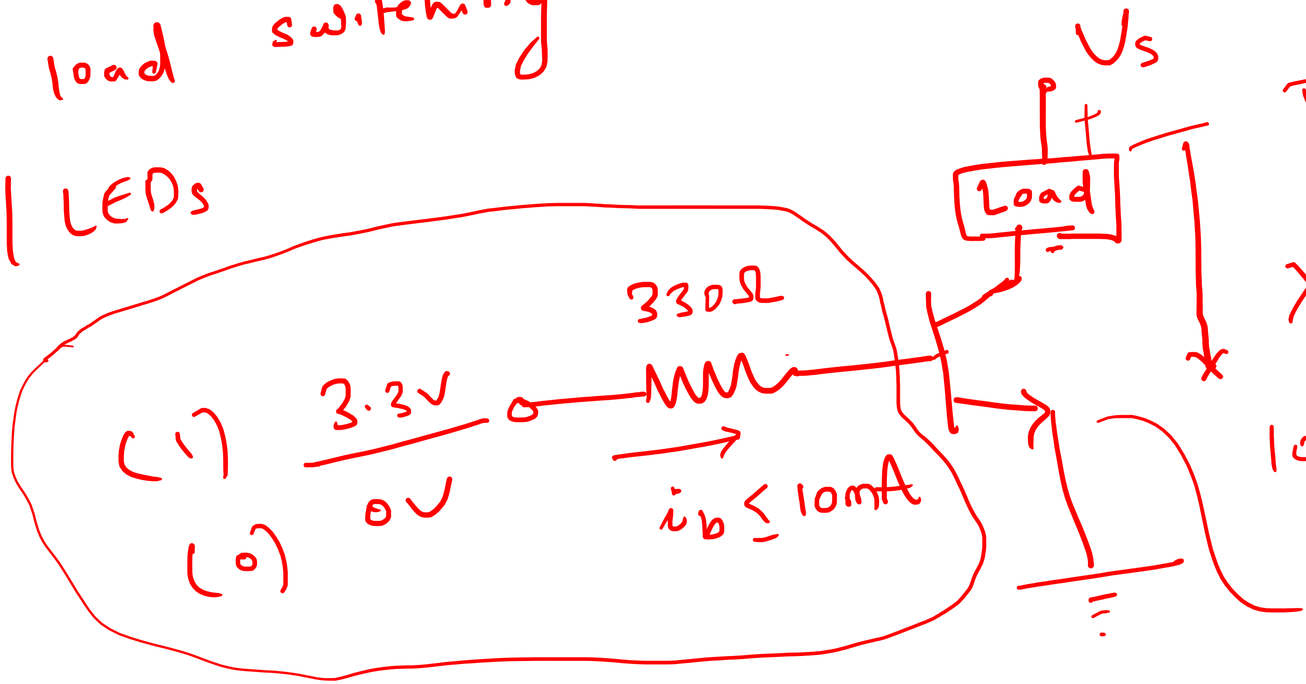
Transistorized Switching



→ DC Load switching

→ AC load switching

Motor| Buzzer| LEDs



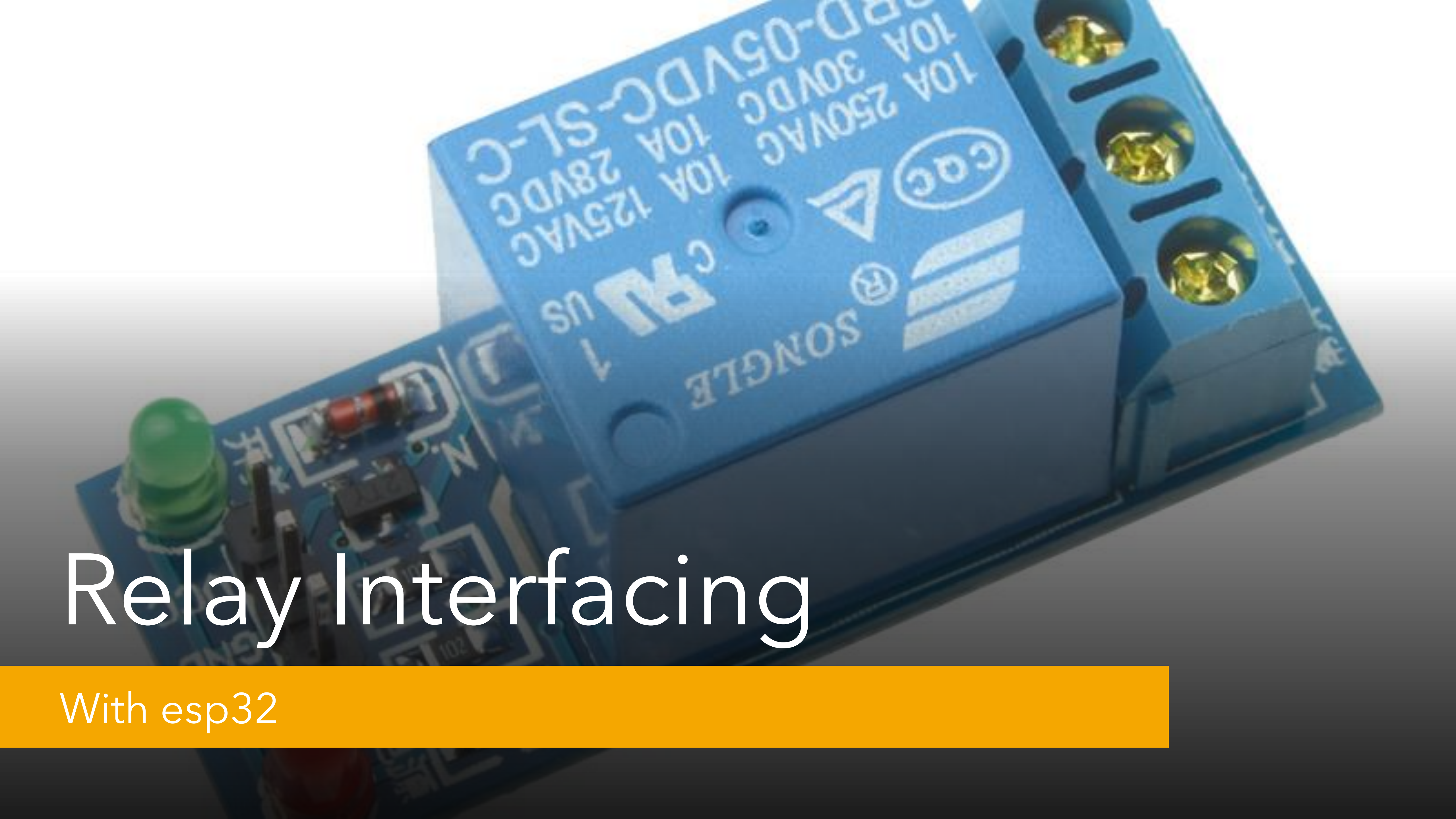
Turn on

X Turns off.

100mA/200mA

BC548 | 2N3904

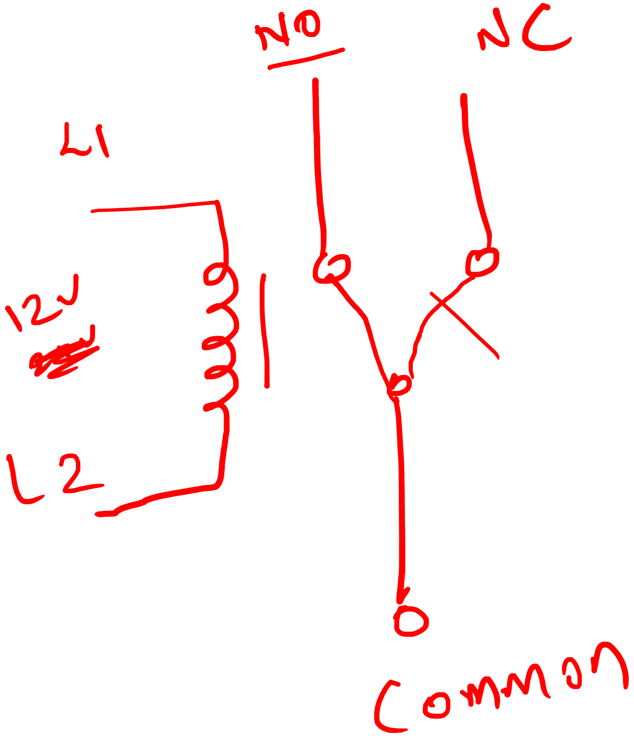
BD139

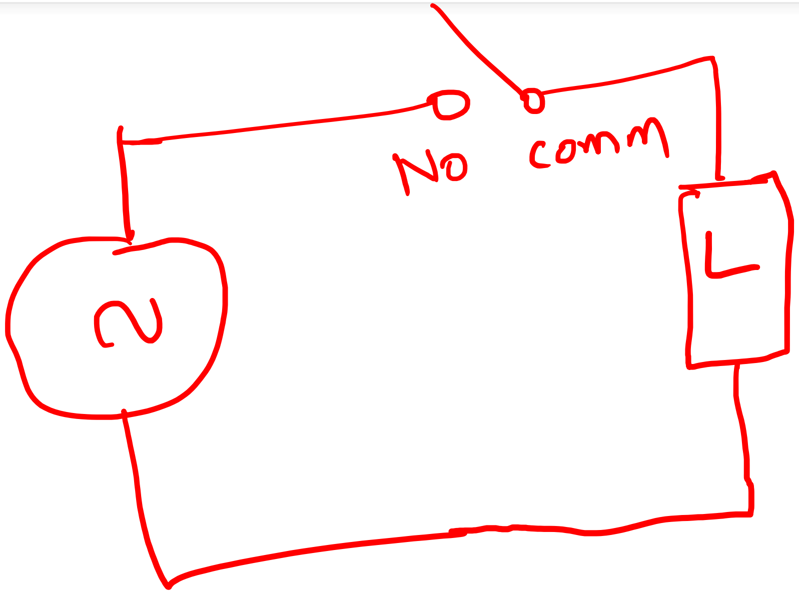


Relay Interfacing

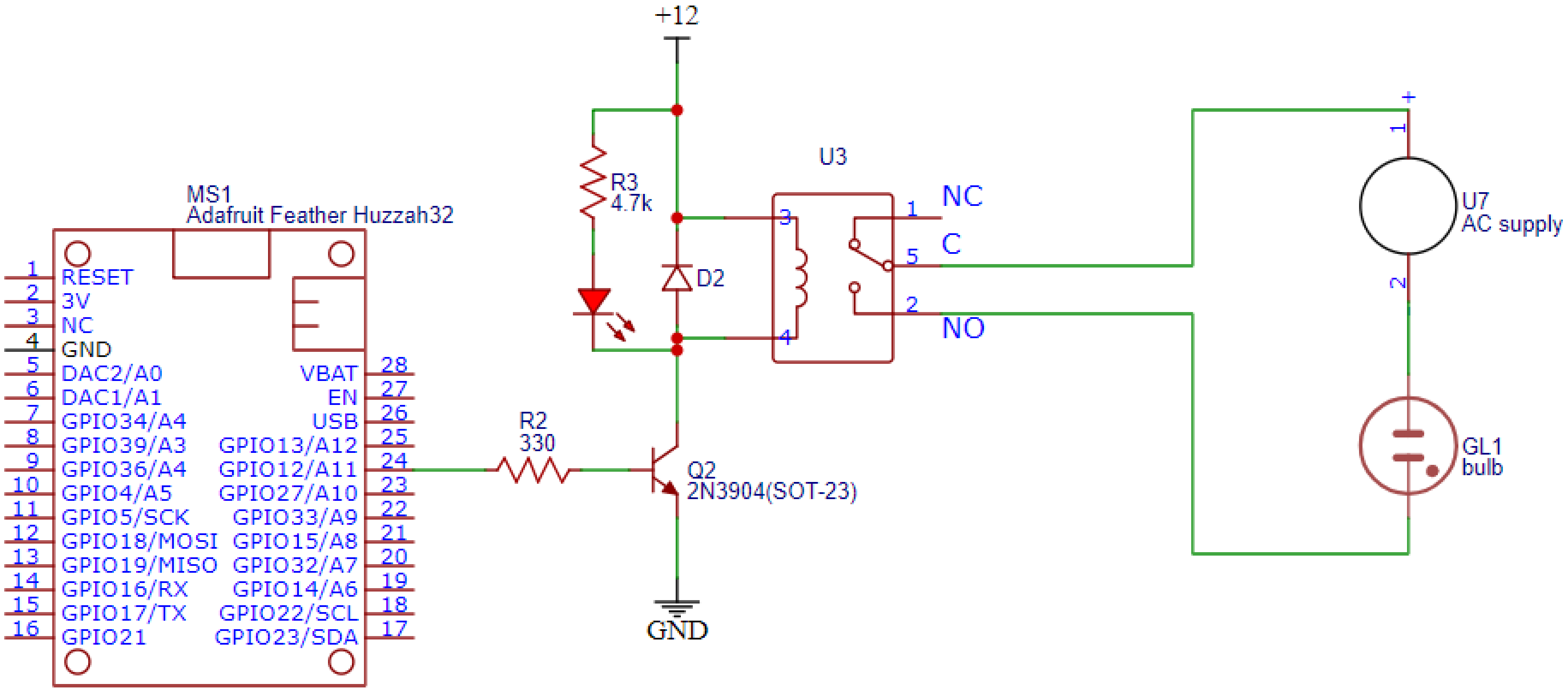
With esp32

Output Relay





Relay Switching Circuit

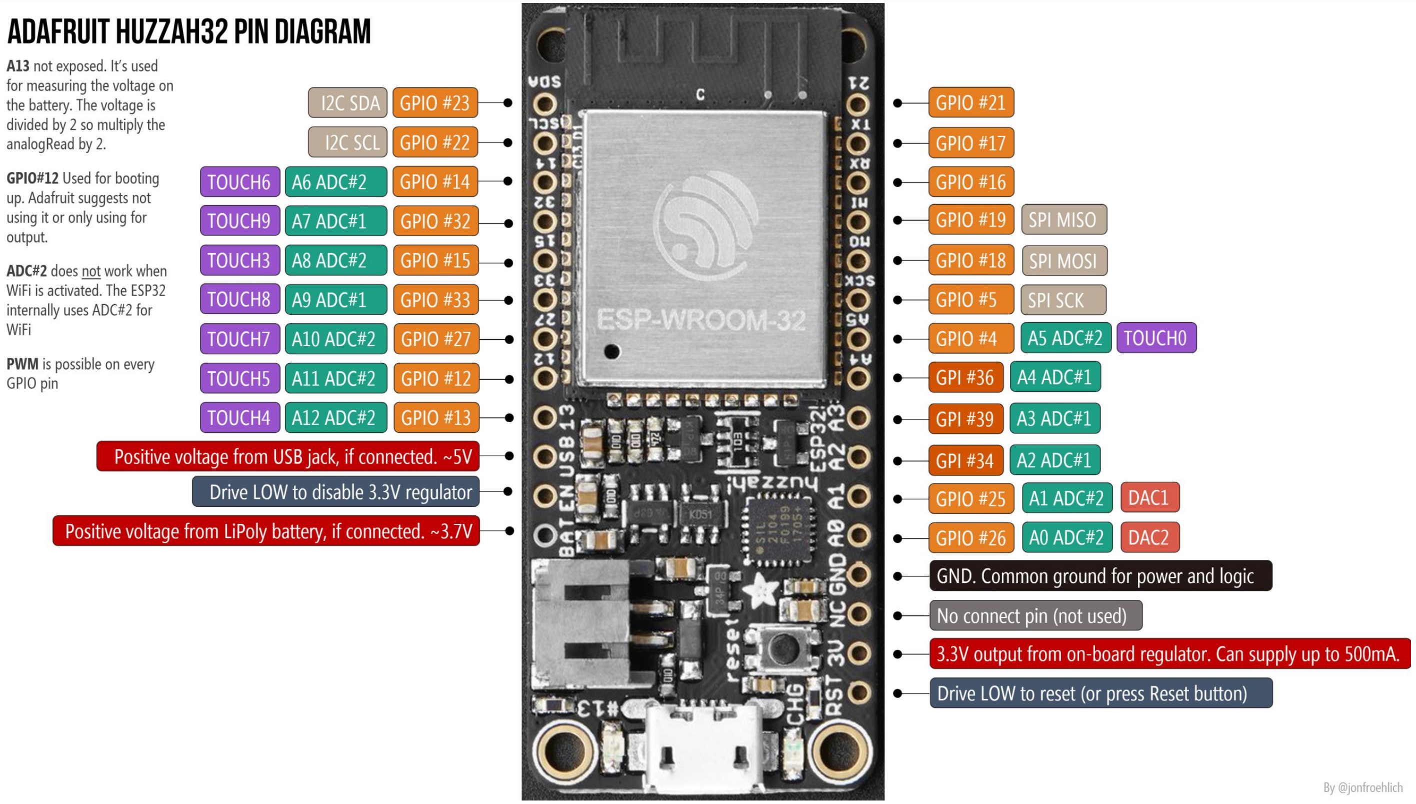


Touch Input on ESP32

Adafruit Huzzah32 Board

How to accept capacitive touch on esp32 Feather Board (touchread)

- 14
- 32
- 15
- 33
- 27
- 12
- 13
- 4



ADAFRUIT HUZZAH32 PIN DIAGRAM

A13 not exposed. It's used for measuring the voltage on the battery. The voltage is divided by 2 so multiply the analogRead by 2.

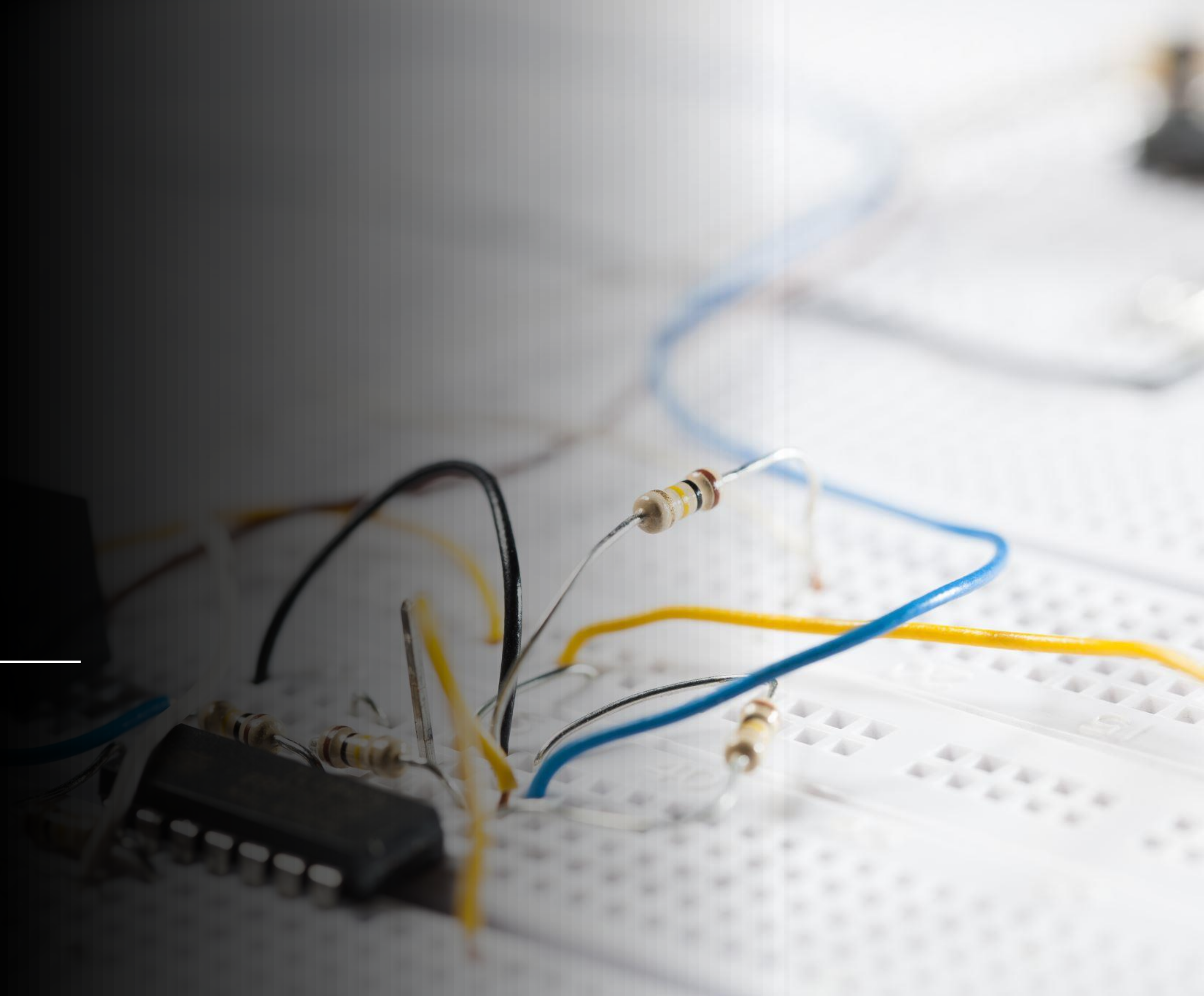
GPIO#12 Used for booting up. Adafruit suggests not using it or only using for output.

ADC#2 does not work when WiFi is activated. The ESP32 internally uses ADC#2 for WiFi

PWM is possible on every GPIO pin

Sensors Interfacing

To esp32

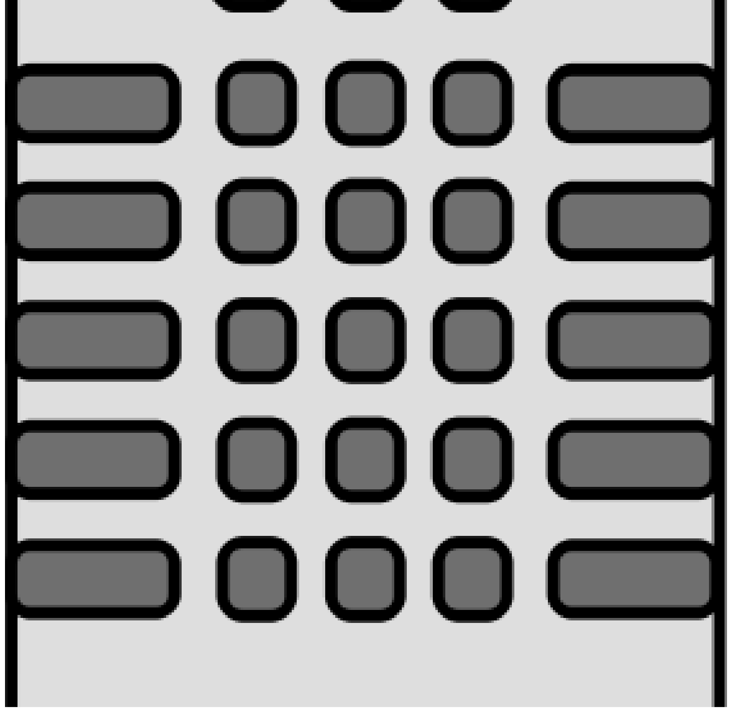


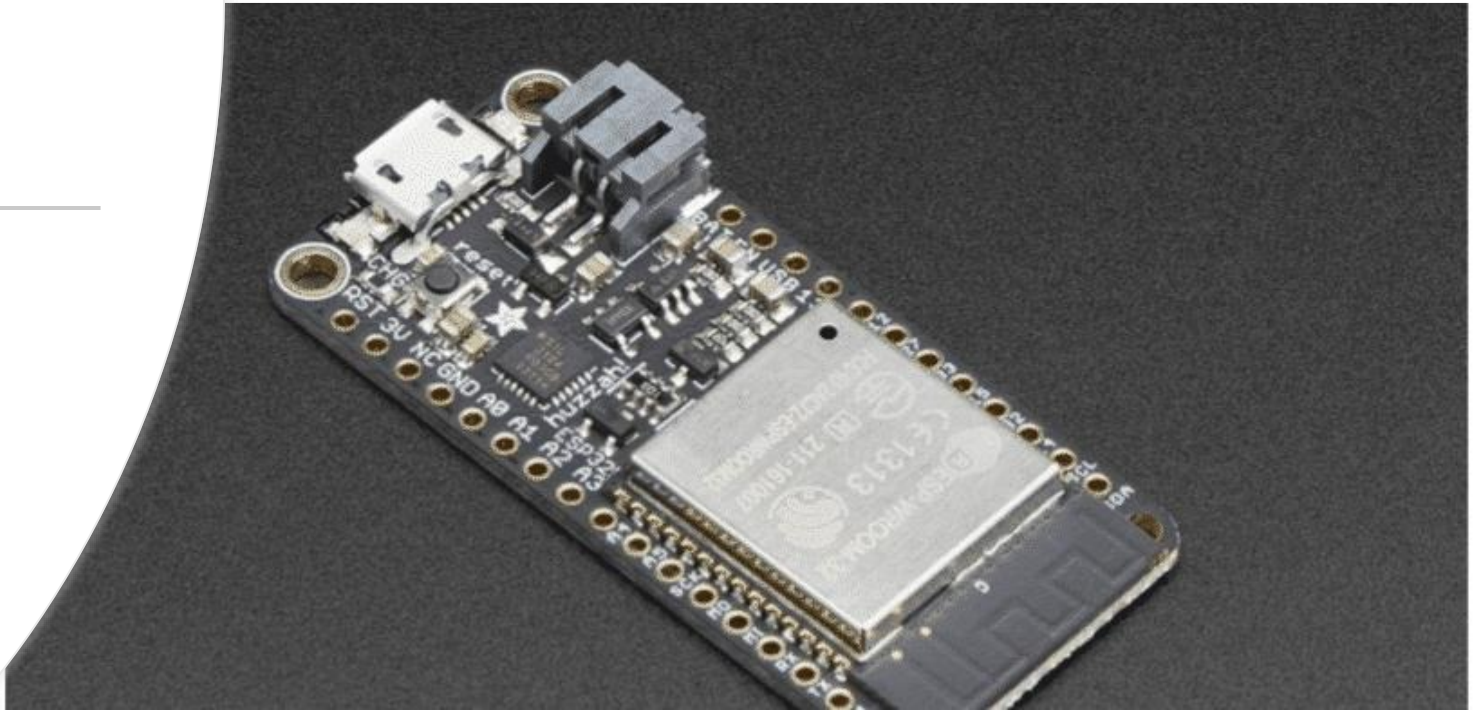
Types of Sensors

- Analog Sensors
  - Give a specified 0-X volt dc output
  - Need adc to use
- Digital Sensors
  - Either give a straight 1/0 output
  - Gives digital data over some protocol
  - Onewire / spi / i2c

Why no Analog?

- ESp32 analog input pins are non-linear
- Lot of confusion in documentation about the usability
- No fixed reference voltage
- If you want to use analog sensors, use an external ADC Chip



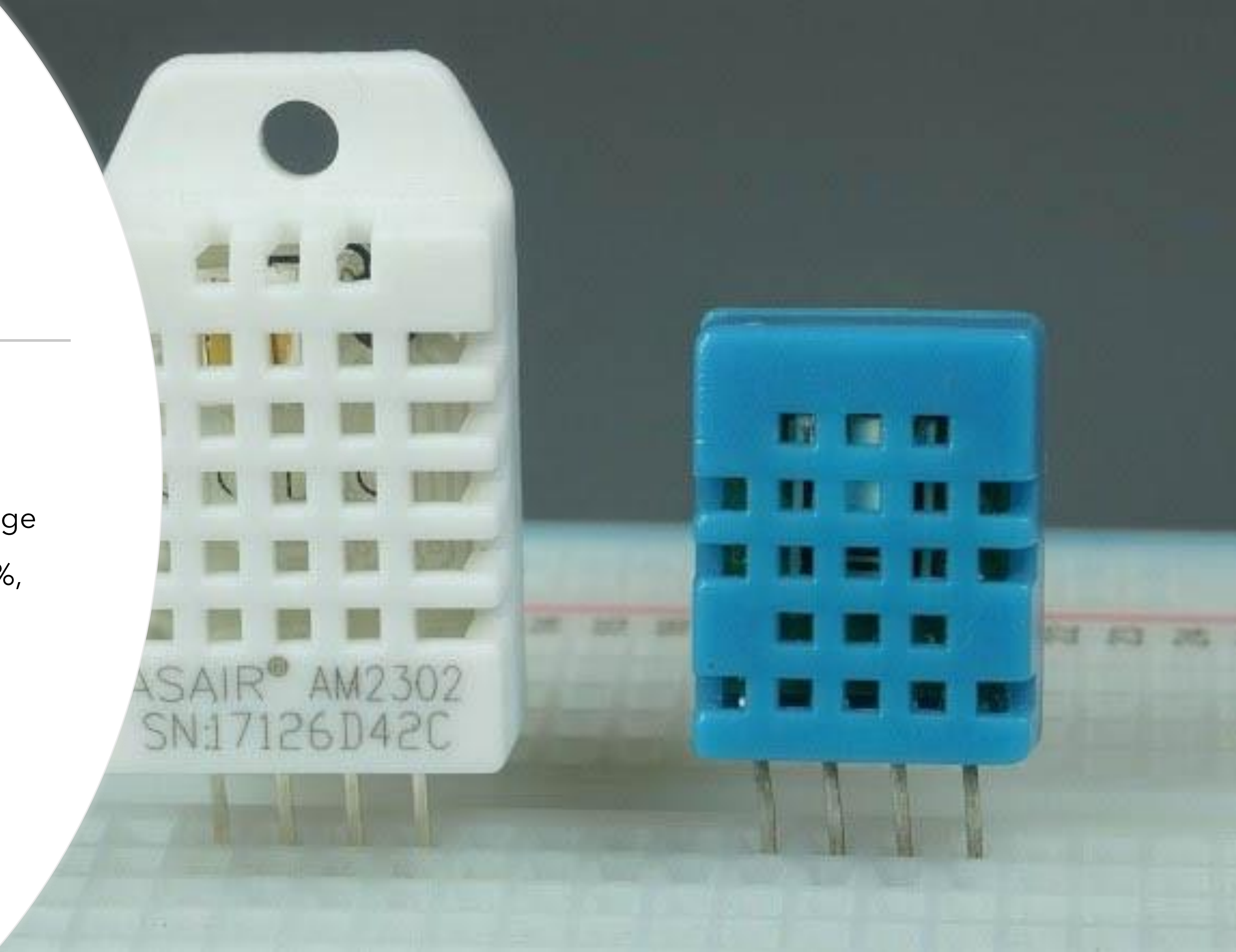


DHT22

Interfacing with ESP32

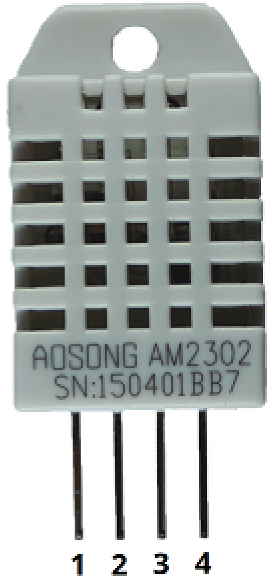
DHT22

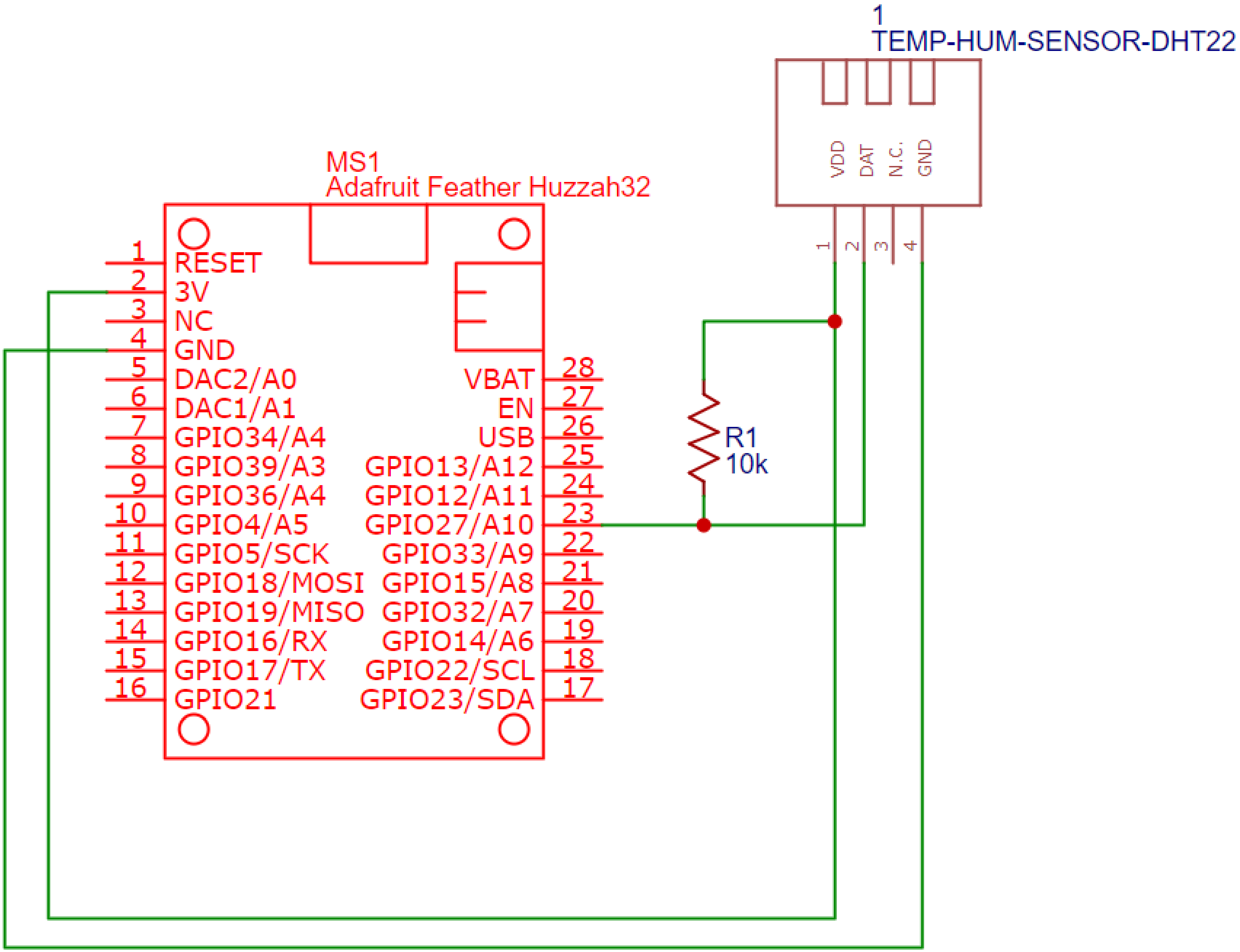
- -40 to 80 °C
- 0 to 100% Humidity
- 3 – 6 V DC Operating voltage
- Resolution : Humidity: 0.1%, Temperature: 0.1°C



DHT22 Pinout

- Pin 1 → 3.3v
- Pin 2 → GPIO with 10k Pullup
- Pin 3 → NC
- Pin 4 → GND




1
TEMP-HUM-SENSOR-DHT22
VDD
DAT
N.C.
GND
1
2
3
4
MS1
Adafruit Feather Huzzah32
1 RESET
2 3V
3 NC
4 GND
5 DAC2/A0
6 DAC1/A1
7 GPIO34/A4
8 GPIO39/A3
9 GPIO36/A4
10 GPIO4/A5
11 GPIO5/SCK
12 GPIO18/MOSI
13 GPIO19/MISO
14 GPIO16/RX
15 GPIO17/TX
16 GPIO21
28 VBAT
27 EN
26 USB
25 GPIO13/A12
24 GPIO12/A11
23 GPIO27/A10
22 GPIO33/A9
21 GPIO15/A8
20 GPIO32/A7
19 GPIO14/A6
18 GPIO22/SCL
17 GPIO23/SDA
R1
10k

Library needed

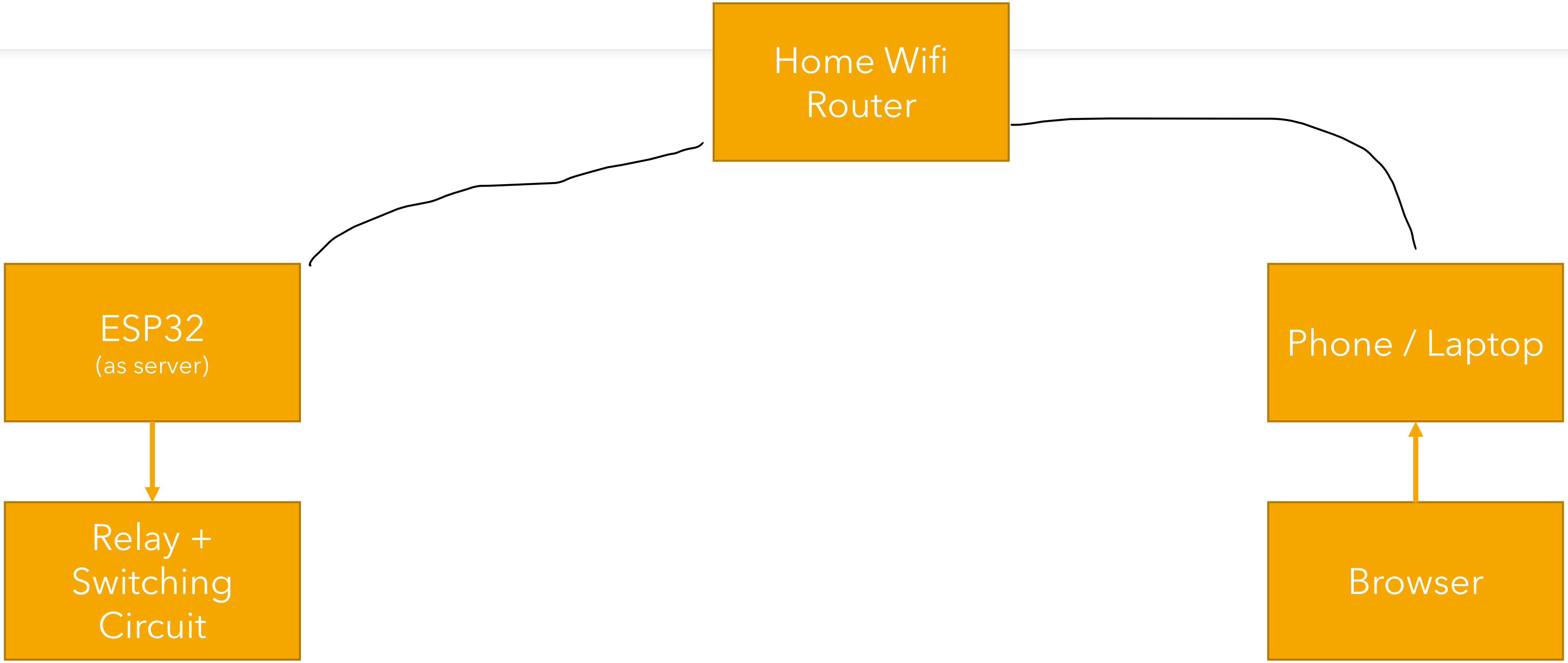
- Open your Arduino IDE and go to Sketch > Include Library > Manage Libraries. The Library Manager should open.
- DHT Sensor Library by Adafruit
- Adafruit Unified Sensor Library

Lets test ☺

Wifi Device Control

Using esp32

Concept



Home Wifi Router

ESP32
(as server)

Relay + Switching Circuit

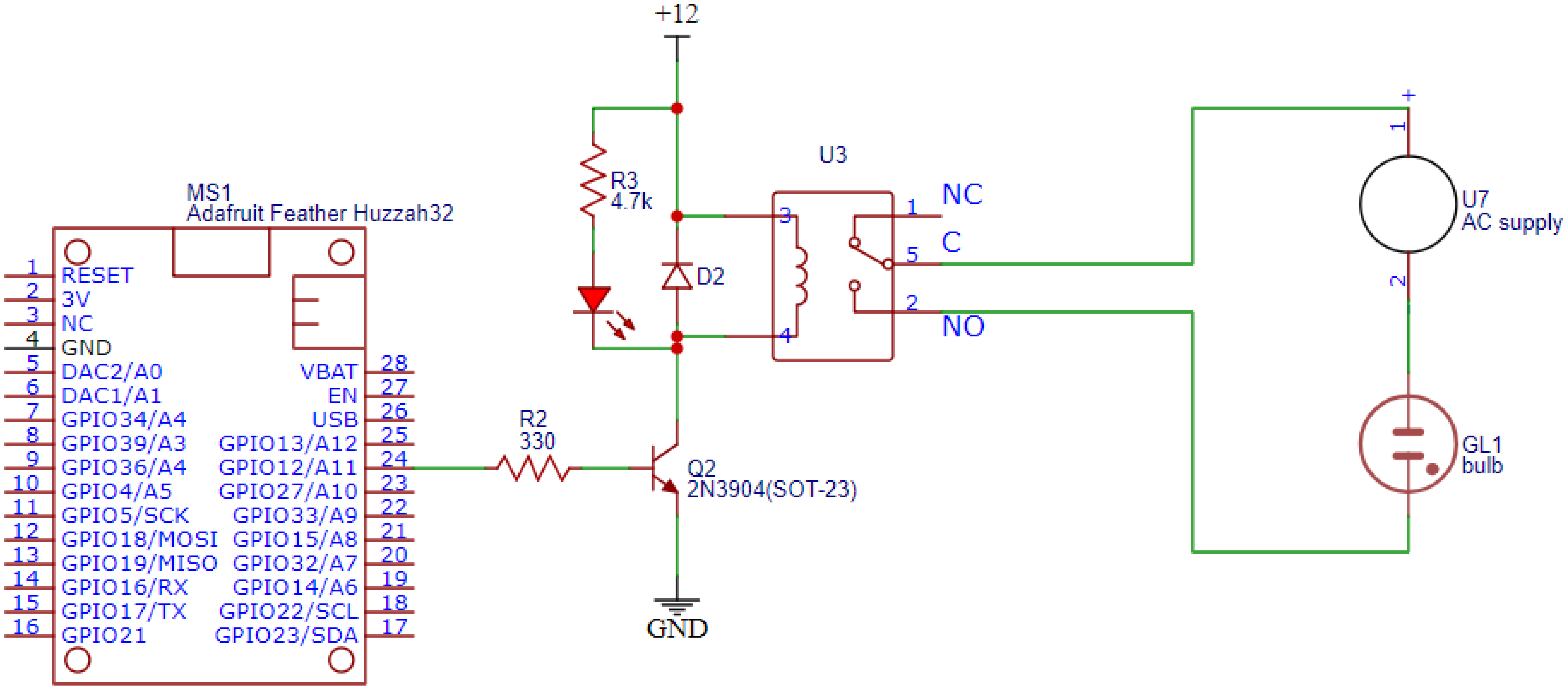
Phone / Laptop

Browser

Wifi Device Control

- Esp32 acts as a mini http server
- Serves a small web page
- You can take actions in this page
- Relay / LED Control

Schematic for this Experiment

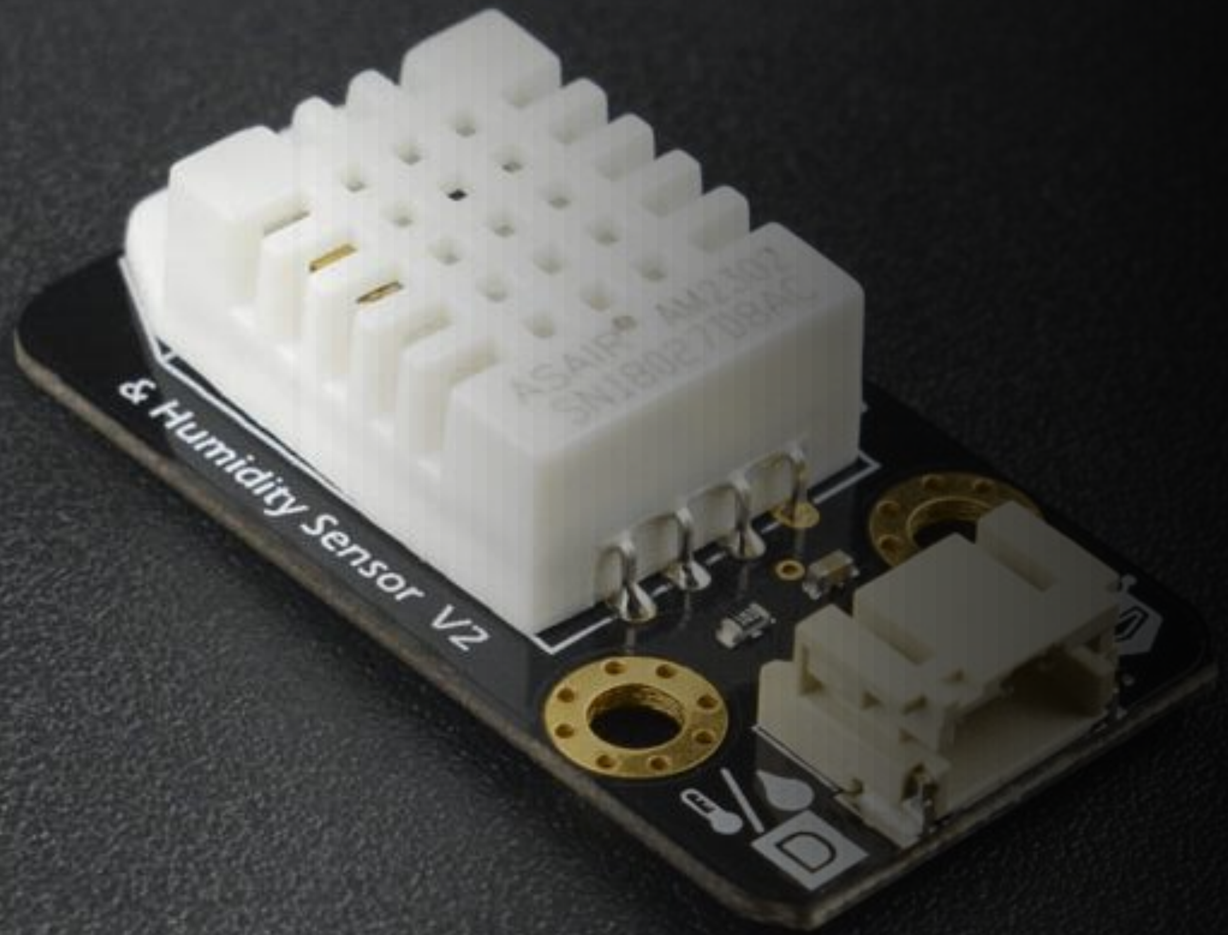


Required Libraries

- AsyncTCP – – Download here AsyncTCP.rar
- ESPAsyncWebserver -- Download here ESPAsyncWebServer.rar
- Extract the library and copy the folder into
- Program files → Arduino (x86) → Libraries

Test the Code

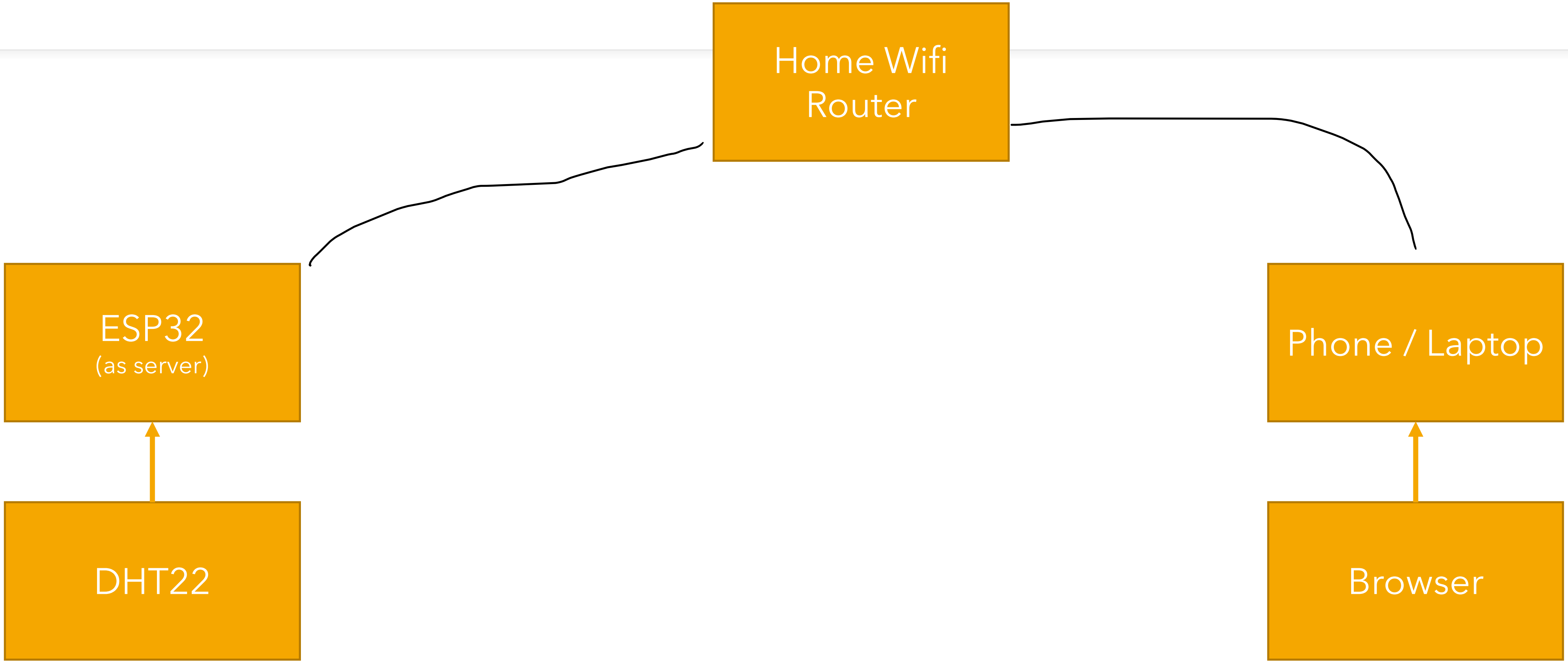
- Insert your Wifi Credentials
- Upload the Code
- Open Serial Terminal with baud rate of 9600
- Reset The ESP32
- Check IP Address assigned to esp32
- Open that IP Address on Phone / computer browser



Weather Monitoring

Using ESP32 and DHT22 in LAN

Concept



Home Wifi Router

ESP32 (as server)

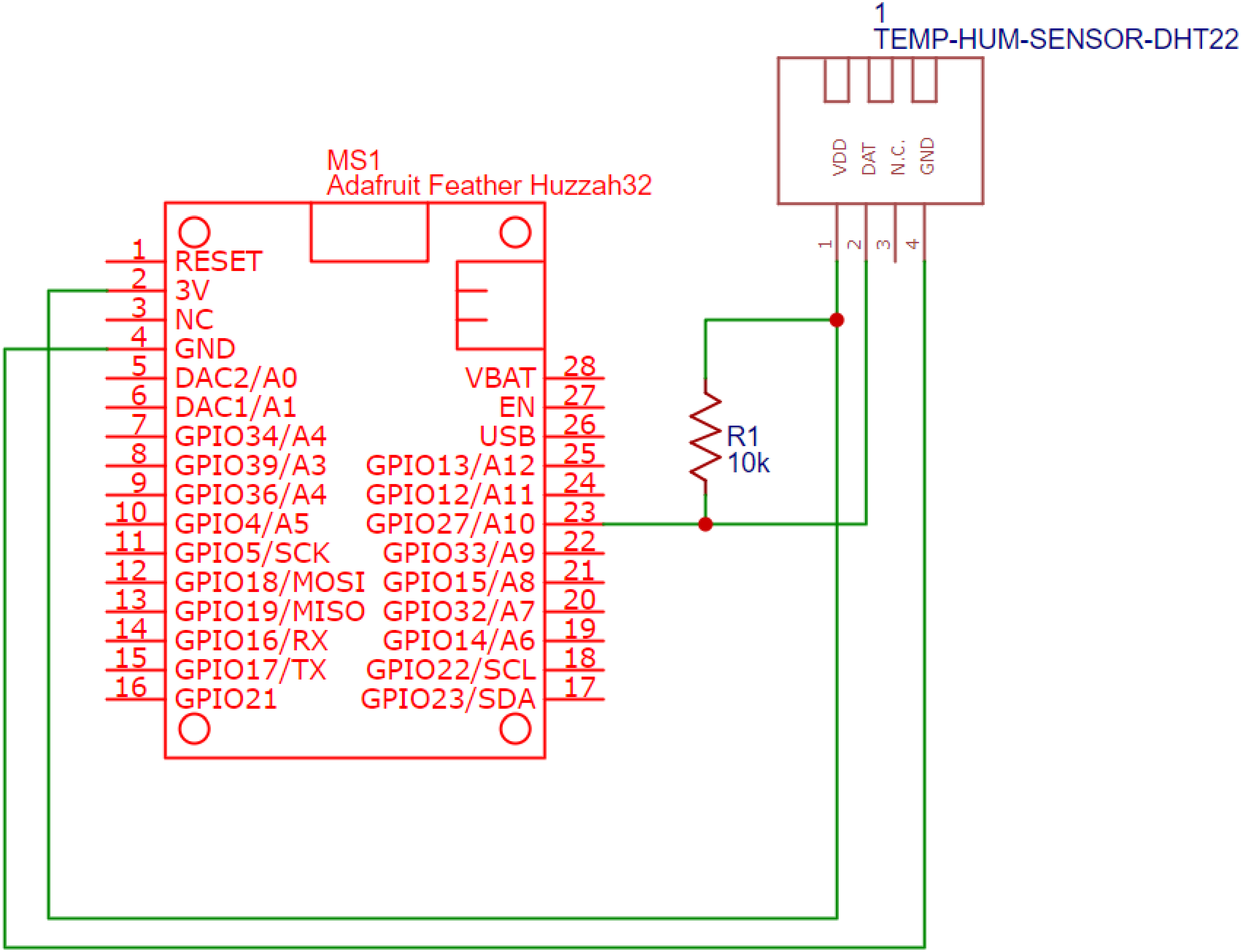
DHT22

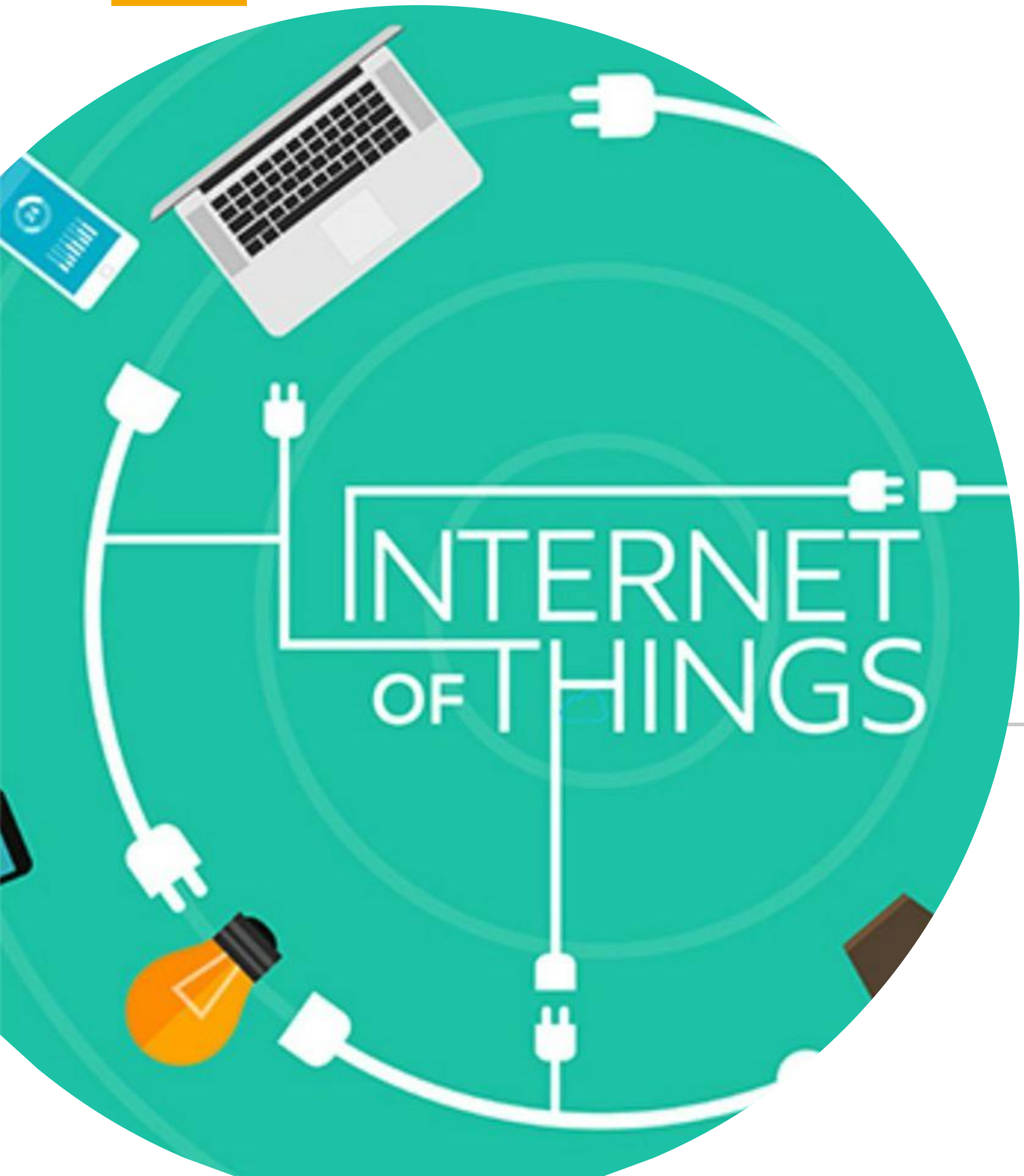
Phone / Laptop

Browser

Required Libraries

- AsyncTCP – – Download here AsyncTCP.rar
- ESPAsyncWebserver -- Download here ESPAsyncWebServer.rar
- DHT Sensor Library by Adafruit
- Adafruit Unified Sensor Library
- Program files → Arduino (x86) → Libraries


1
TEMP-HUM-SENSOR-DHT22
VDD
DAT
N.C.
GND
1
2
3
4
MS1
Adafruit Feather Huzzah32
1 RESET
2 3V
3 NC
4 GND
5 DAC2/A0
6 DAC1/A1
7 GPIO34/A4
8 GPIO39/A3
9 GPIO36/A4
10 GPIO4/A5
11 GPIO5/SCK
12 GPIO18/MOSI
13 GPIO19/MISO
14 GPIO16/RX
15 GPIO17/TX
16 GPIO21
28 VBAT
27 EN
26 USB
25 GPIO13/A12
24 GPIO12/A11
23 GPIO27/A10
22 GPIO33/A9
21 GPIO15/A8
20 GPIO32/A7
19 GPIO14/A6
18 GPIO22/SCL
17 GPIO23/SDA
R1
10k



IoT

Introduction

Internet of Things

Room

Industrial machine

Weather

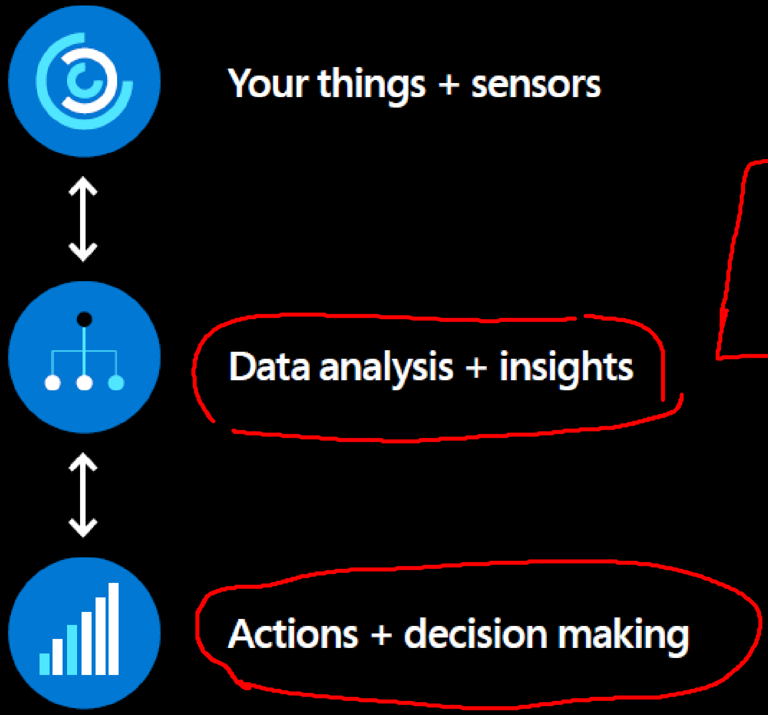
Building

Vehicle

What is IoT?

Reports

Internet

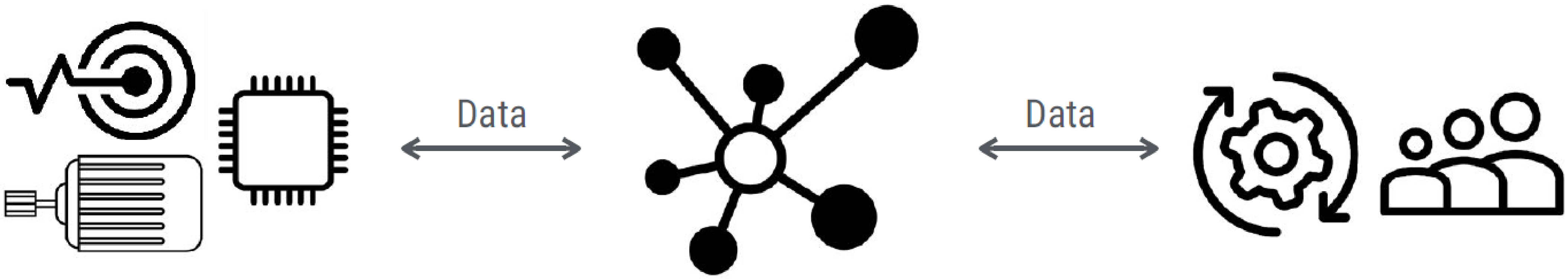


Your things + sensors

Data analysis + insights

Actions + decision making

Anywhere world



Data

Data

Things

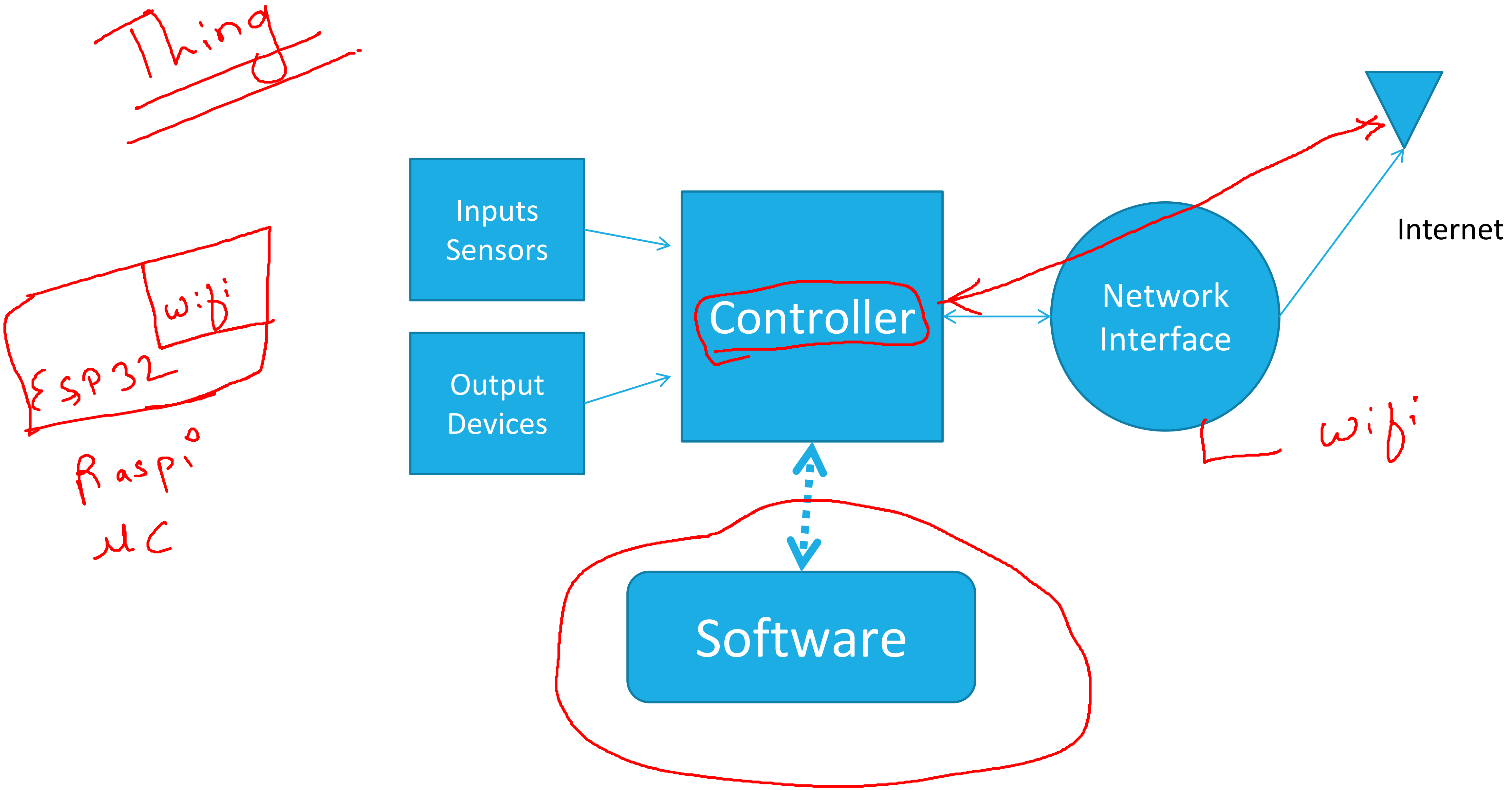
(Sensors, actuators, MCU/MPU, network, energy, firmware)

Connectivity

(PAN, LPWAN, Cellular)

People & Processes

(IoT Cloud, Machine Learning, AI)



Thing

Inputs Sensors

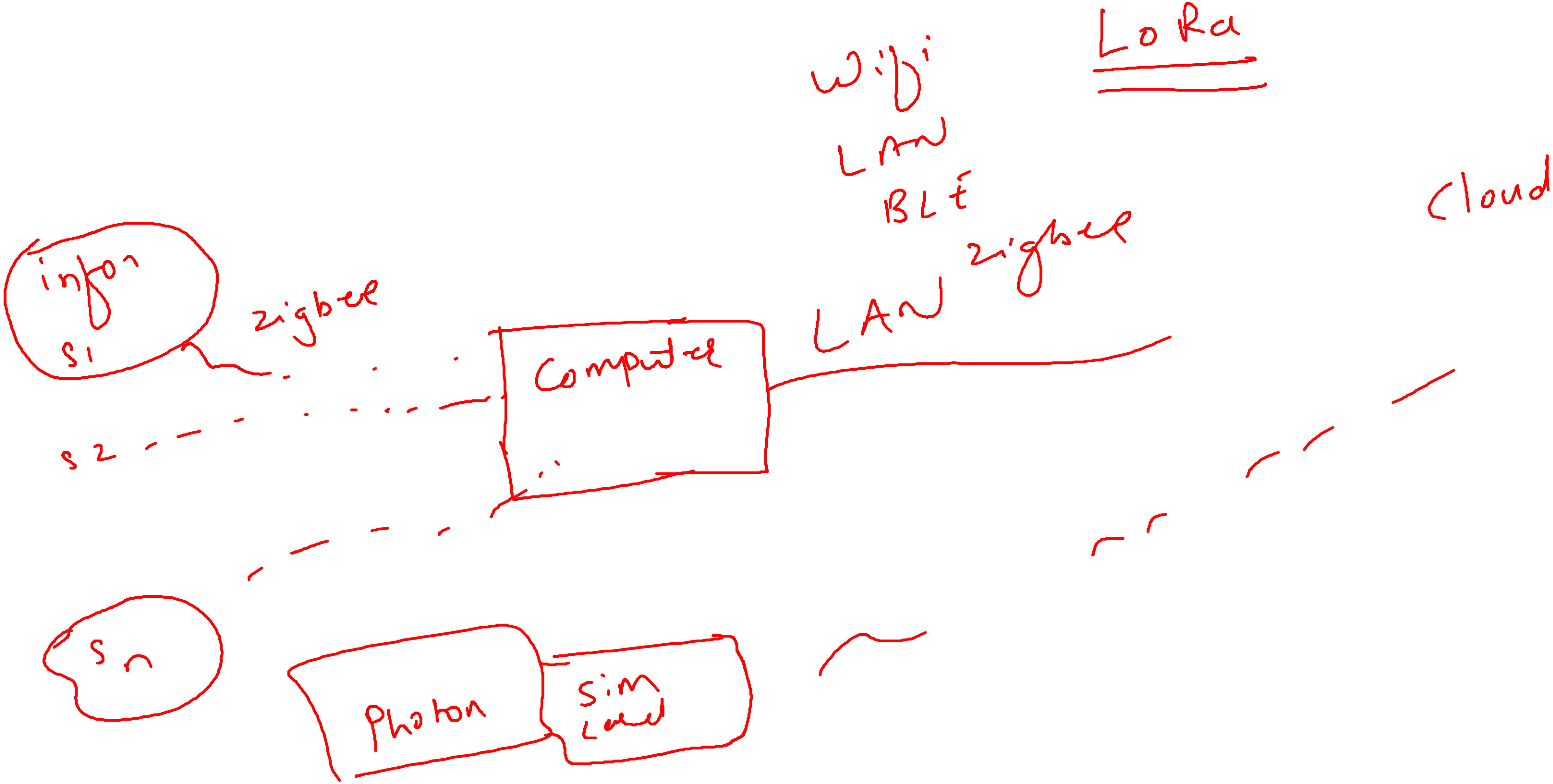
Output Devices

Controller

Network Interface

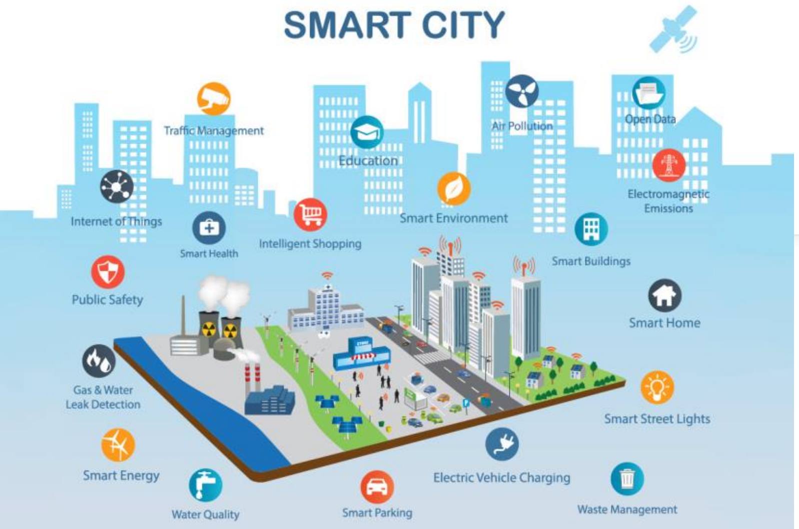
Internet

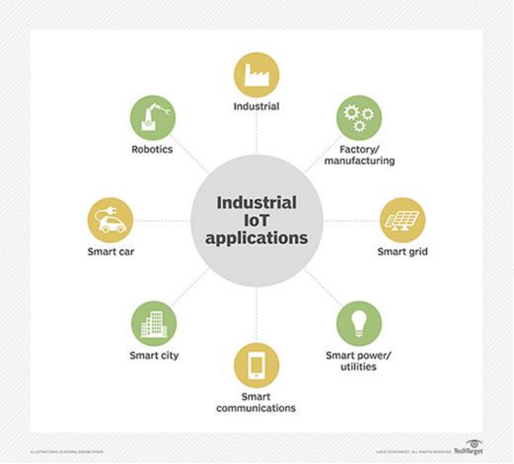
Software

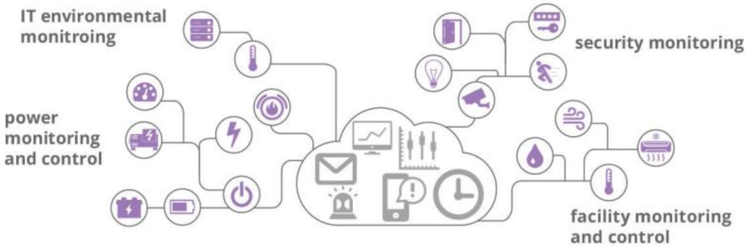

LoRa
Wifi
LAN
BLE
Cloud
info
S1
zigbee
Computer
LAN
zigbee
S2
Sn
Photon
sim card

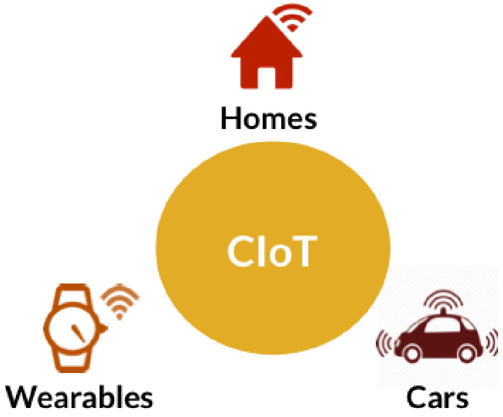
Applications

- Consumer
- Industrial
- Commercial
- infrastructure






IT environmental monitroing
security monitoring
power monitoring and control
facility monitoring and control



IoT Architecture Requirements

- Handle proper hardware and software heterogeneity.
- Reliability
- Scale
- Data latency.
- Be secure by design
- Lower barriers to entry: evaluate -> prototype -> deploy
- Manufacturable

Technologies Involved

Programming in

- C / C++
- Python
- Web development (all web tech)
- Machine Learning
- Cloud software
- Data analysis
- App development

Protocols

- Data sending from device to Cloud
- HTTP
- MQTT

Hardware

- Microcontroller
- Sensors
- Devices

AWS | GCP | Azure

PaaS

Product

SaaS ✓

Choice of Cloud Service

✓ 1. Custom Design with Major Cloud Providers

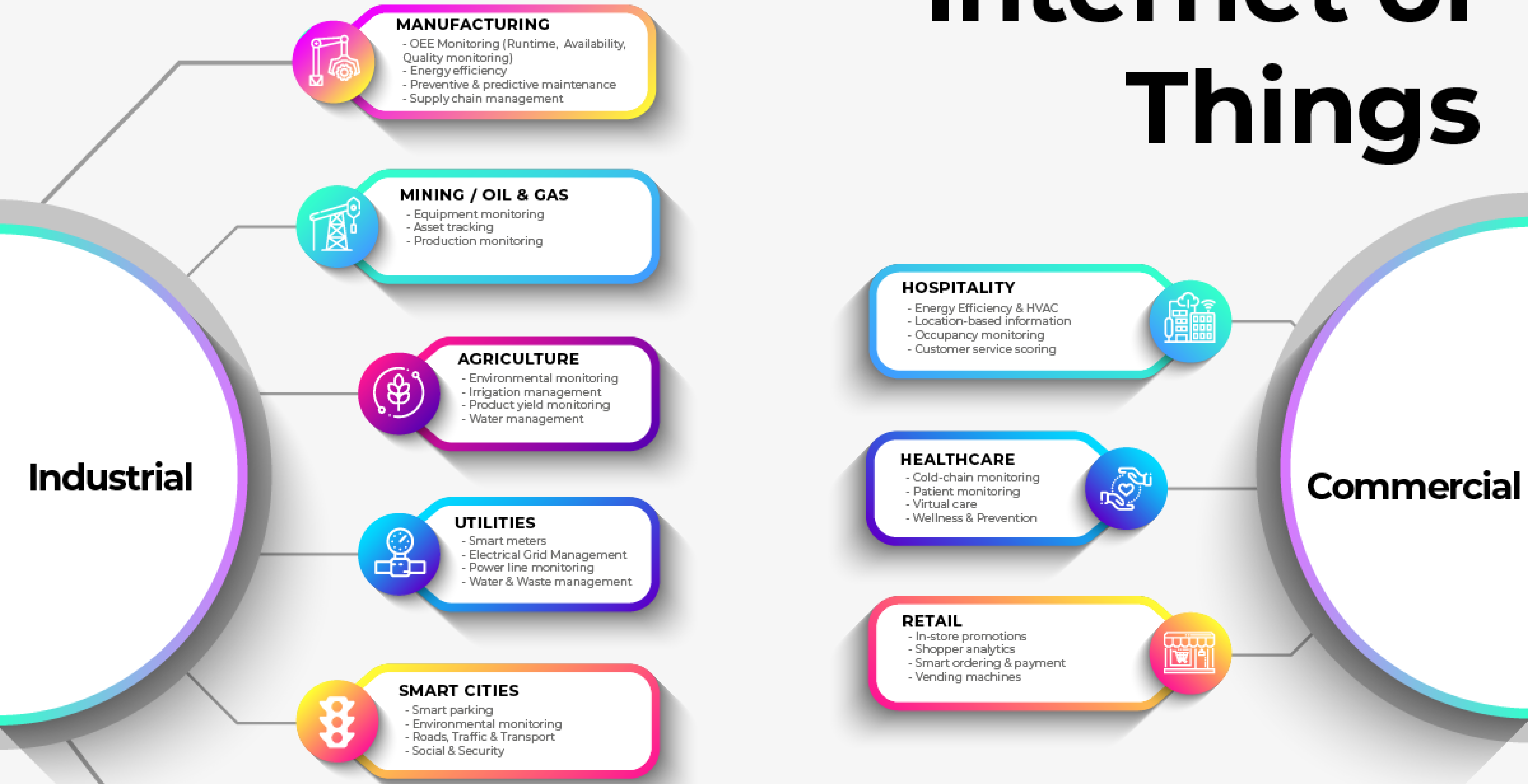
2. Using existing service Providers optimized for IoT

Ready Service Providers

- Thingworx
- Everythng
- Sensorcloud
- Device Cloud
- ThingSpeak
- Numerex

free

Internet of Things



Industrial

MANUFACTURING

- OEE Monitoring (Runtime, Availability, Quality monitoring)
- Energy efficiency
- Preventive & predictive maintenance
- Supply chain management

MINING / OIL & GAS

- Equipment monitoring
- Asset tracking
- Production monitoring

AGRICULTURE

- Environmental monitoring
- Irrigation management
- Product yield monitoring
- Water management

UTILITIES

- Smart meters
- Electrical Grid Management
- Power line monitoring
- Water & Waste management

SMART CITIES

- Smart parking
- Environmental monitoring
- Roads, Traffic & Transport
- Social & Security

Commercial

HOSPITALITY

- Energy Efficiency & HVAC
- Location-based information
- Occupancy monitoring
- Customer service scoring

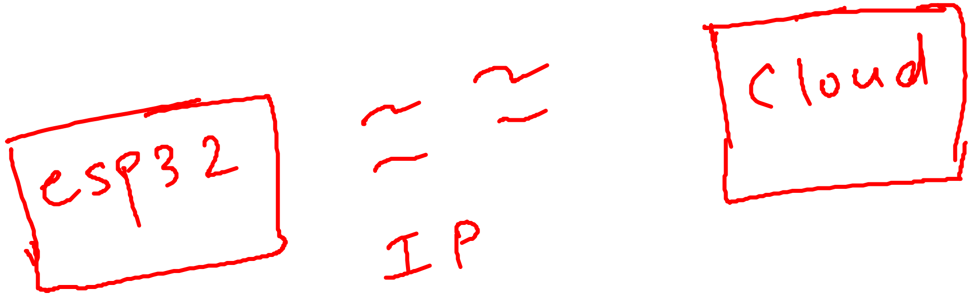
HEALTHCARE

- Cold-chain monitoring
- Patient monitoring
- Virtual care
- Wellness & Prevention

RETAIL

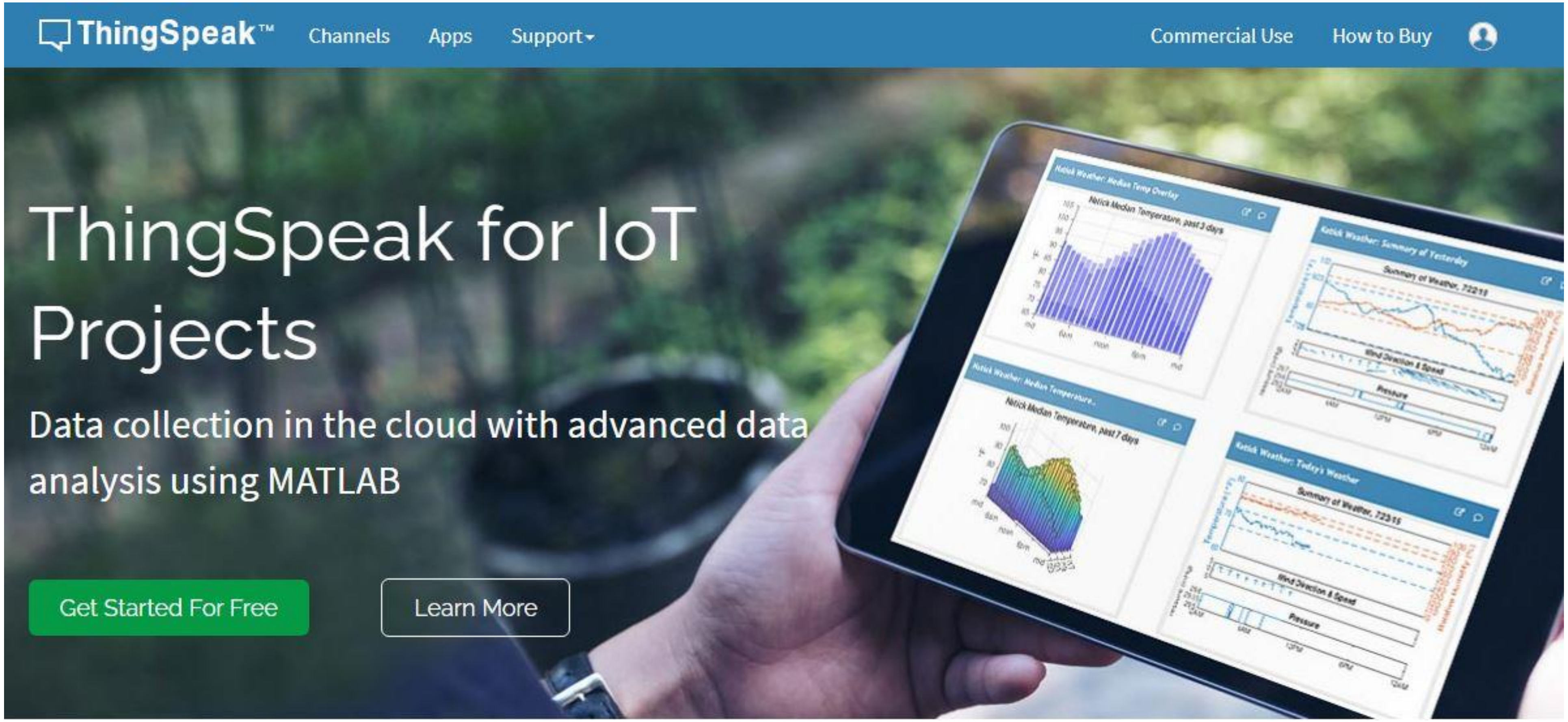
- In-store promotions
- Shopper analytics
- Smart ordering & payment
- Vending machines

Protocols used in IoT



- **AMQP** stands for Advanced Message Queueing Protocol and is an open standard application layer protocol.
- **MQTT** stands for Message Queuing Telemetry Transport machine-to-machine connectivity protocol designed as a lightweight publish/subscribe messaging transport, which makes it very suitable for use with IoT
- **HTTP** stands for Hypertext Transfer Protocol is an application protocol for distributed, collaborative, and hypermedia information systems

Project 1 : Temperature and Humidity Logging in Cloud



ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

Thingspeak

8 x sensors

- Free for up to 8 fields

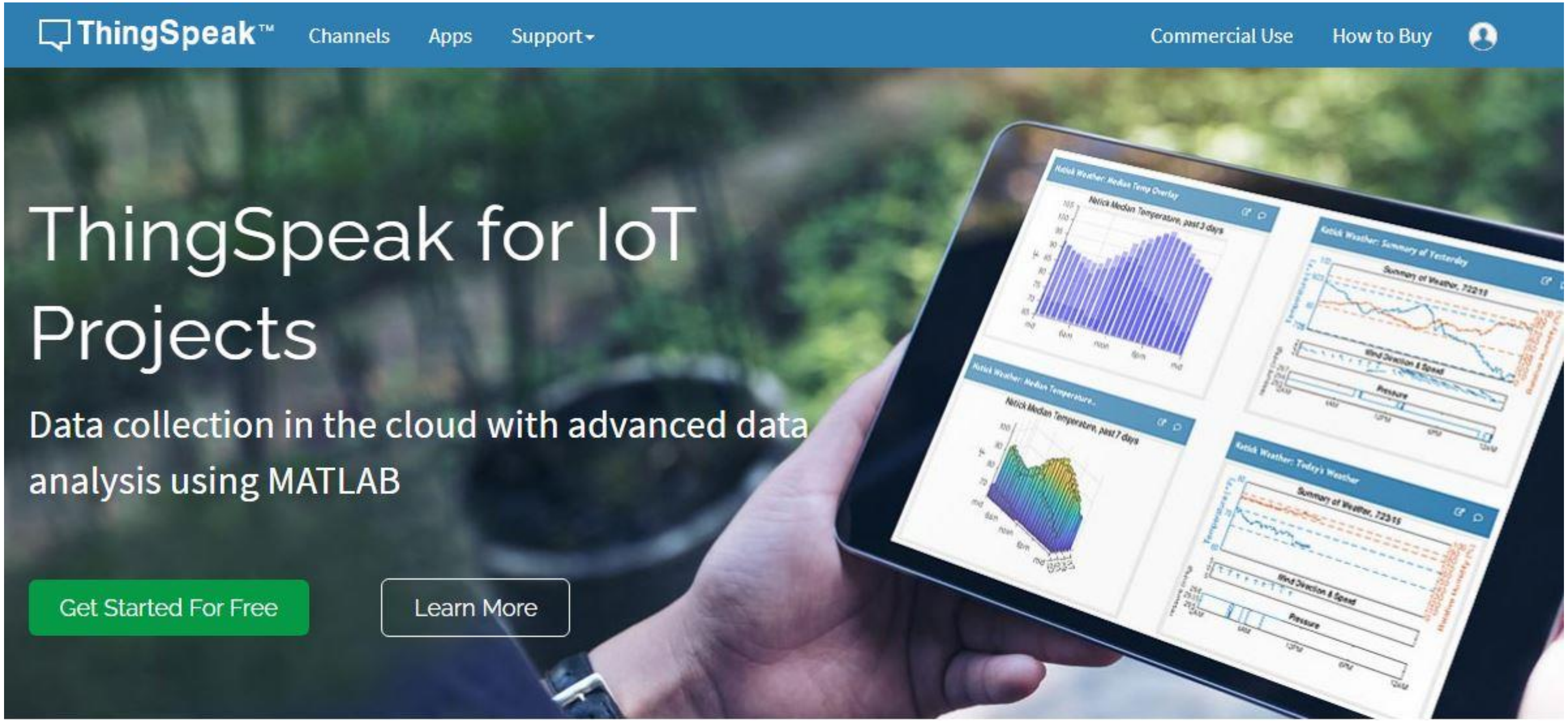
every 15 / sec

- 15 second refresh time
- Lots of online apps can be developed for analytics

Thingspeak and HTTP

- You can send requests to thingspeak server via http requests
- Easiest to create IoT Application
- Thingspeak account + write API Key + esp32
- Thingspeak adafruit library
- DHT22 connected to esp32 board

Project 1 : Temperature and Humidity Logging in Cloud



ThingSpeak for IoT Projects

Data collection in the cloud with advanced data analysis using MATLAB

Required Libraries

- DHT Sensor Library by Adafruit

Thingspeak analytics

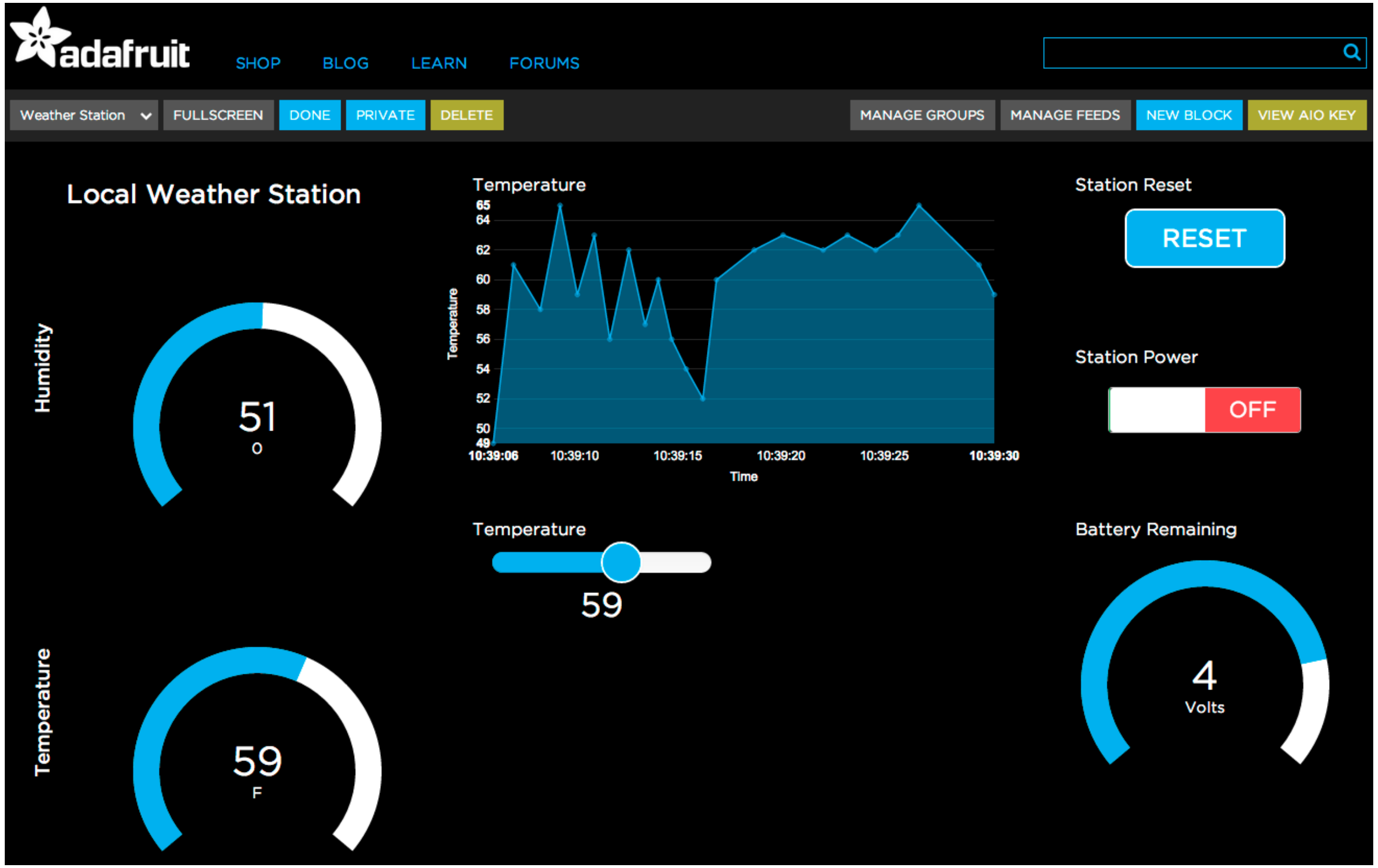
- Goto profile, copy alert API key
- Matlab Analysis Code
- Goto apps → Matlab analysis → New → Paste the code → Save
- Goto apps → React
- Used as A rection for something

Matlab Code (Alert api key needs replacement)

```
alert_body = 'This is the text that will be emailed';
alert_subject = 'This will be the subject of the email';
alert_api_key = 'TAK5Q7V3N5EEH07FXD658';
alert_url= "https://api.thingspeak.com/alerts/send";
jsonmessage = sprintf(['{"subject": "%s", "body": "%s"}'], alert_subject,alert_body);
options = weboptions("HeaderFields", {'Thingspeak-Alerts-API-Key', alert_api_key; 'Content-Type','application/json
result = webwrite(alert_url, jsonmessage, options);
```

What is MQTT

Adafruit IO


adafruit
SHOP
BLOG
LEARN
FORUMS
Weather Station
FULLSCREEN
DONE
PRIVATE
DELETE
MANAGE GROUPS
MANAGE FEEDS
NEW BLOCK
VIEW AIO KEY
Local Weather Station
Humidity
51
o
Temperature
59
F
Temperature
65
64
62
60
58
56
54
52
50
49
10:39:06
10:39:10
10:39:15
10:39:20
10:39:25
10:39:30
Time
Temperature
59
Station Reset
RESET
Station Power
OFF
Battery Remaining
4
Volts

MQTT Fundamentals

- Broker address
- Username and pwd
- Mqtt topic name

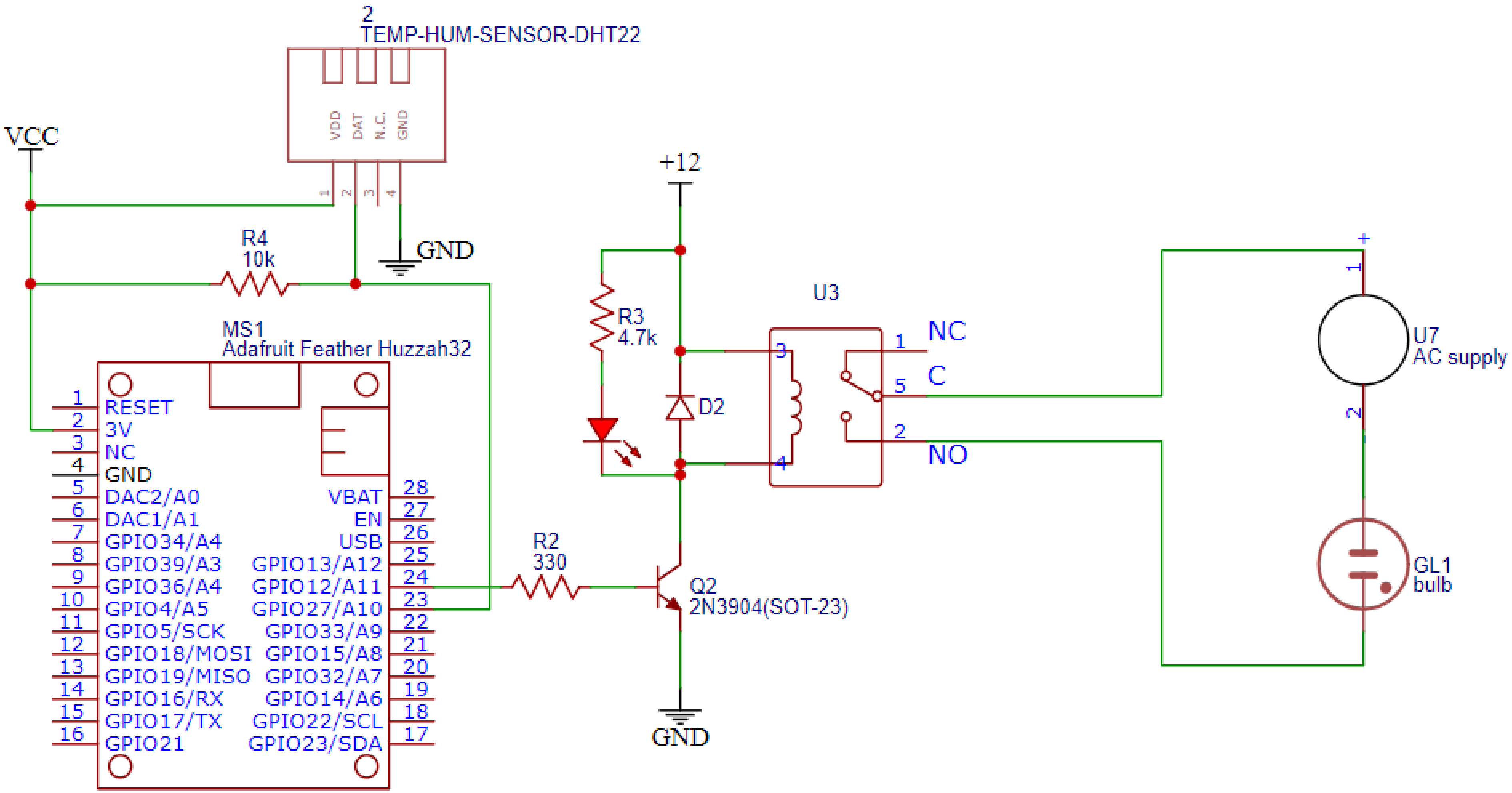
Adafruit IO

- Io.adafruit.com
- Account creation
- Creation of and understanding Feeds
- Creation of Dashboard
- Adafruit MQTT Library
- Free Account supports 5 different fields (or topics)
- Trial Code

Adafruit IO MQTT Example (feeds)

- Temperature
- Humidity
- Relay

Circuit Wiring : DHT22 and Relay



Alexa Controlled Home Lamp

Things you’ll need

- Alexa echo or alexa app
- IFTTT account
- Adafruit IO Account and the Same last Arduino Code

Flow

- Alexa sends command to IFTTT Server
- IFTTT Will send command to adafruit broker
- Adafruit Broker will update the value of relay feed
- Esp32 is subscribed to relay feed to it receives it
- ESP32 turns on/off relay