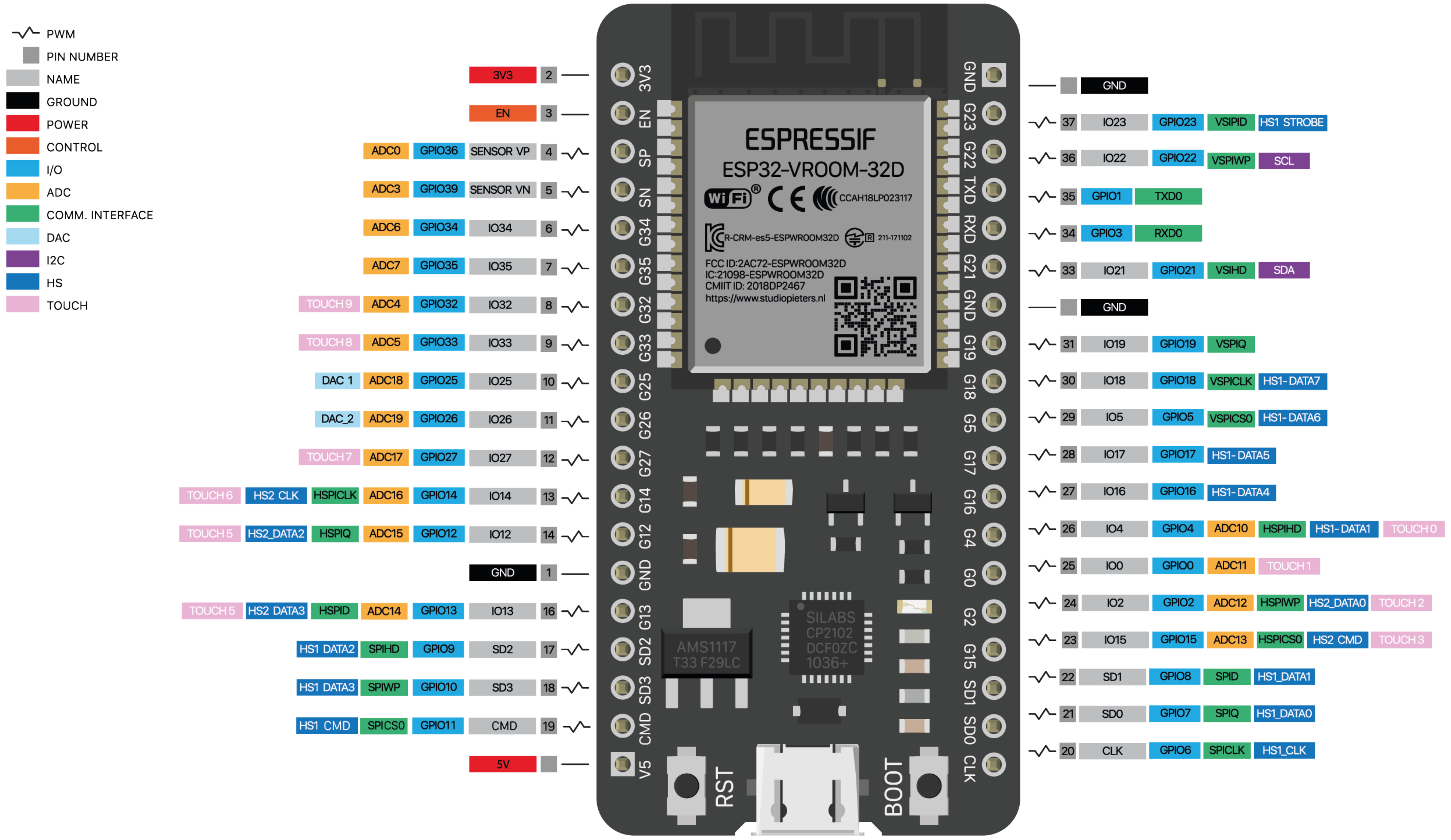
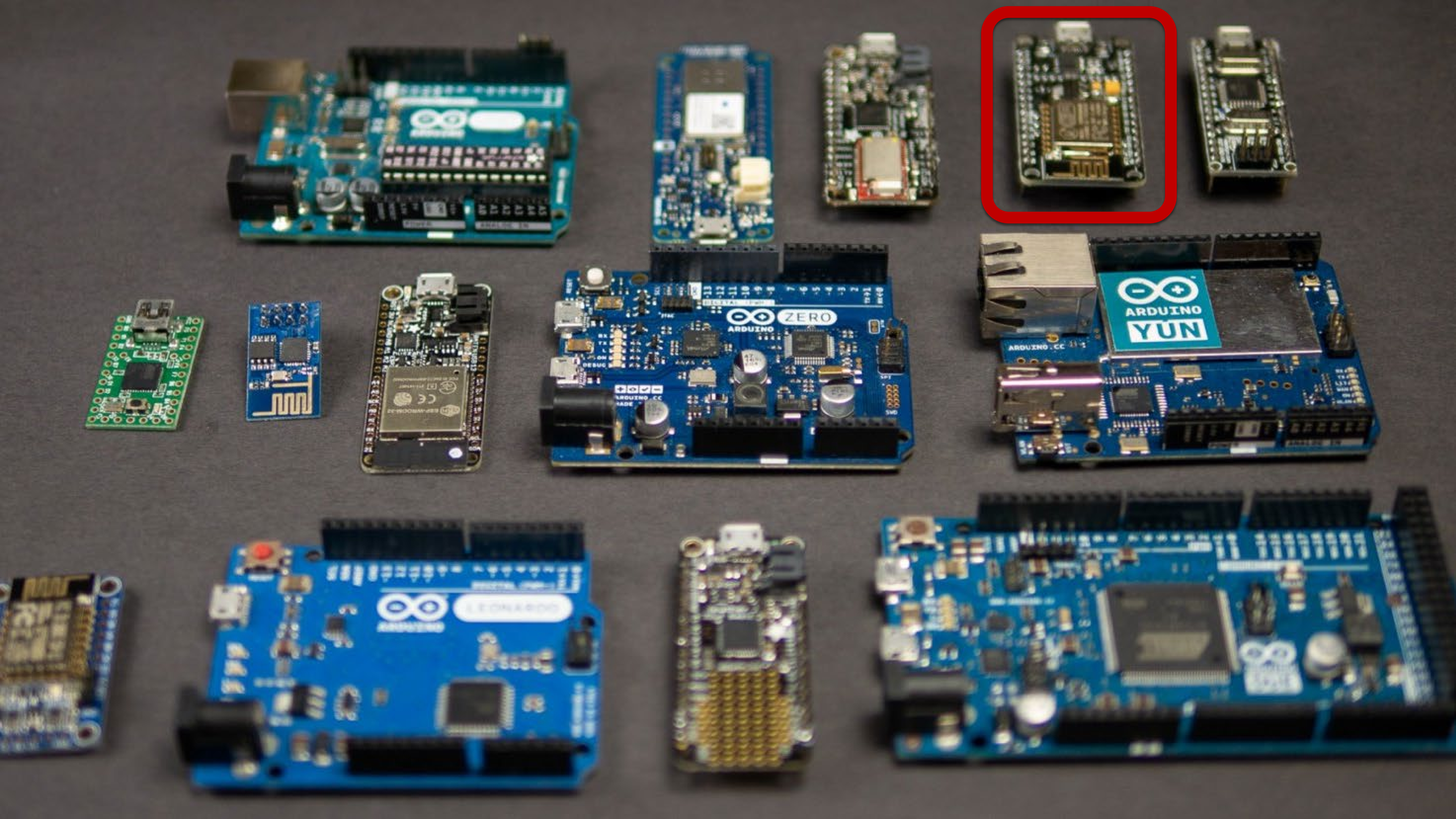
DigitalOutput

Huaishu Peng | UMD CS | Fall 2022

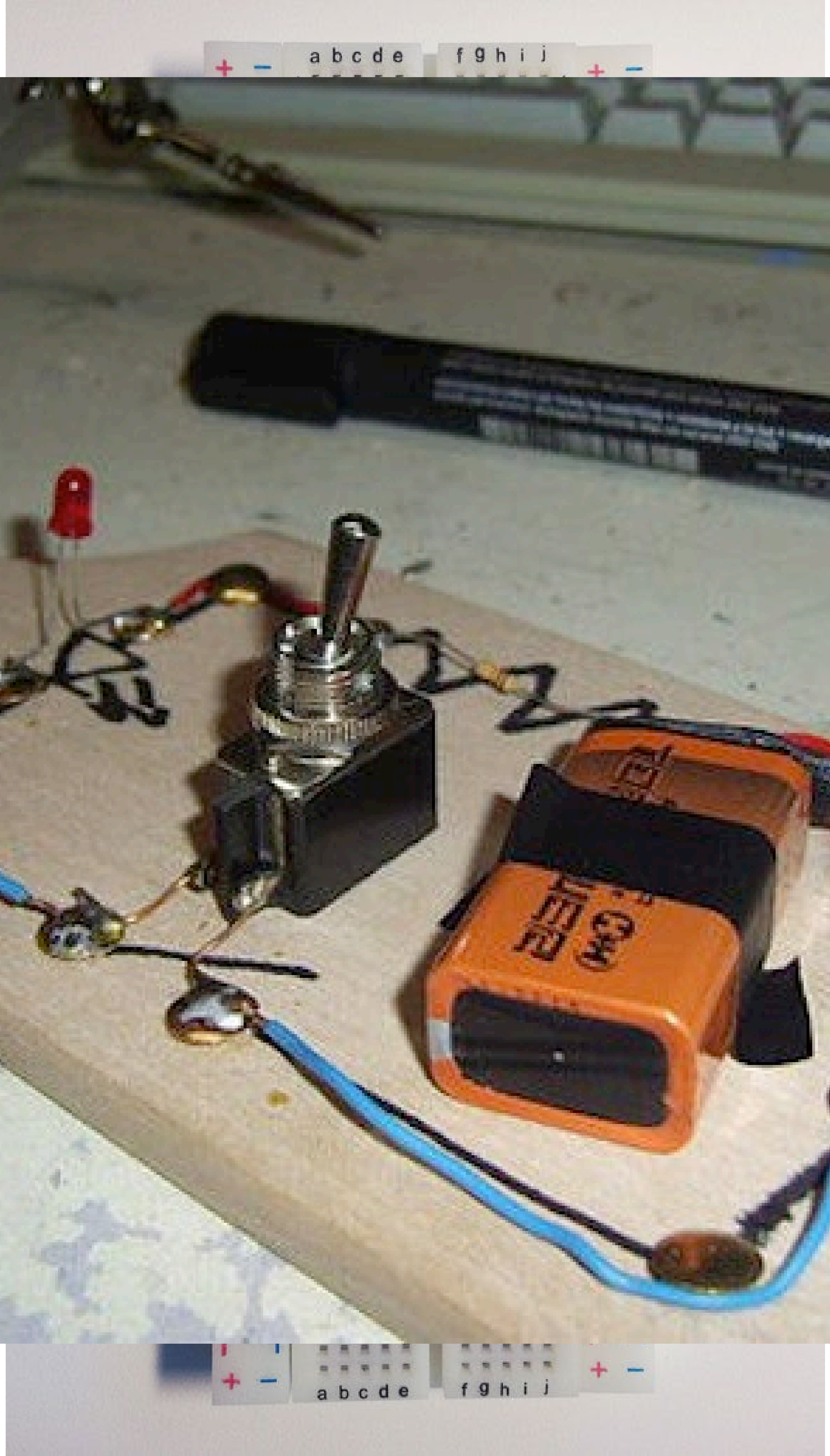

PWM
PIN NUMBER
NAME
GROUND
POWER
CONTROL
I/O
ADC
COMM. INTERFACE
DAC
I2C
HS
TOUCH
3V3 2
EN 3
ADC0 GPIO36 SENSOR VP 4
ADC3 GPIO39 SENSOR VN 5
ADC6 GPIO34 IO34 6
ADC7 GPIO35 IO35 7
TOUCH9 ADC4 GPIO32 IO32 8
TOUCH8 ADC5 GPIO33 IO33 9
DAC 1 ADC18 GPIO25 IO25 10
DAC_2 ADC19 GPIO26 IO26 11
TOUCH7 ADC17 GPIO27 IO27 12
TOUCH6 HS2 CLK HSPICLK ADC16 GPIO14 IO14 13
TOUCH5 HS2_DATA2 HSPIQ ADC15 GPIO12 IO12 14
GND 1
TOUCH5 HS2 DATA3 HSPID ADC14 GPIO13 IO13 16
HS1 DATA2 SPIHD GPIO9 SD2 17
HS1 DATA3 SPIWP GPIO10 SD3 18
HS1 CMD SPICS0 GPIO11 CMD 19
5V
3V3 EN SP SN G34 G35 G32 G33 G25 G26 G27 G14 G12 GND G13 SD2 SD3 CMD V5
GND G23 G22 TXD RXD G21 GND G19 G18 G5 G17 G16 G4 G0 G2 G15 SD1 SD0 CLK
ESPRESSIF
ESP32-VROOM-32D
CCAH18LP023117
R-CRM-es5-ESPWROOM32D
211-171102
FCC ID:2AC72-ESPWROOM32D
IC:21098-ESPWROOM32D
CMIIT ID: 2018DP2467
https://www.studiopieters.nl
AMS1117 T33 F29LC
SILABS CP2102 DCF0ZC 1036+
RST
BOOT
GND
37 IO23 GPIO23 VSPID HS1 STROBE
36 IO22 GPIO22 VSPIWP SCL
35 GPIO1 TXD0
34 GPIO3 RXD0
33 IO21 GPIO21 VSIHD SDA
GND
31 IO19 GPIO19 VSPIQ
30 IO18 GPIO18 VSPICLK HS1-DATA7
29 IO5 GPIO5 VSPICS0 HS1-DATA6
28 IO17 GPIO17 HS1-DATA5
27 IO16 GPIO16 HS1-DATA4
26 IO4 GPIO4 ADC10 HSPIHD HS1-DATA1 TOUCH0
25 IO0 GPIO0 ADC11 TOUCH1
24 IO2 GPIO2 ADC12 HSPIWP HS2_DATA0 TOUCH2
23 IO15 GPIO15 ADC13 HSPICS0 HS2 CMD TOUCH3
22 SD1 GPIO8 SPID HS1_DATA1
21 SD0 GPIO7 SPIQ HS1_DATA0
20 CLK GPIO6 SPICLK HS1_CLK

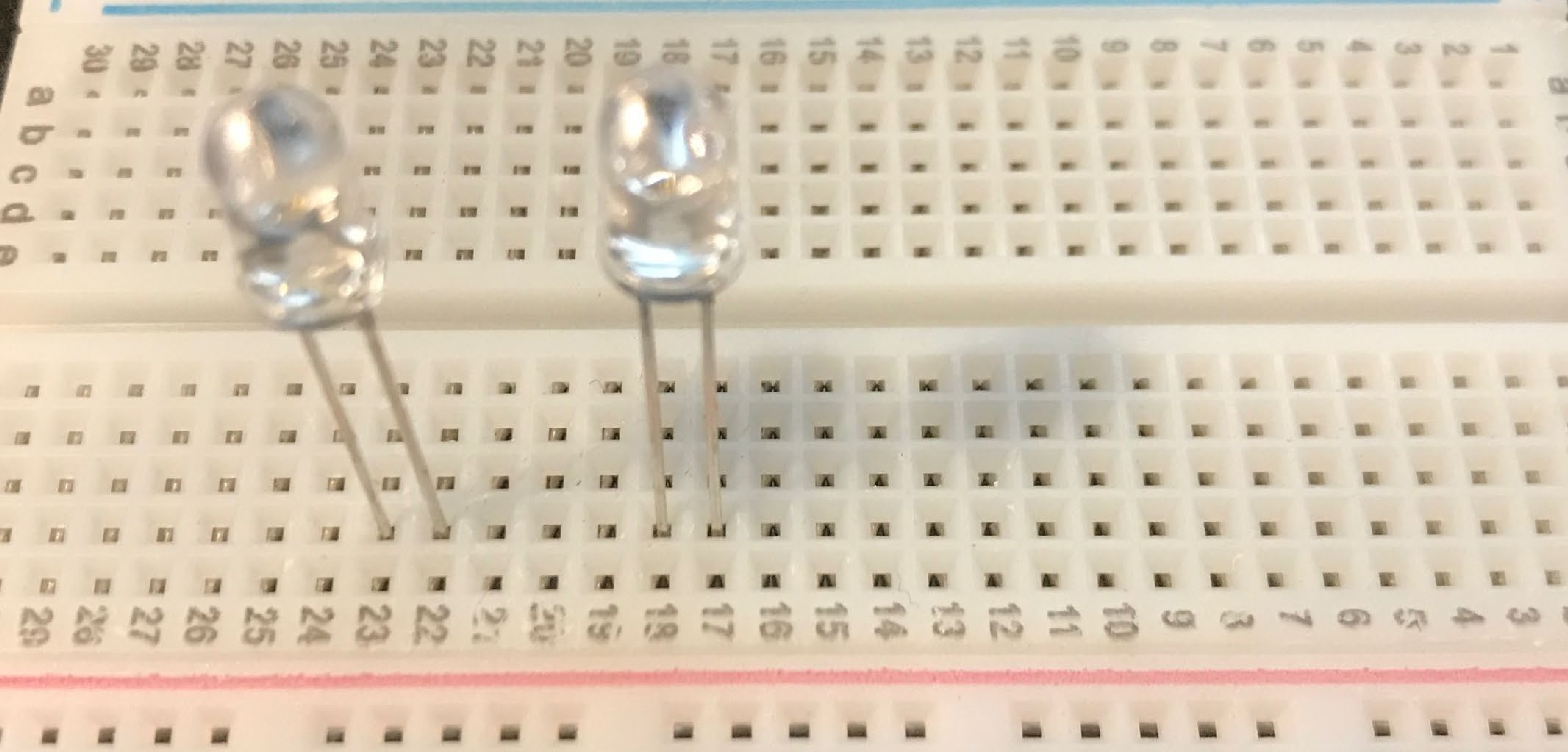
https://www.studiopieters.nl/esp32-pinout/


ARDUINO
ZERO
ARDUINO
YUN
ARDUINO.CC
LEONARDO
ESP-WROOM-32

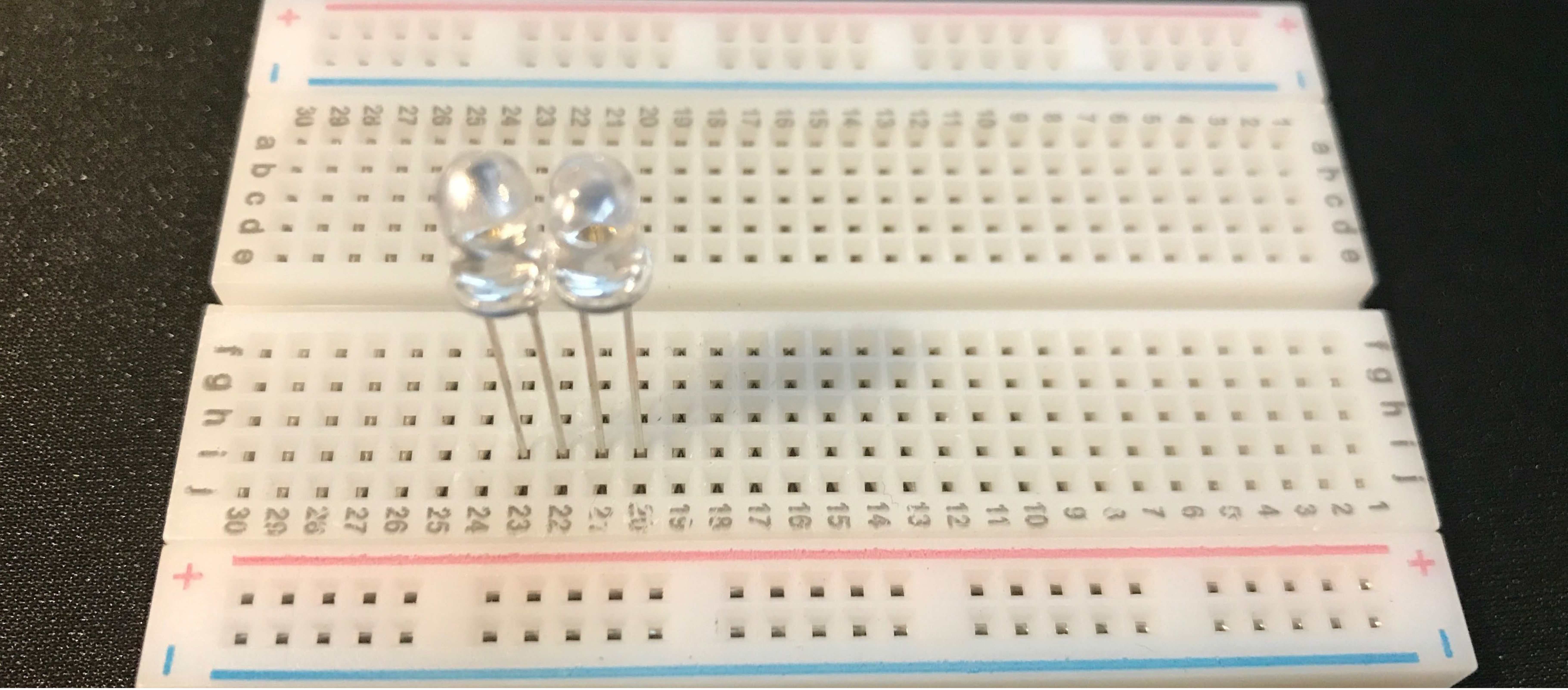
ESP32 (38Pin version)

18 Analog-to-Digital Converter (ADC) channels
3 SPI interfaces
3 UART interfaces
2 I2C interfaces
16 PWM output channels
2 Digital-to-Analog Converters (DAC)
2 I2S interfaces
10 Capacitive sensing GPIO's

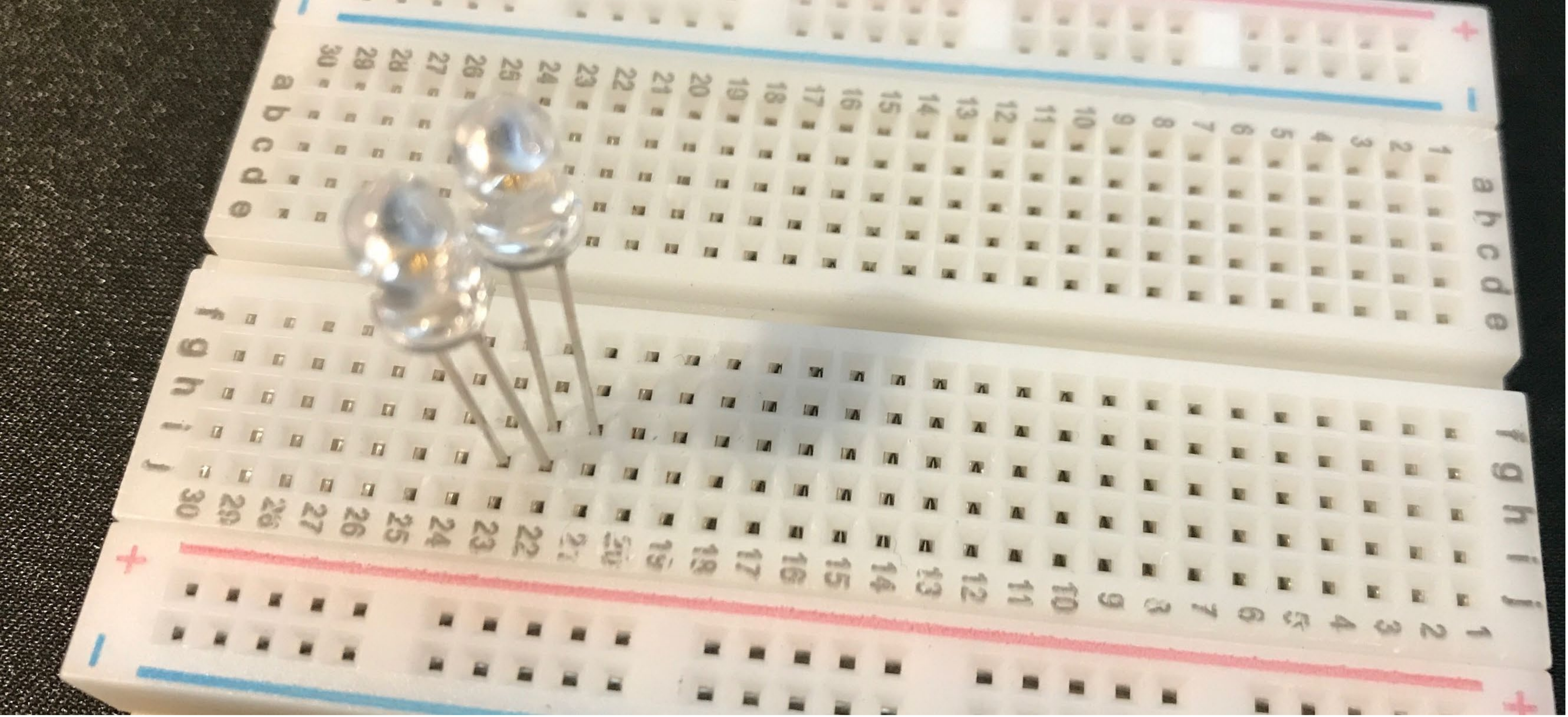

+ −
a b c d e
f g h i j
+ −



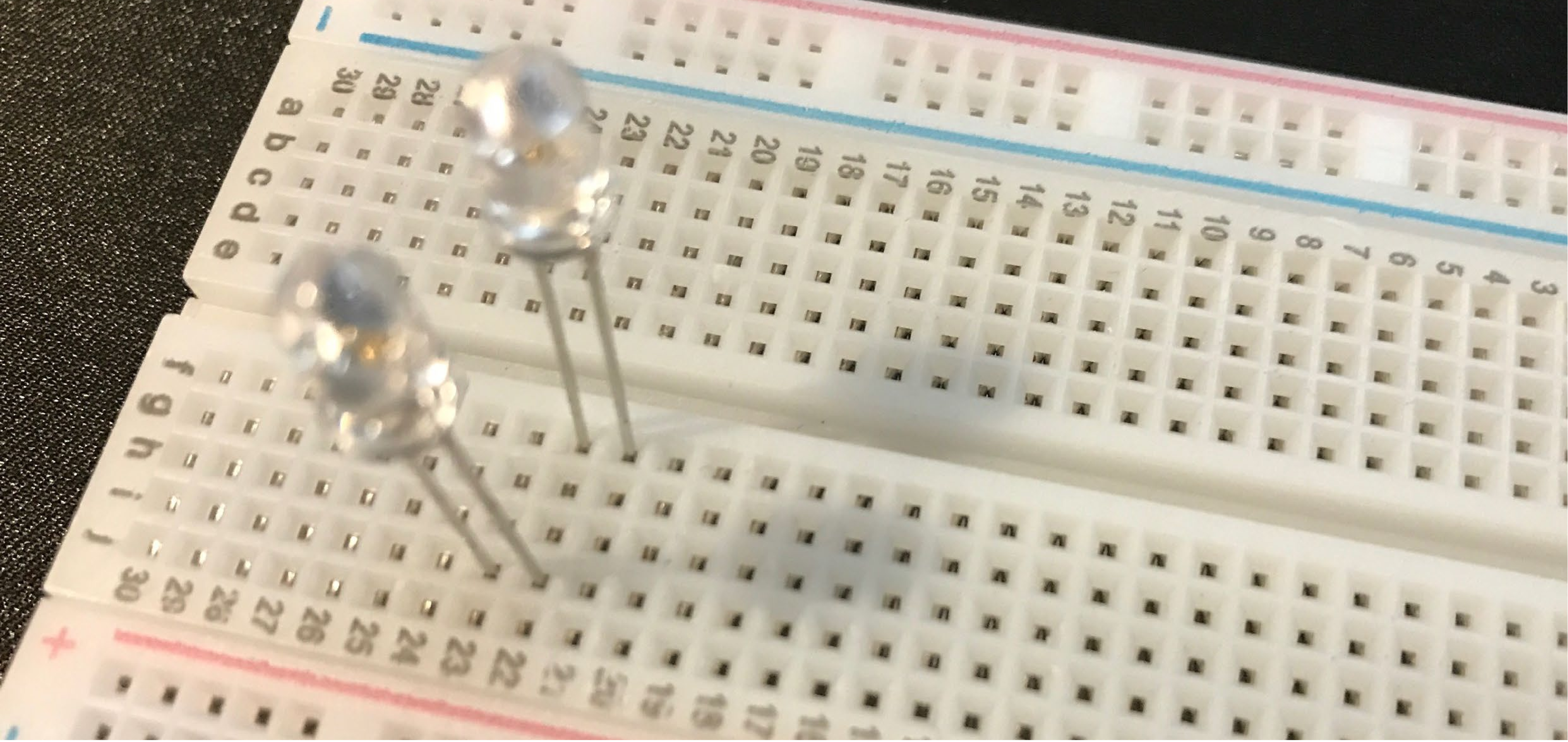
are the LEDs connected with each other?



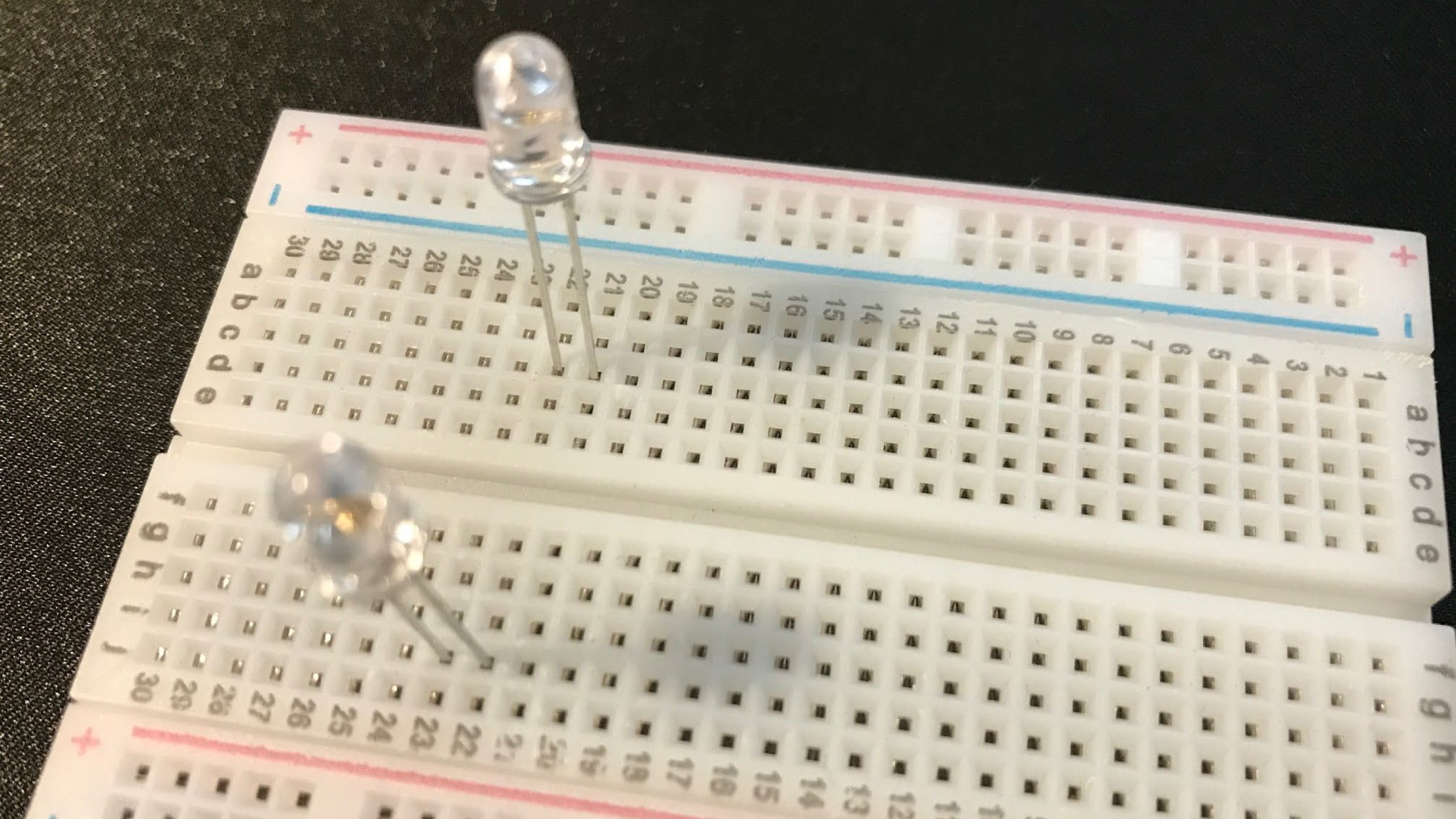
Now?

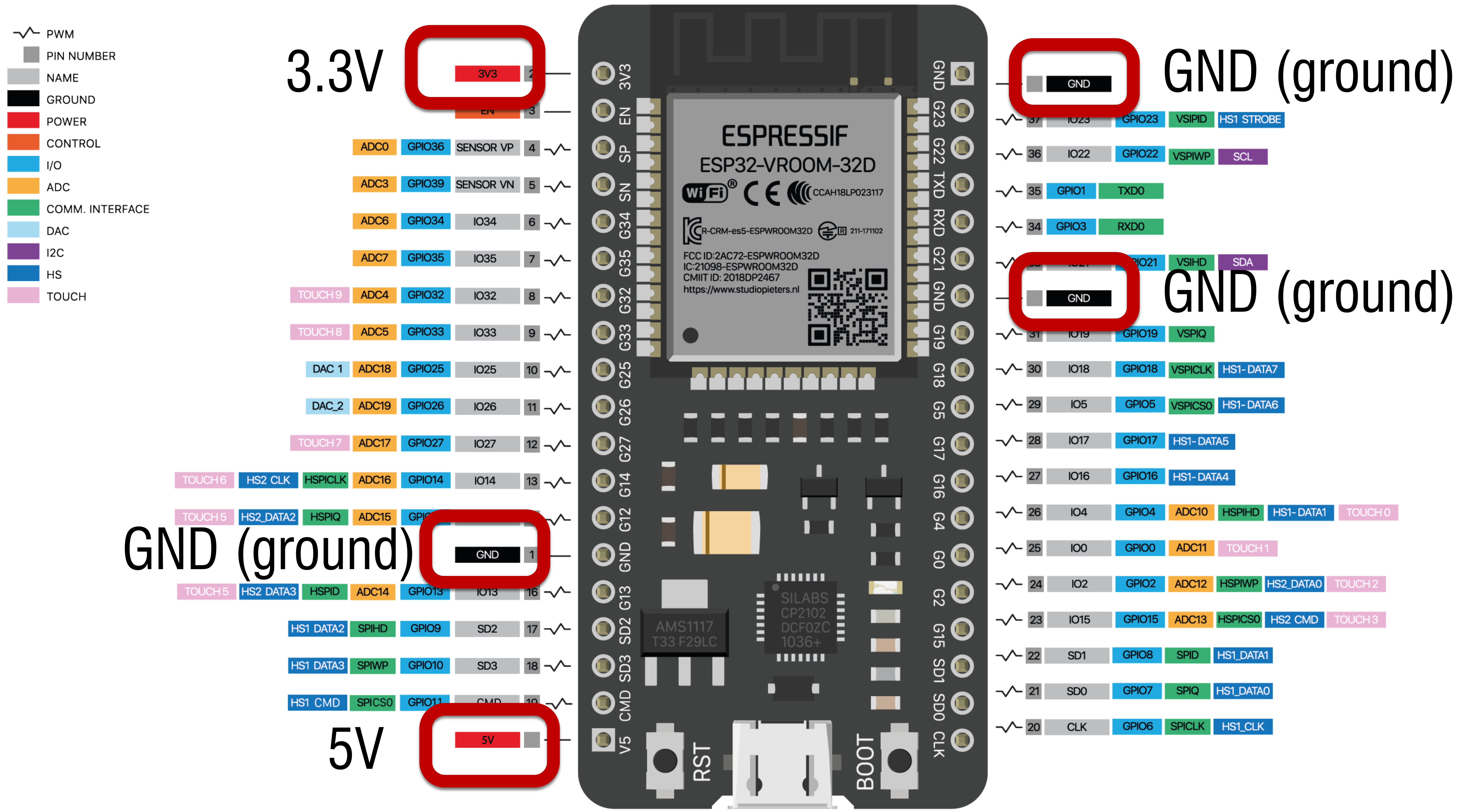


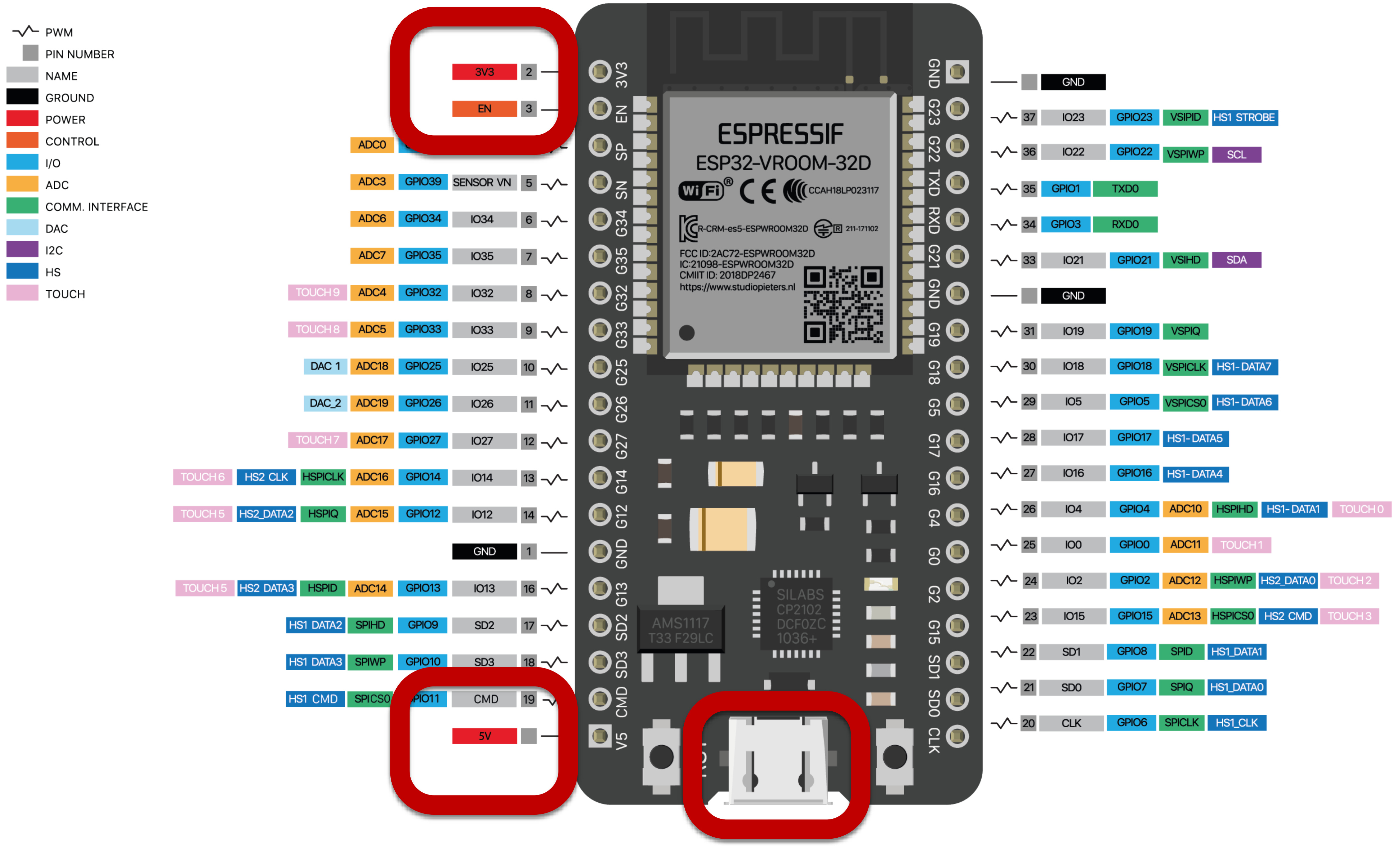
Now?



Now?




PWM
PIN NUMBER
NAME
GROUND
POWER
CONTROL
I/O
ADC
COMM. INTERFACE
DAC
I2C
HS
TOUCH
3.3V
3V3 2
EN 3
ADC0 GPIO36 SENSOR VP 4
ADC3 GPIO39 SENSOR VN 5
ADC6 GPIO34 IO34 6
ADC7 GPIO35 IO35 7
TOUCH 9 ADC4 GPIO32 IO32 8
TOUCH 8 ADC5 GPIO33 IO33 9
DAC 1 ADC18 GPIO25 IO25 10
DAC_2 ADC19 GPIO26 IO26 11
TOUCH 7 ADC17 GPIO27 IO27 12
TOUCH 6 HS2 CLK HSPICLK ADC16 GPIO14 IO14 13
TOUCH 5 HS2_DATA2 HSPIQ ADC15
GND (ground)
GND 1
TOUCH 5 HS2 DATA3 HSPID ADC14 GPIO13 IO13 16
HS1 DATA2 SPIHD GPIO9 SD2 17
HS1 DATA3 SPIWP GPIO10 SD3 18
HS1 CMD SPICS0 GPIO11 CMD 19
5V
5V
3V3 EN SP SN G34 G35 G32 G33 G25 G26 G27 G14 G12 GND G13 SD2 SD3 CMD V5
ESPRESSIF
ESP32-VROOM-32D
Wi Fi
CE
CCAH18LP023117
R-CRM-es5-ESPWROOM32D
211-171102
FCC ID:2AC72-ESPWROOM32D
IC:21098-ESPWROOM32D
CMIIT ID: 2018DP2467
https://www.studiopieters.nl
AMS1117 T33 F29LC
SILABS CP2102 DCF0ZC 1036+
RST
BOOT
GND G23 G22 TXD RXD G21 GND G19 G18 G5 G17 G16 G4 G0 G2 G15 SD1 SD0 CLK
GND (ground)
GND
37 IO23 GPIO23 VSPID HS1 STROBE
36 IO22 GPIO22 VSPIWP SCL
35 GPIO1 TXD0
34 GPIO3 RXD0
IO21 GPIO21 VSIHD SDA
GND (ground)
GND
31 IO19 GPIO19 VSPIQ
30 IO18 GPIO18 VSPICLK HS1- DATA7
29 IO5 GPIO5 VSPICS0 HS1- DATA6
28 IO17 GPIO17 HS1- DATA5
27 IO16 GPIO16 HS1- DATA4
26 IO4 GPIO4 ADC10 HSPIHD HS1- DATA1 TOUCH 0
25 IO0 GPIO0 ADC11 TOUCH 1
24 IO2 GPIO2 ADC12 HSPIWP HS2_DATA0 TOUCH 2
23 IO15 GPIO15 ADC13 HSPICS0 HS2 CMD TOUCH 3
22 SD1 GPIO8 SPID HS1_DATA1
21 SD0 GPIO7 SPIQ HS1_DATA0
20 CLK GPIO6 SPICLK HS1_CLK


PWM
PIN NUMBER
NAME
GROUND
POWER
CONTROL
I/O
ADC
COMM. INTERFACE
DAC
I2C
HS
TOUCH
3V3 2
EN 3
ADC0
ADC3 GPIO39 SENSOR VN 5
ADC6 GPIO34 IO34 6
ADC7 GPIO35 IO35 7
TOUCH9 ADC4 GPIO32 IO32 8
TOUCH8 ADC5 GPIO33 IO33 9
DAC 1 ADC18 GPIO25 IO25 10
DAC_2 ADC19 GPIO26 IO26 11
TOUCH7 ADC17 GPIO27 IO27 12
TOUCH6 HS2 CLK HSPICLK ADC16 GPIO14 IO14 13
TOUCH5 HS2_DATA2 HSPIQ ADC15 GPIO12 IO12 14
GND 1
TOUCH5 HS2 DATA3 HSPID ADC14 GPIO13 IO13 16
HS1 DATA2 SPIHD GPIO9 SD2 17
HS1 DATA3 SPIWP GPIO10 SD3 18
HS1 CMD SPICS0 PIO11 CMD 19
5V
3V3 EN SP SN G34 G35 G32 G33 G25 G26 G27 G14 G12 GND G13 SD2 SD3 CMD V5
GND G23 G22 TXD RXD G21 GND G19 G18 G5 G17 G16 G4 G0 G2 G15 SD1 SD0 CLK
ESPRESSIF
ESP32-VROOM-32D
CCAH18LP023117
R-CRM-es5-ESPWROOM32D
211-171102
FCC ID:2AC72-ESPWROOM32D
IC:21098-ESPWROOM32D
CMIIT ID: 2018DP2467
https://www.studiopieters.nl
AMS1117 T33 F29LC
SILABS CP2102 DCF0ZC 1036+
GND
37 IO23 GPIO23 VSPID HS1 STROBE
36 IO22 GPIO22 VSPIWP SCL
35 GPIO1 TXD0
34 GPIO3 RXD0
33 IO21 GPIO21 VSIHD SDA
GND
31 IO19 GPIO19 VSPIQ
30 IO18 GPIO18 VSPICLK HS1-DATA7
29 IO5 GPIO5 VSPICS0 HS1-DATA6
28 IO17 GPIO17 HS1-DATA5
27 IO16 GPIO16 HS1-DATA4
26 IO4 GPIO4 ADC10 HSPIHD HS1-DATA1 TOUCH0
25 IO0 GPIO0 ADC11 TOUCH1
24 IO2 GPIO2 ADC12 HSPIWP HS2_DATA0 TOUCH2
23 IO15 GPIO15 ADC13 HSPICS0 HS2 CMD TOUCH3
22 SD1 GPIO8 SPID HS1_DATA1
21 SD0 GPIO7 SPIQ HS1_DATA0
20 CLK GPIO6 SPICLK HS1_CLK

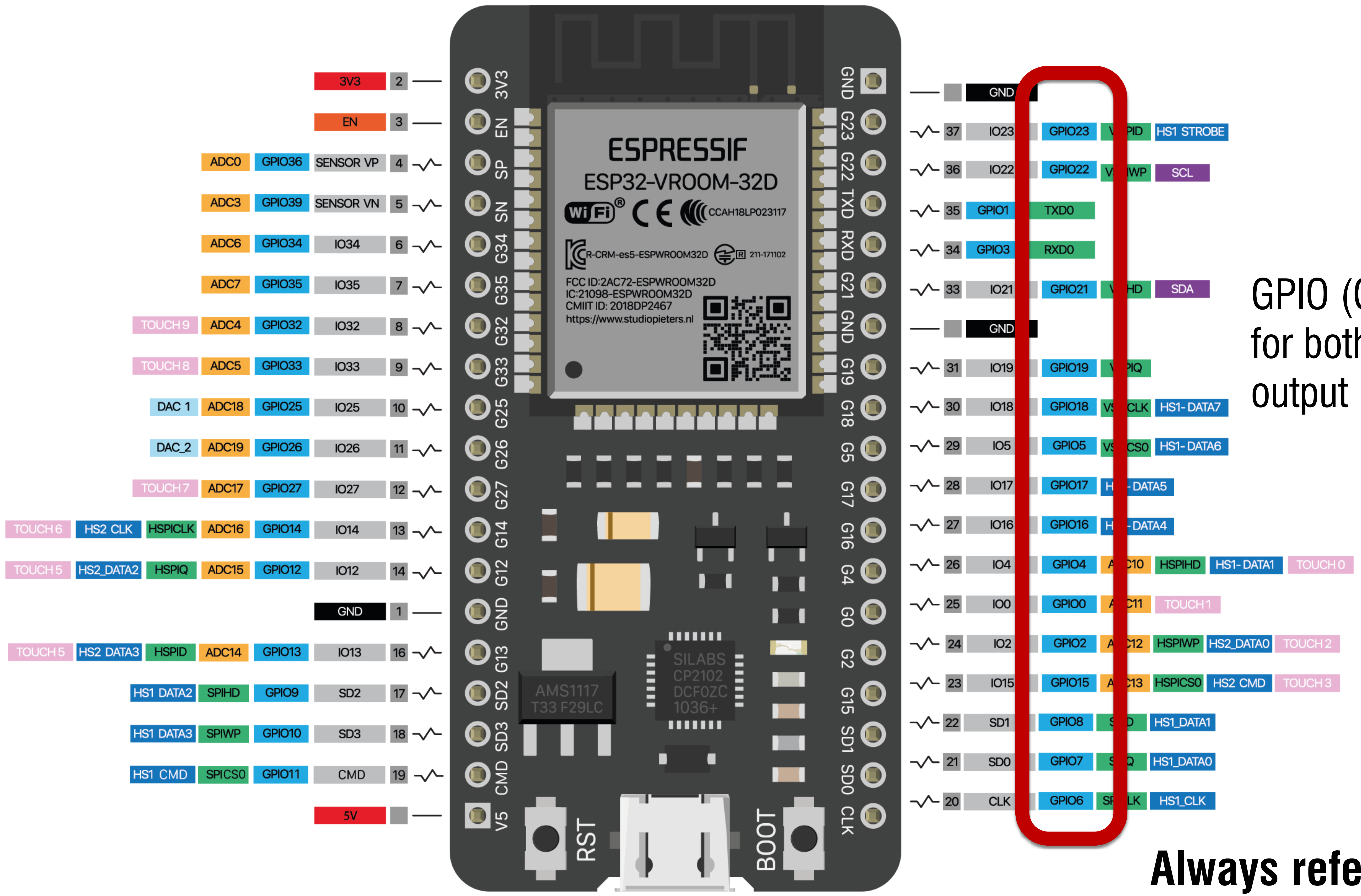
3 options to power up ESP32.

1. Directly via micro-USB port.

2. Unregulated power to GND and 5V pins (Between 5 to 12 v)

3. Regulated power to GND and 3.3V pins (ONLY 3.3v!)

Always only power the microcontroller with one option



GPIO (General Purpose IO) for both digital input and output

Always refer to the pin layout

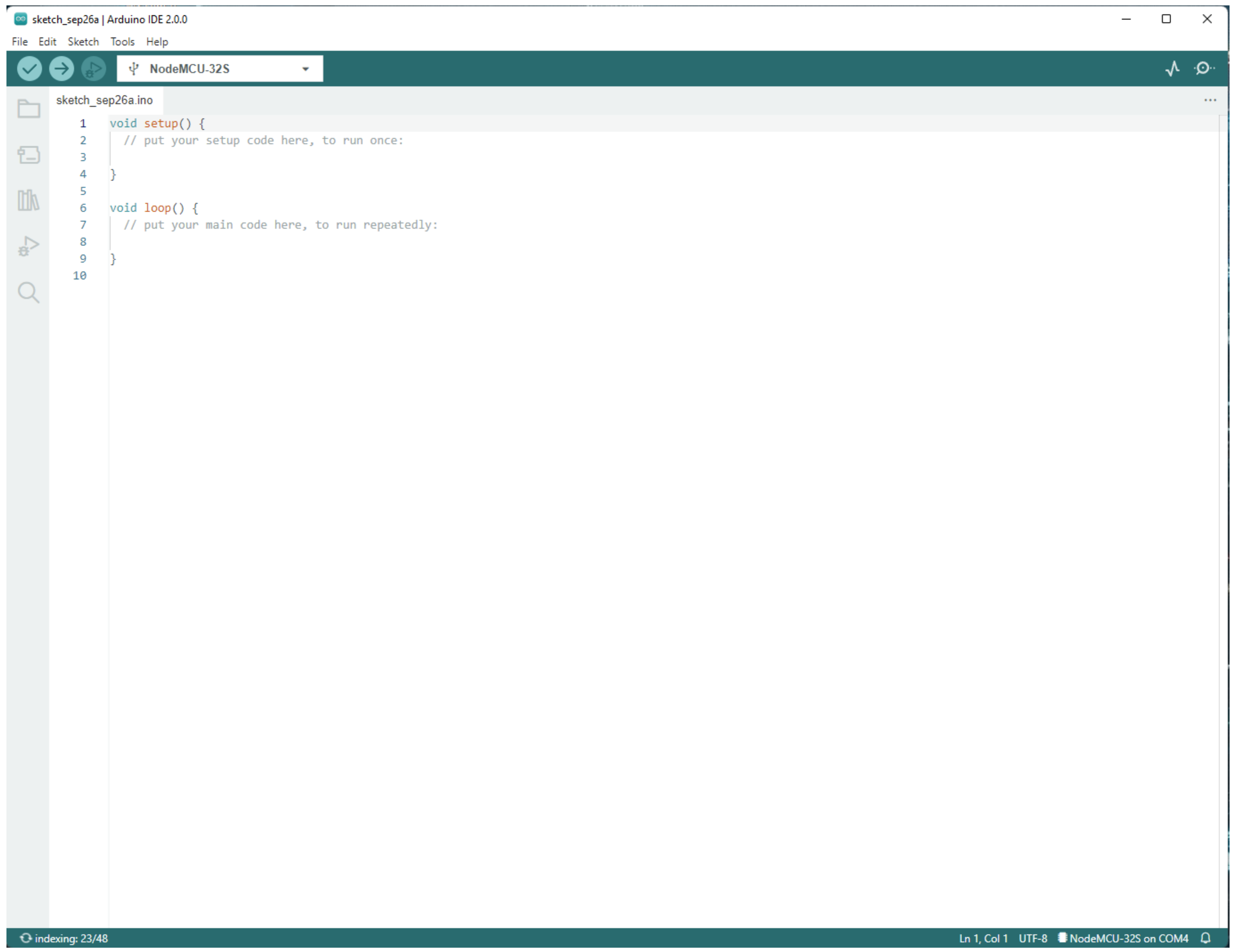
Digital Output – Blink an LED

Digital Output

Set the logic value of a pin
– **LOW** (0V) or **HIGH** (3.3V)

Arduino functions
– **pinMode(pin, OUTPUT)** to set the pin direction
*Often in the **setup()** function*
– **digitalWrite(pin, value)** to write the current value of a pin

Limitations
– Only 0 or 3.3 V with limited current;

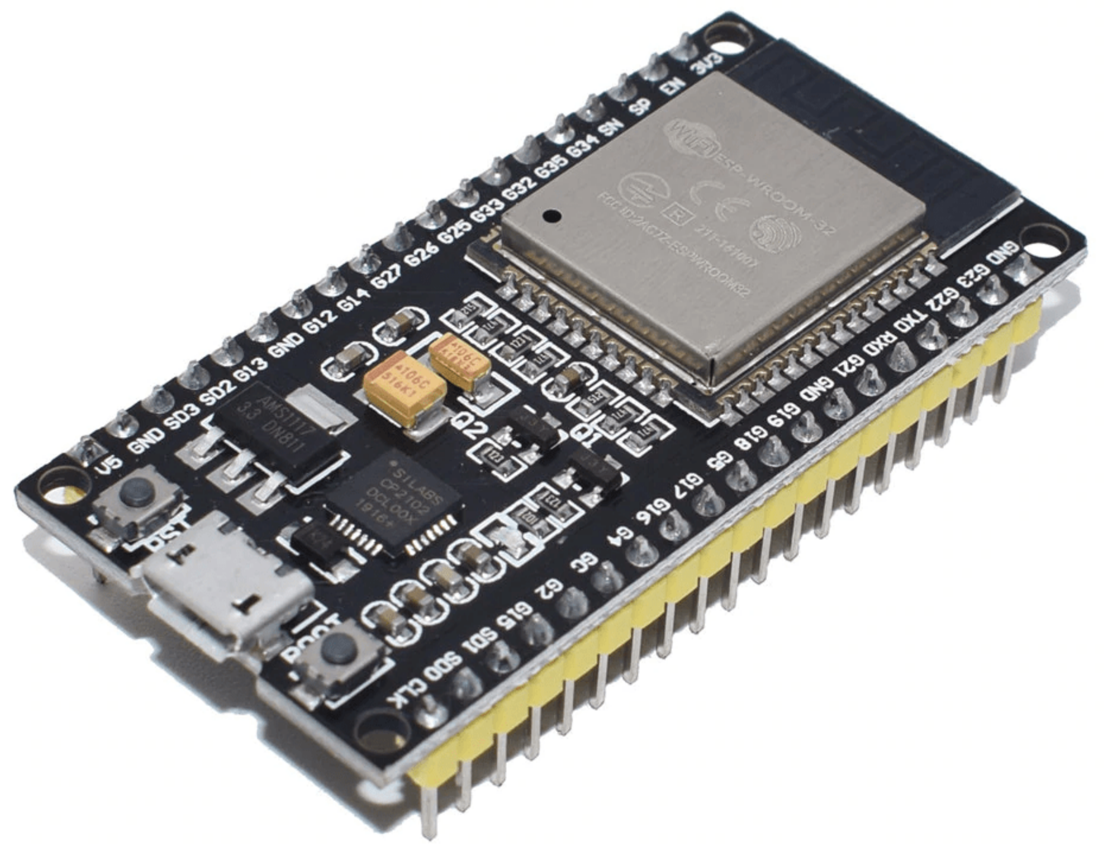


sketch_sep26a | Arduino IDE 2.0.0
File Edit Sketch Tools Help
NodeMCU-32S
sketch_sep26a.ino
indexing: 23/48
Ln 1, Col 1 UTF-8 NodeMCU-32S on COM4

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

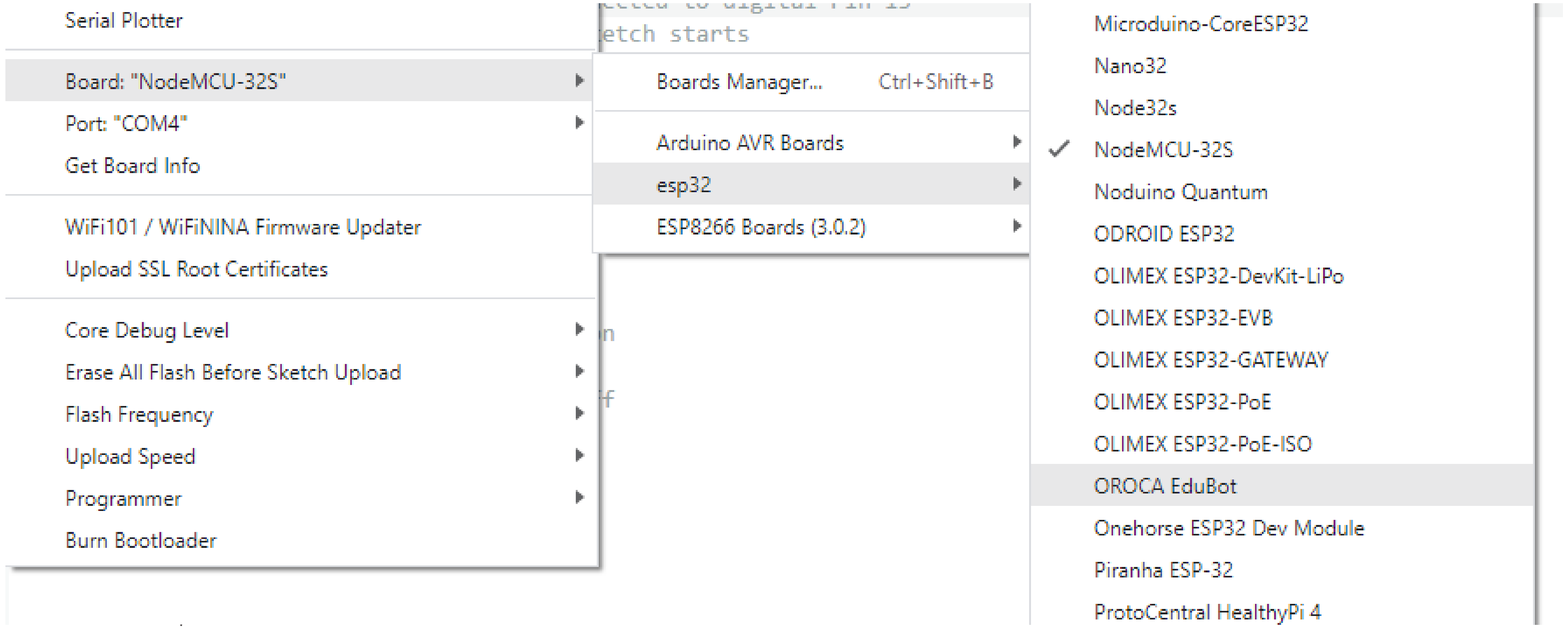
Blink the built-in LED

```
// constants definition
const int ledPin = 2; // Default LED is connected to GPIO 2
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(5000); // wait for 5 second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(5000); // wait for 5 second
}
```



Select
Board: -> esp32 -> NodeMCU-32S

Hit **Upload**



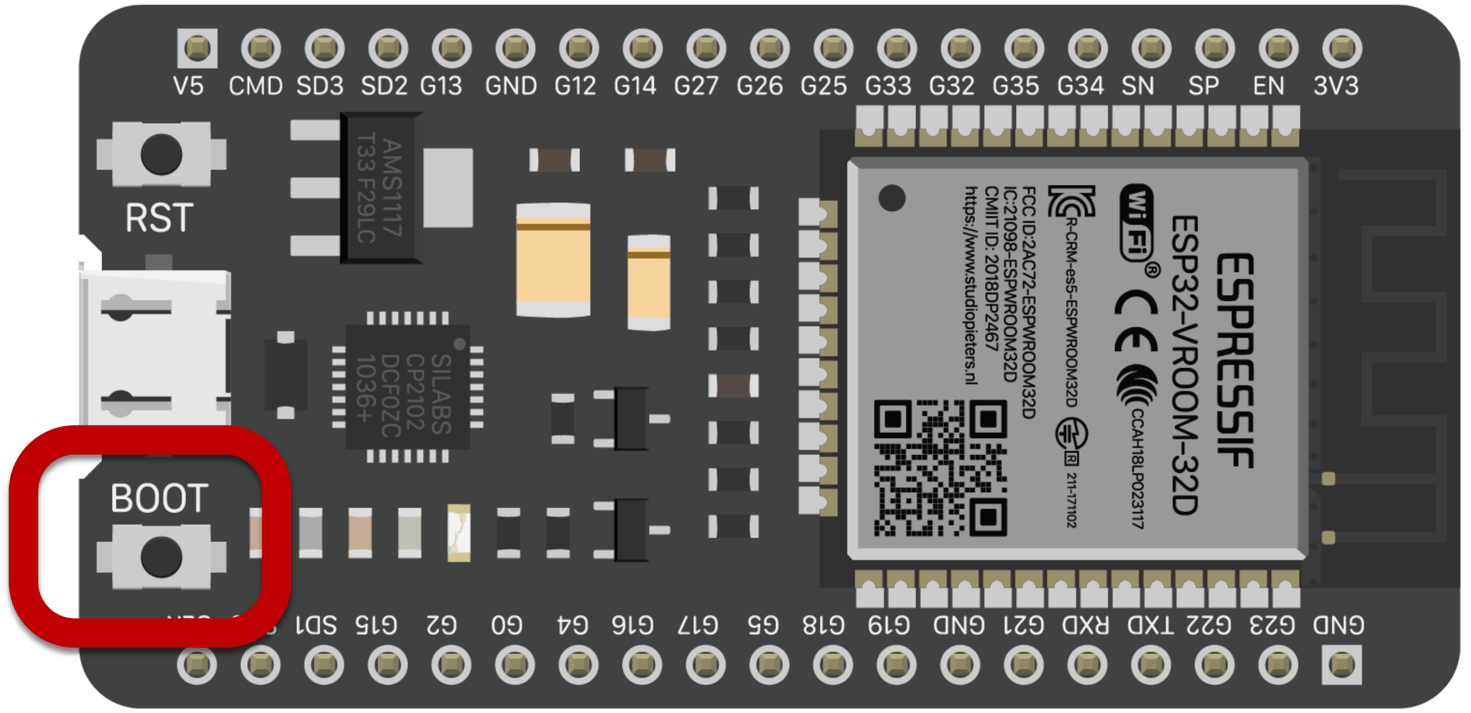
Serial Plotter
Board: "NodeMCU-32S"
Port: "COM4"
Get Board Info
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Core Debug Level
Erase All Flash Before Sketch Upload
Flash Frequency
Upload Speed
Programmer
Burn Bootloader

Boards Manager... Ctrl+Shift+B
Arduino AVR Boards
esp32
ESP8266 Boards (3.0.2)

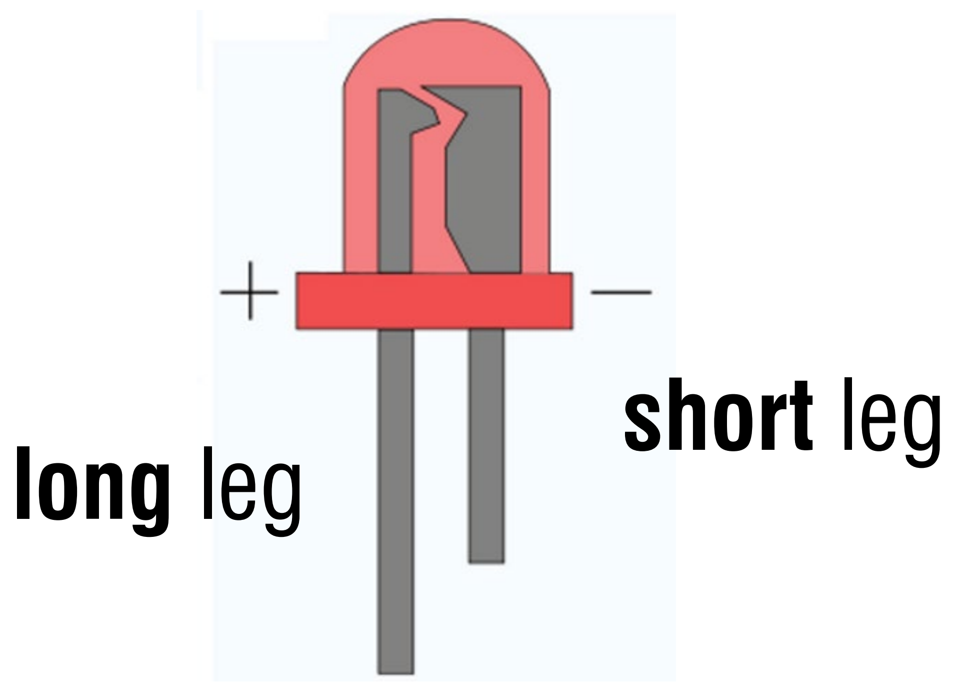
Microduino-CoreESP32
Nano32
Node32s
✓ NodeMCU-32S
Noduino Quantum
ODROID ESP32
OLIMEX ESP32-DevKit-LiPo
OLIMEX ESP32-EVB
OLIMEX ESP32-GATEWAY
OLIMEX ESP32-PoE
OLIMEX ESP32-PoE-ISO
OROCA EduBot
Onehorse ESP32 Dev Module
Piranha ESP-32
ProtoCentral HealthyPi 4

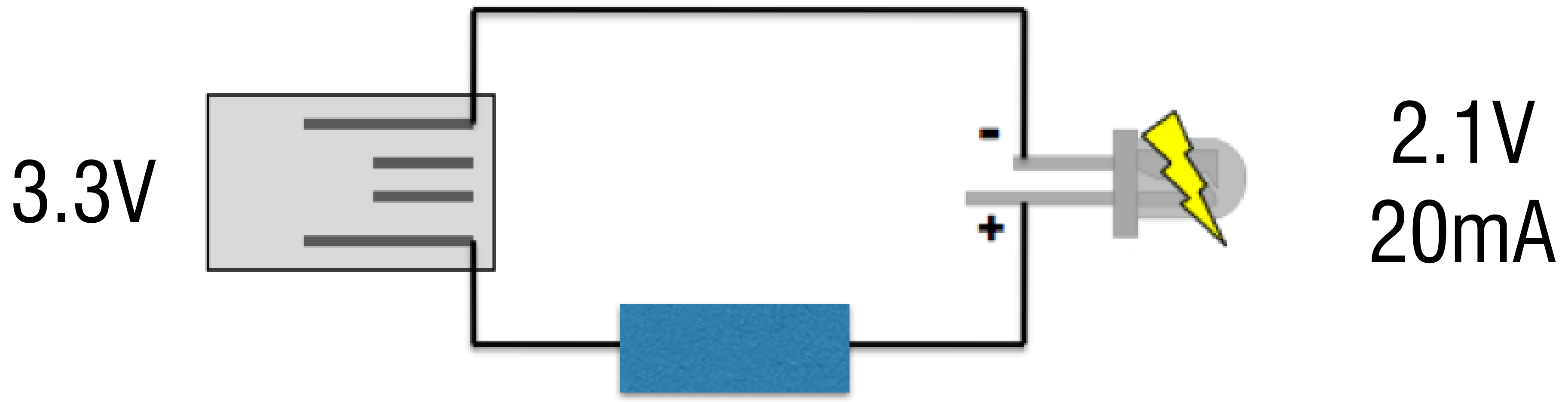
On the ESP32,

Press and hold the BOOT button until you see the code starts uploading

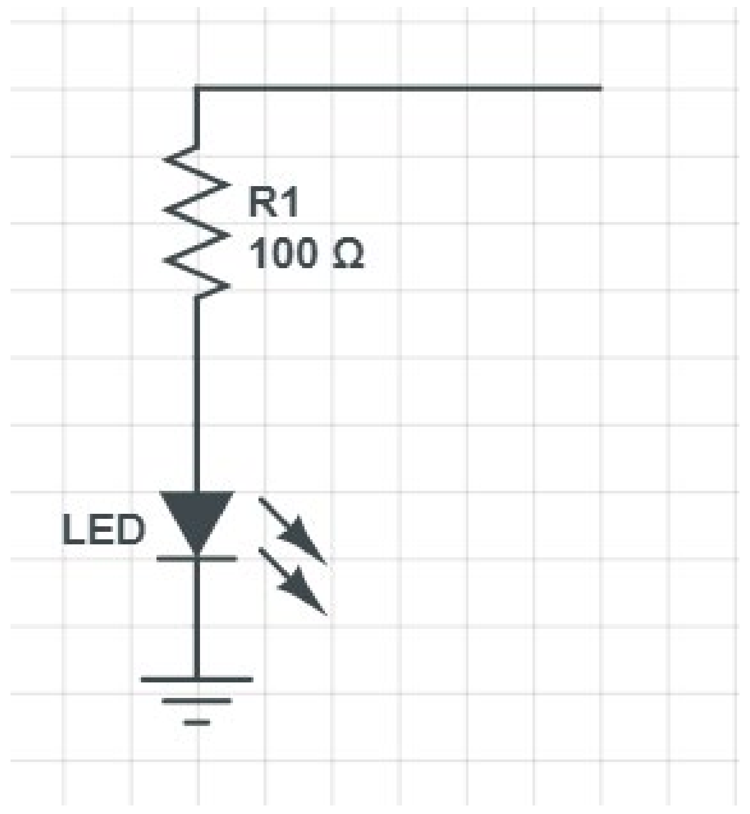


Practice: Light up the RED Led





add a **resistor**
(lowers the current)

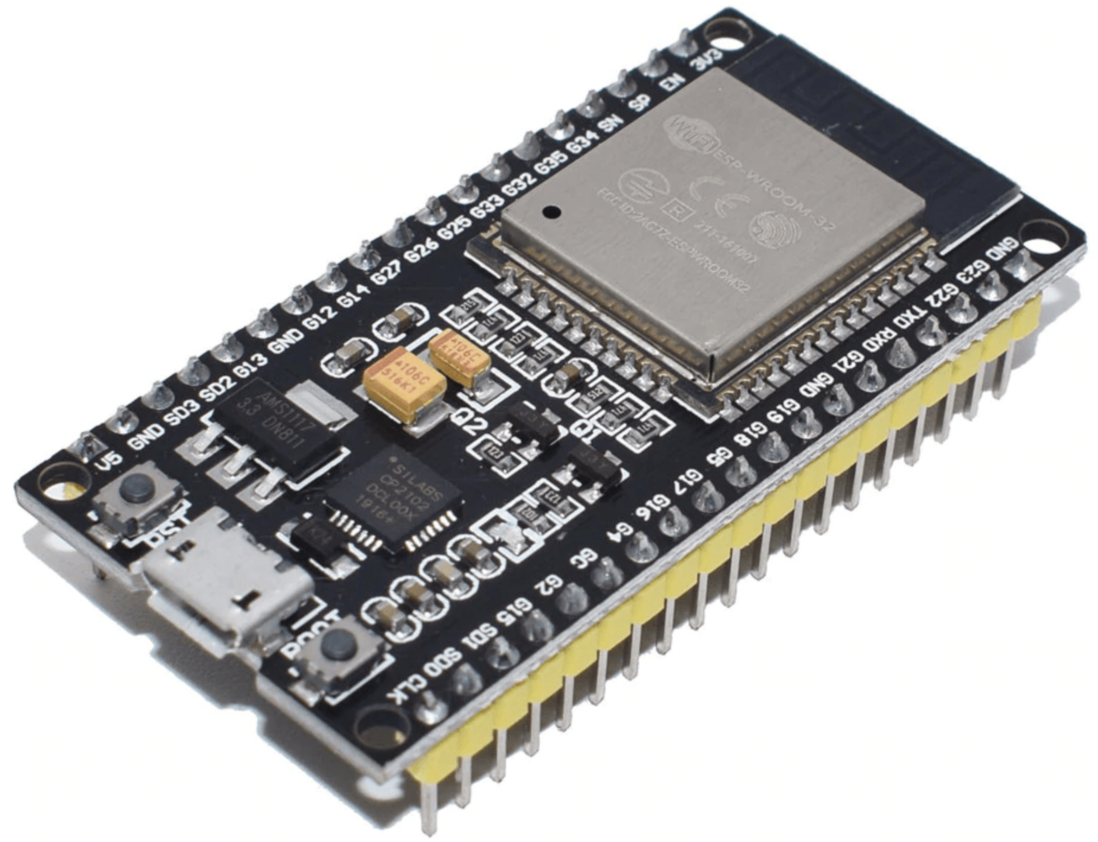


Ohm's Law

$\Delta V = R * I$

Blink an external LED

```
// constants definition
const int ledPin = 23; // Default LED is connected to GPIO 23
// The setup() method runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(5000); // wait for 5 second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(5000); // wait for 5 second
}
```



Serial Communication – talk to PC

Serial Communication

Setup
– **Serial.begin(<baud_speed>)**//9600

Receiving information
– Test is data is available
  Serial.available()
– Read one byte
  Serial.read()

Sending information
– Raw data transfer
  Serial.write(val) or Serial.write(buf, len)

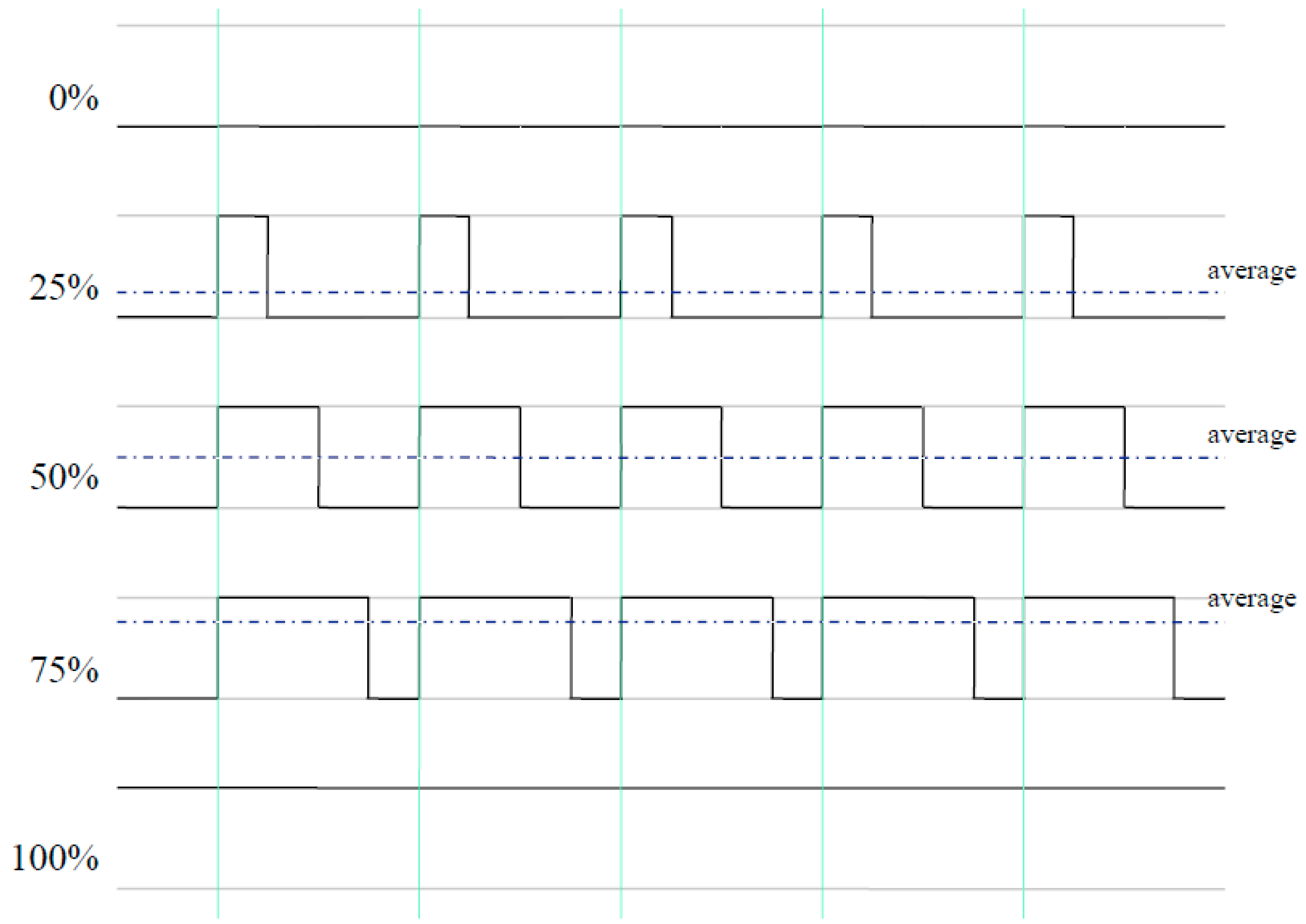
Other commands -> https://www.arduino.cc/reference/en

– Formatted output
  Serial.print (x,{BIN,OCT,DEC,HEX})
– Read formatted data
  Serial.parseFloat()
  Serial.parseInt()

Echo program

```
// setup performs initializations
void setup()
{
  // initialize the serial port setting its speed to 9600 Baud:
  Serial.begin(9600);
}
// the loop() method runs over and over again,
// as long as the Arduino has power
void loop()
{
  // Temporary buffer
  byte incoming_byte;
  // check if the something is pending
  if (Serial.available() > 0)
  {
    // read the pending byte;
    incoming_byte = Serial.read();
    // Sending it back;
    Serial.write(incoming_byte);
  }
}
```

Pulse Width Modulation (PWM)



analogWrite() is on a scale of 0 - 255

Now modify your program to

blink the LED with 100% light intensity when type ‘1’ from the PC

turn it off when type ‘0’

light up with 50% light intensity when type ‘2’