Tested IOT DEVICES Diagram Documentation

Updated 2025.12

Author: David Malawey

This slide deck is to be posted in one of my open source repositories including:

- OpenLabProject ► https://qr.net/openlabproject
- SCUTTLE Tech Guide ► https://qr.net/scuttleproject
- Engineering Resources ► https://qr.net/mxet

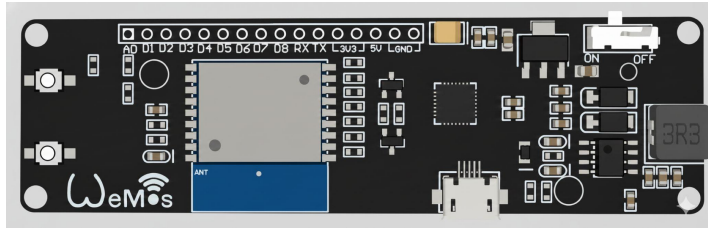
Selected Devices

Default device

ESP8266 “nodeMCU”

Purpose:

- Well documented online
- Minimal cost among MCUs ($5)
- ships with header pins – short time to prototype

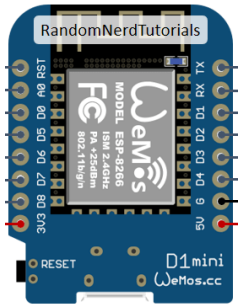


Secondary device

ESP-WROOM with 18650 integration

Purpose:

- Standalone battery operation for wireless needs
- Nearly identical programming requirements

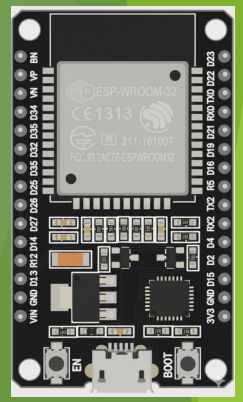


Lightweight

ESP8266 on D1 Mini

Purpose:

- For the most minimal size requirement
- For lowest cost ($4)



Not currently in use:

Listed here:

- ESP32
- Texas Instruments CC3200 Launchpad

Reasoning:

- Unnecessary cost for most projects.
- Higher performance can be met by our microcomputer (pi or beagle)
- Lower volumes or availability

Purpose

[Purpose] (2025)

- This document is a consolidated pack of my notes for self-study of wireless IoT microcontrollers and integration of sensors.
- For each sensor I wished to test & understand, I built my own setup, usually following a guide online. Some designs have pivoted from the examples to reach new outcomes, and I needed to take my notes visually to record how the devices are setup.
- For those self-teaching with sensors:
  - Use this PPTX to have a collection of clean graphics and an easy way to draw your wiring configurations.
  - With this template, you can record a step that may have taken 2 hours of troubleshooting, in 10 seconds.
  - In the end, you will thank yourself for years to come if you perform your own projects with records and save time on the next projects.
- December 2025 is the first time I considered publishing this document which had been very messy and never complete, only serving to tie together my other notes for a long time.
- By now, the diagrams are improved enough for many of them to be intelligible guides for engineers who are learning on their own.

Preface

[Takeaways] (2024)

- This document originated as simple notes drawn for MCU testing at the desktop. Errors & incomplete diagrams are in these slides; the purpose is to use as a template for clean final documentation on ESP related projects in the future.
- wiring diagrams: for simple diagrams, draw.io is now preferred over powerpoint. (2025)
- tests aim to isolate one function, such as control of an output device or sampling of an input device
- tests containing two or more functions, such a MQTT + buzzer are relevant because MCU sometimes unexpectedly occupies output pins to perform internal functions.

About Documentation:
► This pptx is made with metric units in Windows 10 or 11
► The "Earth Ground" symbol (⏚) is a Unicode character (U+23DA)
► code refers to Arduino program, usually 1 file, with .ino extension
► for text in shapes, set text box margins to 0cm
► Links: include links in full text where possible, for accessibility
► Wemos and ESP-WROOM-02 may be used interchangeably in early tests

Formatting:
Headings: DS ISO, bold
Paragraphs: Bahnschrift, Segoe UI Symbol,
External diagrams: drawio format, file: esp_diagrams.drawio

[Usability]
Shortkeys
Ω = ALT+234
⏚ = found in Font Segoe UI symbol
Ribbon show/hide = ctrl+F1

Sources

Description of selected sources and example provided. Convert the initial list into clear table with examples noted. DM 2025.12

Tutorial Sources:
- RandomNerdTutorials.com
- LastMinuteEngineers.com
- Install.wled.me for LED light tutorials
- fastled.io for LED code library
- microcontrollerslab.com

Source	Example Offered	Device
RandomNerdTutorials.com		
Install.wled.me	LED light tutorials	
Fastled.io	LED code libraries	
howToMechatronics.com	Stepper motor & arduino demo	
microcontrollerslab.com	Rotary encoder with ESP8266	
How2electronics.com	Capacitive soil moisture sensor & arduino	Arduino

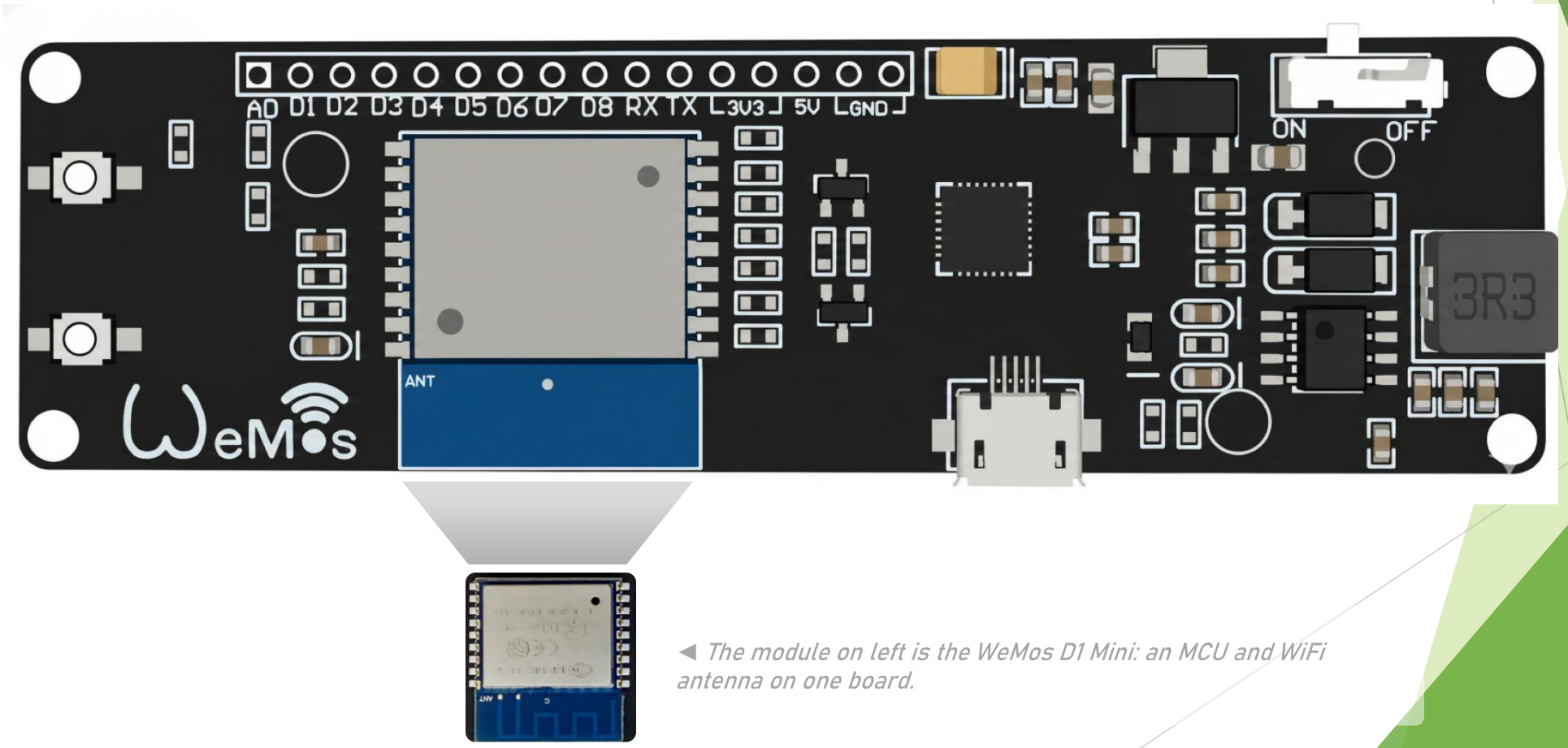
ESP8266 WEMOS Section

Containing ESP8266 chip with added circuitry for battery & charging

Wemos Board

This board is called Wemos D1 ESP Wroom 02 board
The MCU is based on ESP8266.
The first PCB module contains wifi antenna and the MCU board.
This board is mounted on the larger board with a battery holder and more circuitry.

Purchased in 2020, Malaysia from Lazada online.
Tested during 2020

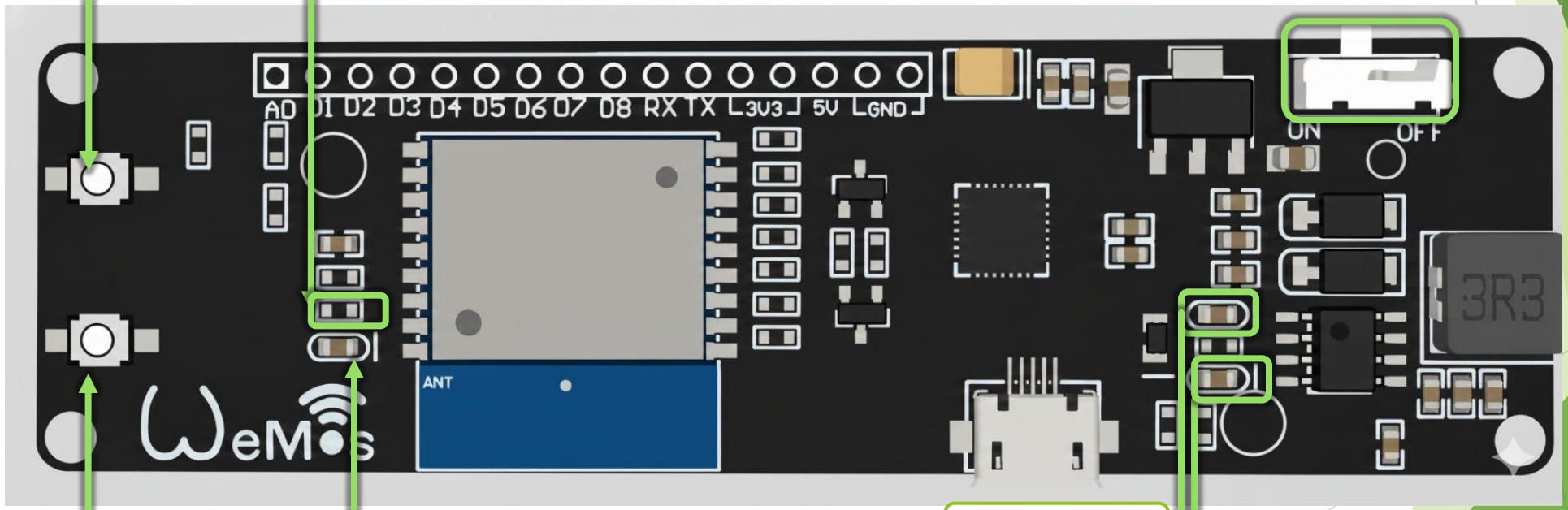


◄ *The module on left is the WeMos D1 Mini: an MCU and WiFi antenna on one board.*

WeMos Board

[2025.06] Use this base image for cleanup of diagrams in future projects with WeMos board. WeMos contains ESP-WROOM-02 on a circuit board with pinouts and onboard battery cell, 18650 size, li-ion.

Diagram Notes:
(1) junction to bridge by soldering, for sleep mode implementation. This junction is tied to GPIO16, controls reset function.



WeMos Board Components

Documentation compiled from online sources including Arduino Forum:
https://arduino.stackexchange.com/questions/60419/
Last accessed 2025.07

Notes for diagram below

1. LED, blue, attached to GPIO16
2. ADC, input pin for analog inputs, tied to analog-to-digital converter on MCU
3. Resistors, 220k & 110k onboard make voltage divider circuit. This circuit is connected to the pin labeled "AD" for analog inputs.
4. Auto-Programming, circuit for setting MCU input pins (flash and reset) when the user flashes the MCU. This circuit eliminates requirement for user to hold flash or reset button when sending command to flash the board from the Arduino IDE, connected over the USB cable.
5. GPIO16, for control of RESET function, tied to these solder pads
6. LDO, low dropout regulator, AMS 11173, provides 3.3v dc power to the board. The li-ion battery is in the 3.7v range.
7. CP2102, a USB-to-serial converter chip, allows PC to communicate to the MCU from USB to UART (serial)
8. TP5410, a controller for recharging lithium battery cell using 5.0v from the USB port.
9. PWR switch - disconnects the battery from the circuit. The board can still be powered by the USB input when OFF.

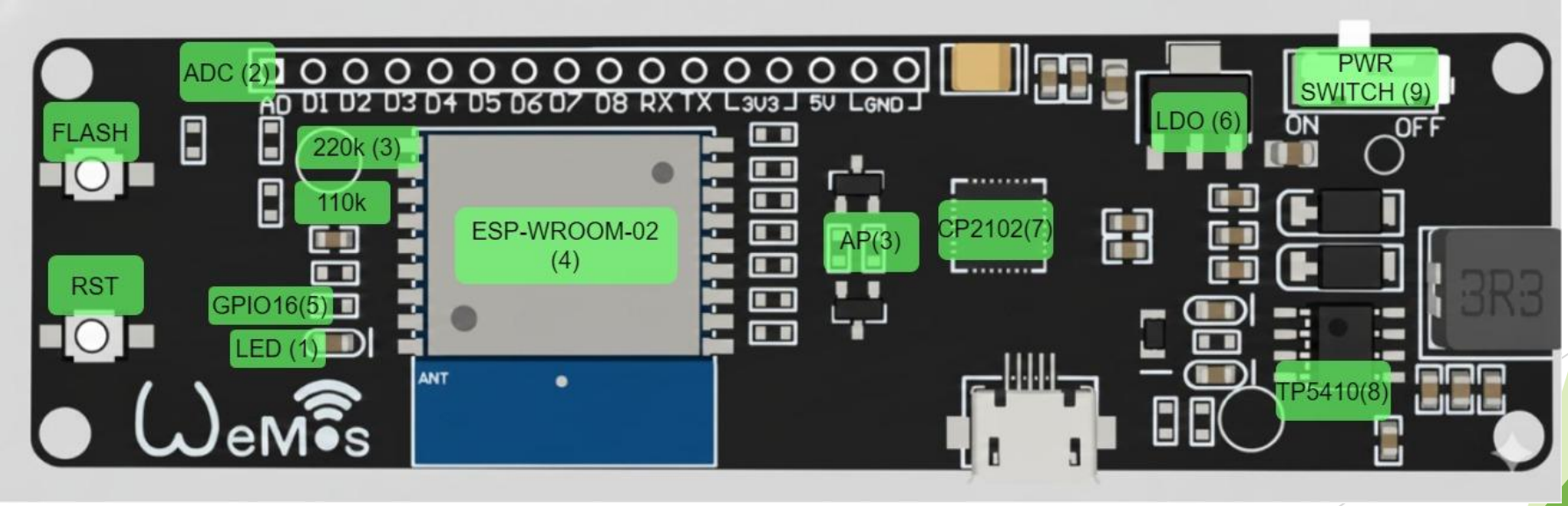


Diagram compiled in drawio file 2025_ESP_diagrams.drawio

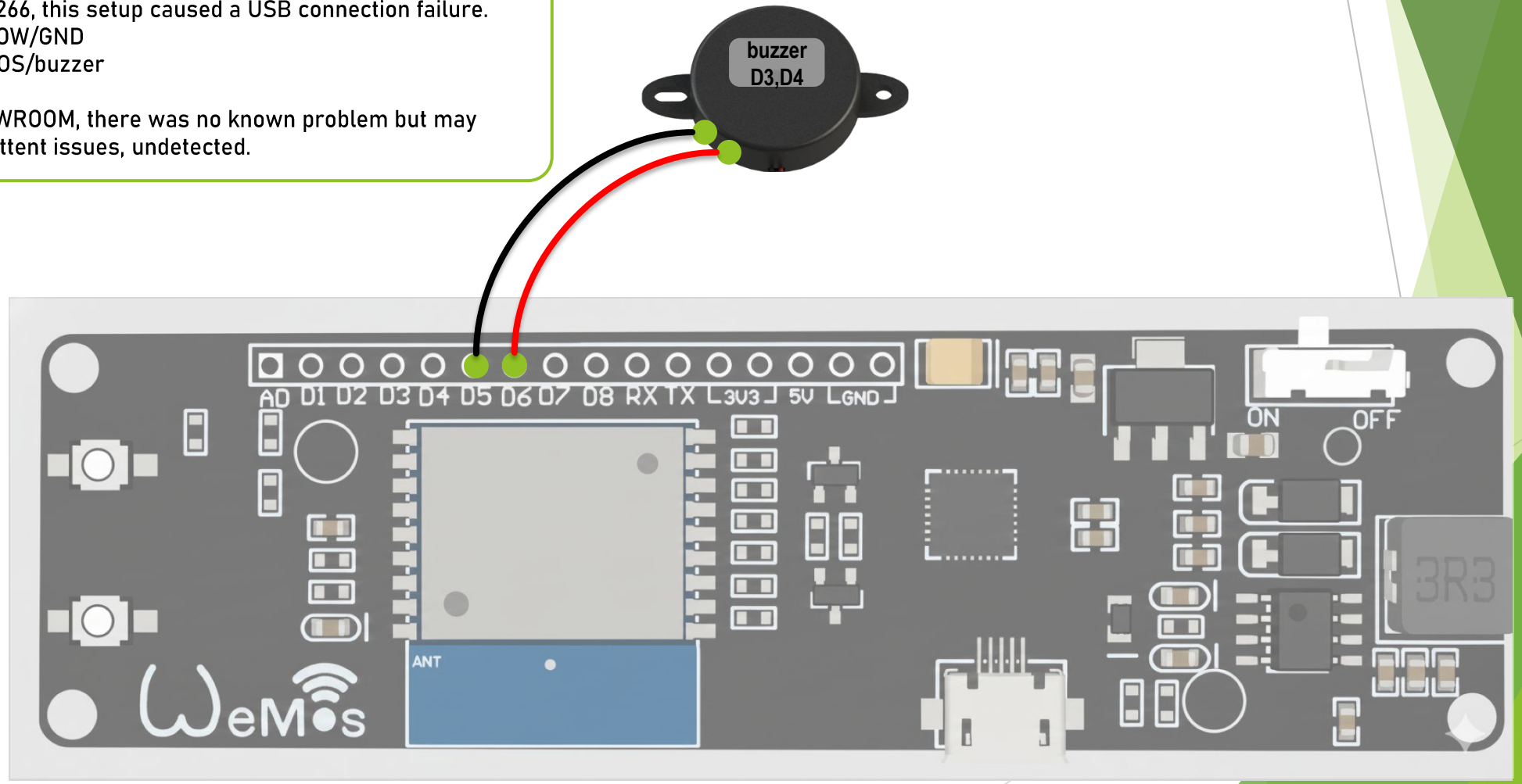
ESP-WROOM Buzzer test (successful)

Caution: be careful about using pins “0” and “2” for assignment to buzzer. If possible, find other pins that are not tied to LED or buttons in any condtion.

When using ESP8266, this setup caused a USB connection failure.
D3 = using 0 for LOW/GND
D4 = using 2 for POS/buzzer

When using ESP-WROOM, there was no known problem but may have been intermittent issues, undetected.

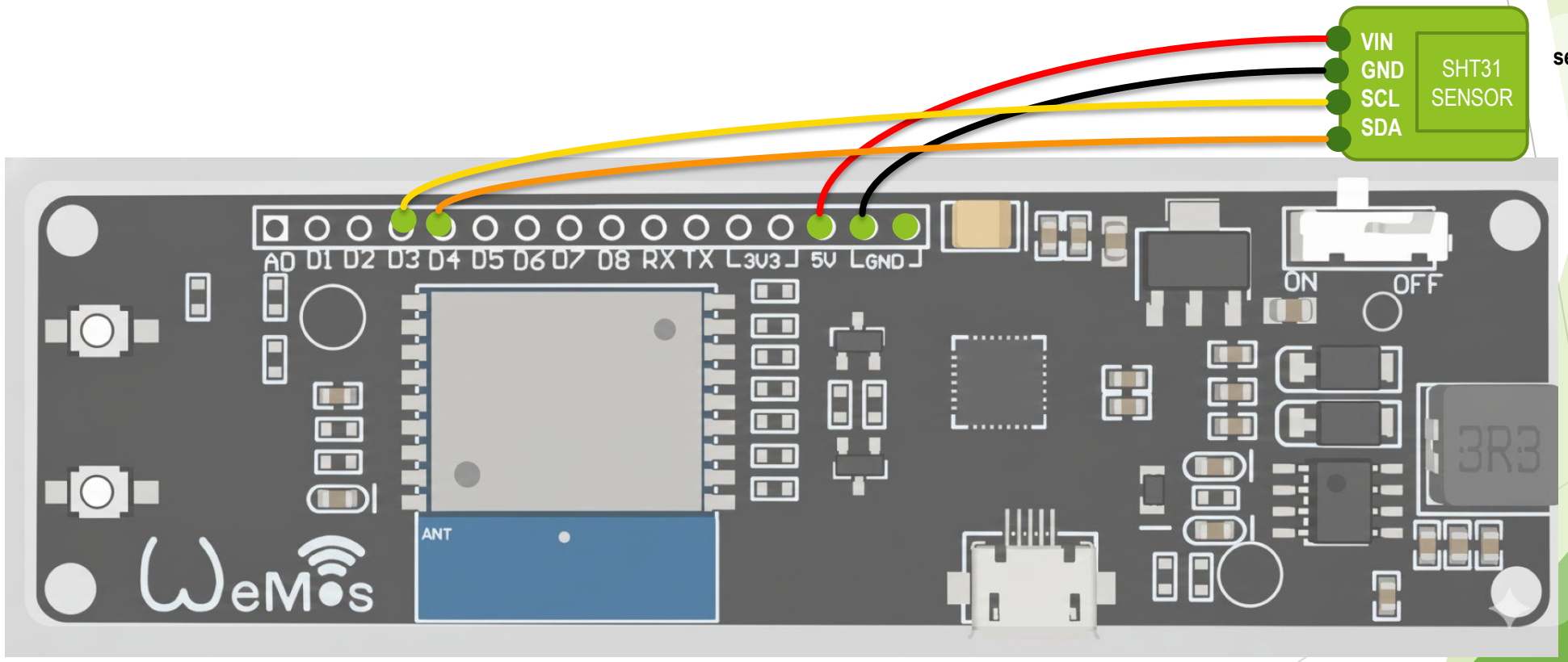
Test Notes:
(This test was initiated in 2020 for verifying which pins can operate for our desired inputs/outputs. Treat these slides as notes from the desk, some are incomplete or have errors. Successful & repeatable setups have more specific & clear notes.



ESP-WROOM SHT-31 sensor, i2c

incomplete documentation, from 2020
Attempt data sampling from temp/humidity sensor SHT31
►SCL,SDA connects to D3, D4 on wemos board
► This device used many times on esp8266 boards

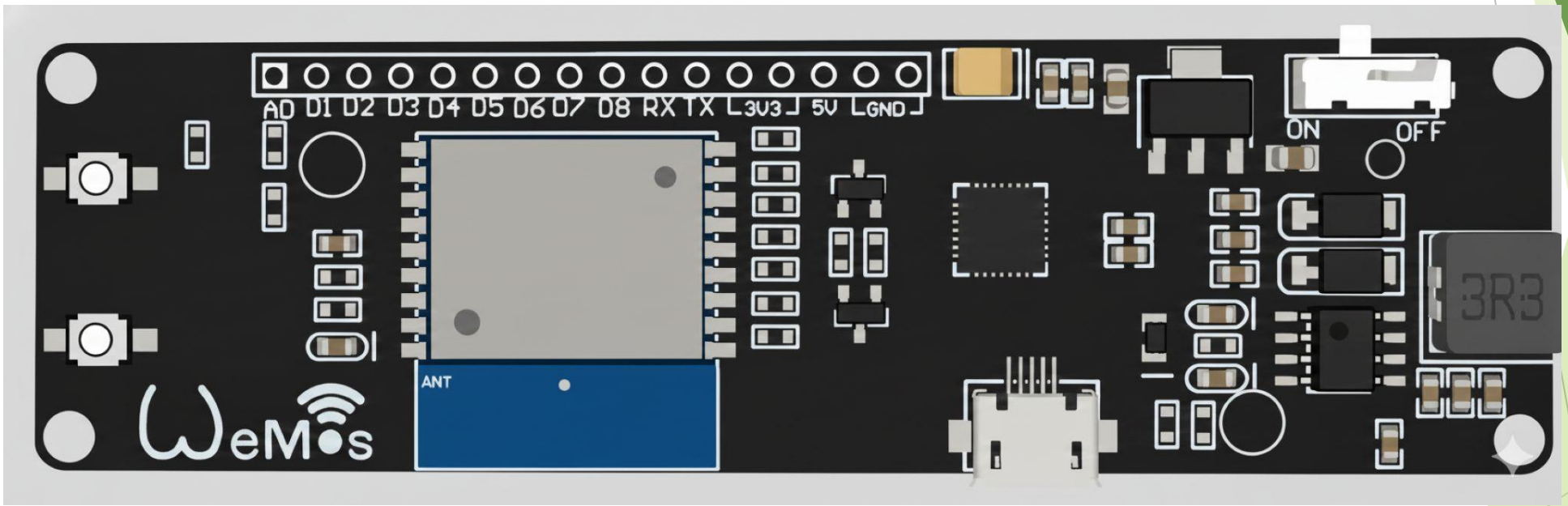
This test was initiated in 2020, graphics were incomplete.



sensor connections:
D3,D4,5v,GND

ESP-WROOM IMAGE

High Quality Photo for implementing diagrams with WeMos board.
Recommended: overlay white shape with 30% transparency over this image to begin diagram.
2025 photo replaced with gemini-generated vector graphic.



ESP32 Section

Devices integrated on ESP32 boards – ESP-WROOM-02

The ESP32 maybe the most popular wifi board for makers, found near $9 per board in 2020.

This device has

ESP-WROOM 02 Device Pin Data

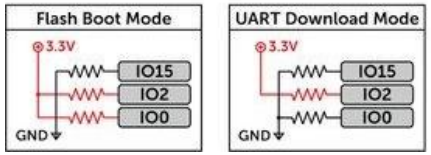
Data gathered ~2020 to understand the details of ESP8266 Devices.

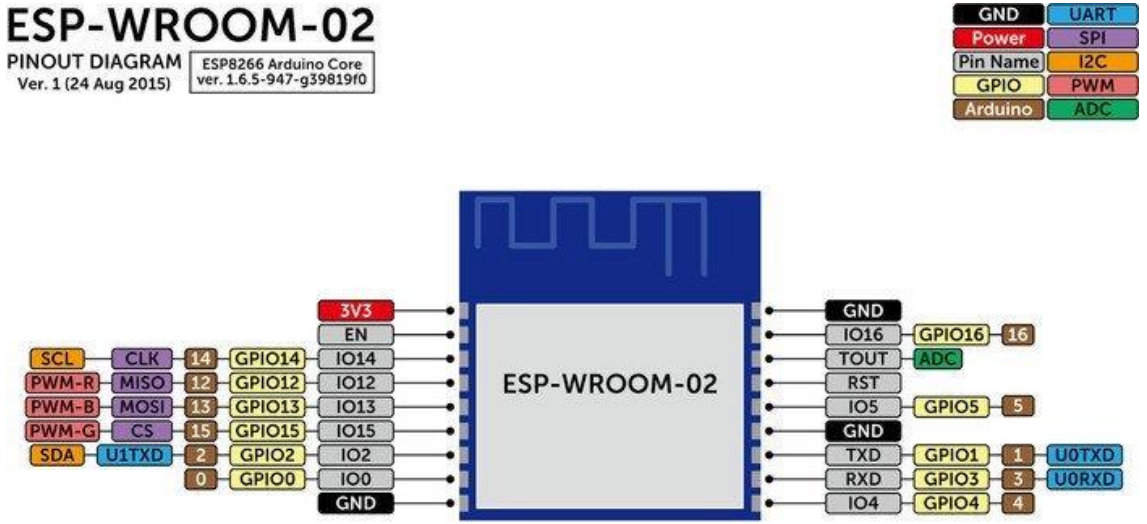
WROOM02 is a version of an integrated ESP8266 mcu on a minimal PCB. Unlike the larger boards in this document, the WROOM-02 device has pads for soldering rather than the male pins ready for wire connections.

This page helps define which circuitry is inside the IC versus on the development boards on other slides.

Knowing where is the "internal pullup" is key to integrating simple sensors like switches, and get them working.

Onboard Resistors





My summary on pullups from datasheet

Pins	Internal Pullup?	WROOM Silkscreen label
GPIO2	YES	D4
GPIO1	YES, but cannot use simultaneously with serial	TX
GPIO15	Pulldown	
GPIO0	Pulldown & pullup	FLASH (existing button)
GPIO14	NO	D5

Directly from datasheet 0C-ESP- WROOM-02 By Expressif Systems

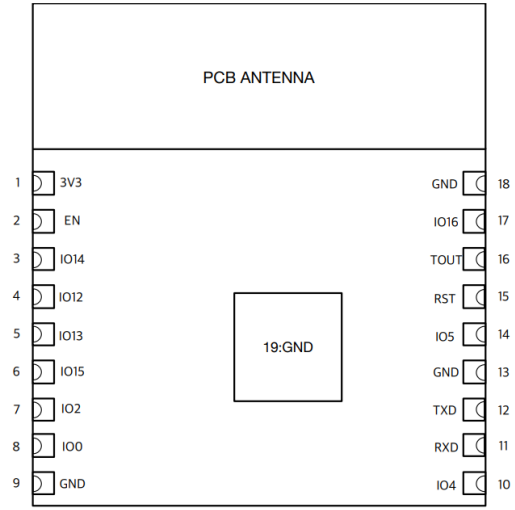
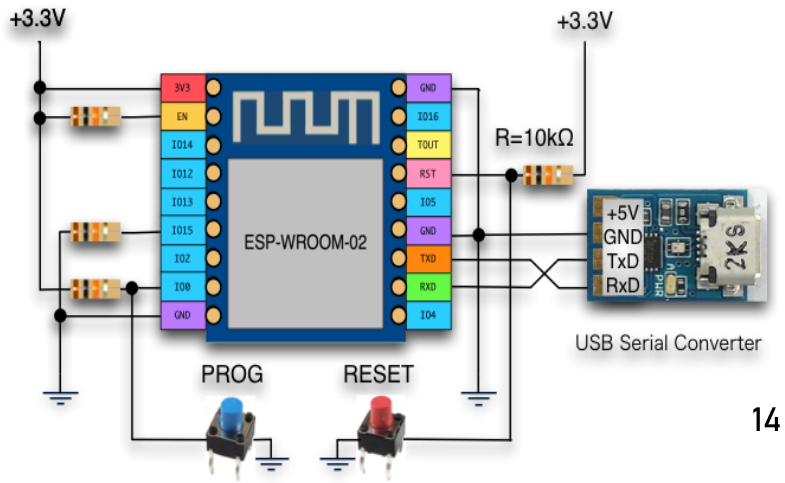


Figure 2-1. ESP-WROOM-02 Pin Layout (Top View)

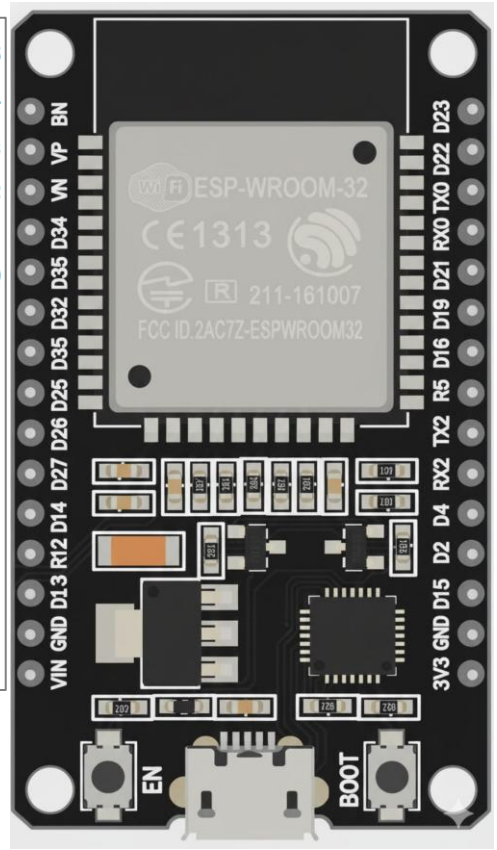
Onboard Resistors



ESP32 Pinout (part1)

This graphic has professional level of detail but we need a reduced pinout description for making simple wiring descriptions. Images retained for reference & troubleshooting.

				Chip-enable signal,Active High.		EN	pin15
				ADC PA	RTC_GPIO0	ADC1_CH0	SENSOR_VP	GPIO36	pin14
				ADC PA	RTC_GPIO3	ADC1_CH3	SENSOR_VN	GPIO39	pin13
					RTC_GPIO4	ADC1_CH6	VDET1	GPIO34	pin12
					RTC_GPIO5	ADC1_CH7	VDET2	GPIO35	pin11
			XTAL_32kHz	Touch9	RTC_GPIO9	ADC1_CH4		GPIO32	pin10
			XTAL_32kHz	Touch8	RTC_GPIO8	ADC1_CH5		GPIO33	pin9
				DAC_1	RTC_GPIO6	ADC2_CH8	EMAC_RXD0	GPIO25	pin8
				DAC_2	RTC_GPIO7	ADC2_CH9	EMAC_RXD1	GPIO26	pin7
				Touch7	RTC_GPIO17	ADC2_CH7	EMAC_RX_DV	GPIO27	pin6
HS2_CLK	SD_CLK	HSPI_CLK	MTMS	Touch6	RTC_GPIO16	ADC2_CH6	EMAC_TXD2	GPIO14	pin5
HS2_DATA2	SD_DATA2	HSPI_MISO	MTDI	Touch5	RTC_GPIO15	ADC2_CH5	EMAC_TXD3	GPIO12	pin4
HS2_DATA3	SD_DATA3	HSPI_MOSI	MTCK	Touch4	RTC_GPIO14	ADC2_CH4	EMAC_RX_ER	GPIO13	pin3
								GND	pin2
POWER								VIN	pin1

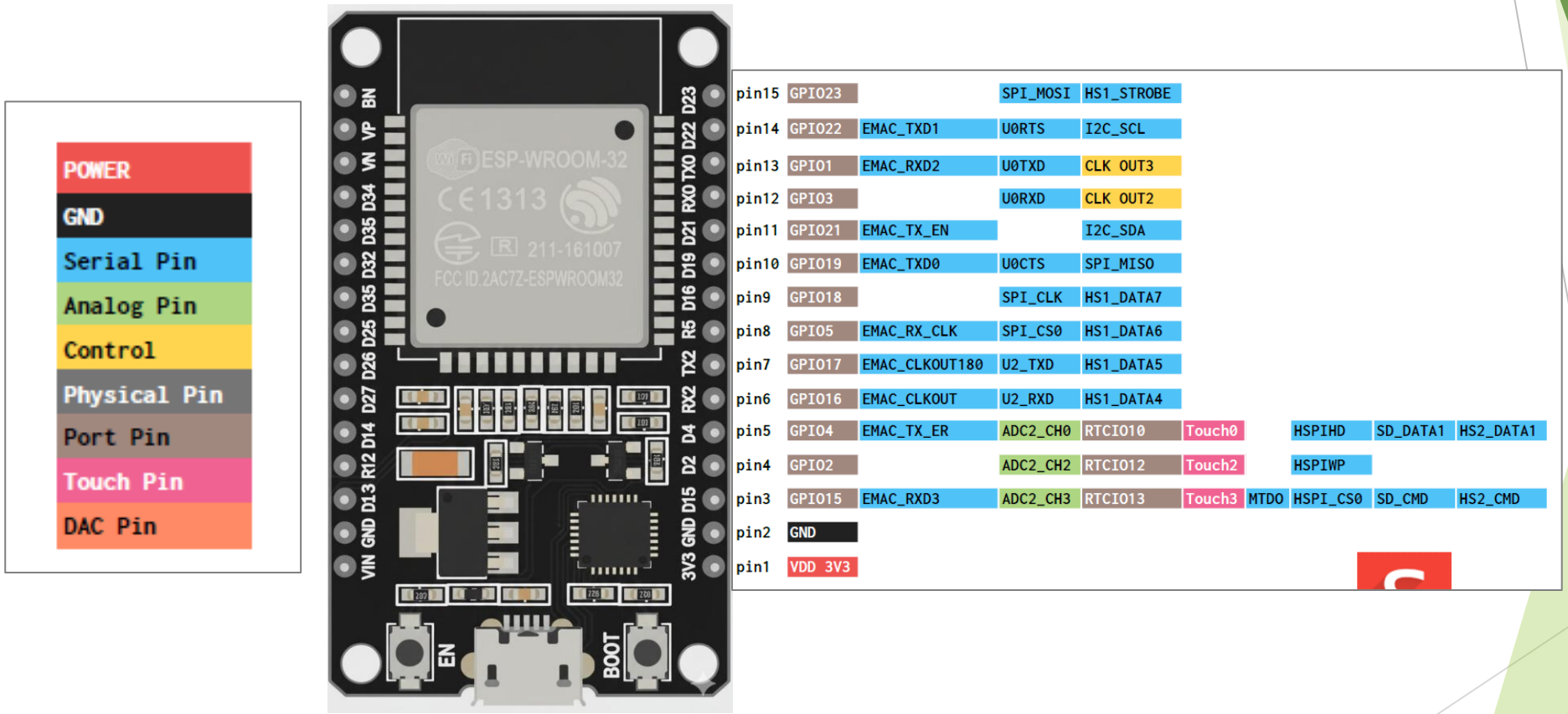


POWER
GND
Serial Pin
Analog Pin
Control
Physical Pin
Port Pin
Touch Pin
DAC Pin

Originally sourced from Playelek.com, with 2016-made graphic.

ESP32 Pinout (part2)

ESP32 Dev Kit v1 Pinout – originally sourced from playelek.com, 2016 graphic



Originally sourced from Playelek.com, with 2016-made graphic.

ESP8266 Section

Devices integrated on ESP8266 boards.

ESP8266 is the simplest among popular WiFi enabled mcu devices, and lowest cost, often below $6 per board.

ESP-8266MOD

Introduction to ESP8266-MOD. This is a board designed as an evaluation board for ESP8266-MOD commonly copied into products with minimal documentation & available online.

Onboard Resistors

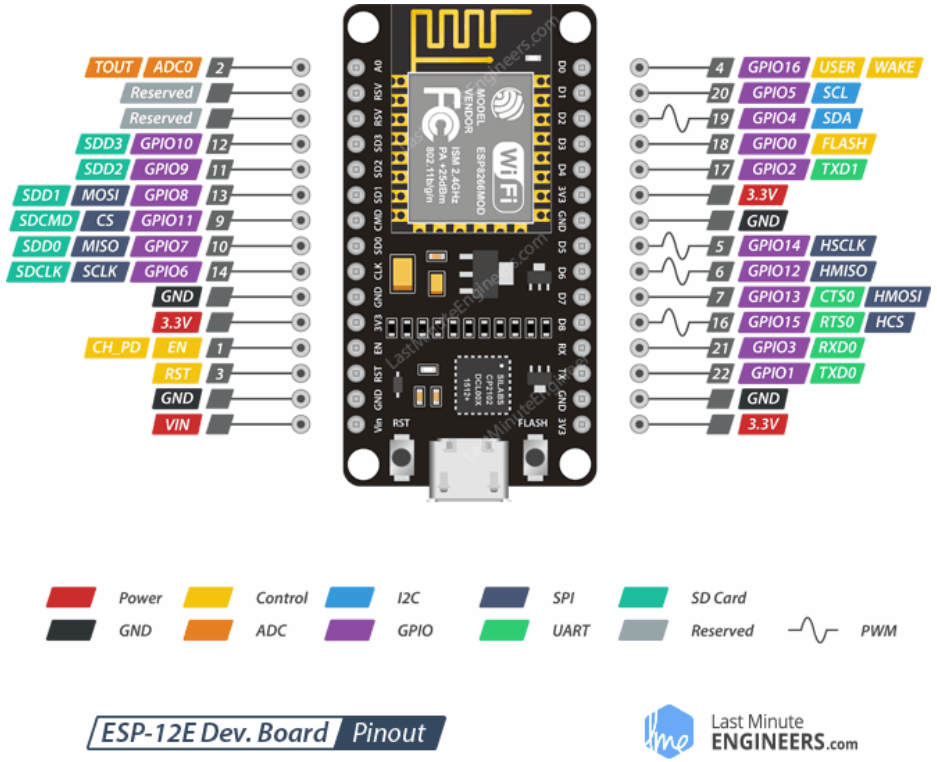


Diagram from LastMinuteEngineers.com, retrieved 2020

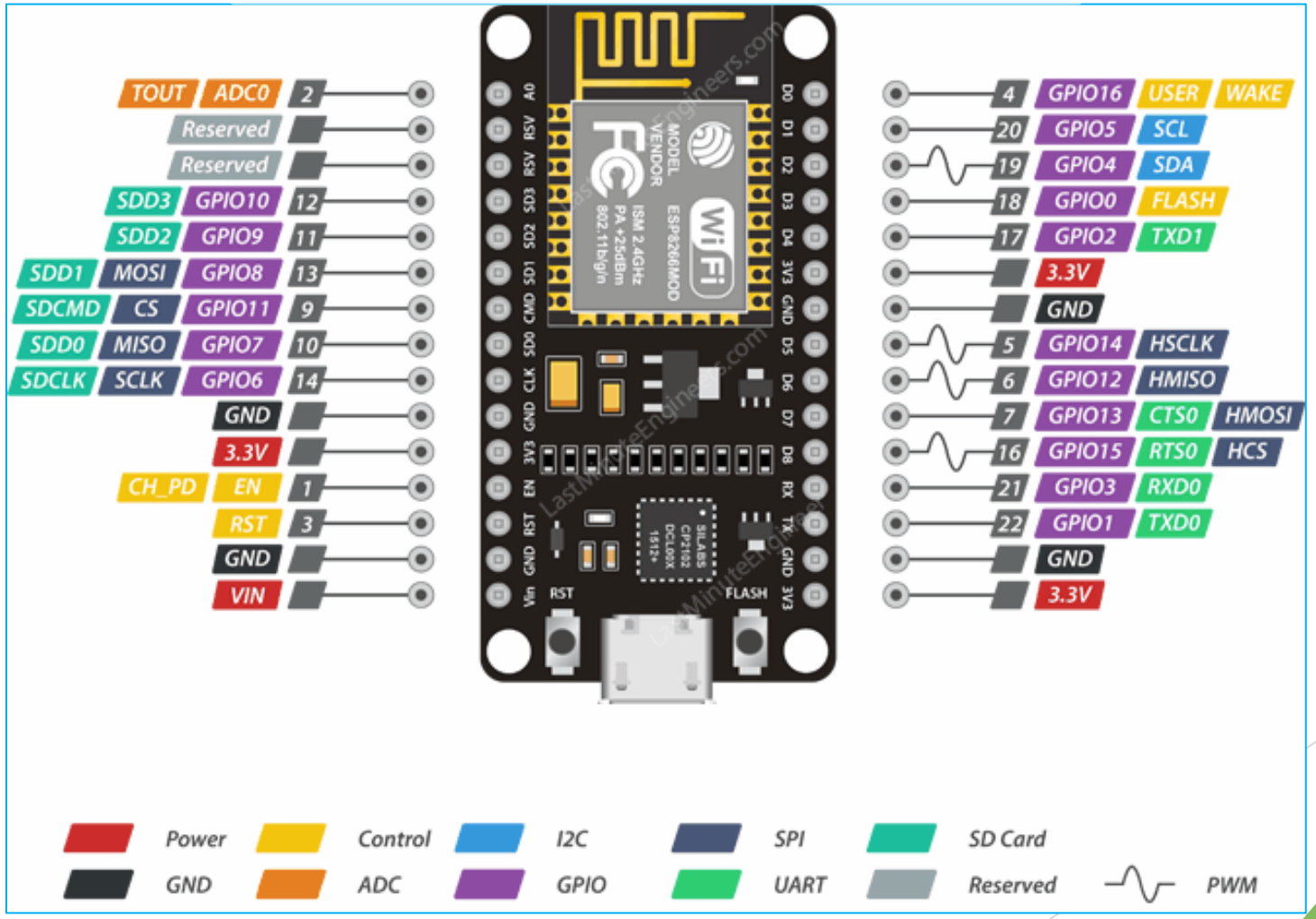
Special notes:

Pins	Note	SILKSCREEN
GPIO16 PIN 4	WAKE pin	D0
GPIO2 PIN 17	~Pulled up ~High at boot ~Connected to onboard LED	D4
GPIO0	~Pulled up ~Tied to FLASH button ~boot fails if low	D3
GPIO15	~Pulled to GND ~boot fail if HIGH	
PIN 3	~ Tied to RST	

ESP-8266MOD

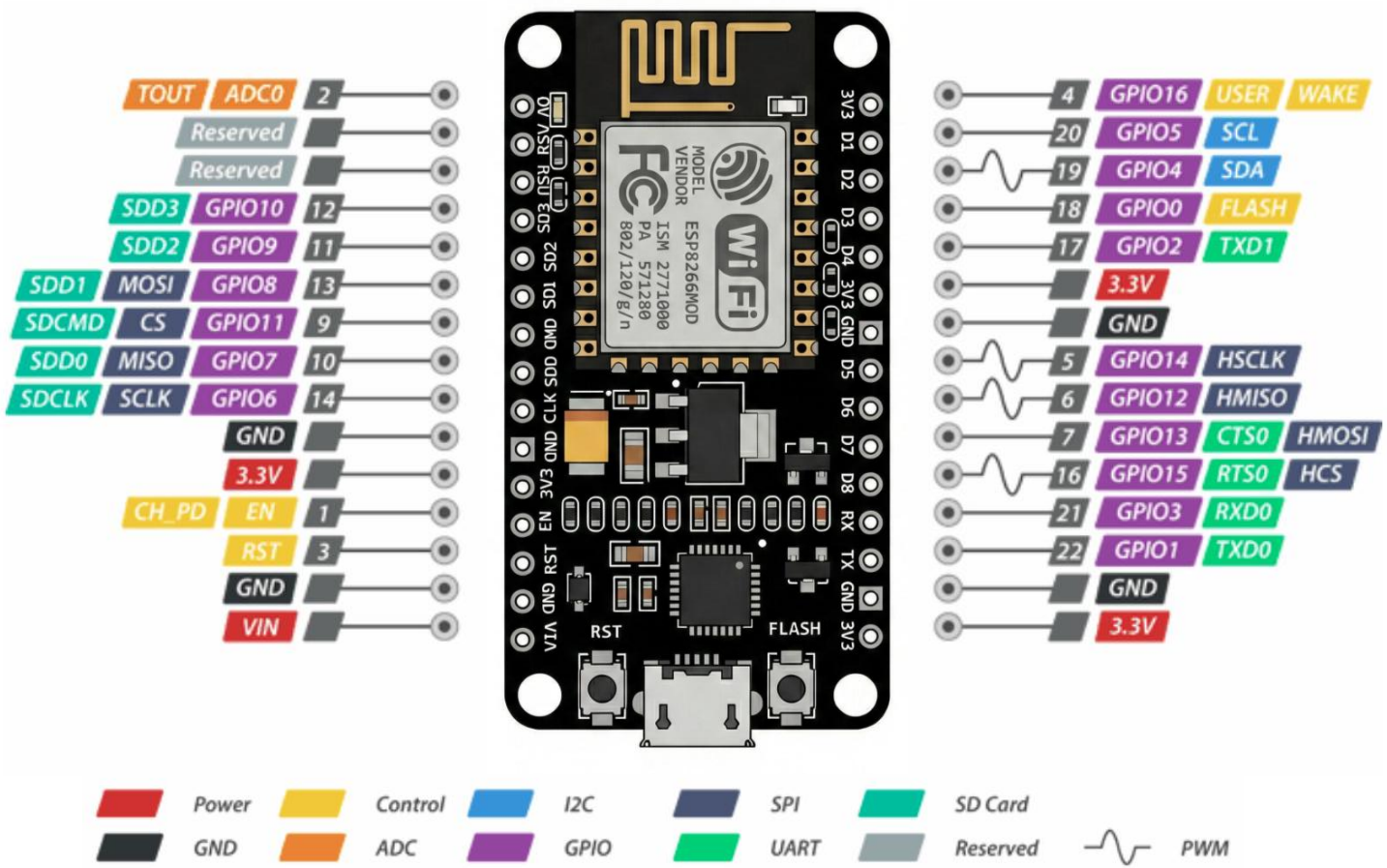
This image is the best starting point for producing simple connection diagrams.
Made by lastminuteEngineers, and presently in process as a vector graphic in .drawio for open publication. (DM 2025.12)

Titled "ESP-12E Dev. Board Pinout" by LastMinute Engineers.com



ESP 8266MOD (nodeMCU)

(incomplete) notes for connecting sensors on D3, D4 ect. Depending on the arduino libraries installed, pin numbers in software must define the pins with the pin number of the chip (0, 2, etc) or pin number of the boards (GPIO14 = 14 and so forth).



ESP-12E Dev. Board Pinout

Last Minute ENGINEERS.com

D3 = must use 0
D4 = must use 2
HAVING ERRORS:
Attempt:
14 = gnd
12 = pos (buzzer)

Piezo Buzzer + MQTT(successful)

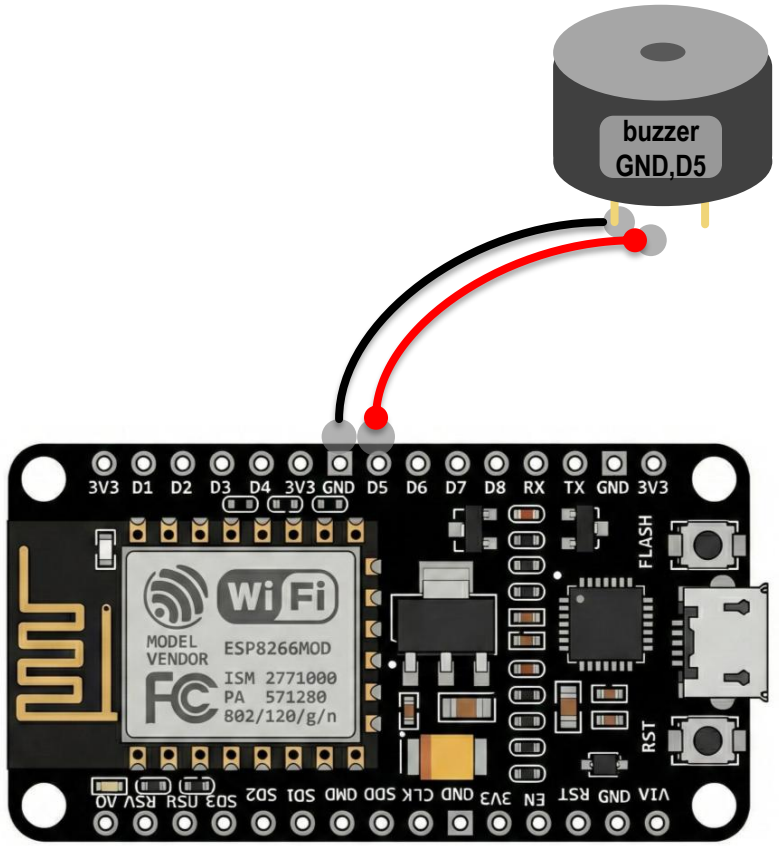
Using ESP8266MOD

Latest test code: 07.12buzzer.ino

Buzzer PN: AT-1438-TWT-R
Spec: 5v, 3.8kHz
Manufacturer: PUI Audio, inc

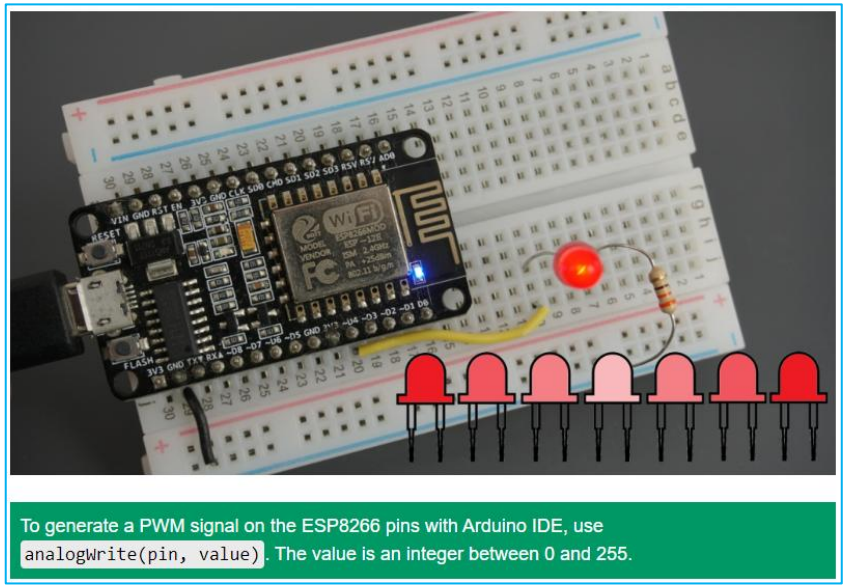
Successful Test using:
GND = buzzer gnd
D5 = buzzer positive

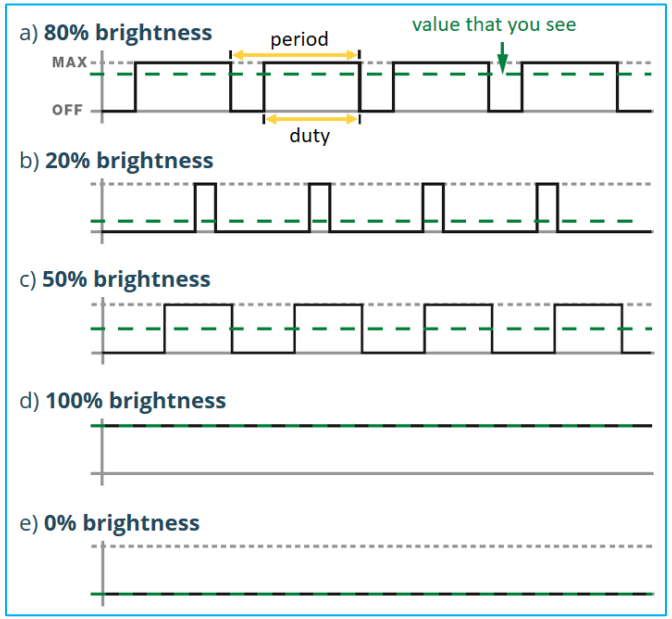
Notes: no resistor in series with buzzer pin, as found in some online examples.



PWM control demo (in progress)

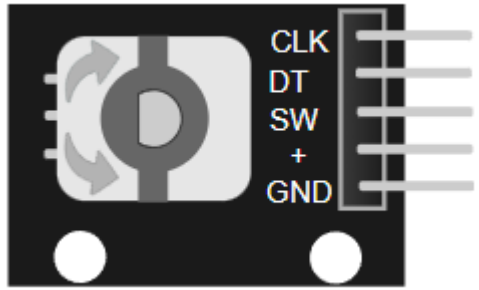
- From RandomNerdTutorials
- https://randomnerdtutorials.com/esp8266-pwm-arduino-ide/
- output range of analogWriteFreq() is from 100Hz to 40,000Hz
- common PC fan range of operation is 21-28kHz, spec is 25kHz
- use analogWrite(pin, value) to command PWM output.

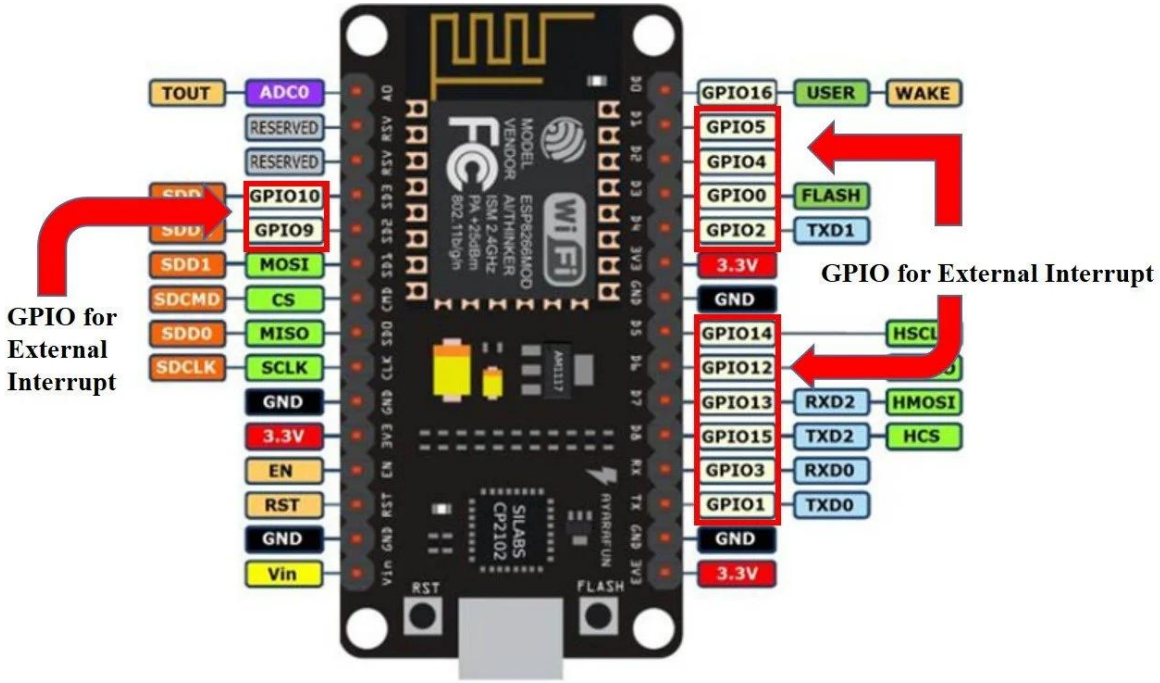




Rotary Encoder (in progress)

- encoder PN: KY-040
- reference about interrupts here (https://microcontrollerslab.com/esp8266-interrupts-timers-arduino-ide-nodemcu/)
- example for encoder here https://docs.wokwi.com/parts/wokwi-ky-040





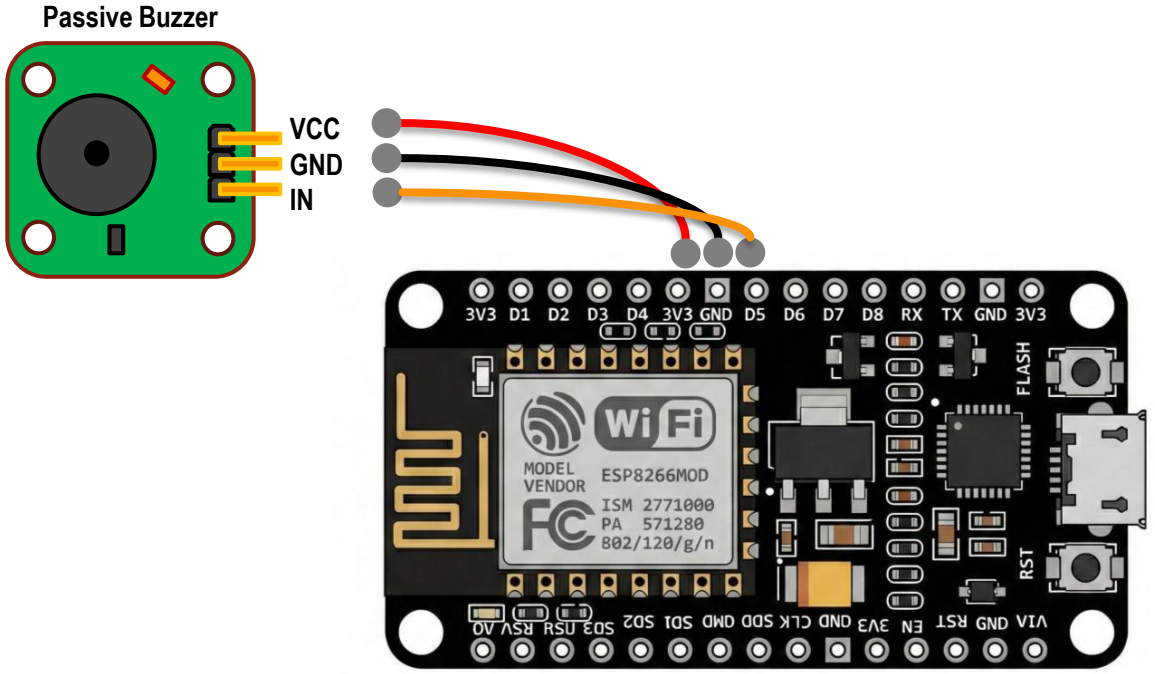
Name	Description
CLK	Rotary encoder pin A
DT	Rotary encoder pin B
SW	Push button pin. Normally open, shorted to GND on press
VCC	Voltage supply
GND	Ground

- pins that support interrupts

ESP8266MOD | buzzer

Planned test: 08.11mqtt_BUZ_v3
Using hardware: BTE16-13P

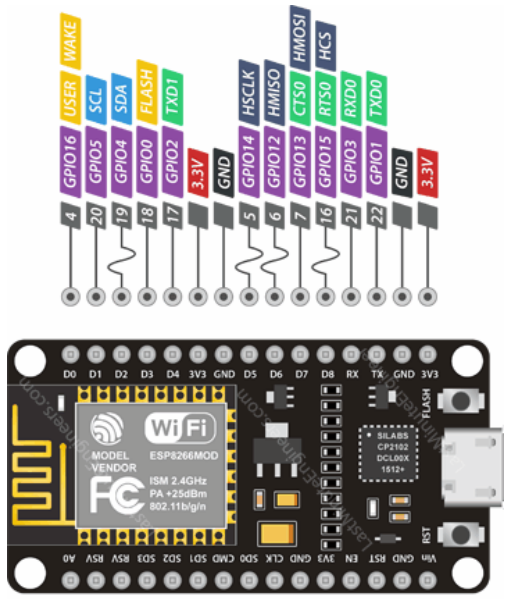
Successful Test using:
GND = buzzer gnd
D5 = buzzer positive
(can use D3, D4, or D5?)

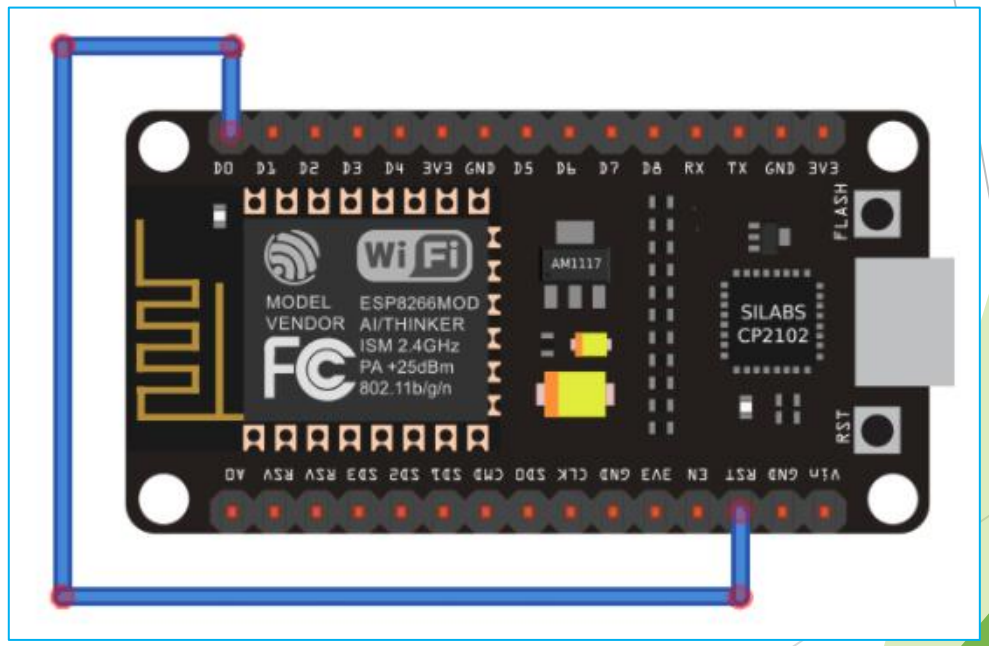


ESP8266MOD | SLEEP

Purpose: test a code that sets ESP8266 to sleep mode for low power consumption.
Wiring: must connect pin 16 to reset pin.

Tutorial at RandonNerdTutorials

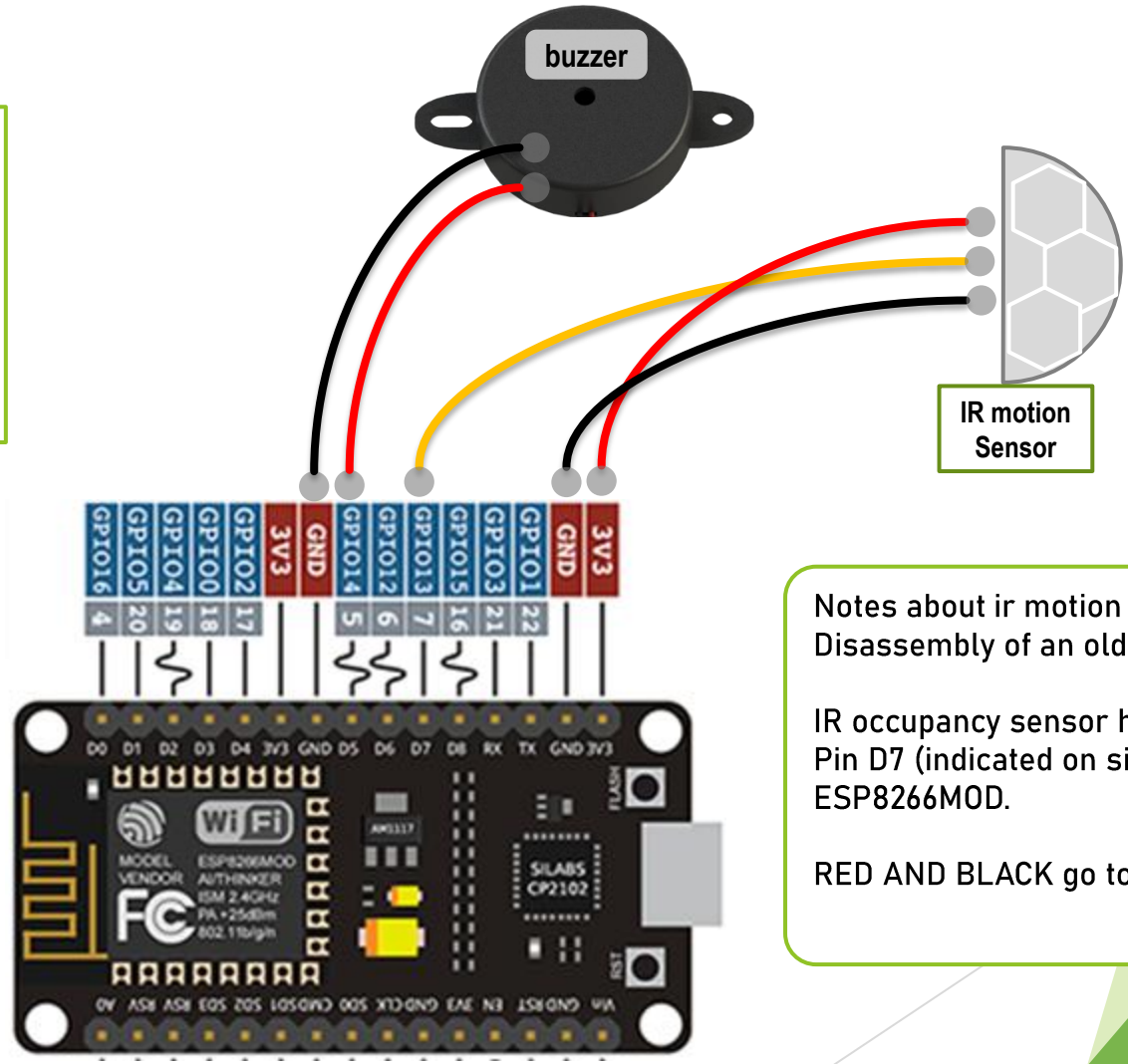




ESP8266MOD | buzzer + Motion Sensor test

Purpose: configure the device to sense motion with IR sensor & report motion with a buzzer.
Test using:
GND = buzzer gnd
D5 = buzzer positive

D7 = Motion Sensor Output



esp8266mod

Notes about ir motion sensor setup:
Disassembly of an old prototype 2020.09.22

IR occupancy sensor has signal connected to Pin D7 (indicated on silkscreen) on ESP8266MOD.

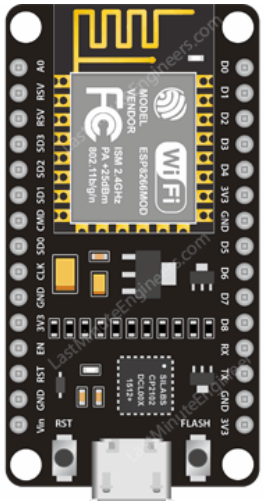
RED AND BLACK go to 3.3 and GND

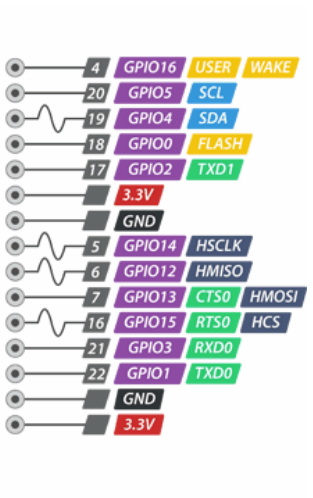
ESP8266MOD DC motor driver Test

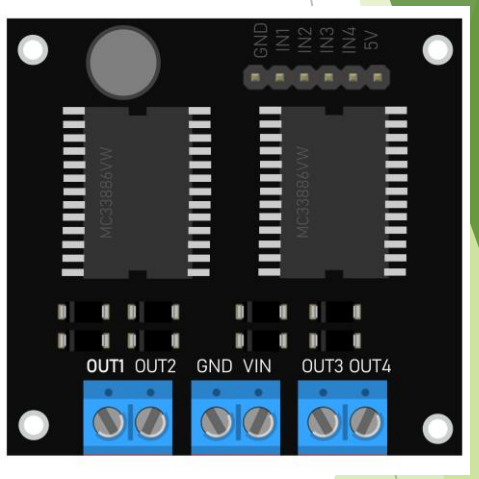
2020.10.04 Test results:
Goal: 1 motor, 1 direction speed control

GPIO	Assigned
4	= IN1 (PWM always=0)
14	= IN2
12	= IN3 (PWM always=0)
15	= IN4

Good starting example at LastMinuteEngineers.com

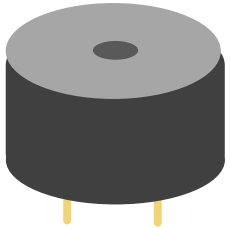




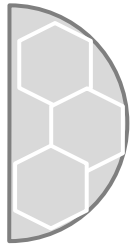


Graphics Slide

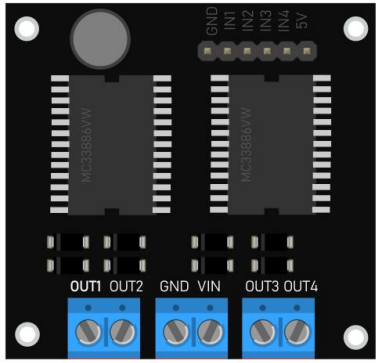
- for modifying vector graphics as needed.



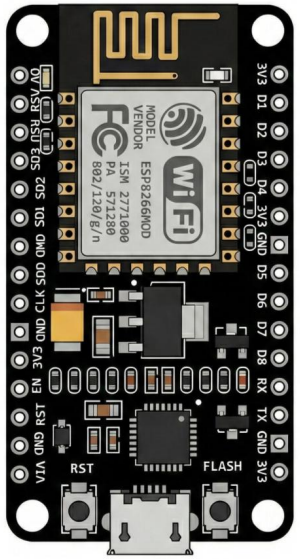
Buzzer, piezo



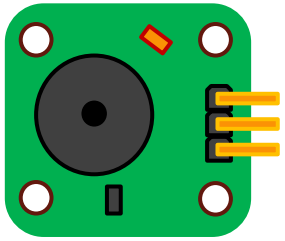
PIR motion sensor



dual motor driver

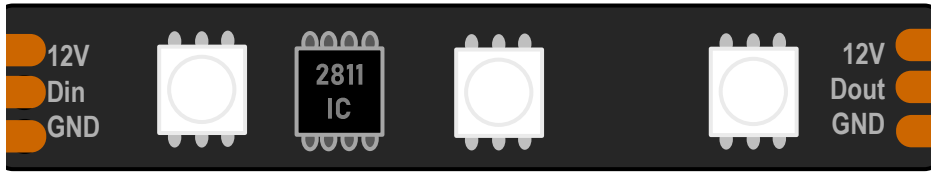


esp8266 board



Passive buzzer board

Light sensor



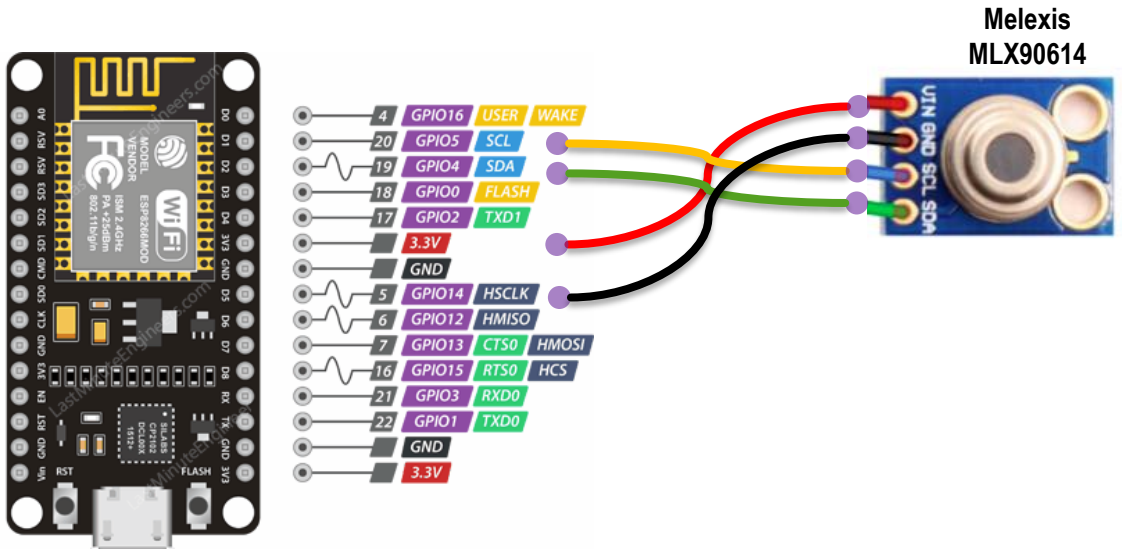
LED strip

ESP8266MOD | gy-906 MLX90614ESF

2020.06.25 Test results:
Goal: get ambient and surface measurements in degrees C

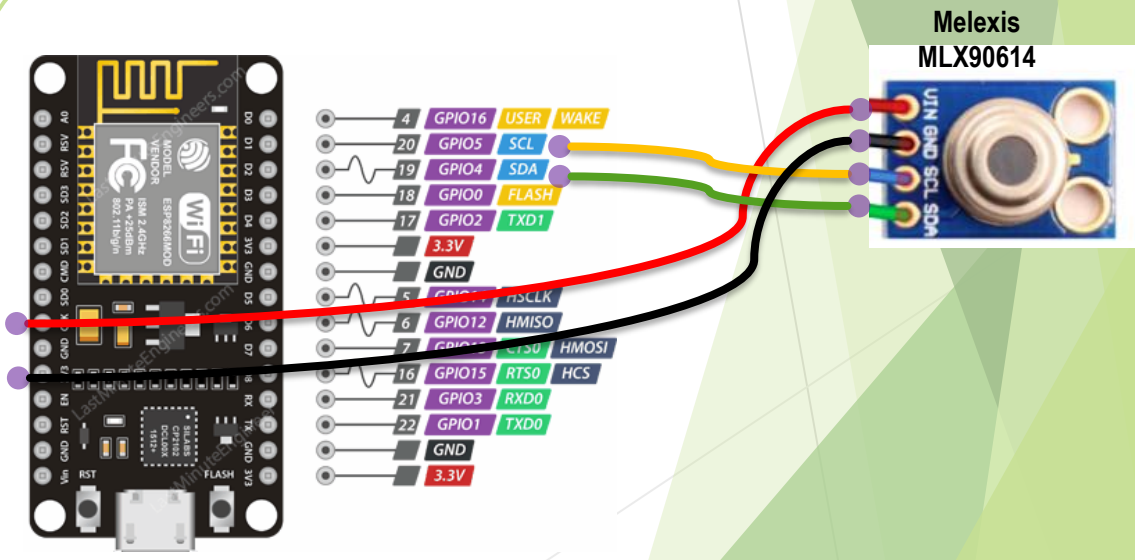
Successful first communications.

Software: 06.25testMLX.ino



2020.08.18 Test results:
Goal: change GND pin to leave this D6 Pin open

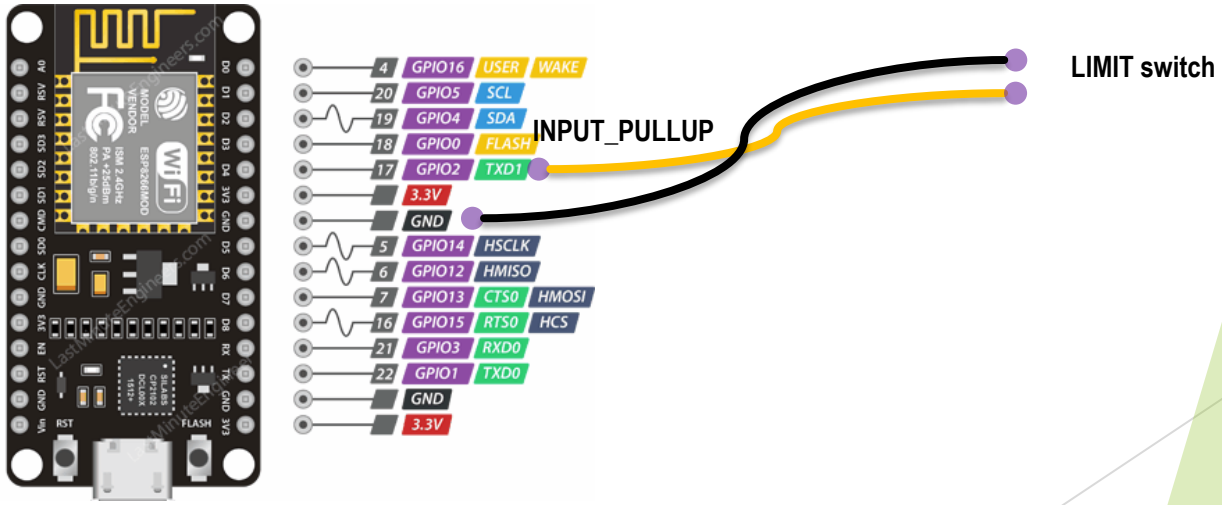
Result: cannot flash new programs while CLK (pin 14) is in connected to the MLX sensor.



ESP8266MOD | Limit switch

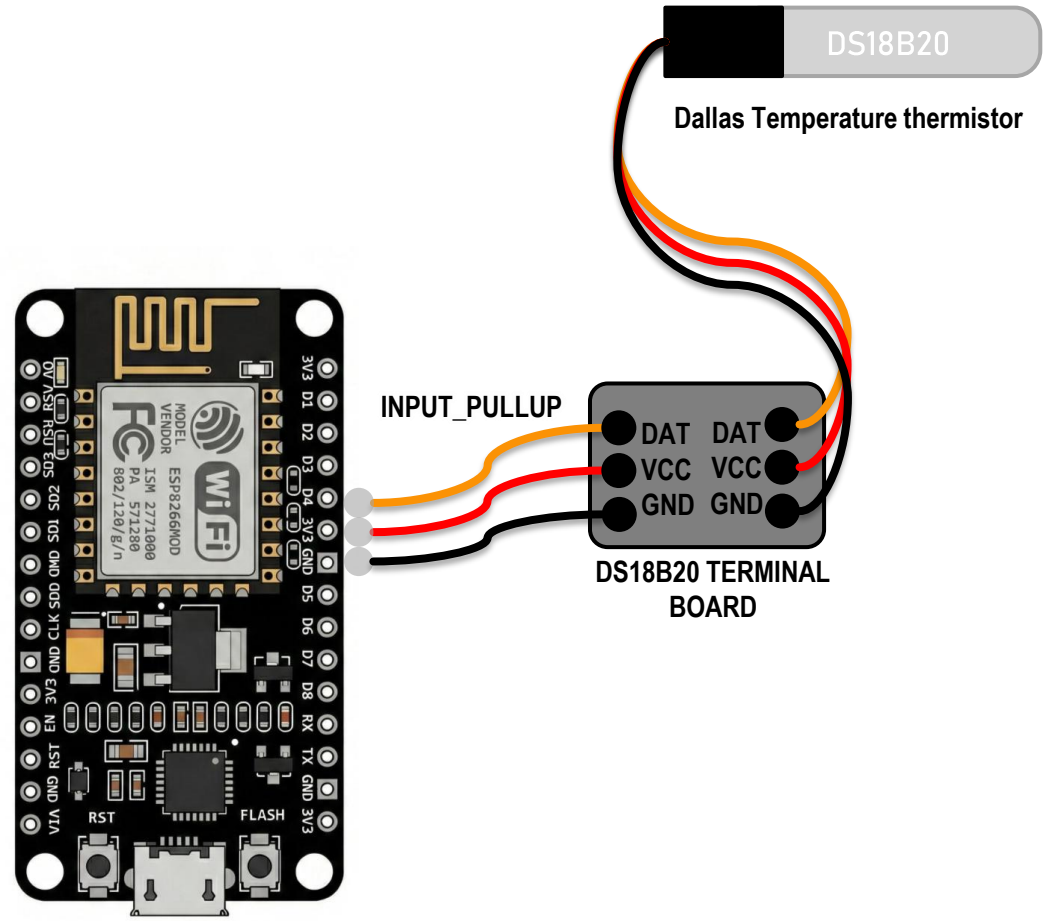
The simplest test to validate software which defines an input pin and detects a rising/falling voltage input.

Purpose: configure pin with an internal pullup resistor to detect the switch of a limit switch.



Thermistor DS18B20

The most popular temperature sensor for makers is the thermistor, DS18B20
The most popular DS18B20 offerings include a steel/plastic housing and 3-wire leads in a round cable.



2021.08

Followed tutorial at lastMinuteEngineers.com

- Used accessory board instead of 4.7k resistor
- Used 3.3v instead of 5v
- board connections: D4, 3V3, GND

Multiple Float Switches, relays

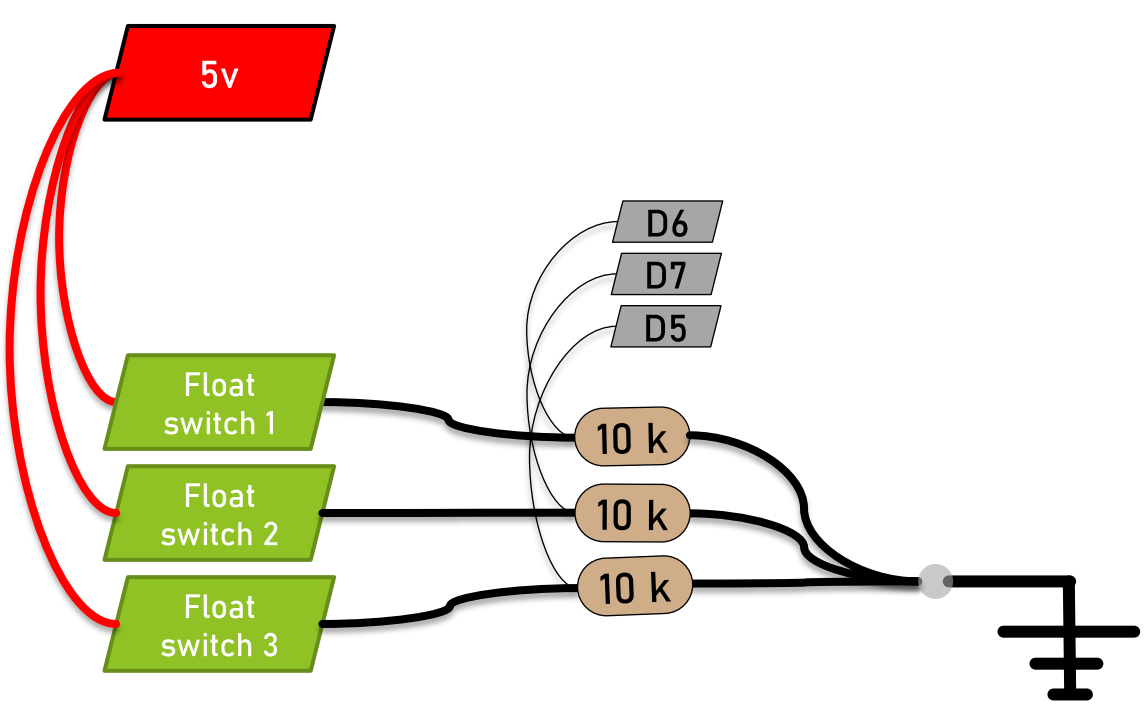
2021.03.23

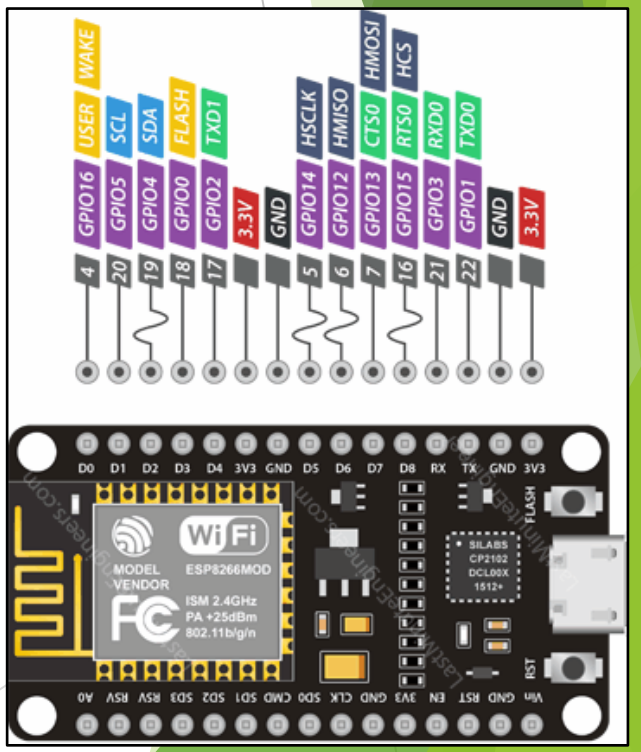
Sketch: 03.08valveTrial

GPIO Assigned
12 = FS1
13 = FS2
14 = FS3

Switch conditions: Normally LOW, active HIGH

Best Pins selection





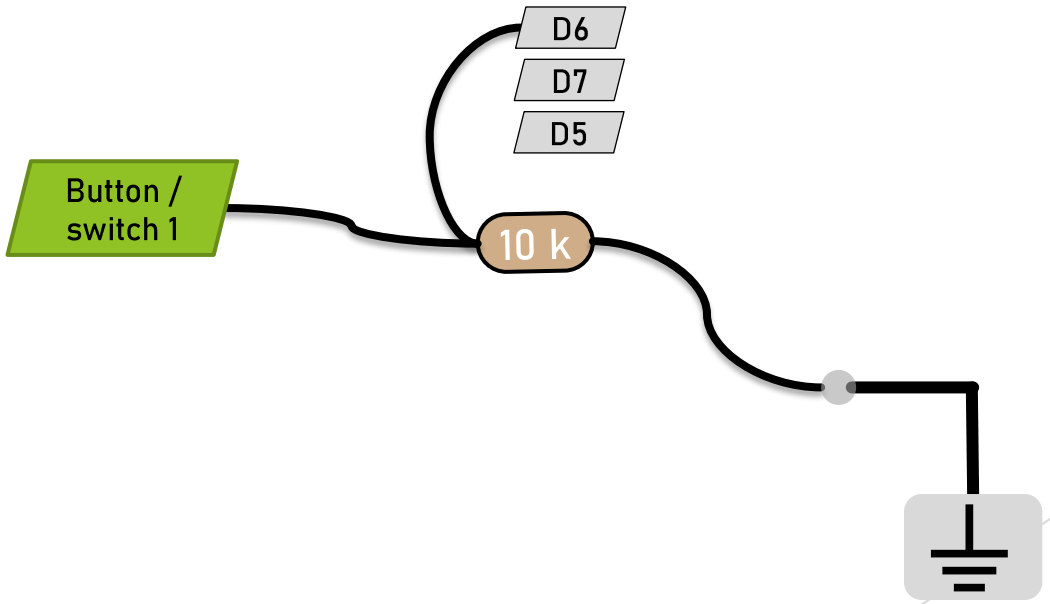
ESP8266MOD | simple button w 3.3v LED

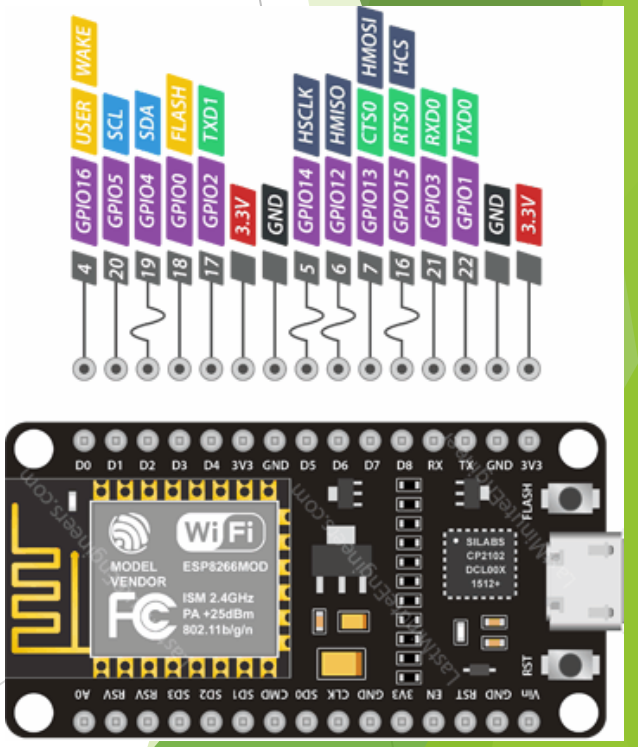
Integrating one button to verify input in code. The button is most often normally open. The simplest integration includes a pullup (or pulldown?) resistor connecting to one leg of the button. Button-press gives closed circuit and input moves (to LOW state?)

2021.09

Sketch: tbd

⏚

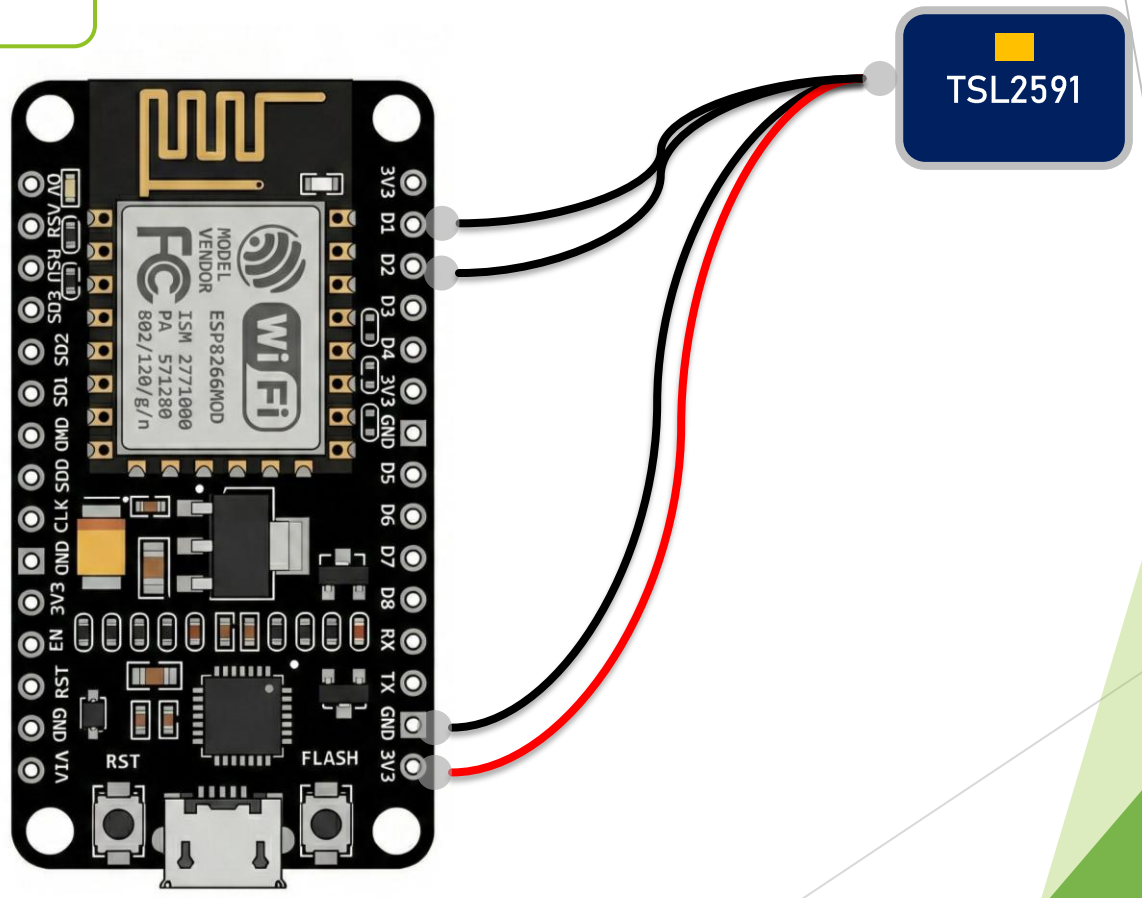




Ambient Light Sensor

2021.03.23
MCU: ESP8266
Purpose: Get Samples from ambient light sensor TSL2591
Sketch: UPDATE HERE

[PINS]
D1
D2
GND
3.3V

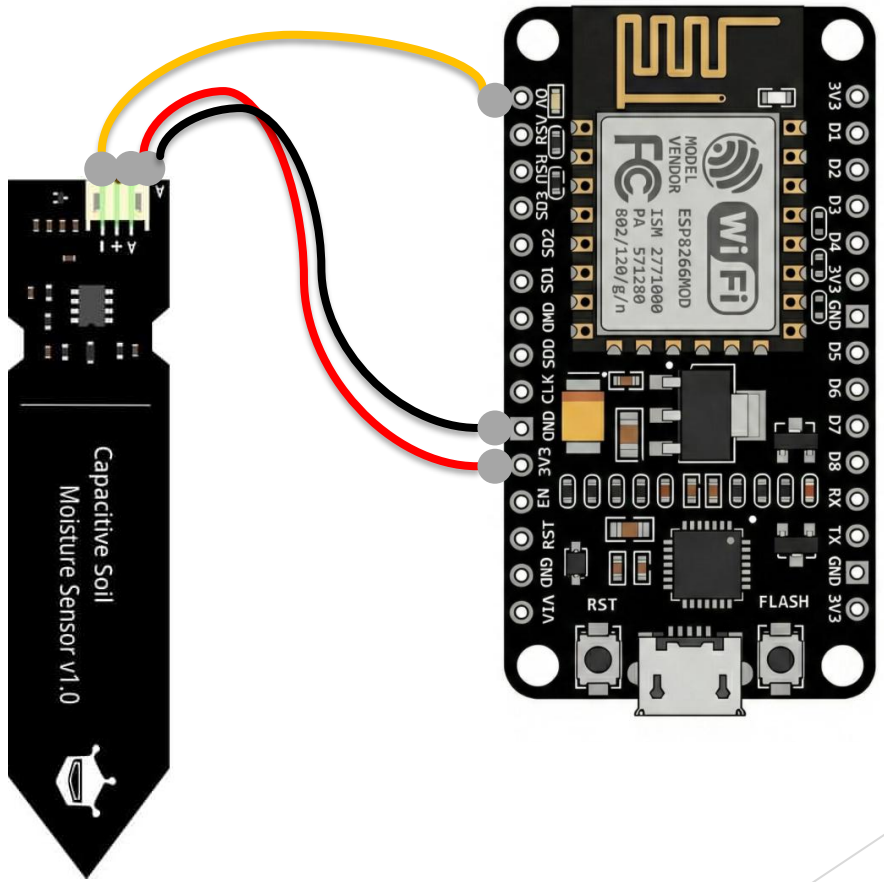


Soil moisture sensor v1.2

Purpose: test and characterize the soil moisture meter.
Simple Example for (using analog pin A0)

- Pins:
- GND
- 3.3V
- A0 (analog)

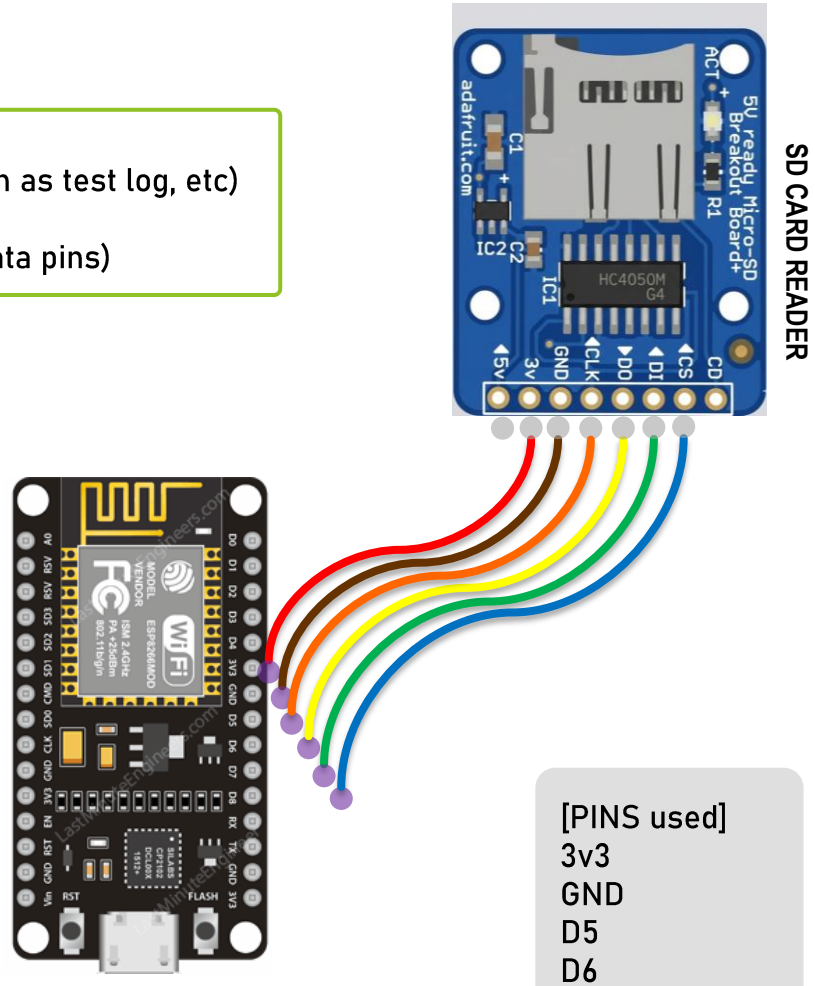
Process: Map the analog voltage from pure water to pure air.
Tested 2021.09 successfully



Find Example: https://how2electronics.com/interface-capacitive-soil-moisture-sensor-arduino/

SD card reader

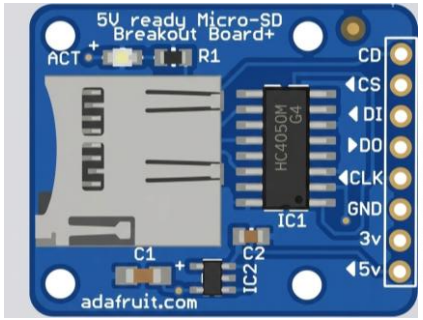
Test, 2021
►Purpose: attempt to save and retrieve files of string data (such as test log, etc) on the SD card.
►Pins Used: 3.3v, GND, and D5 thru D8 (corresponding to SPI data pins)



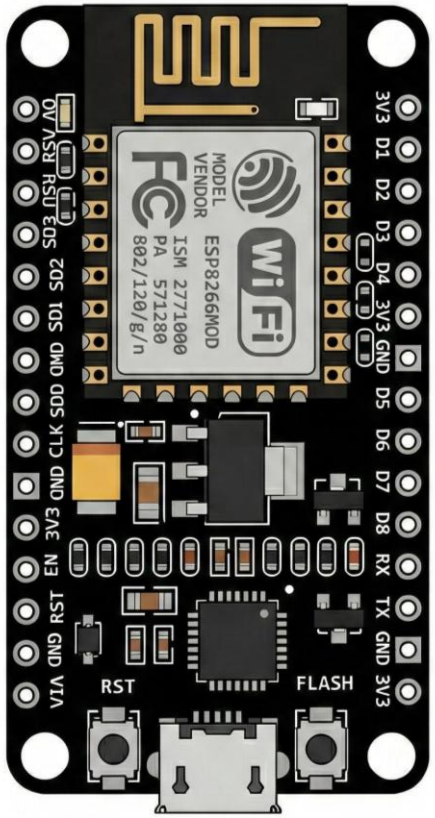
[PINS used]
3v3
GND
D5
D6
D7
D8

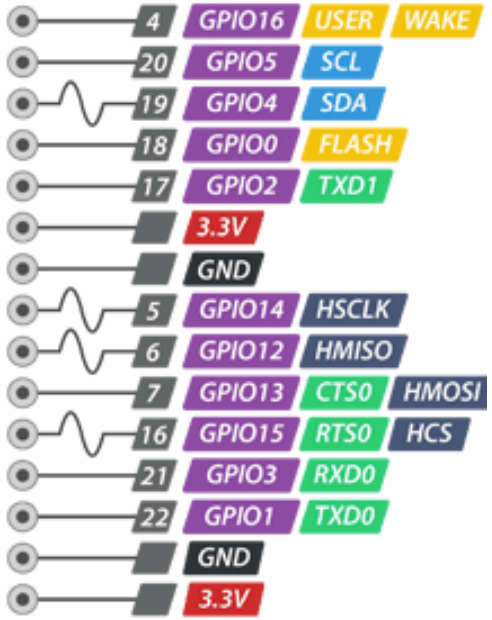
SD Card Reader (2)

- identification of names for each pin, on the popular SD card reader board.
- the ESP has two sets of SPI pins, and our selected set is on the right-hand as shown.
- lastminuteengineers designates these as HSCLK instead of SCLK



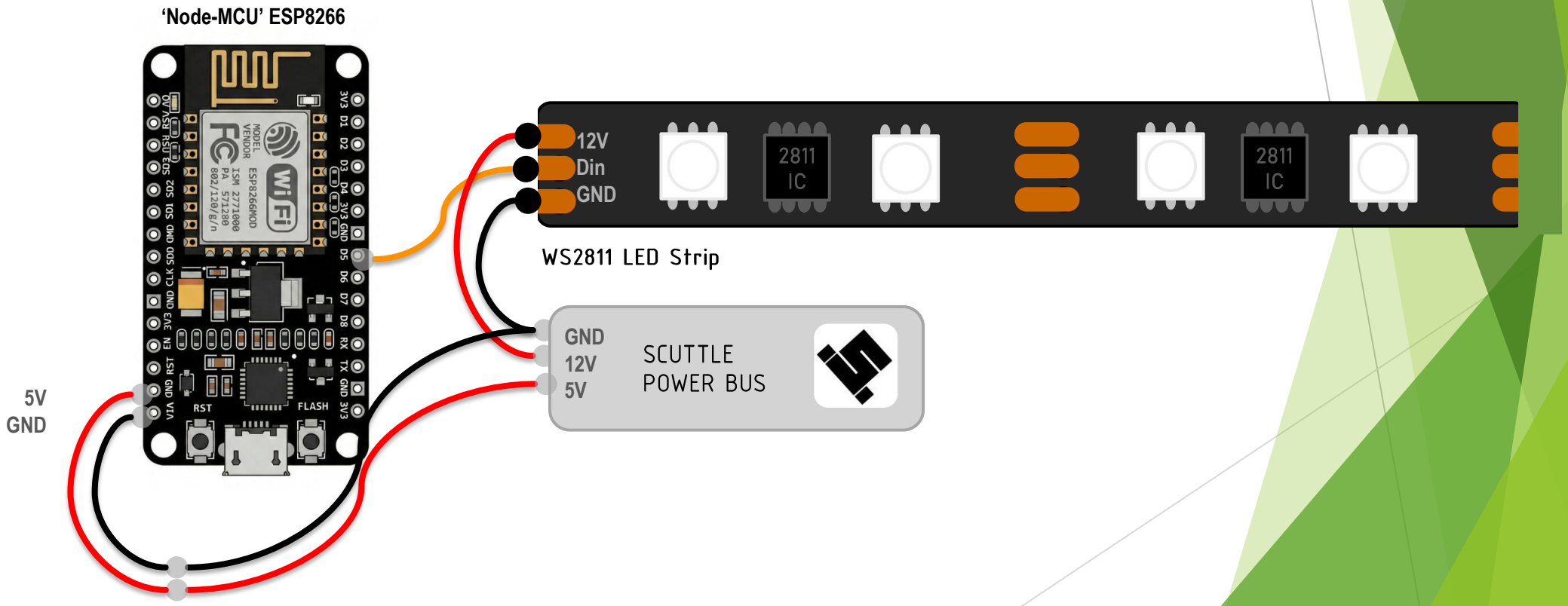
CS
(MOSI) DI
(MISO) DO
CLK
GND
3V
5V





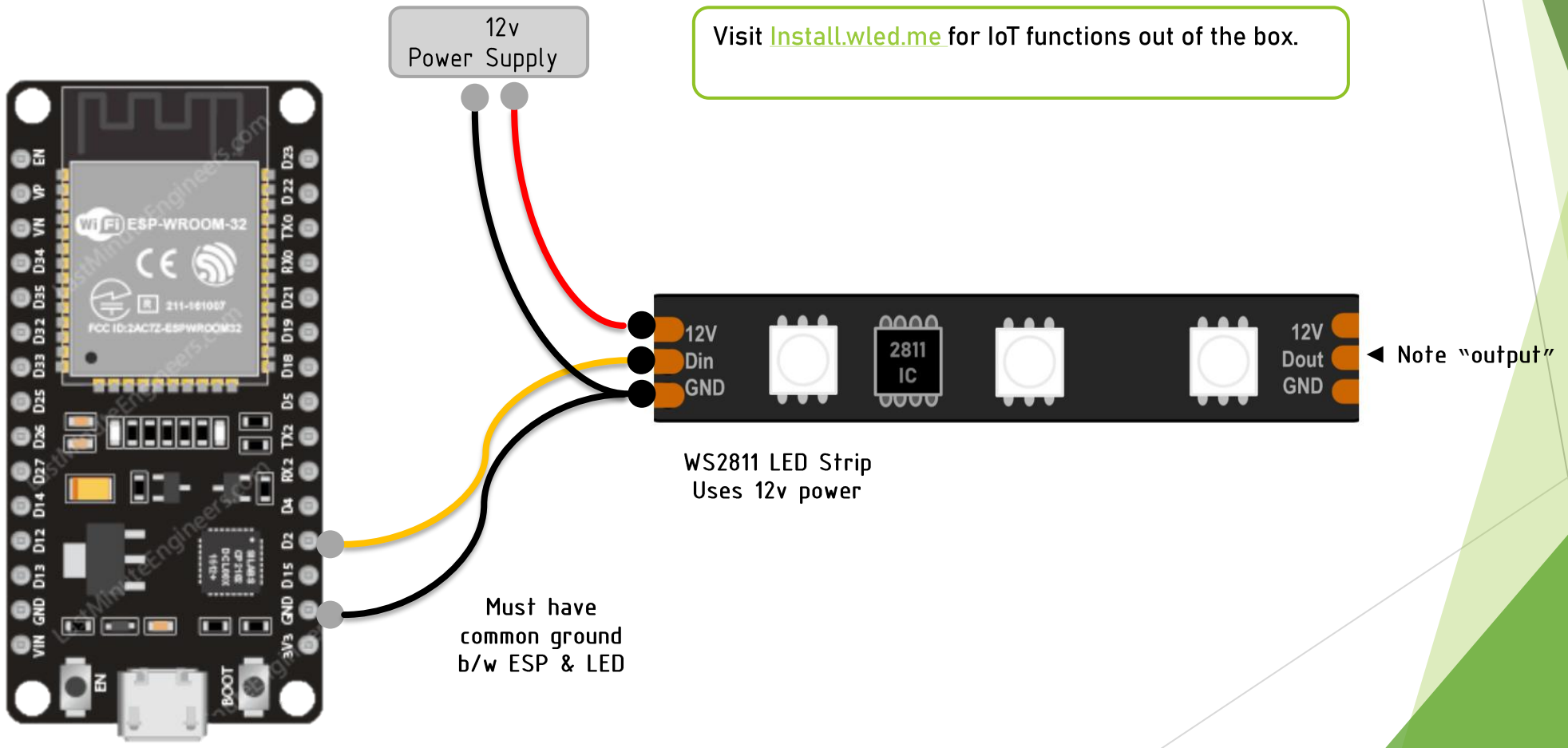
'neoPixel' WS8211

2021.12 Updated Notes
This test is for driving addressable LED strip with the ESP8266MOD, and neopixel library for Arduino.
Example Video on Youtube
Example Product on Amazon (WS2811)
Example Code on Github based on https://fastled.io/



RGB lights WS2811

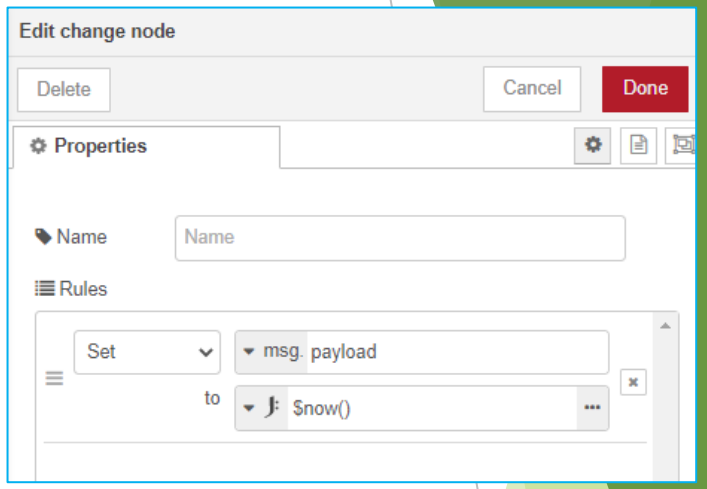
The general configuration for power and communication to the programmable RGB lights with MCU. This setup uses ESP32 for the MCU.
Tested in 2021.12



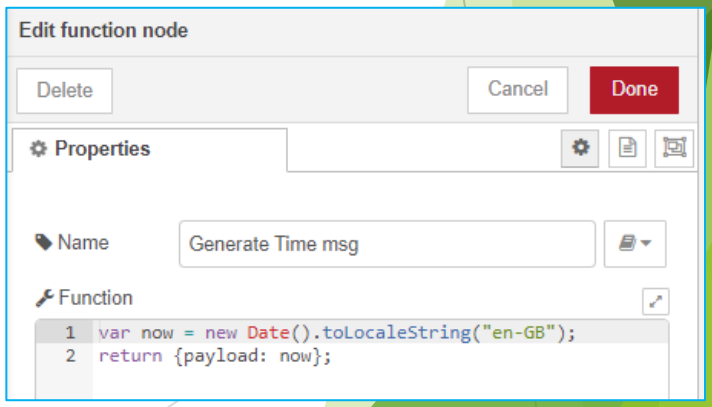
Capture time from NodeRed

Purpose: get the device to connect to wifi and retrieve a timestamp from a nodered script operating on a web server. For keeping true time on the edge device.

09/02/2021



2021-09-02T16:34:26.940Z

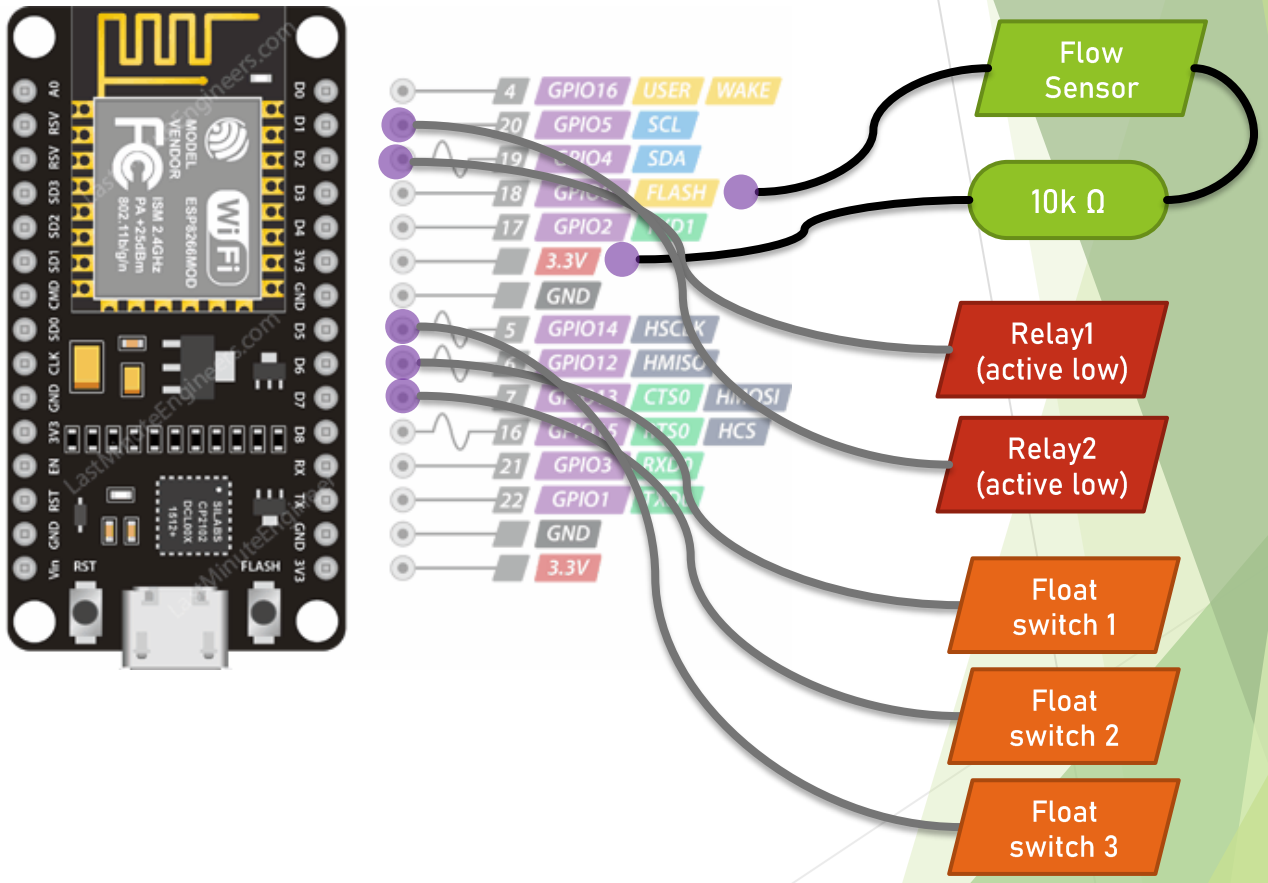


9/2/2021, 11:36:17 AM

ESP8266MOD | float switches, relays, flow sensor

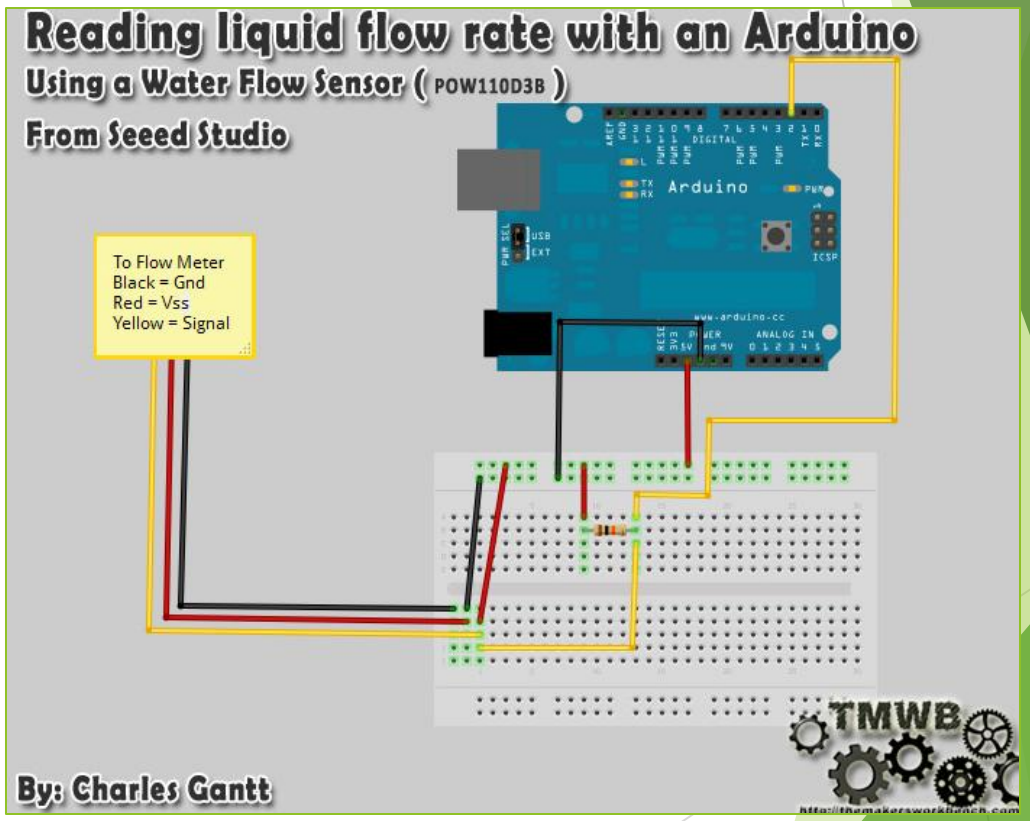
[2021.03] The following slides relate to a hydro-control system, beginning with a flow-sensor. Float switches are included for a water reservoir (low, medium, and high levels)

Float switch connections:
- ► normally open switch
- ► one terminal tied to GND
- ► other terminal connected as shown
- ► pins set to input_pullup condition



Wiring suggestion from Arduino Forum

- Link to the topic here
- 5v is supplied to the sensor
- 5v pullup resistor on the signal line
- Using 10k resistor



Best Pins to Use – RandomNerdTutorials

- Link to the writeup from RandomNerdTutorials

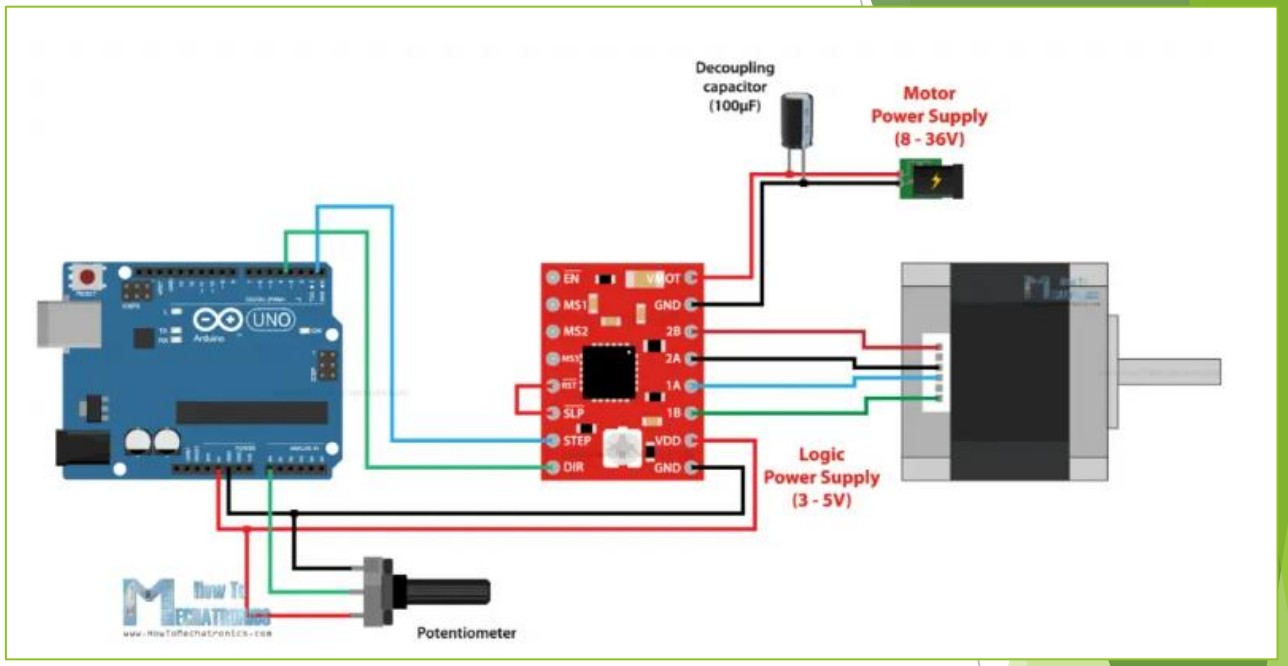
Label	GPIO	Input	Output	Notes
D0	GPIO16	no interrupt	no PWM or I2C support	HIGH at boot used to wake up from deep sleep
D1	GPIO5	OK	OK	often used as SCL (I2C)
D2	GPIO4	OK	OK	often used as SDA (I2C)
D3	GPIO0	pulled up	OK	connected to FLASH button, boot fails if pulled LOW
D4	GPIO2	pulled up	OK	HIGH at boot connected to on-board LED, boot fails if pulled LOW
D5	GPIO14	OK	OK	SPI (SCLK)
D6	GPIO12	OK	OK	SPI (MISO)
D7	GPIO13	OK	OK	SPI (MOSI)
D8	GPIO15	pulled to GND	OK	SPI (CS) Boot fails if pulled HIGH
RX	GPIO3	OK	RX pin	HIGH at boot
TX	GPIO1	TX pin	OK	HIGH at boot debug output at boot, boot fails if pulled LOW
A0	ADC0	Analog Input	X	

Stepper Motor

Original Demo
- ►Video explanation HowToMechatronics_stepperMotor
- ►Written guide StepperMotorsGuide
- ►12v for the motor driver input
- ►DIR/STEP/ENA controls motor with 3 pins
- ►Using 10k resistor

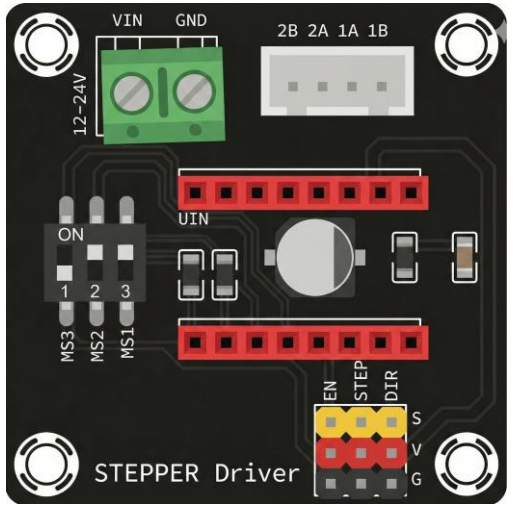
2025.08 Demo
Convert the example to use the now-available motor driver breakout board. Use ESP8266 to produce step signals for the motor.

Jeanoko DRV8825/A4988 Stable 42 Stepper Motor Driver Expansion Board



Example Motor Setup

Expansion Board (2025)

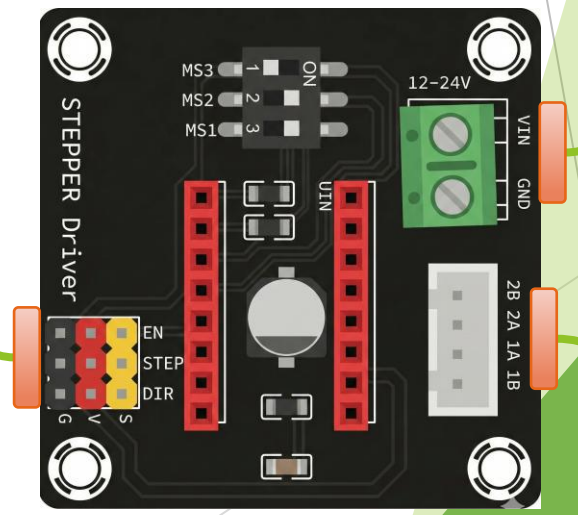


PC Fan Tachometer

- To test a stepper motor, we need an oscillating signal on the STEP input.
- We can generate a square wave using a common PC fan.
- Then, we can eliminate the microcontroller from this demonstration.

Fan connector Hookup:
1 GND – scope ground
2 5V – connect pullup resistor to tach
3 Tach – scope signal
4 NC – no connection

tachometer signal



12v DC supply

stepper motor

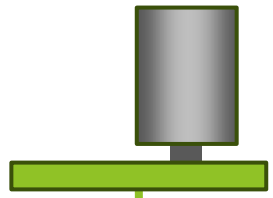
Rotary Knob

Purpose: adapt the trial using fan tachometer to deliver step inputs from a rotary knob to the stepper driver.

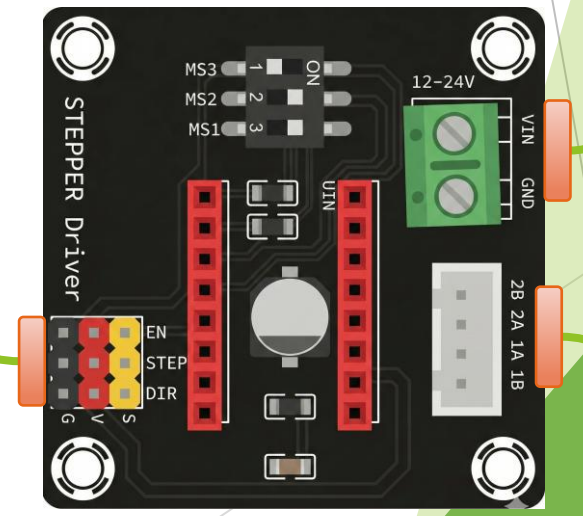
- We can generate a square wave using a common PC fan.
- Then, we can eliminate the microcontroller from this demonstration.

Fan connector Hookup:
1 GND – scope ground
2 5V – connect pullup resistor to tach
3 Tach – scope signal
4 NC – no connection

rotary encoder
KY-040



pulse signal



12v DC supply

Coil 1 = (1A, 1B)
Coil 2 = (2A, 2B)

stepper motor