The ESP32 devkit

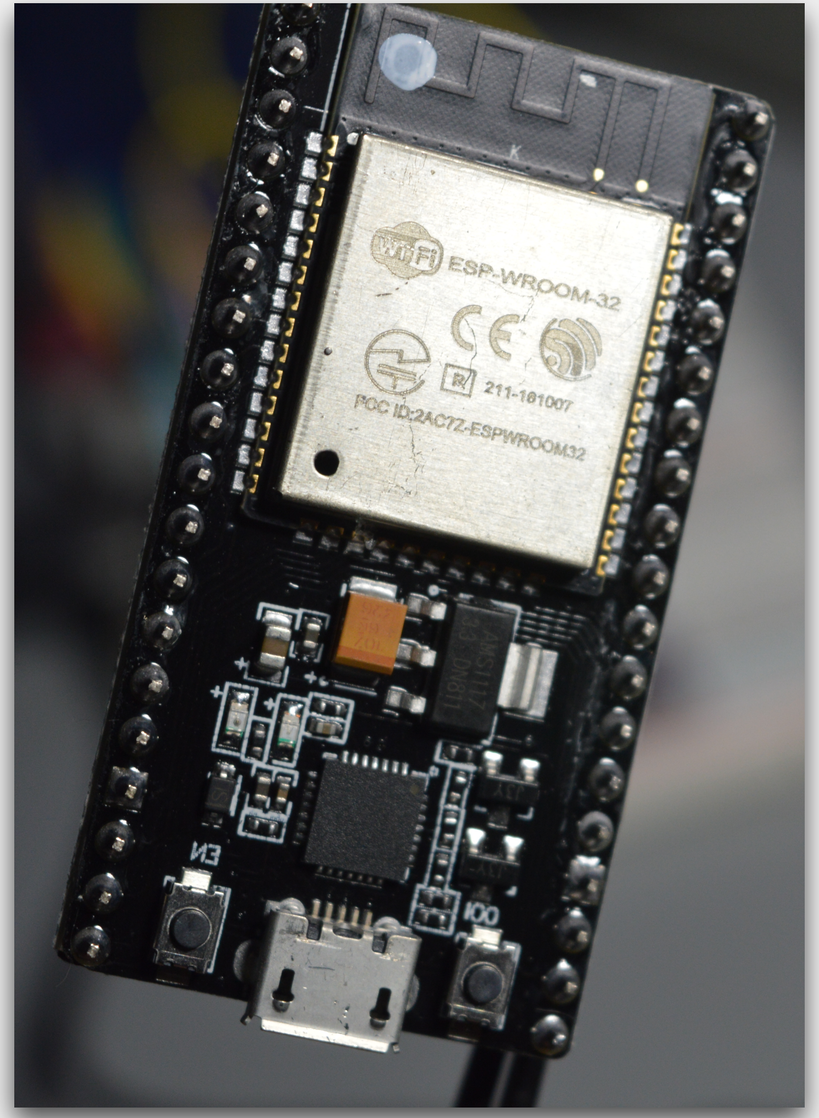
Notes for Makers

Working with the ESP32?

You'll love these notes :-)

What's inside:

1. The ESP32 module
2. The ESP32 Devkit
3. ESP32 vs Arduino (a comparison)
4. ESP32 GPIOs (includes GPIO map)
5. ESP32 communications
6. ESP32 devkit power options
7. ESP32 courses from Tech Explorations



The ESP32 module

A close-up look

The “ESP32” is a reference name to a variety of boards and modules based on the core ESP32 chip.

For example:

- ESP32-WROOM-32 module contains the ESP32-D0WDQ6 chip
- ESP32-WROOM-32D module contains the ESP32-D0WD chip
- ESP32-WROVER-IB module contains the ESP32-D0WD but with added PSRAM
- etc.

The “ESP32” is a reference name to a variety of boards and modules based on the core ESP32 chip.

Each module and chip combination has a unique set of characteristics.

- Amount of flash memory (typically 4MB)
- Presence and amount of PSRAM (pseudo-static RAM[1])
  - 8 MB Available in WROVER modules
- Type of antenna
  - MIFA: Meandered Inverted-F Antenna[2]
  - U.FL: antenna connector for an external antenna
- Number of processing cores
  - Chips with “D” after “ESP32” denote dual core
  - Chips with “S” after “ESP32” denote single core

1 | https://en.wikipedia.org/wiki/Dynamic_random-access_memory#PSRAM
2 | https://en.wikipedia.org/wiki/Inverted-F_antenna
3 | https://docs.espressif.com/projects/esp-idf/en/latest/hw-reference/modules-and-boards.html#wroom-solo-and-wrover-modules

ESP32-WROOM-32

The most commonly used module.

- Contains the **ESP32-D0WDQ6** chip
- **4 MB Flash** (some variants go up to 16MB)
- No PSRAM
- **MIFA** antenna

ESP32-WROOM-32D
ESP32-WROOM-32U

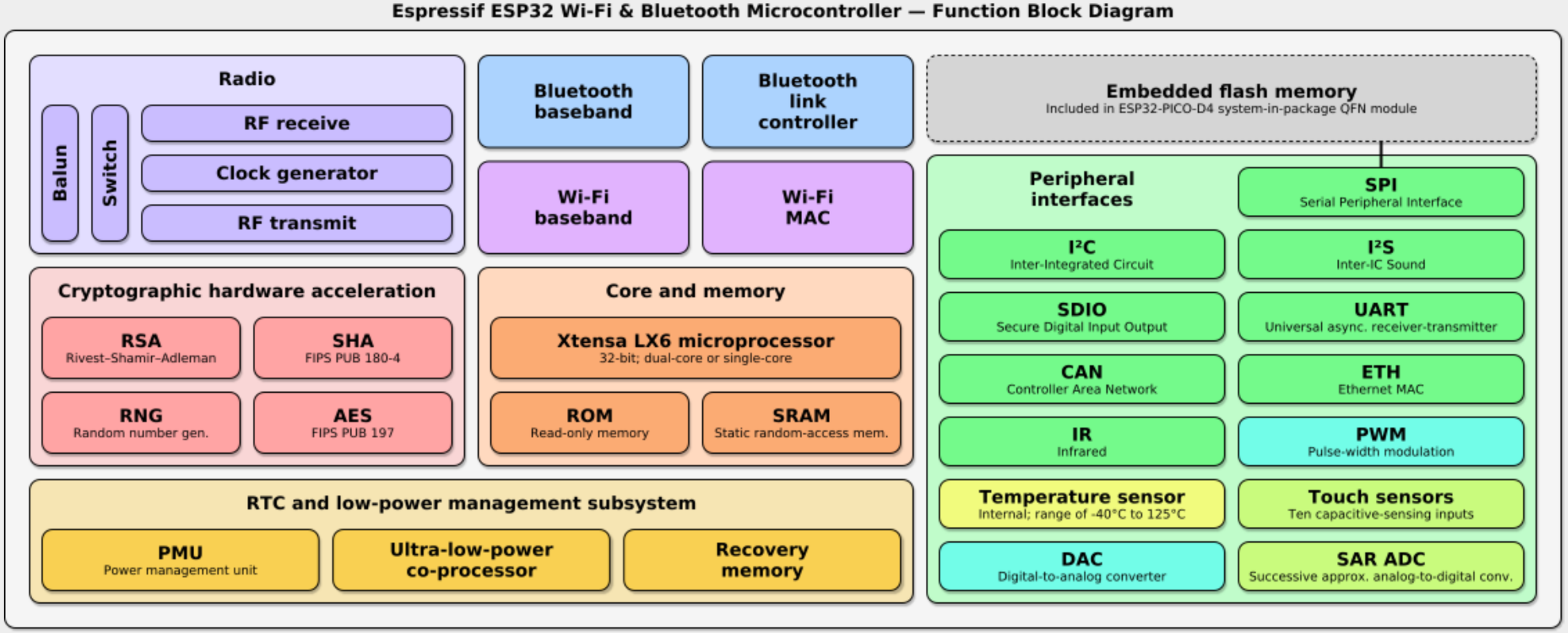
- Contains the **ESP32-D0WD** chip
- **4 MB Flash** (some variants go up to 16MB)
- No PSRAM
- **MIFA** antenna for the “D” model
- **U.FL** antenna connector for the “U” model
- Smaller footprint than the ESP32-WROOM-32

ESP32-WROVER

More powerful compared to the WROOM models

- **ESP32-WROVER** and **ESP32-WROVER-I** use the **ESP32-D0WDQ6** chip (same as ESP32-WROOM-32)
- **ESP32-WROVER-B** and **ESP32-WROVER-IB** use the **ESP32-D0WD** chip (same as ESP32-WROOM-32D and U)
- **4 MB Flash** (similar to WROOM modules)
- **8 MB SPI PSRAM** (WROOM have none)
- **MIFA** or **U.FL** antenna
- Depending on the model, can operate at **1.8V**, and up to **144MHz** clock speed

Focus: ESP32-D0WDQ6



1 | https://en.wikipedia.org/wiki/ESP32#/media/File:Espressif_ESP32_Chip_Function_Block_Diagram.svg

ESP32 module common features

All ESP32 modules share these features (only a summary):

- CPU cores (one or two)
- Internal memory (ROM, SRAM)
- External SRAM
- Timers and watchdogs
  - Four general-purpose 64-bit timers
  - Three watchdog timers (used to recover from faults)
- RTC clock
- 2.4 GHz receiver and transmitter radio
- Wifi, 802.11 b/g/n
- Bluetooth, classic and BLE
- RTC (co-processor) and Low-Power management with multiple power modes.
- 34 GPIO pins
- Analog to Digital Converter (ADC)
- Hall Sensor, capable to detect a magnetic field without additional hardware
- Digital to Analog Converter (DAC)
- Touch sensor via 10 capacitive-sensing pins.
- Ethernet MAC interface.
- SD/SDIO/MMC host controller
- SDIO/SPI slave controller
- UART
- I^2C
- Infrared Remote Controller
- Pulse Counter
- Pulse Width Modulation (PWM)
- LED PWM
- SPI
- Hardware acceleration of algorithms such as AES, RSA and ECC

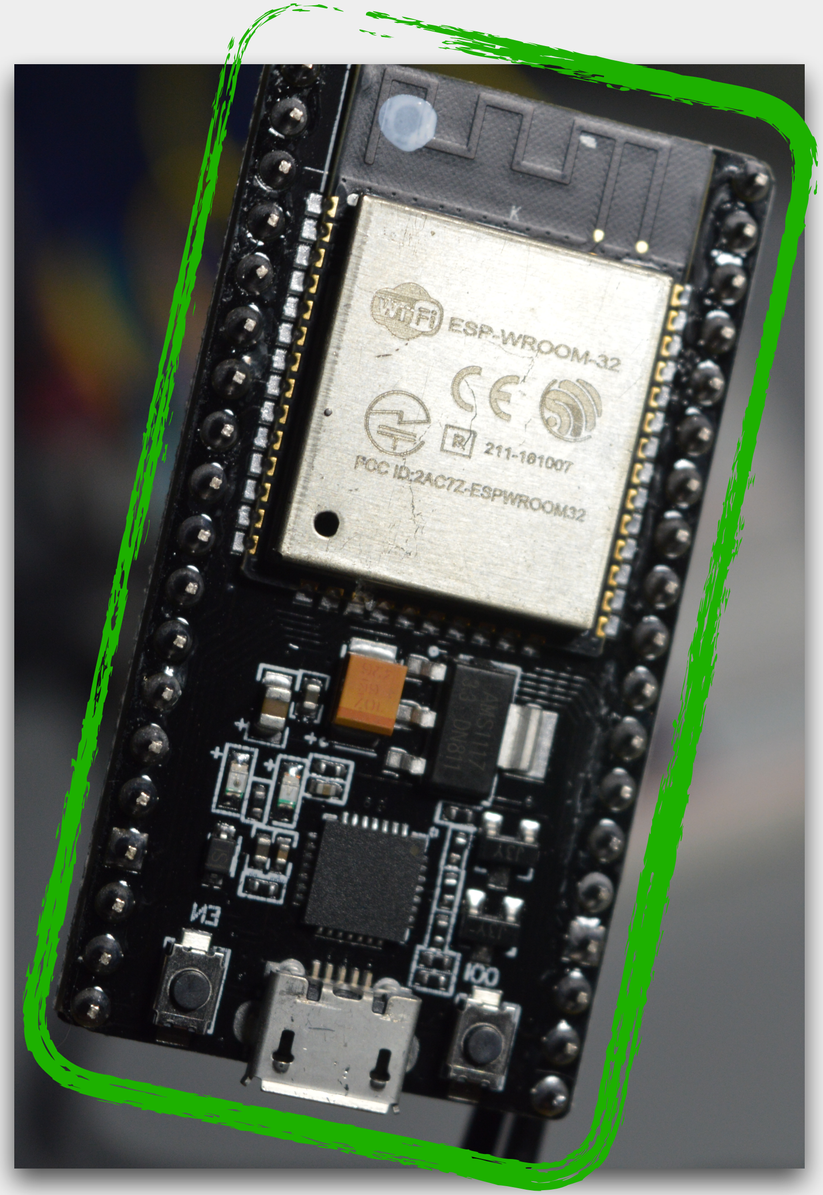
ESP32 DevKitC V4

ESP32 modules require supporting hardware

Hardware like:

- A USB to serial programming interface
- A power subsystem
- Pushbuttons for reset and setting the upload mode
- Indicator lights
- And more

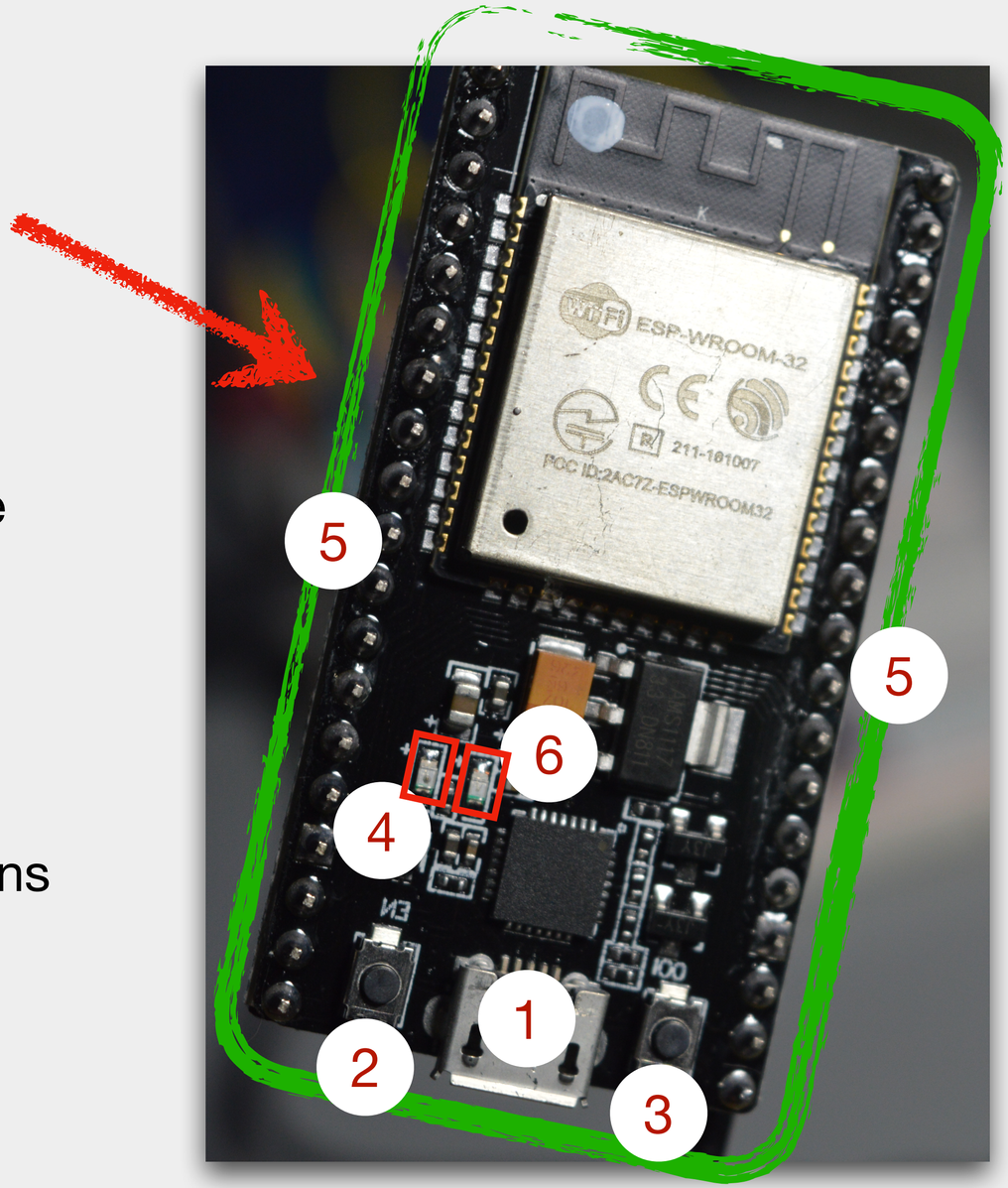
Various boards implement ESP32 development kits with a variety of features.



ESP32 DevKitC V4

Supports the ESP32-WROOM-32 module with:

1. A micro USB port to serial programming interface
    a. Also provides power
2. Pushbutton for reset (“EN”)
3. Pushbutton to enable firmware download mode (“BOOT”)
4. Power on LED
5. Two rows of headers that breakout the module pins
    a. Compatible with regular breadboards 👍
6. A programmable LED (attached to GPIO2)

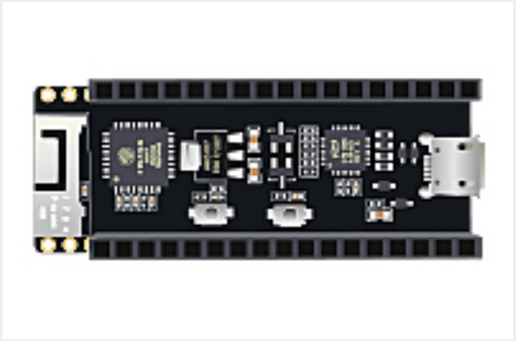


Many other ESP32 boards...

Just some examples

ESP32-LyraTD-MSC

ESP32-PICO-KIT



ESP-WROVER-KIT

ESP32-LyraT

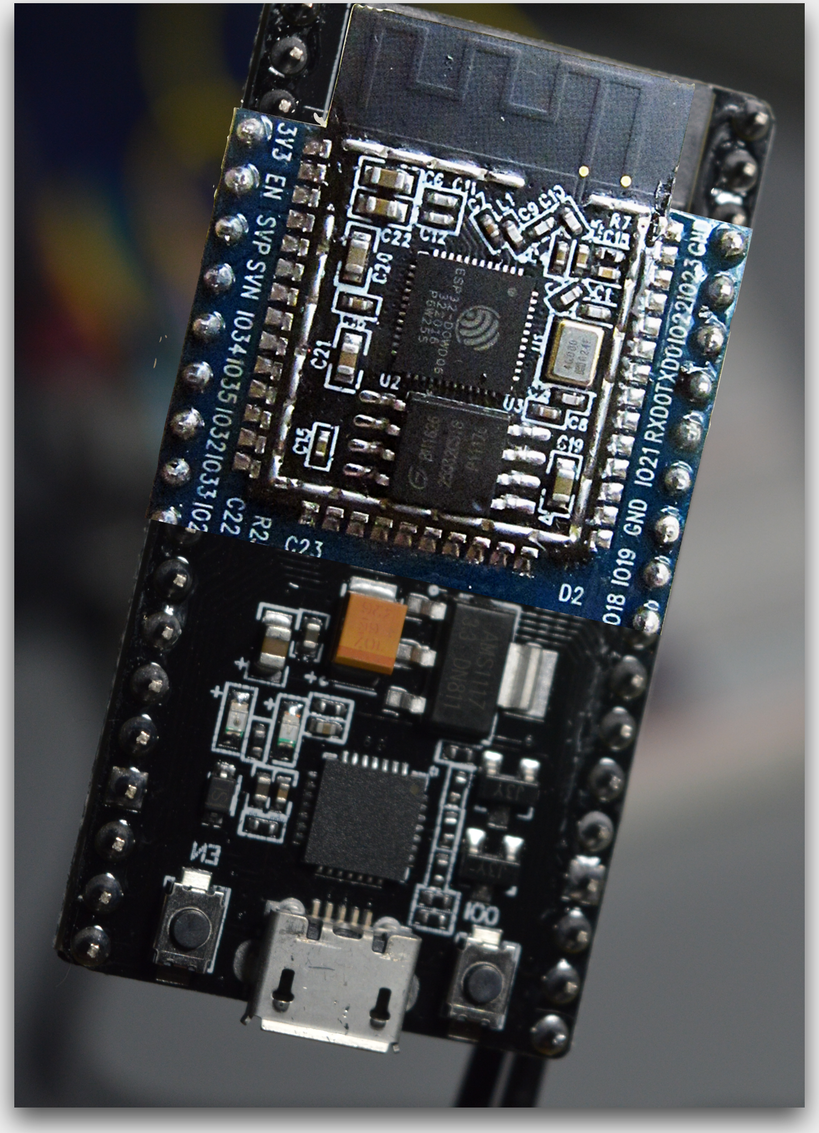
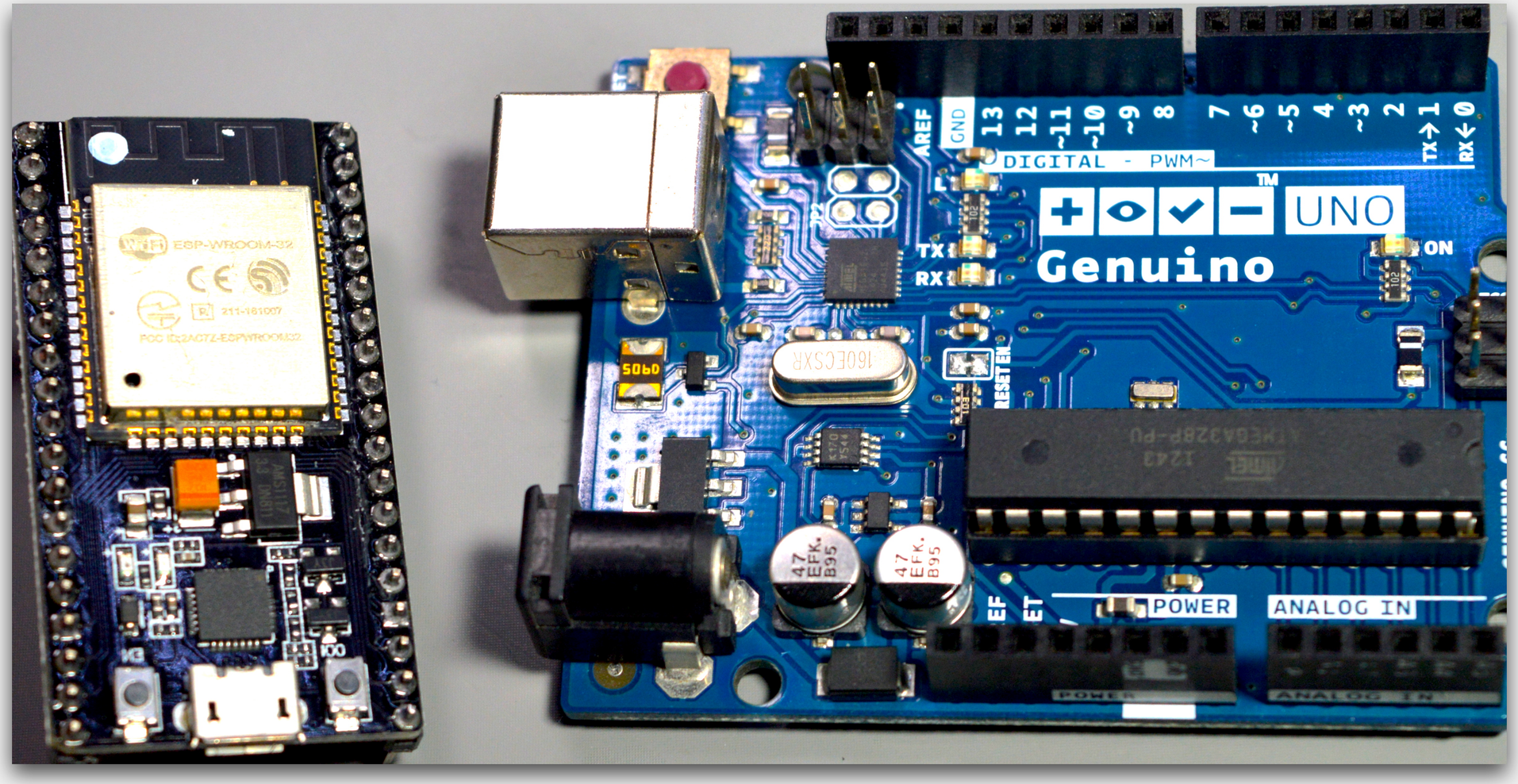


Photo of ESP32-D0WDQ6 by Brian Krent - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=57745131

docs.espressif.com/projects/esp-idf/en/latest/hw-reference/modules-and-boards.html

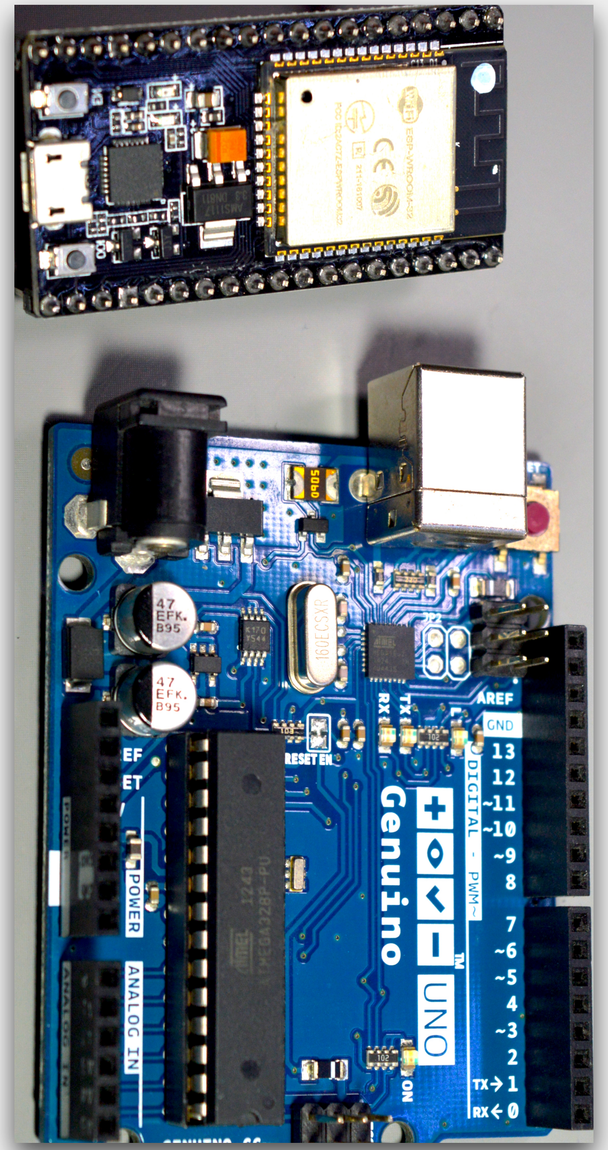
ESP32 vs Arduino



ESP32 vs Arduino

Almost as different as Black and White

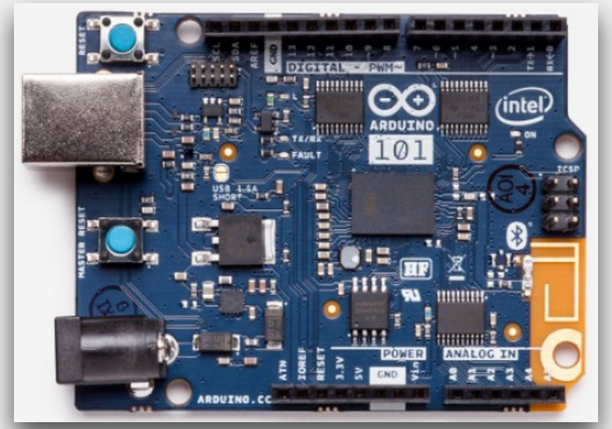
- Hardware architecture
- Capabilities
  - Build-in features
  - Memory
  - Processing
  - Number of GPIOs
  - Communications
  - Etc etc.

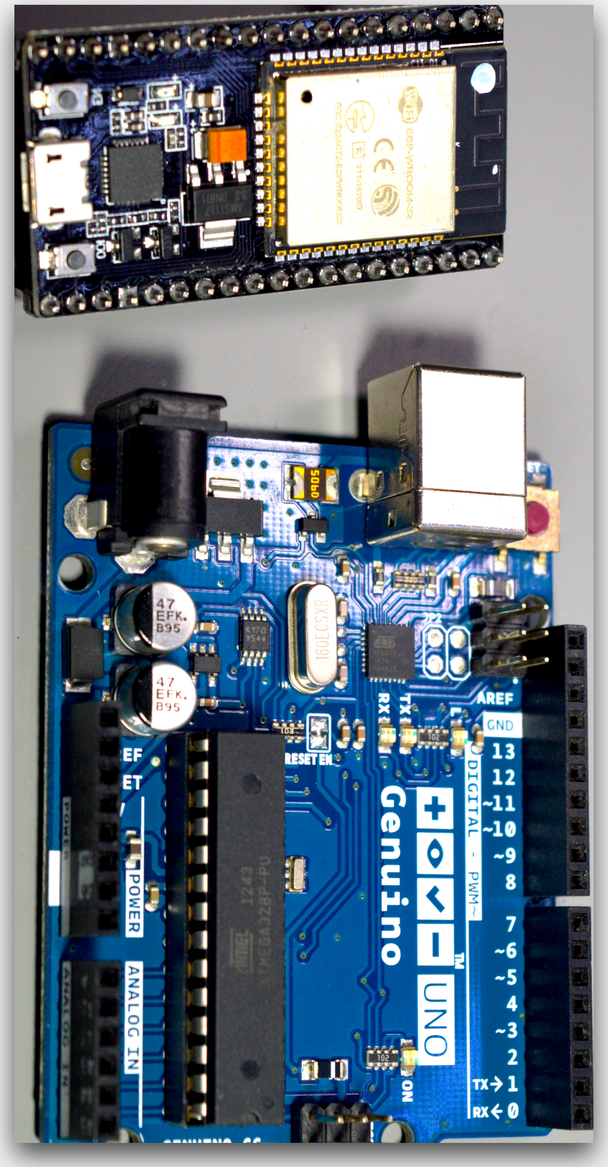


ESP32 vs Arduino

Almost as different as Black and White

The closest “thing” to an ESP32 board is, perhaps, an Arduino 101 or Arduino Zero.



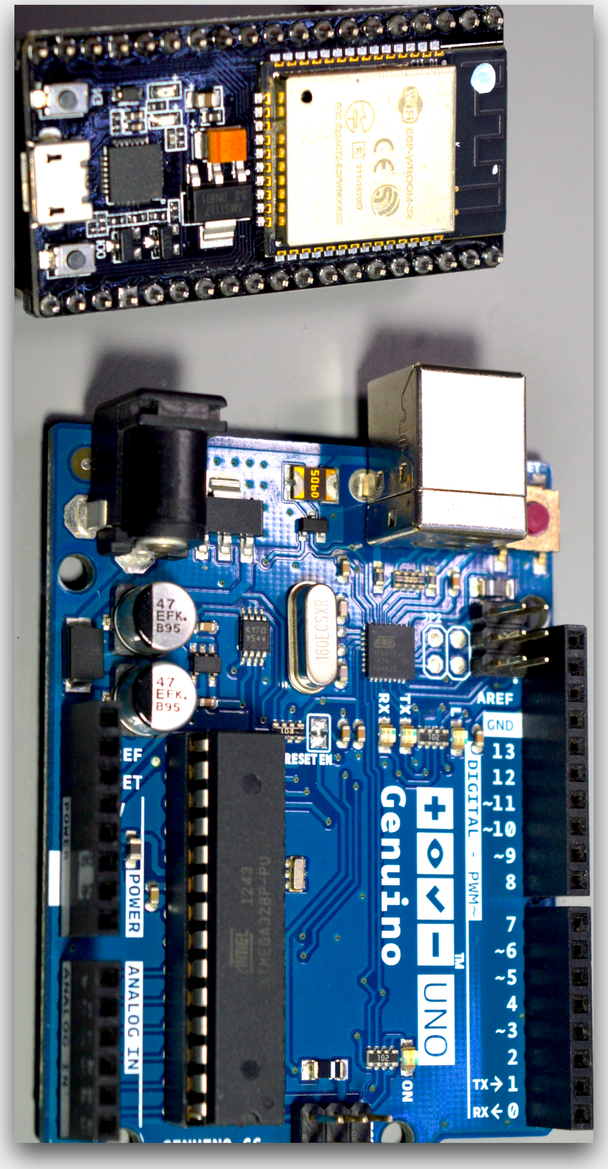


ESP32 vs Arduino

Where the two meet, is in the software

- ESP32 is compatible with the Arduino…
  - Development environment
  - Programming language
  - Libraries

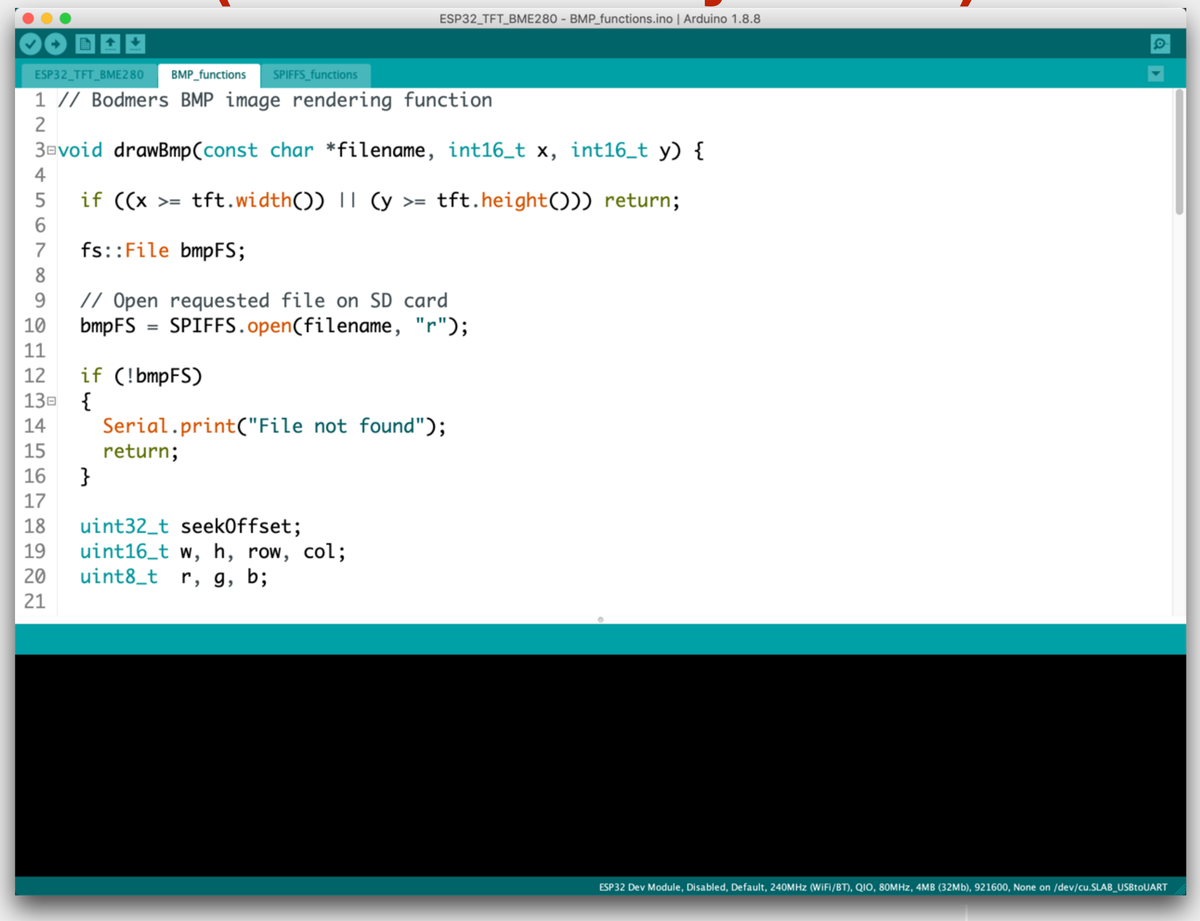
But adds amazing capabilities in every area.

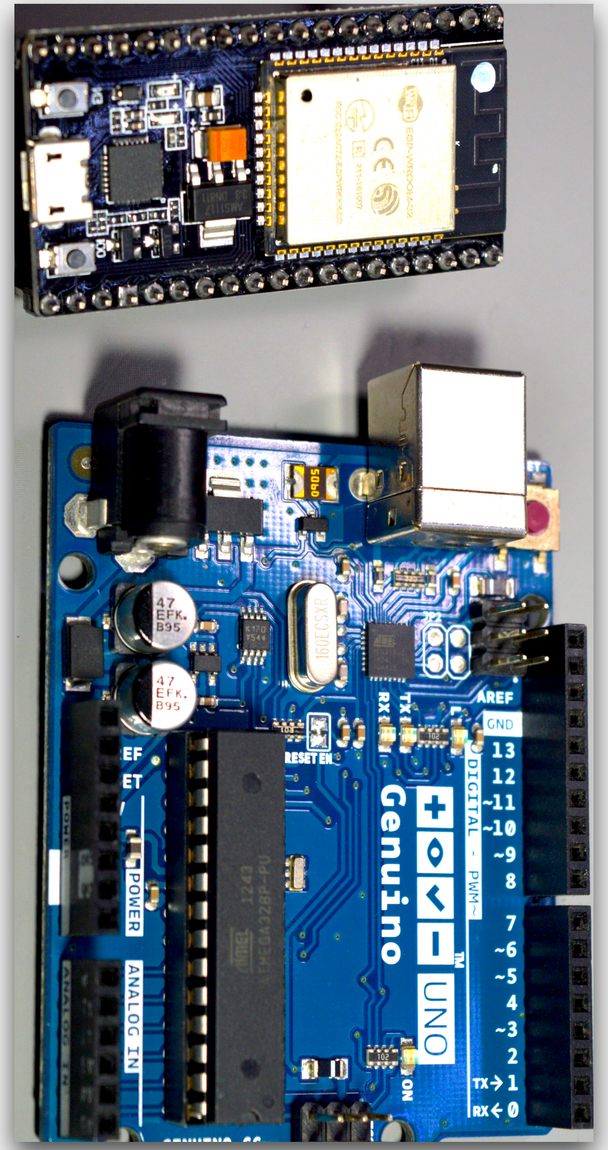


ESP32 vs Arduino

ESP32 works with the Arduino IDE

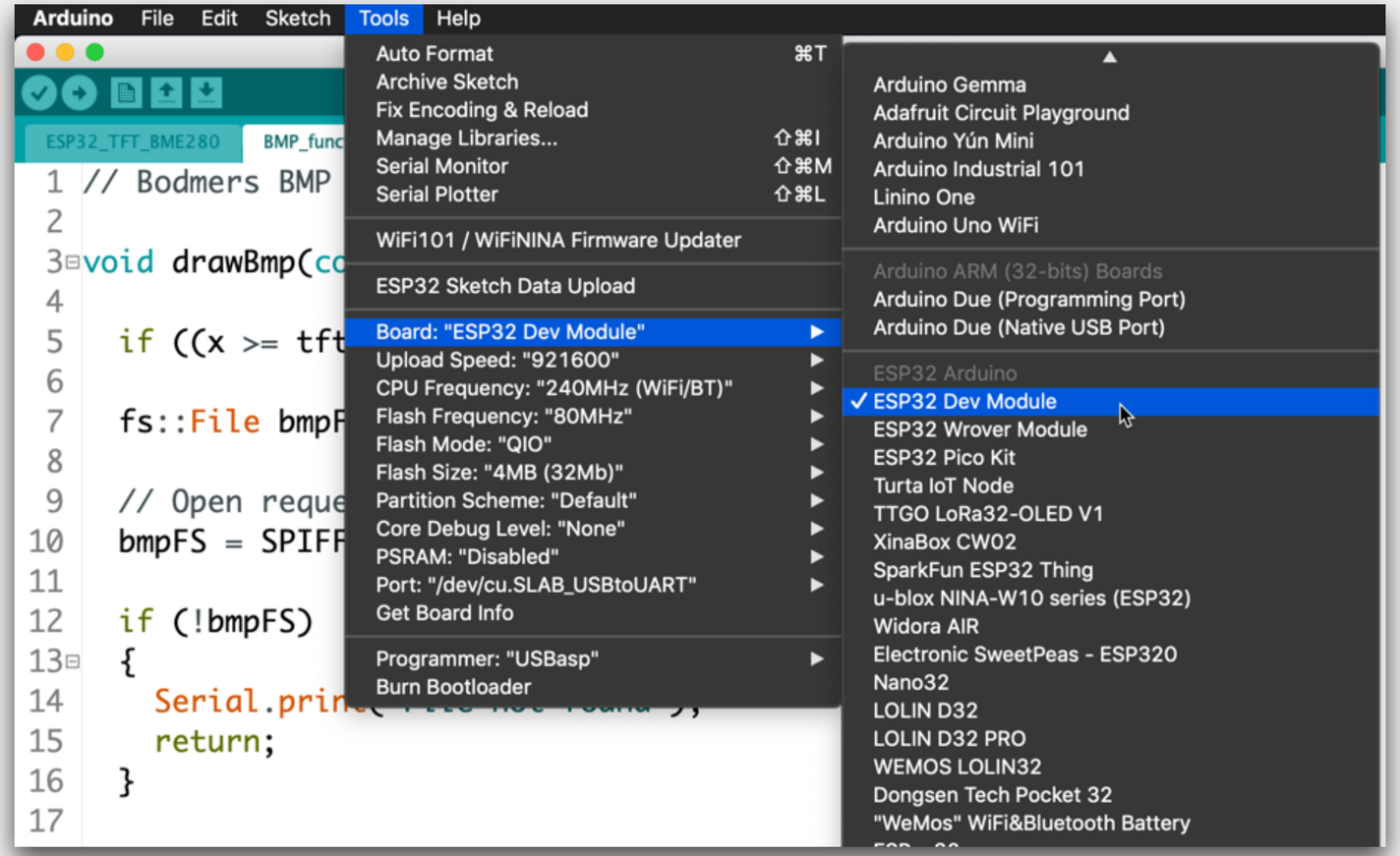
(as well as many others)

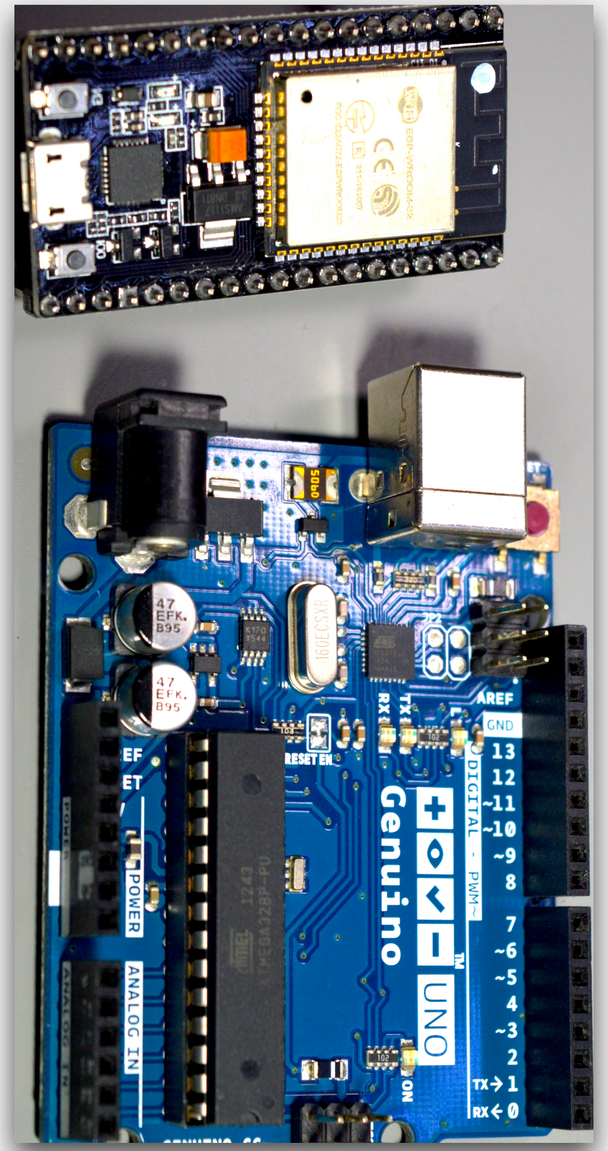




ESP32 vs Arduino

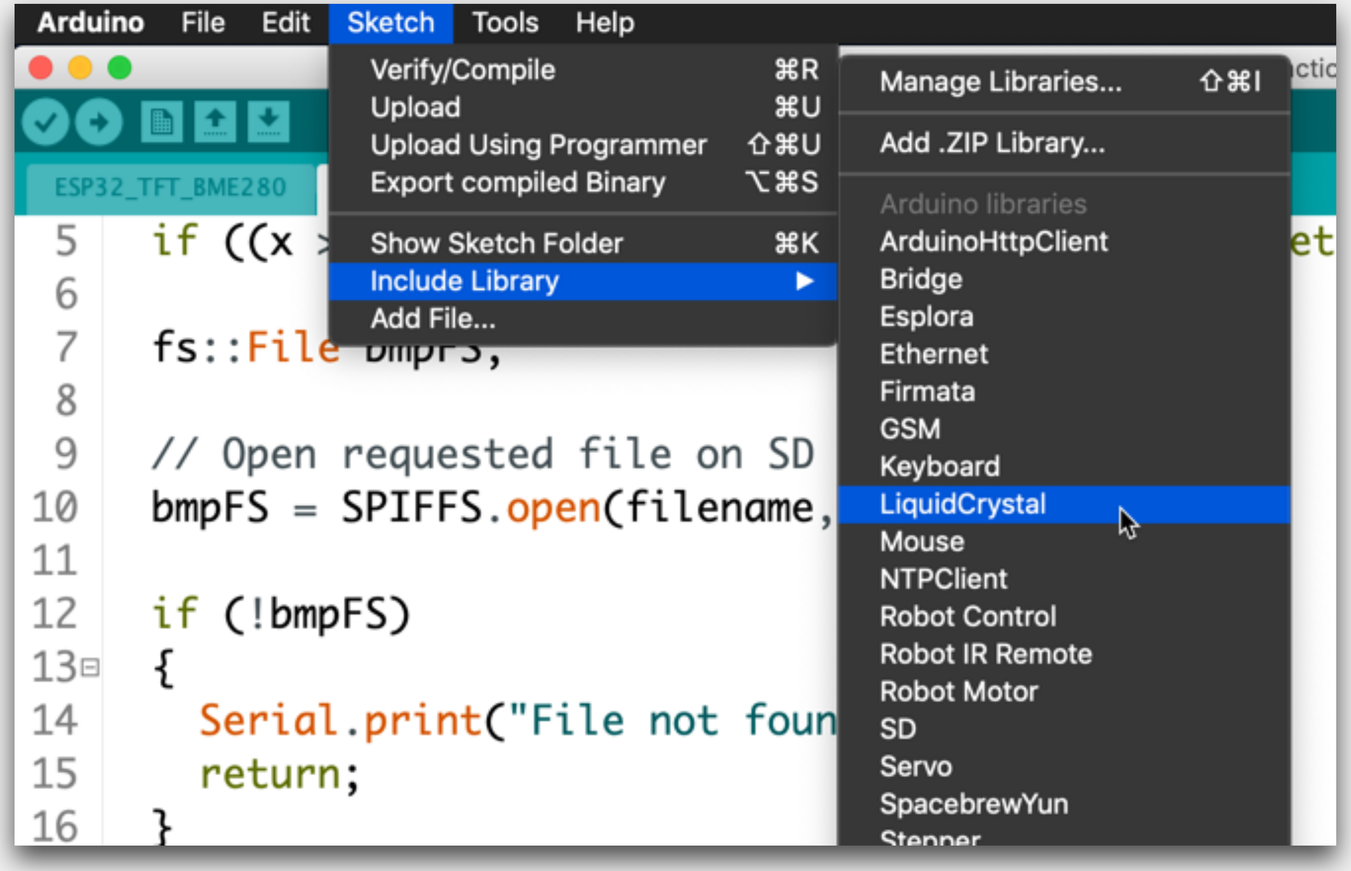
ESP32 support is impressive

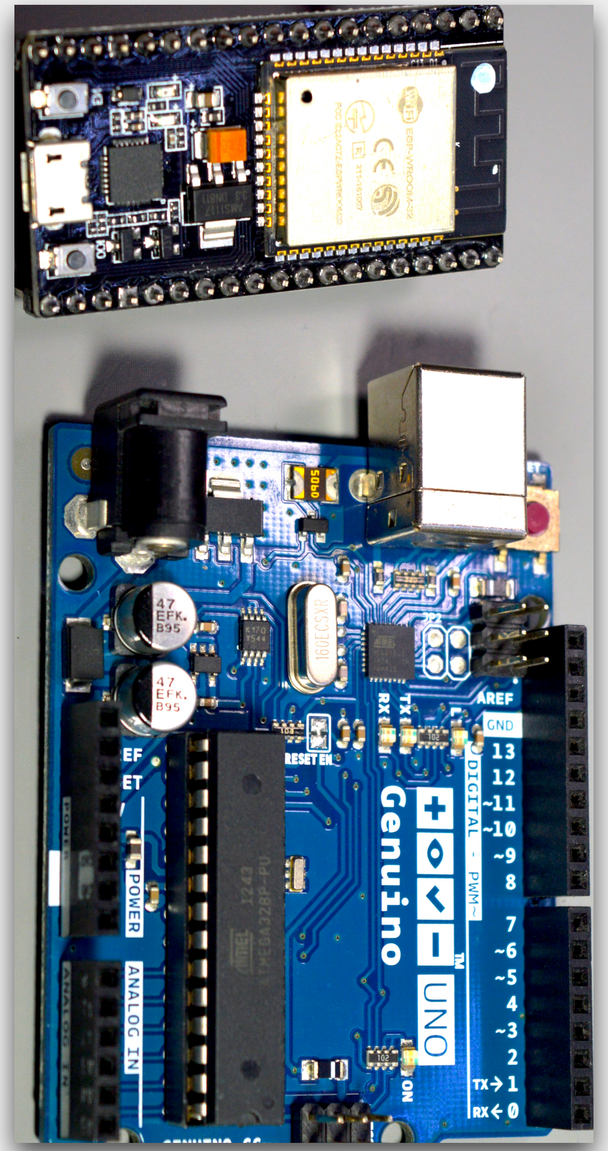




ESP32 vs Arduino

Most libraries, just work

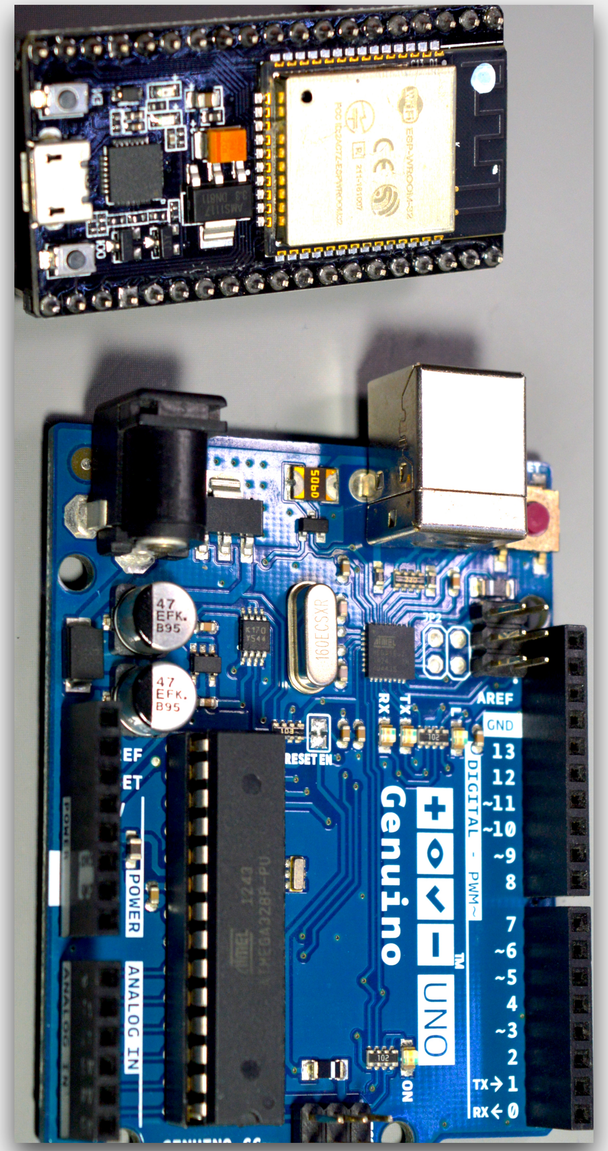




ESP32 vs Arduino

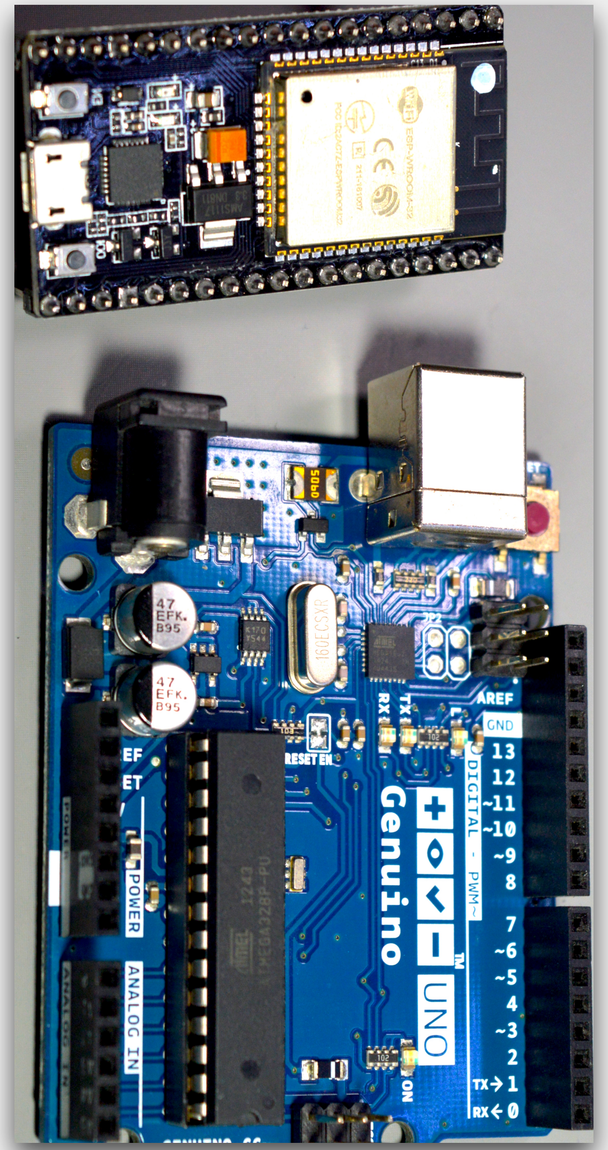
Unique ESP32 features, such as the SPIFS, is accessible via familiar Arduino libraries

```
7  //=============================================================================================
8  //                  Print a SPIFFS directory list (root directory)
9  //=============================================================================================
10
11 void listFiles(void) {
12   Serial.println();
13   Serial.println("SPIFFS files found:");
14
15 #ifdef ESP32
16   listDir(SPIFFS, "/", true);
17 #else
18   fs::Dir dir = SPIFFS.openDir("/"); // Root directory
19   String  line = "=====================================";
20
21   Serial.println(line);
22   Serial.println("  File name                Size");
23   Serial.println(line);
24
25   while (dir.next()) {
26     String fileName = dir.fileName();
27     Serial.print(fileName);
```



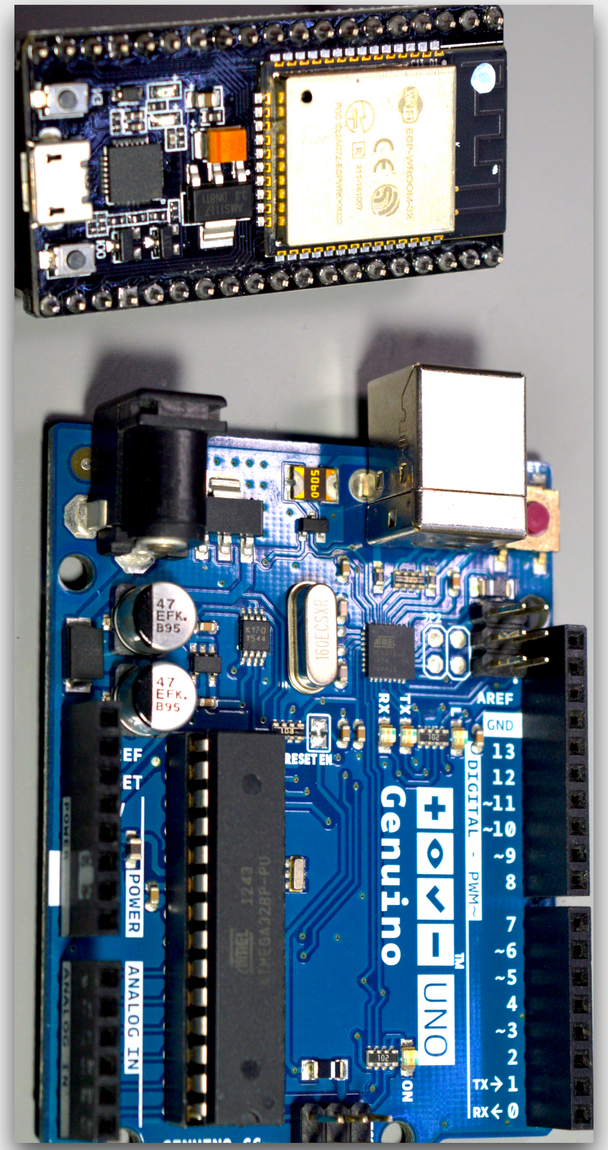
ESP32 vs Arduino

Who is the ESP32 for?



ESP32 vs Arduino

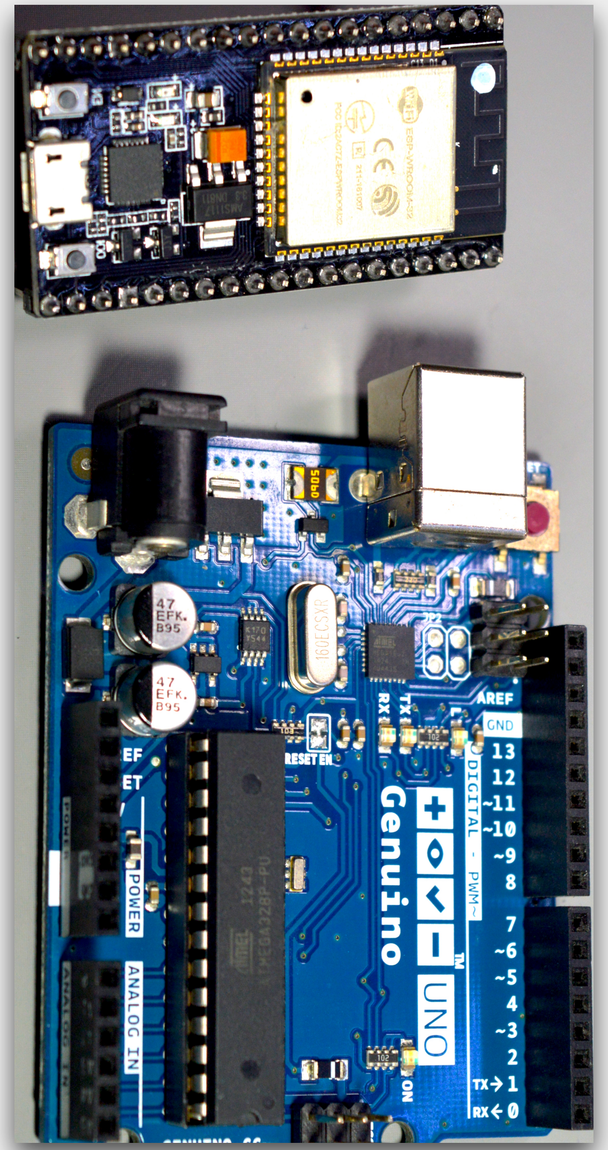
Not for beginners!



ESP32 vs Arduino

The Arduino is a better choice for new Makers.

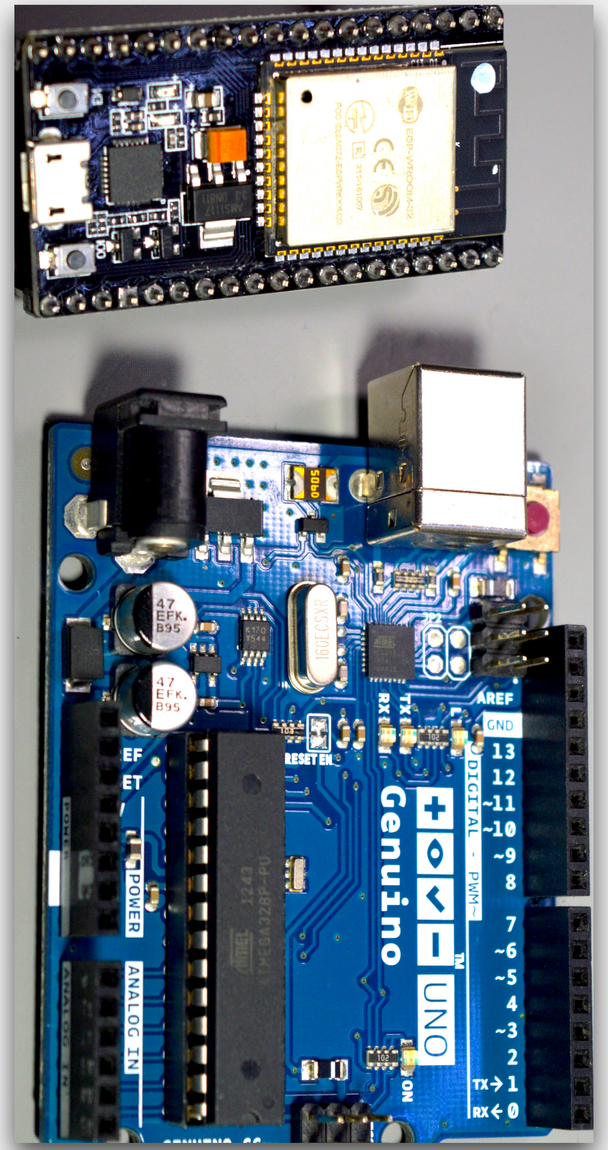
- Much simpler architecture (gentler learning curve)
- Easier to setup (works out of the box)
- Works with everything in the Arduino world
  - An amazing accumulated body of knowledge to learn from.
  - The ESP32 requires the ability to adapt your Arduino knowledge.
- More robust, able to take a lot of misuse



ESP32 vs Arduino

The ESP32 is perfect for Maker with at least intermediate Arduino skills.

- Any ESP32 capability that matches the Arduino, has no learning curve.
- Unique capabilities can be learned incrementally.
- You get Wifi, Bluetooth, lots of memory and speed for “free”.
- You can treat the ESP32 as a supercharged Arduino Uno
- You can also grow your skills to a totally new class.
- You can finally move away from the Arduino IDE to a more complete IDE.

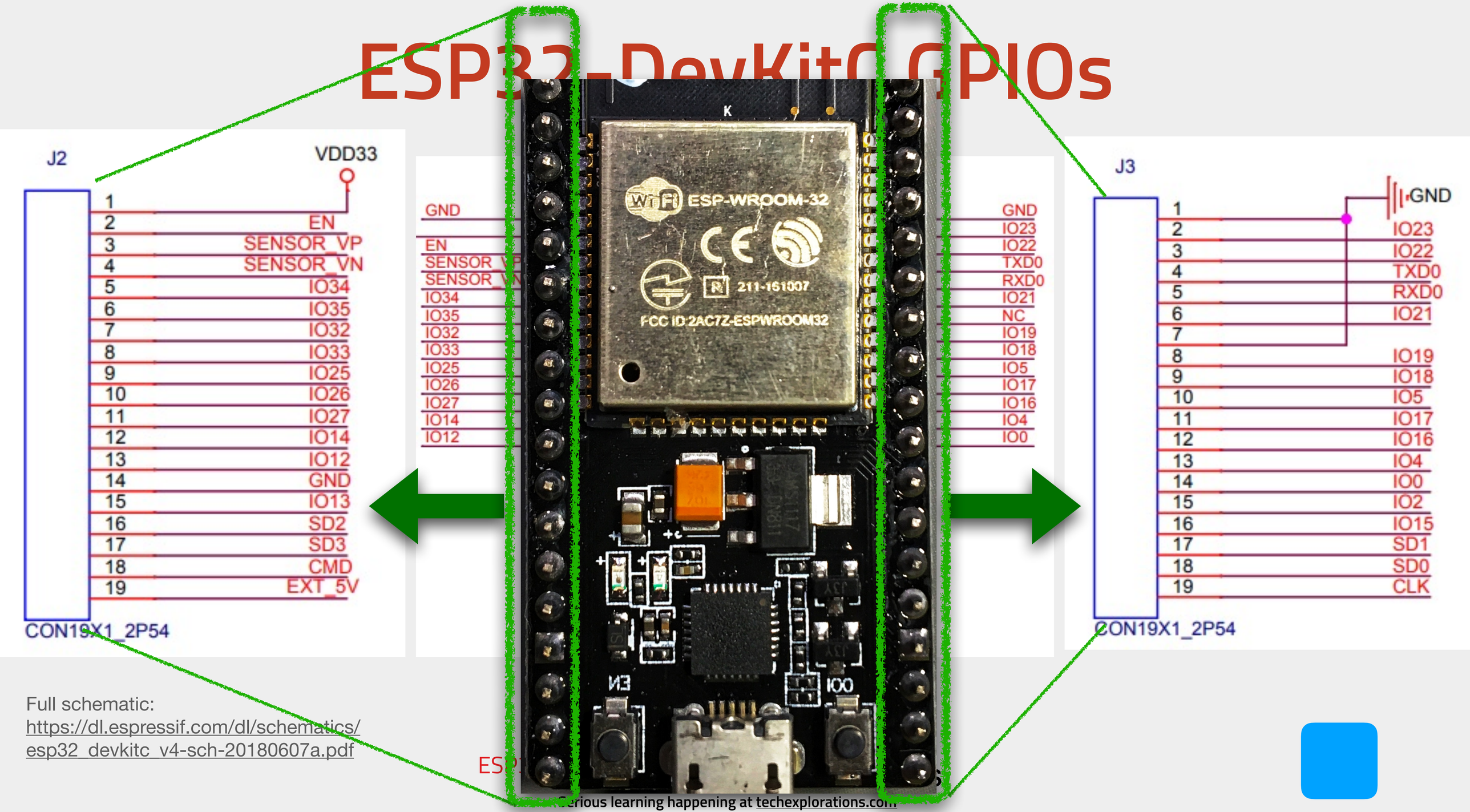


ESP32 GPIOs

ESP32-DevKitC GPIOs

Most of the 38 pins of the ESP32-WROOM-32 module are broken out in two rows of pins in the ESP32 Dev Kit

ESP32-DevKitC GPIOs



Full schematic:
https://dl.espressif.com/dl/schematics/esp32_devkitc_v4-sch-20180607a.pdf

ESP32-DevKitC GPIOs

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS

Name	No.	Type	Function
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Original: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
Page 3

ESP32-DevKitC GPIOs

Most pins have multiple roles.

Name	No.	Type	Function
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
			GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0,

Original: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
Page 3

ESP32-DevKitC GPIOs

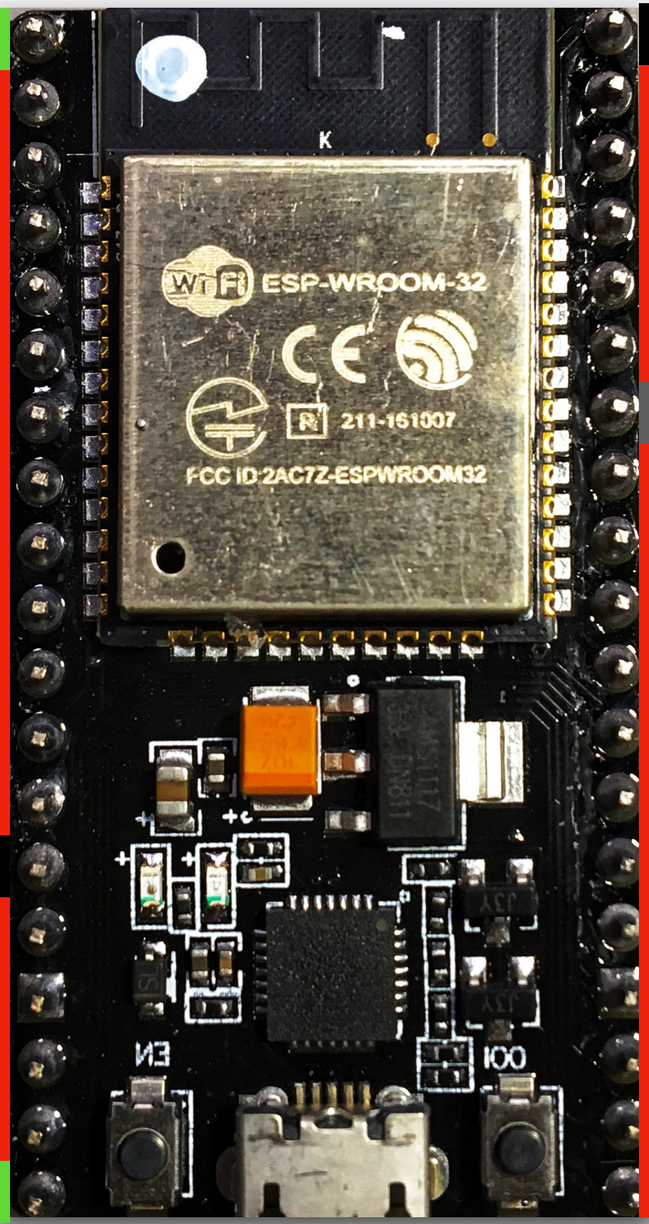
Beware: GPIOs 6-11 are connected to the module's integrated SPI flash and can't be used for external connections

These GPIOs are not broken out in the Dev kit.

Original: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
Page 8

ESP32-DevKitC Pins

				3.3V	GND				
			EN	09	36 ~	IO23	SPI MOSI		
Input only	SVP	ADC1_0	IO36	05	39 ~	IO22	I²C SCL		
Input only	SVN	ADC1_3	IO39	08	41 ~	IO1	TXD0		
Input only		ADC1_6	IO34	10	40 ~	IO3	RXD0		
Input only		ADC1_7	IO35	11	42 ~	IO21	I²C SDA		
	TOUCH9	ADC1_4	IO32	~ 12	NC				
	TOUCH8	ADC1_5	IO33	~ 13	38 ~	IO19	SPI MISO		
	DAC_1	ADC2_8	IO25	~ 14	35 ~	IO18	SPI SCK		
	DAC_2	ADC2_9	IO26	~ 15	34 ~	IO5	SPI SS		
	TOUCH7	ADC2_7	IO27	~ 16	27 ~	IO17	TXD		
	TOUCH6	ADC2_6	IO14	~ 17	25 ~	IO16	RXD		
	TOUCH5	ADC2_5	IO12	~ 18	24 ~	IO4	ADC2_0	TOUCH0	
				GND	23 ~	IO0	ADC2_1	TOUCH1	
	TOUCH4	ADC2_4	IO13	~ 20	22 ~	IO2	ADC2_2	TOUCH2	
			IO9	~ 28	21 ~	IO15	ADC2_3	TOUCH3	
			IO10	~ 29	33 ~	IO8	SD1		
	CMD		IO11	~ 30	32 ~	IO7	SD0		
				5V	31 ~	IO6	CLK		



~ PWM

~ PWM

Based on information from https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

ESP32 for Busy People

Tech Explorations

Serious learning happening at techexplorations.com

You can download a printable version of this map from the lecture page.

ESP32-DevKitC GPIOs

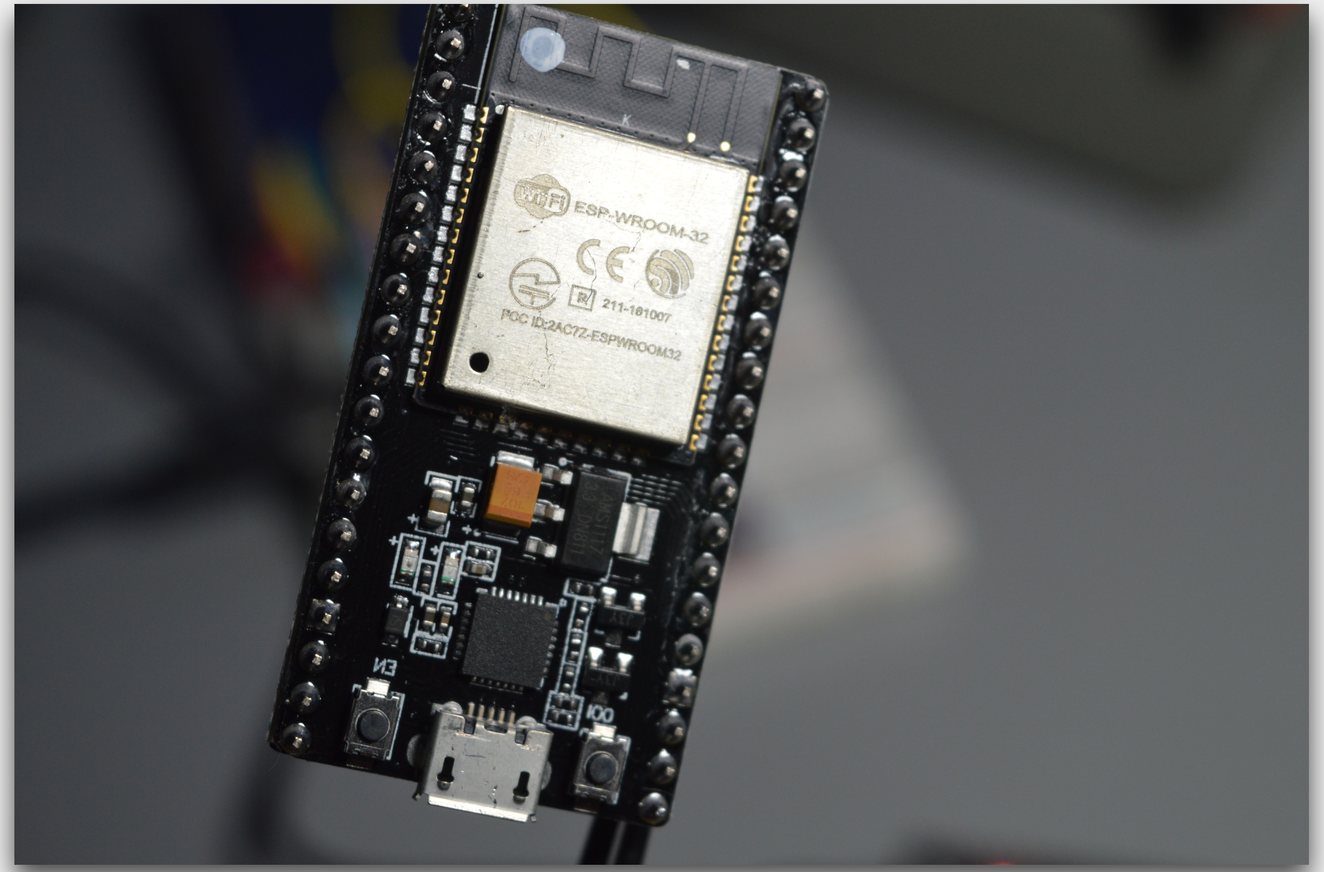
Pin No.	Power Supply Pin	Analog Pin	Digital Pin	Power Domain	Analog Function1	Analog Function2	Analog Function3	RTC Function1	RTC Function2	Function1	Type	Function2	Type	Function3	Type	Function4	Type	Function5	Type	Function6	Type	Drive Strength (2'd2: 20 mA)	At Reset	After Reset
1	VDDA			VDDA supply in																				
2		LNA_IN		VDD3P3																				
3	VDD3P3			VDD3P3 supply in																				
4	VDD3P3			VDD3P3 supply in																				
5		SENSOR_VP		VDD3P3_RTC	ADC_H	ADC1_CH0		RTC_GPIO0		GPIO36	I			GPIO36	I								oe=0, ie=0	oe=0, ie=0
6		SENSOR_CAPP		VDD3P3_RTC	ADC_H	ADC1_CH1		RTC_GPIO1		GPIO37	I			GPIO37	I								oe=0, ie=0	oe=0, ie=0
7		SENSOR_CAPN		VDD3P3_RTC	ADC_H	ADC1_CH2		RTC_GPIO2		GPIO38	I			GPIO38	I								oe=0, ie=0	oe=0, ie=0
8		SENSOR_VN		VDD3P3_RTC	ADC_H	ADC1_CH3		RTC_GPIO3		GPIO39	I			GPIO39	I								oe=0, ie=0	oe=0, ie=0
9		CHIP_PU		VDD3P3_RTC																				
10		VDET_1		VDD3P3_RTC		ADC1_CH6		RTC_GPIO4		GPIO34	I			GPIO34	I								oe=0, ie=0	oe=0, ie=0
11		VDET_2		VDD3P3_RTC		ADC1_CH7		RTC_GPIO5		GPIO35	I			GPIO35	I								oe=0, ie=0	oe=0, ie=0
12		32K_XP		VDD3P3_RTC	XTAL_32K_P	ADC1_CH4	TOUCH9	RTC_GPIO9		GPIO32	I/O/T			GPIO32	I/O/T							2'd2	oe=0, ie=0	oe=0, ie=0
13		32K_XN		VDD3P3_RTC	XTAL_32K_N	ADC1_CH5	TOUCH8	RTC_GPIO8		GPIO33	I/O/T			GPIO33	I/O/T							2'd2	oe=0, ie=0	oe=0, ie=0
14			GPIO25	VDD3P3_RTC	DAC_1	ADC2_CH8		RTC_GPIO6		GPIO25	I/O/T			GPIO25	I/O/T					EMAC_RXD0	I	2'd2	oe=0, ie=0	oe=0, ie=0
15			GPIO26	VDD3P3_RTC	DAC_2	ADC2_CH9		RTC_GPIO7		GPIO26	I/O/T			GPIO26	I/O/T					EMAC_RXD1	I	2'd2	oe=0, ie=0	oe=0, ie=0
16			GPIO27	VDD3P3_RTC		ADC2_CH7	TOUCH7	RTC_GPIO17		GPIO27	I/O/T			GPIO27	I/O/T					EMAC_RX_DV	I	2'd2	oe=0, ie=0	oe=0, ie=1
17			MTMS	VDD3P3_RTC		ADC2_CH6	TOUCH6	RTC_GPIO16		MTMS	I0	HSPICLK	I/O/T	GPIO14	I/O/T	HS2_CLK	O	SD_CLK	I0	EMAC_TXD2	O	2'd2	oe=0, ie=0	oe=0, ie=1
18			MTDI	VDD3P3_RTC		ADC2_CH5	TOUCH5	RTC_GPIO15		MTDI	I1	HSPIQ	I/O/T	GPIO12	I/O/T	HS2_DATA2	I1/O/T	SD_DATA2	I1/O/T	EMAC_TXD3	O	2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd
19	VDD3P3_RTC			VDD3P3_RTC supply in																				
20			MTCK	VDD3P3_RTC		ADC2_CH4	TOUCH4	RTC_GPIO14		MTCK	I1	HSPID	I/O/T	GPIO13	I/O/T	HS2_DATA3	I1/O/T	SD_DATA3	I1/O/T	EMAC_RX_ER	I	2'd2	oe=0, ie=0	oe=0, ie=1
21			MTDO	VDD3P3_RTC		ADC2_CH3	TOUCH3	RTC_GPIO13	I2C_SDA	MTDO	O/T	HSPICS0	I/O/T	GPIO15	I/O/T	HS2_CMD	I1/O/T	SD_CMD	I1/O/T	EMAC_RXD3	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
22			GPIO2	VDD3P3_RTC		ADC2_CH2	TOUCH2	RTC_GPIO12	I2C_SCL	GPIO2	I/O/T	HSPIWP	I/O/T	GPIO2	I/O/T	HS2_DATA0	I1/O/T	SD_DATA0	I1/O/T			2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd
23			GPIO0	VDD3P3_RTC		ADC2_CH1	TOUCH1	RTC_GPIO11	I2C_SDA	GPIO0	I/O/T	CLK_OUT1	O	GPIO0	I/O/T					EMAC_TX_CLK	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
24			GPIO4	VDD3P3_RTC		ADC2_CH0	TOUCH0	RTC_GPIO10	I2C_SCL	GPIO4	I/O/T	HSPIHD	I/O/T	GPIO4	I/O/T	HS2_DATA1	I1/O/T	SD_DATA1	I1/O/T	EMAC_TX_ER	O	2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd
25			GPIO16	VDD_SDIO						GPIO16	I/O/T			GPIO16	I/O/T	HS1_DATA4	I1/O/T	U2RXD	I1	EMAC_CLK_OUT	O	2'd2	oe=0, ie=0	oe=0, ie=1
26	VDD_SDIO			VDD_SDIO supply out/in																				
27			GPIO17	VDD_SDIO						GPIO17	I/O/T			GPIO17	I/O/T	HS1_DATA5	I1/O/T	U2TXD	O	EMAC_CLK_OUT_180	O	2'd2	oe=0, ie=0	oe=0, ie=1
28			SD_DATA_2	VDD_SDIO						SD_DATA2	I1/O/T	SPIHD	I/O/T	GPIO9	I/O/T	HS1_DATA2	I1/O/T	U1RXD	I1			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
29			SD_DATA_3	VDD_SDIO						SD_DATA3	I0/O/T	SPIWP	I/O/T	GPIO10	I/O/T	HS1_DATA3	I1/O/T	U1TXD	O			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
30			SD_CMD	VDD_SDIO						SD_CMD	I1/O/T	SPICS0	I/O/T	GPIO11	I/O/T	HS1_CMD	I1/O/T	U1RTS	O			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
31			SD_CLK	VDD_SDIO						SD_CLK	I0	SPICLK	I/O/T	GPIO6	I/O/T	HS1_CLK	O	U1CTS	I1			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
32			SD_DATA_0	VDD_SDIO						SD_DATA0	I1/O/T	SPIQ	I/O/T	GPIO7	I/O/T	HS1_DATA0	I1/O/T	U2RTS	O			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
33			SD_DATA_1	VDD_SDIO						SD_DATA1	I1/O/T	SPID	I/O/T	GPIO8	I/O/T	HS1_DATA1	I1/O/T	U2CTS	I1			2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
34			GPIO5	VDD3P3_CPU						GPIO5	I/O/T	VSPICS0	I/O/T	GPIO5	I/O/T	HS1_DATA6	I1/O/T			EMAC_RX_CLK	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
35			GPIO18	VDD3P3_CPU						GPIO18	I/O/T	VSPICLK	I/O/T	GPIO18	I/O/T	HS1_DATA7	I1/O/T					2'd2	oe=0, ie=0	oe=0, ie=1
36			GPIO23	VDD3P3_CPU						GPIO23	I/O/T	VSPID	I/O/T	GPIO23	I/O/T	HS1_STROBE	I0					2'd2	oe=0, ie=0	oe=0, ie=1
37	VDD3P3_CPU			VDD3P3_CPU supply in																				
38			GPIO19	VDD3P3_CPU						GPIO19	I/O/T	VSPIQ	I/O/T	GPIO19	I/O/T	U0CTS	I1			EMAC_TXD0	O	2'd2	oe=0, ie=0	oe=0, ie=1
39			GPIO22	VDD3P3_CPU						GPIO22	I/O/T	VSPIWP	I/O/T	GPIO22	I/O/T	U0RTS	O			EMAC_TXD1	O	2'd2	oe=0, ie=0	oe=0, ie=1
40			U0RXD	VDD3P3_CPU						U0RXD	I1	CLK_OUT2	O	GPIO3	I/O/T							2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
41			U0TXD	VDD3P3_CPU						U0TXD	O	CLK_OUT3	O	GPIO1	I/O/T					EMAC_RXD2	I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
42			GPIO21	VDD3P3_CPU						GPIO21	I/O/T	VSPIHD	I/O/T	GPIO21	I/O/T					EMAC_TX_EN	O	2'd2	oe=0, ie=0	oe=0, ie=1
43	VDDA			VDDA supply in																				
44		XTAL_N		VDDA																				
45		XTAL_P		VDDA																				
46	VDDA			VDDA supply in																				
47		CAP2		VDDA																				
48		CAP1		VDDA																				
Total Number	8	14	26																					

Notes:

- wpu: weak pull-up;
- wpd: weak pull-down;
- ie: input enable;
- oe: output enable;
- Please see *Table: Notes on ESP32 Pin Lists* for more information.（请参考表：管脚清单说明。）

Original, Page 49: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

ESP32 Communications



ESP32 for Busy People

Tech Explorations

Serious learning happening at techexplorations.com

ESP32 Communications

ESP32 offers multiple communications options

Wireless	Wired
Wifi	3 x SPI (Serial Peripheral Interface)
Bluetooth	2 x I^2C
	2 x I^2S
	3 x UART
	Ethernet MAC interface
	CAN 2.0
	IR (TX/RX)

ESP32 Communications

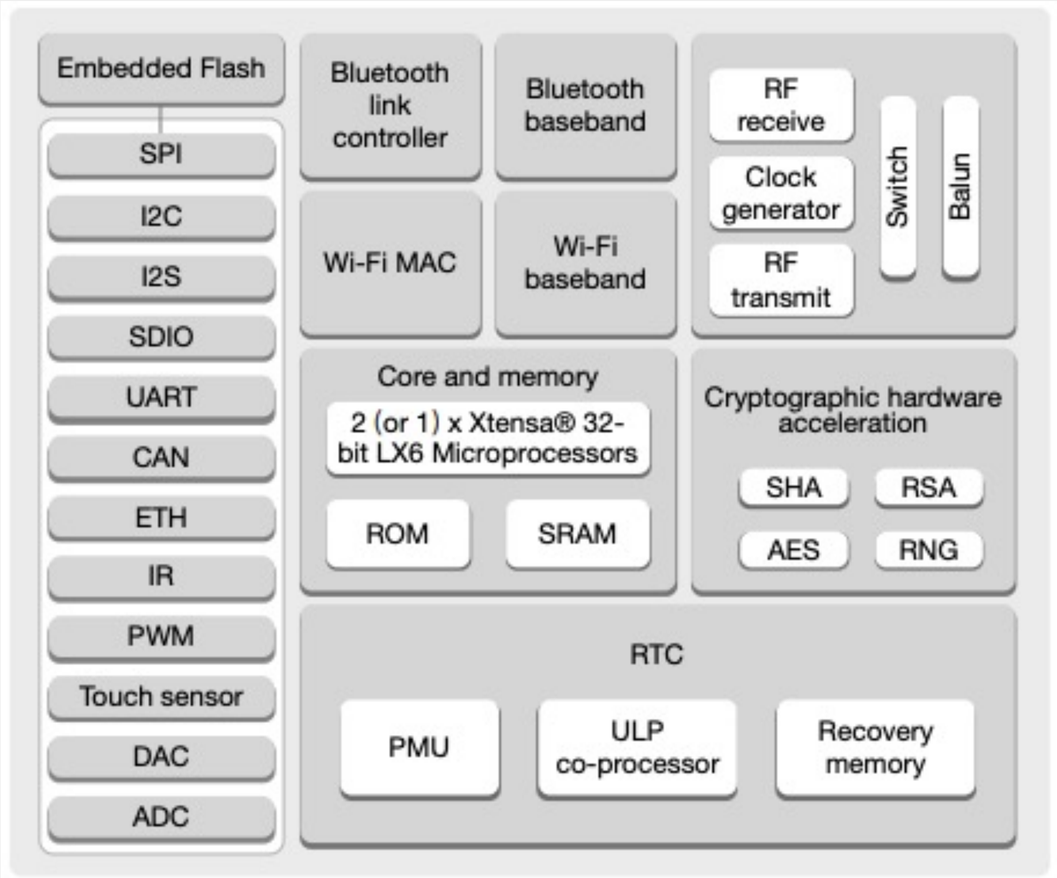


Figure 1: Functional Block Diagram

Section 1.6 in the Datasheet: https://txplo.re/033e8

ESP32 Communications

Wifi
All hardware integrated in the module (antenna, amplifier, filters, power management etc.)
802.11 b/g/n, 802.11 n (2.4 GHz), up to 150 Mbps
WMM (**W**ifi **M**ulti-**M**edia)
TX/RX A-MPDU, RX A-MSDU
4 × virtual Wi-Fi interfaces
Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes Note that when ESP32 is in Station mode,
More details Datasheet, section 3.5

Datasheet: https://txplo.re/033e8

ESP32 Communications

Bluetooth
Compliant with Bluetooth v4.2 BR/EDR and BLE specifications
Class-1, class-2 and class-3 transmitter without external power amplifier
Standard HCI (Host-to-Controller-Interface) based on SDIO/SPI/ UART[1]
Multi-connections in Classic BT and BLE
Simultaneous advertising and scanning
+12 dBm transmitting power
More details Datasheet, section 3.5

1 | https://iotbreaks.com/understand-bluetooth-hci-commands-and-events/

ESP32 Communications

SPI: Serial Peripheral Interface
3 SPIs: SPI, HSPI, VSPI
Up to 80 MHz
Up to 64-byte FIFO
Four modes of SPI transfer format
SPI: SPIHD (28), SPIWP (29), SPICS0 (30), SPICLK (31), SPIQ (32), SPID (33)
HSPI: HSPICLK (17), HSPIQ (18), HSPID (20), HSPICS0 (21), HSPIWP (22)
VSPI: VSPICS0 (34), VSPICLK (35), VSPID (36), VSPIQ (32), VSPIWP (29)

Section 4.1.17 in Datasheet: https://txplo.re/033e8

ESP32 Communications

I²C: Inter-integrated Circuit
Two I²C bus interfaces
Master or slave
Standard (100 Kbits/s) or Fast (400 Kbits/s)
Up to 5 MHz
7-bit or 10-bit addressing
Dual addressing

Section 4.1.11 in Datasheet: https://txplo.re/033e8

ESP32 Communications

I²S: Inter-integrated Circuit
2 I²S interfaces
Master or Slave
Full or half duplex
Up to 40 MHz

Section 4.1.12 in Datasheet: https://txplo.re/033e8

I²C is unrelated to I²S. Learn more at https://en.wikipedia.org/wiki/I%C2%B2S

ESP32 Communications

Universal Asynchronous Receiver Transmitter (UART)
3 UART interfaces
UART0, UART1 and UART2
Asynchronous communications (RS232, RS485)
Up to 5 Mbps
UART0: U0TX (GPIO1), U0RX (GPIO3) UART1: U1TX (SD_DATA_3), U1RX (SD_DATA_2) UART2: U2TX (GPIO17), U2RX (GPIO16)

Section 4.1.10 in Datasheet: https://txplo.re/033e8

ESP32 Communications

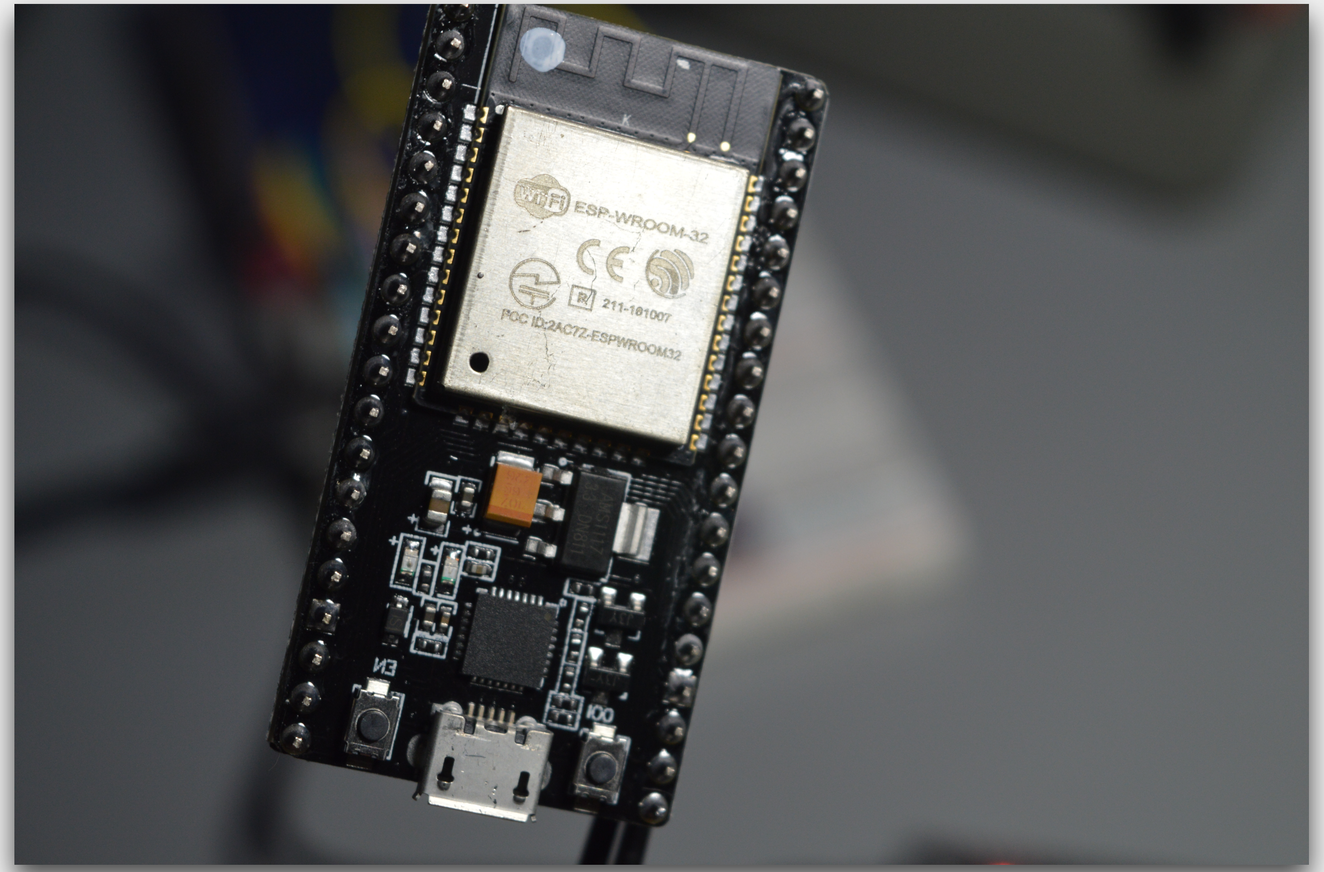
Other communications capabilities

IR (TX/RX): Section 4.1.13

Ethernet MAC: Section 4.1.7

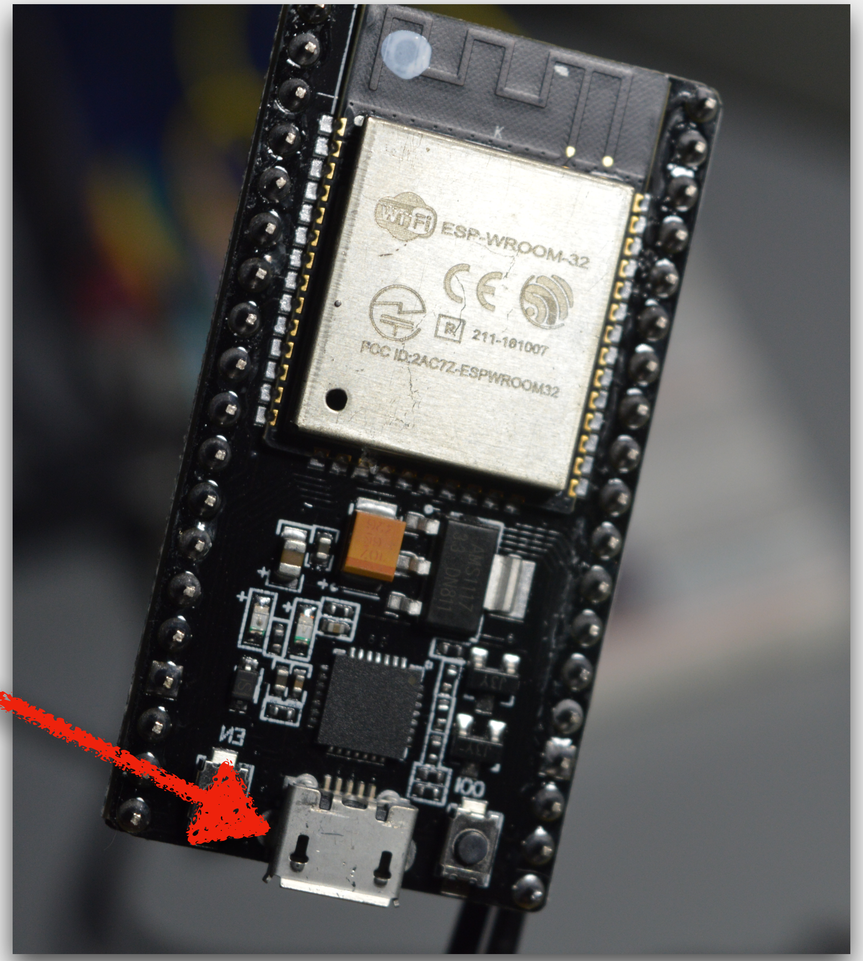
Datasheet: https://txplo.re/033e8

ESP32 Dev Kit v4 power options



ESP32 Power options

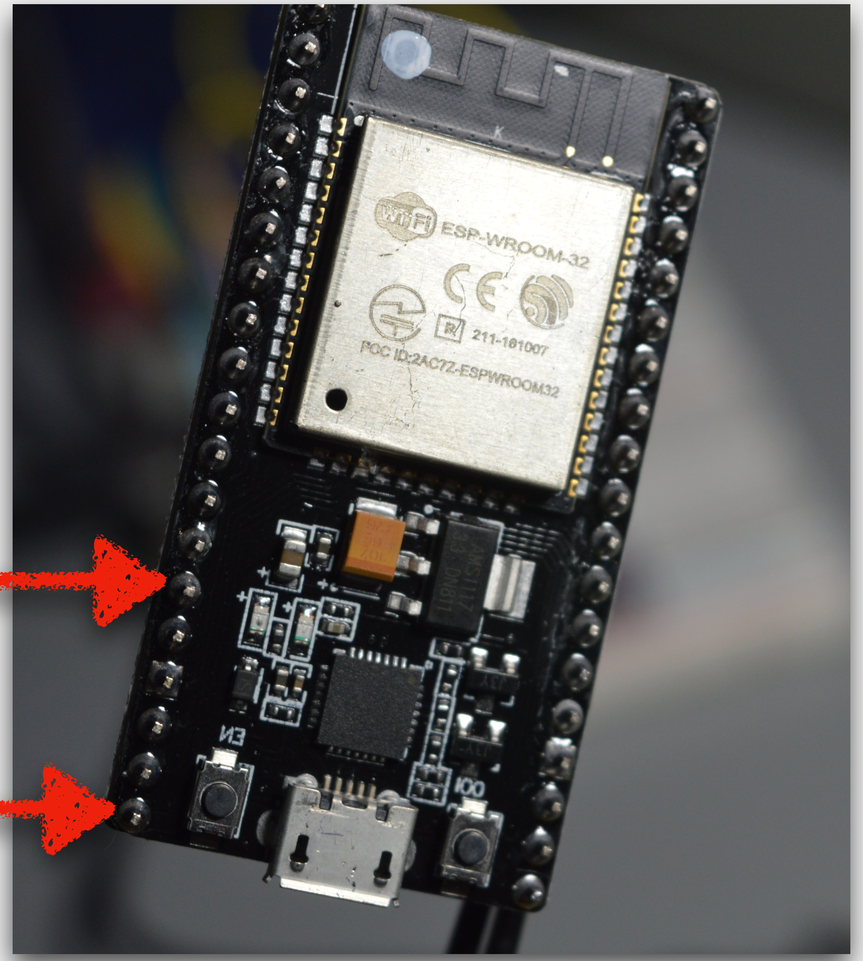
1: USB



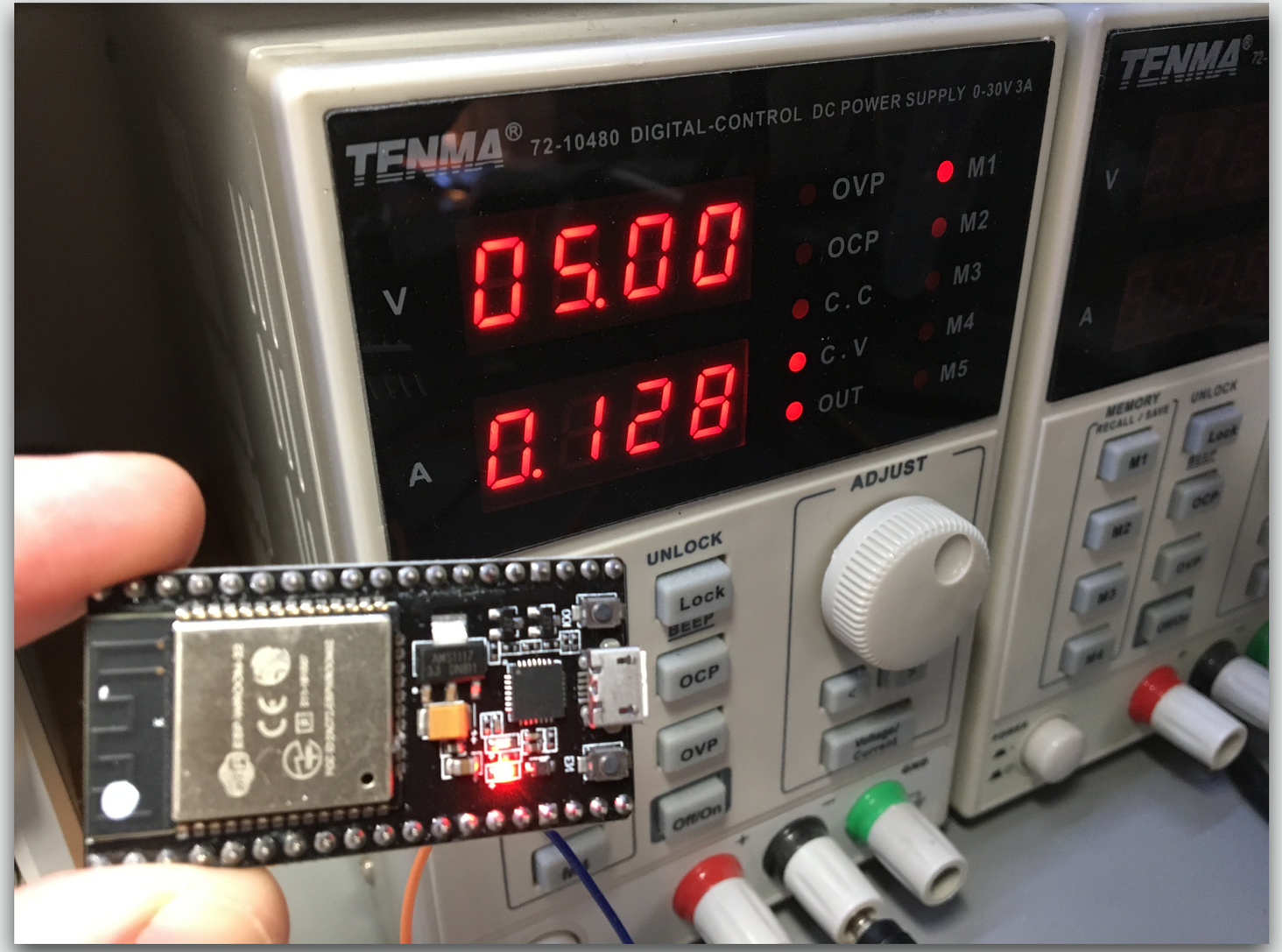
ESP32 Power options

2: 5V / GND header pins

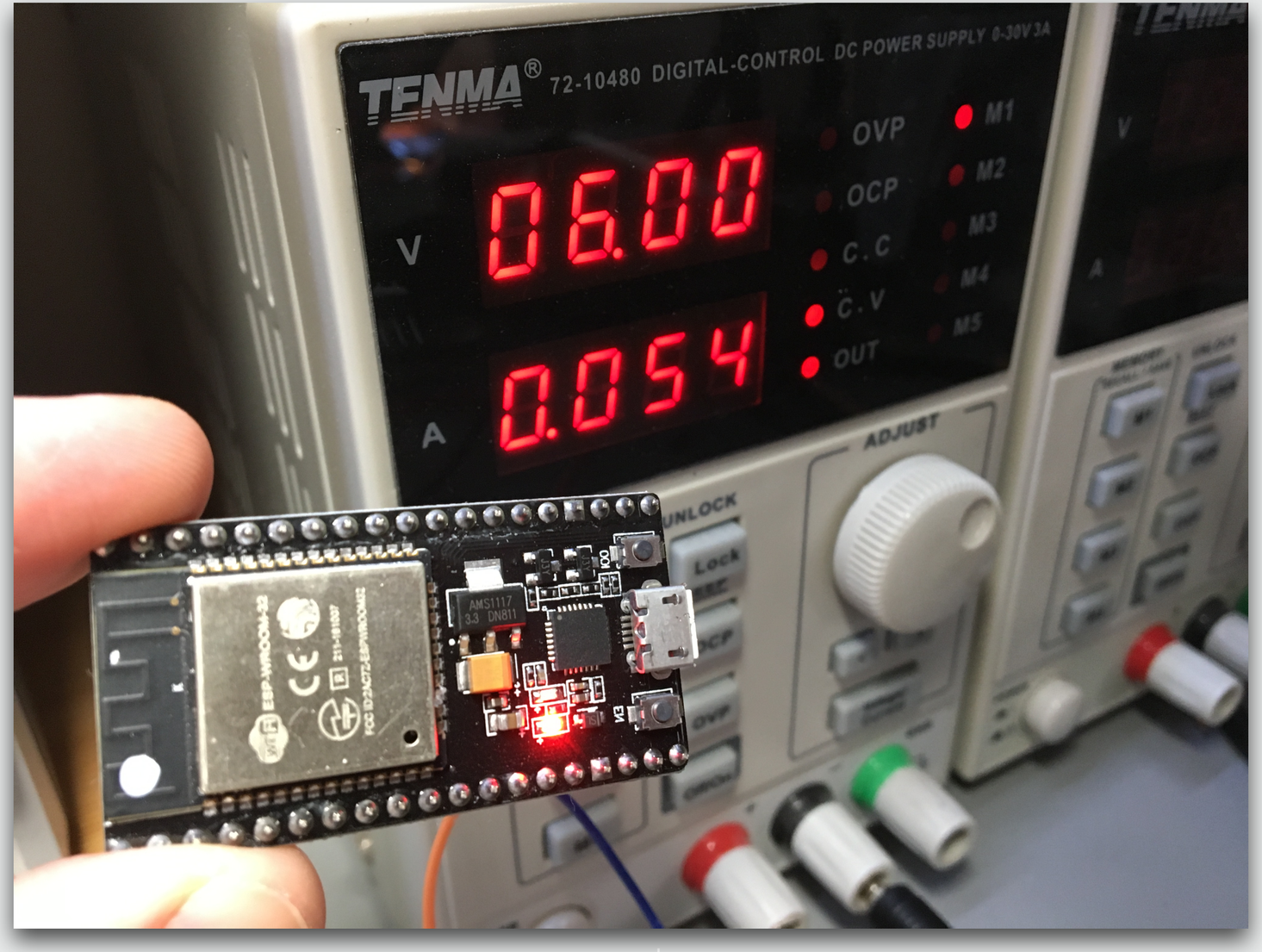
CAUTION: Keep input voltage below 12V to reduce heat on the voltage regulator



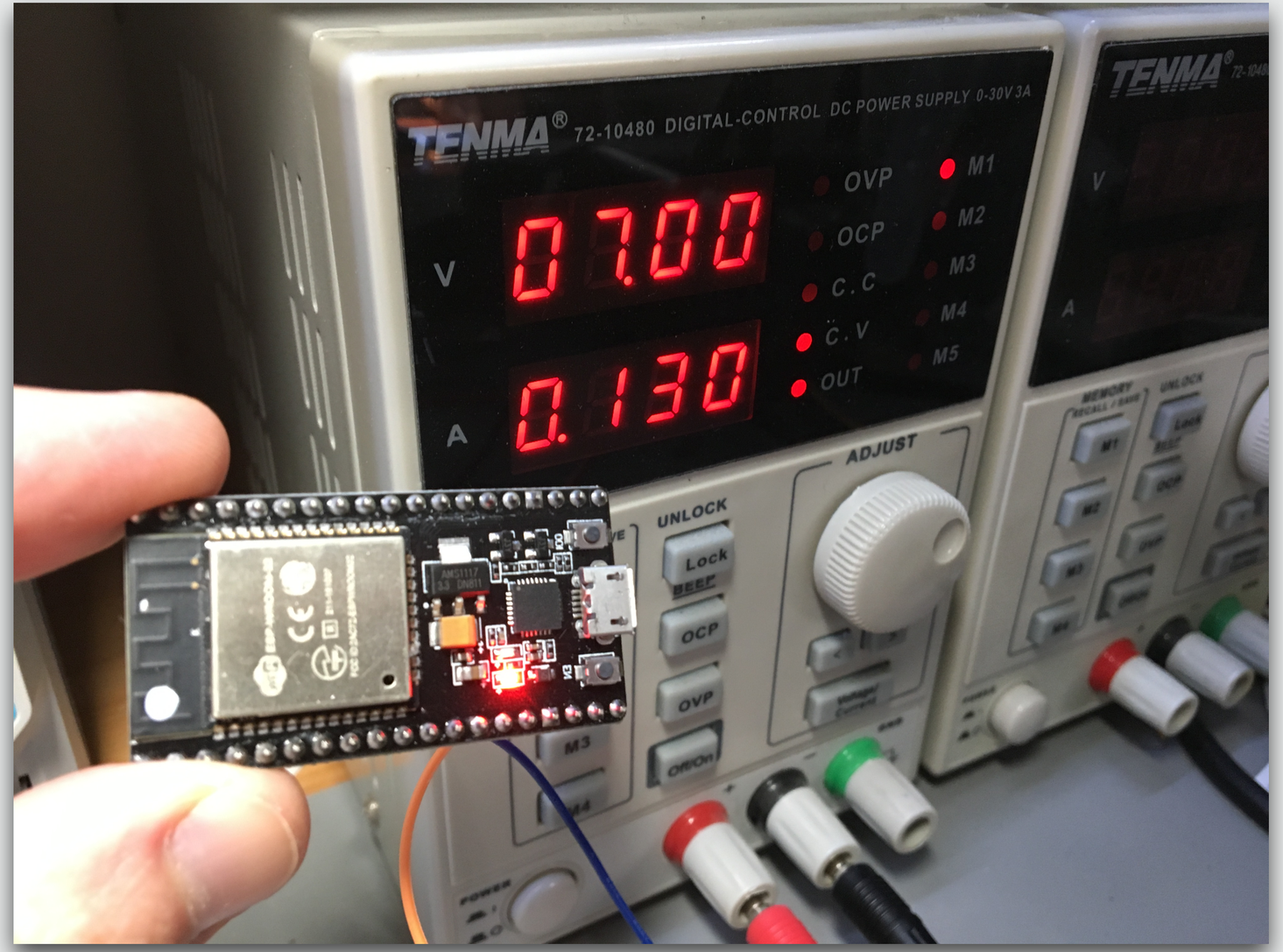
5V input


TENMA® 72-10480 DIGITAL-CONTROL DC POWER SUPPLY 0-30V 3A
V
05.00
A
0.128
OVP
OCP
C.C
C.V
OUT
M1
M2
M3
M4
M5
ADJUST
UNLOCK
Lock
OCP
OVP
On/Off
MEMORY
RECALL / SAVE
UNLOCK
TENMA

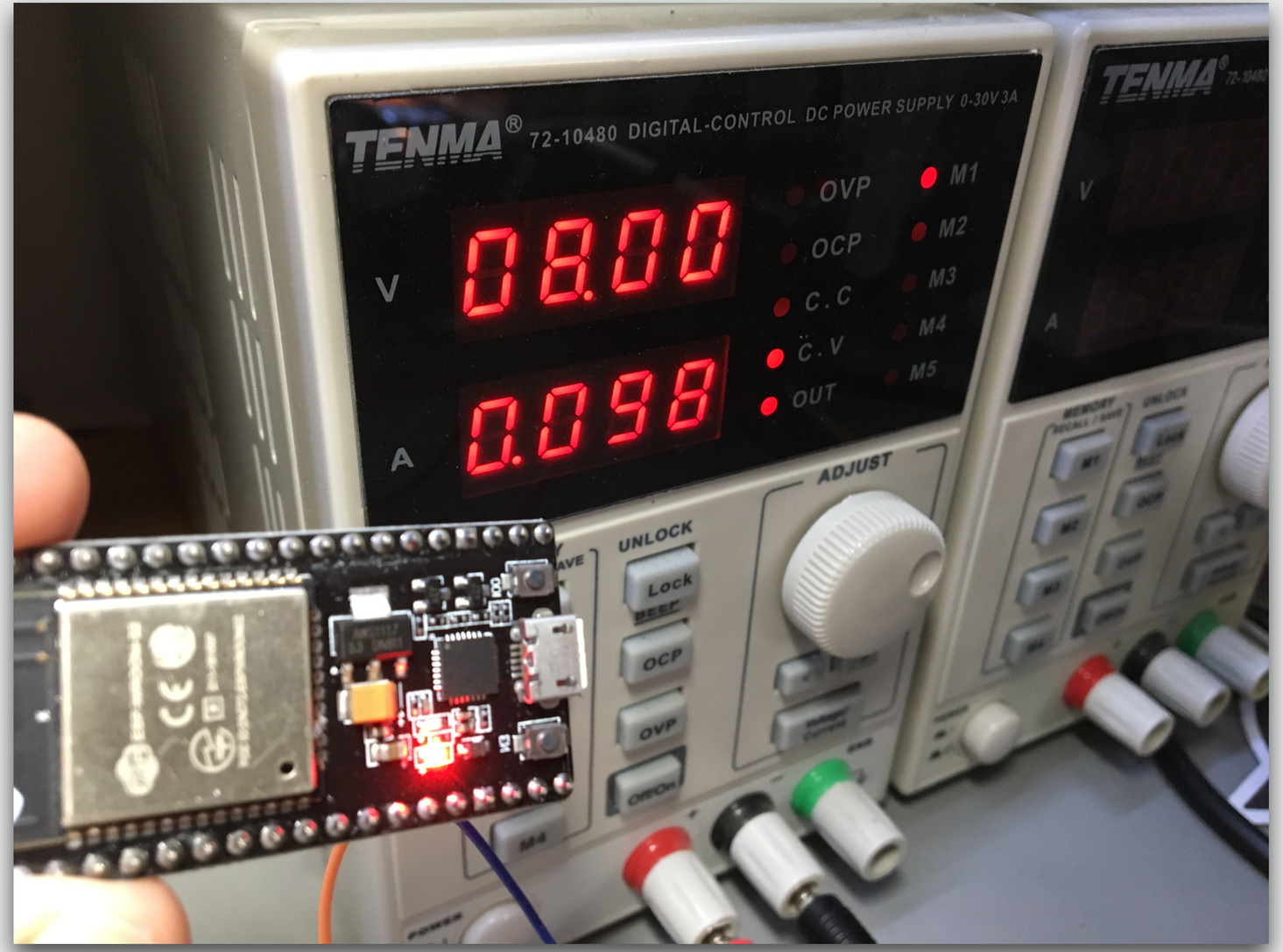
6V input


TENMA
72-10480 DIGITAL-CONTROL DC POWER SUPPLY 0-30V 3A
V
06.00
A
0.054
OVP
OCP
C.C
C.V
OUT
M1
M2
M3
M4
M5
ADJUST
UNLOCK
AMS1117
3.3 DN811

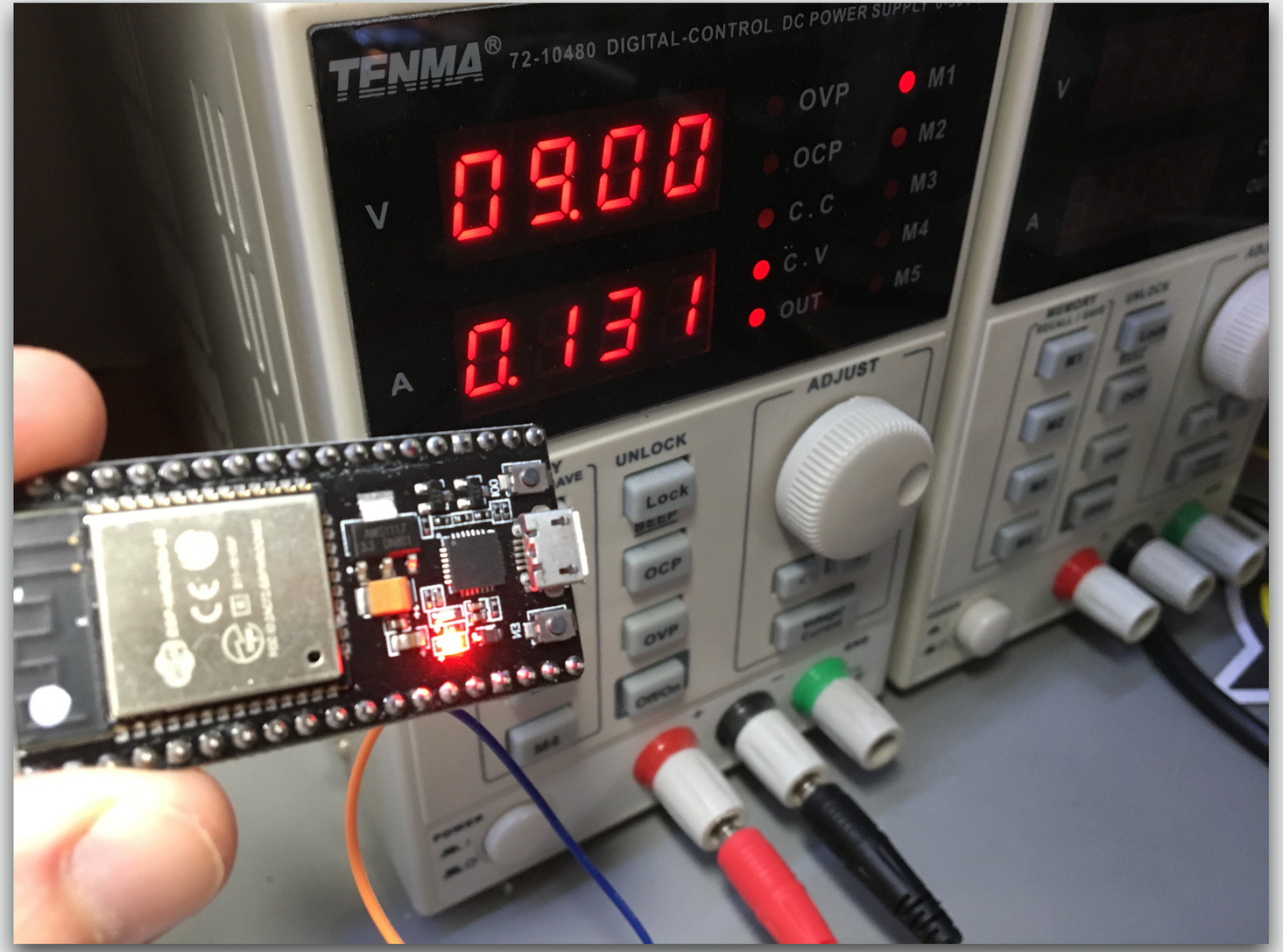
7V input


TENMA
72-10480 DIGITAL-CONTROL DC POWER SUPPLY 0-30V 3A
V
07.00
A
0.130
OVP
OCP
C.C
C.V
OUT
M1
M2
M3
M4
M5
ADJUST
UNLOCK
Lock
BEEP
OCP
OVP
ON/Off
TENMA

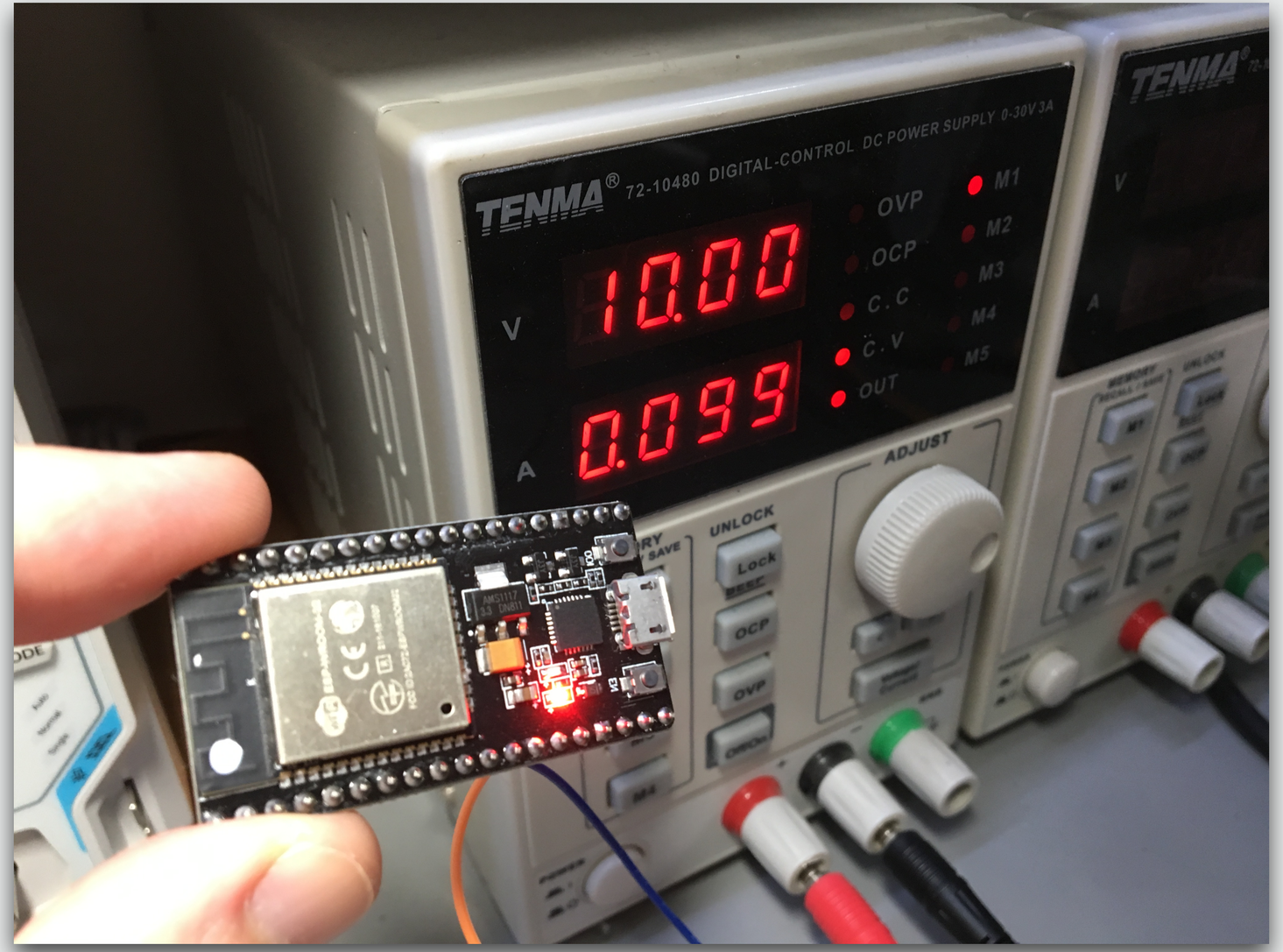
8V input


TENMA® 72-10480 DIGITAL-CONTROL DC POWER SUPPLY 0-30V 3A
V
08.00
A
0.098
OVP
OCP
C.C
C.V
OUT
M1
M2
M3
M4
M5
UNLOCK
Lock
OCP
OVP
ADJUST
TENMA

9V input


TENMA®
72-10480 DIGITAL-CONTROL DC POWER SUPPLY
V
09.00
A
0.131
OVP
OCP
C.C
C.V
OUT
M1
M2
M3
M4
M5
ADJUST
UNLOCK
Lock
OCP
OVP

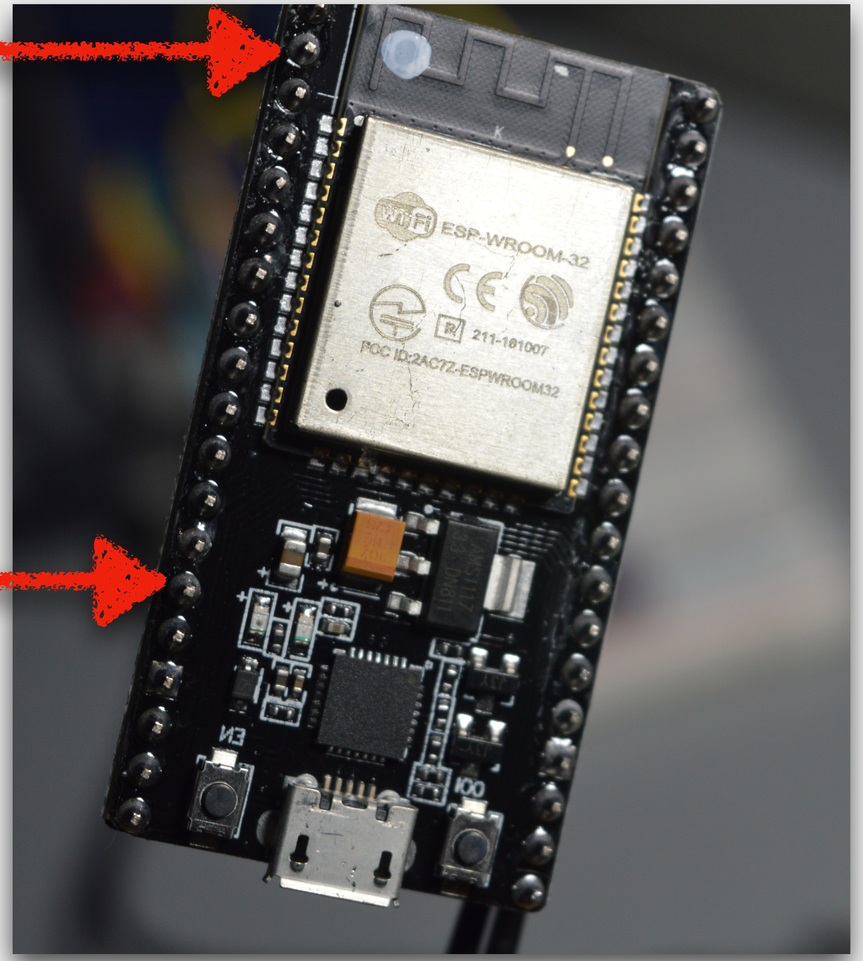
10V input


TENMA
72-10480 DIGITAL-CONTROL DC POWER SUPPLY 0-30V 3A
10.00
0.099
OVP
OCP
C.C
C.V
OUT
M1
M2
M3
M4
M5
ADJUST
UNLOCK
Lock
OCP
OVP

ESP32 Power options

3.3V

GND



3: 3.3V / GND header pins

CAUTION: Voltage must be regulated externally. Do not provide more than 3.3V on the 3.3V pin!

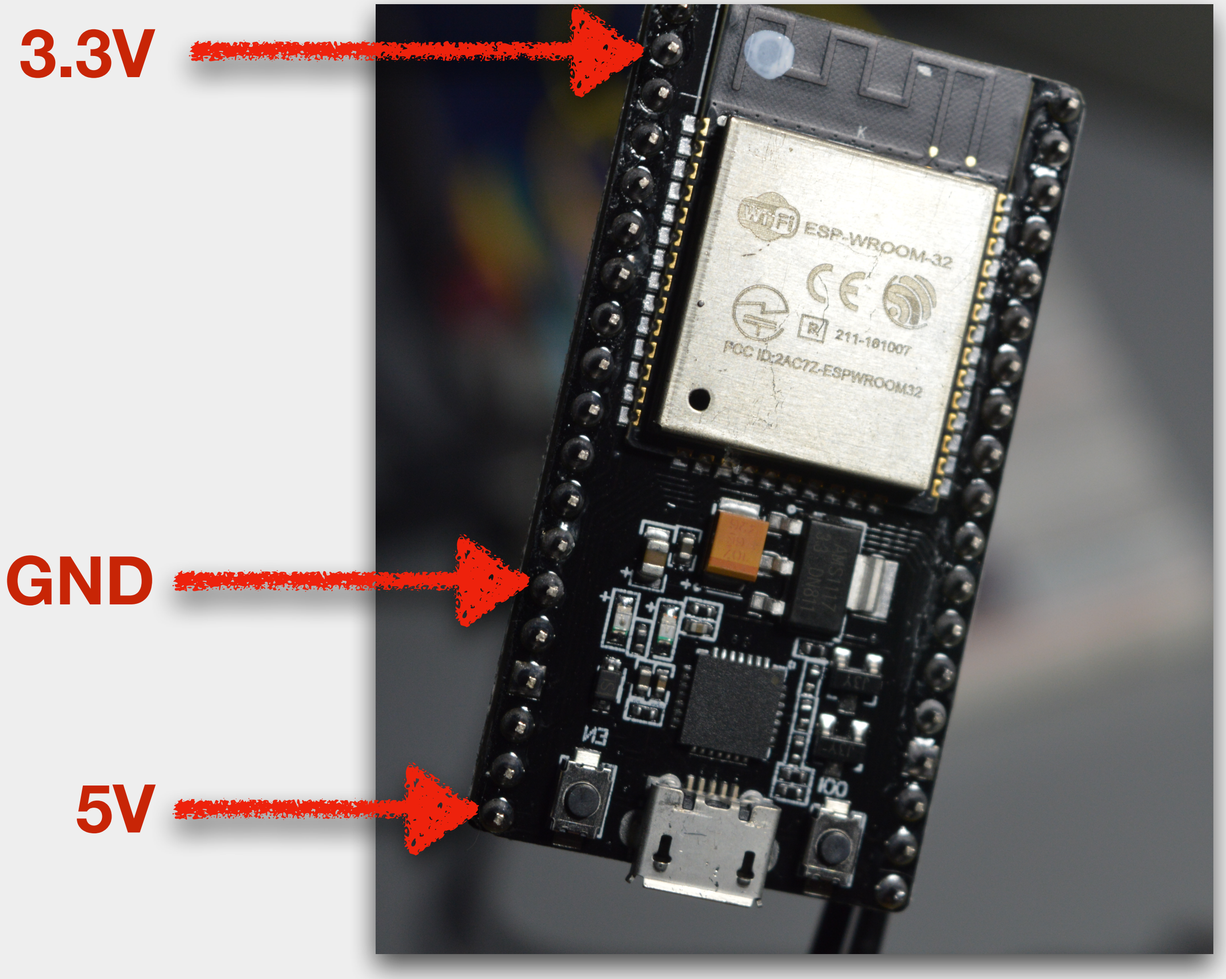
ESP32 Power options

1: USB

2: 5V / GND header pins

3: 3.3V / GND header pins

CAUTION: only use one option at a time!



ESP32 courses at Tech Explorations