SparkFun Thing Plus - ESP32 WROOM (USB-C) Hookup Guide

SparkFun Electronics®

Table of contents

1. Getting Started

Info

This guide is specific to the ESP32 Thing Plus (USB-C) board variant. For the variants with the USB micro-B connector, please refer to the ESP32 Thing Plus hookup guide.

1.1 Introduction

Tip

The CH340C serial-to-UART bridge is used on this board. Therefore, a different driver installation is required from previous versions of the ESP32 Thing Plus.

Warning

Not Yet Implemented: The Arduino core for the ESP32 microcontroller is still a work in progress. There are a handful of peripherals and features that have yet to be implemented, including:

- Analog Output (`analogWrite([pin], [value])`)
  - Alternative: LED Control API
- Pulse Counter
- SDIO
- ~~Timer~~/Real-Time Clock
  - Alternative: **ESP32Time** Arduino library
- TWAI

The peripherals are available (if, also, still in their infancy) in the IoT Development Framework for the ESP32. If your application requires any of the features above, consider giving the ESP-IDF a try! *(Updated: June 2022.)*

1.1.1 Required Materials

To get started, users will need a few items. Now some users may have a few of these items, feel free to modify your cart accordingly.

- SparkFun Thing Plus - ESP32 WROOM (USB-C)
- USB 3.1 Cable A to C - 3 Foot - The USB interface serves two purposes: it powers the board and allows users to upload programs. (**If your computer doesn't have a USB-A slot, then choose an appropriate cable or adapter.*)
- Computer with an operating system (OS) that is compatible with all the software installation requirements.

USB 3.1 Cable A to C - 3 Foot	SparkFun Thing Plus - ESP32 WROOM (USB-C)
CAB-14743	WRL-20168

Headers & Accessories

Headers are great for development purposes, letting users swap parts with just a set of jumper wires. If you would like to add headers to your board, check out some of the options for the Thing Plus or Feather form factor boards below. For a full selection of our available **Headers** or **Soldering Tools**, click on the associated links.





Break Away Headers - Straight	SparkFun Beginner Tool Kit	Feather Stackable Header Kit	Photon Header - 12 Pin Female	Header - 8-pin Female (PTH, 0.1")
PRT-00116	TOL-14681	PRT-15187	PRT-14321	PRT-11895

Tip

New to soldering? Check out our Through-Hole Soldering Tutorial for a quick introduction!

How to Solder: Through-Hole Soldering

Li-Po Battery

For mobile applications, users will want to pick up a single-cell LiPo battery from our catalog. Below, are a few available options:




Lithium Ion Battery - 2Ah	Lithium Ion Battery - 400mAh	Lithium Ion Battery - 1Ah	Lithium Ion Battery - 110mAh
PRT-13855	PRT-13851	PRT-13813	PRT-13853

Jumper Modification

To modify the jumpers, users will need soldering equipment and/or a knife.

Solder Lead Free - 100-gram Spool

TOL-09325

Weller WLC100 Soldering Station

TOL-14228

Chip Quik No-Clean Flux Pen - 10mL

TOL-14579

Hobby Knife

TOL-09200

Tip

New to jumper pads? Check out our Jumper Pads and PCB Traces Tutorial for a quick introduction!

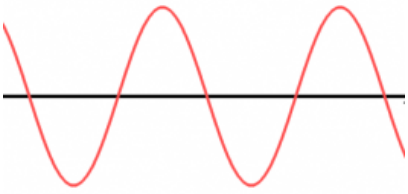
How to Work with Jumper Pads and PCB Traces

1.1.2 Suggested Reading

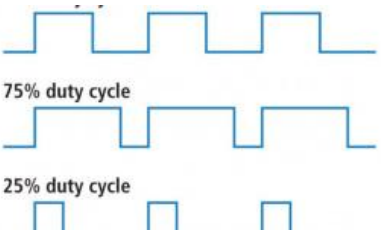
As a more advanced development board, we will skip over the more fundamental tutorials (i.e. **Ohm's Law** and **What is Electricity?**). However, below are a few tutorials that may help users familiarize themselves with various aspects of the board.

Serial Communication

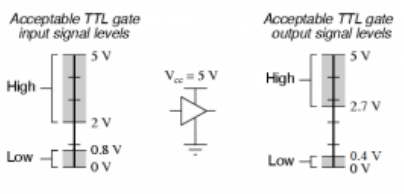
Analog vs. Digital



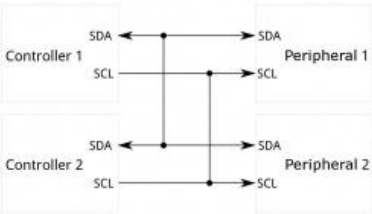
Pulse Width Modulation



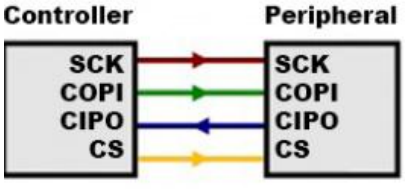
Logic Levels



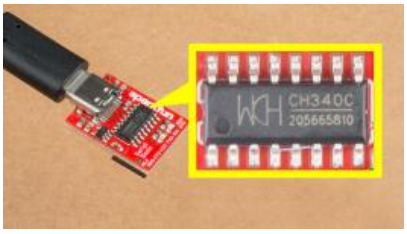
I2C



Serial Peripheral Interface (SPI)



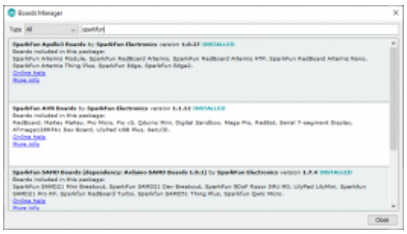
How to Install CH340 Drivers



ESP32 Thing Plus Hookup Guide

Installing the Arduino IDE

Installing Board Definitions in the Arduino IDE



How to Solder: Through-Hole Soldering

How to Work with Jumper Pads and PCB Traces

One of the new, advanced features of the board is that it takes advantage of the Qwiic connect system. We recommend familiarizing yourself with the **Logic Levels** and **I^{2}C** tutorials. Click on the banner above to learn more about Qwiic products.

2023-03-22

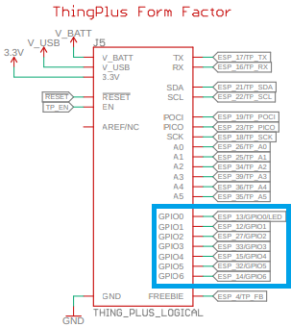
2023-03-22

santaimpersonator

1.2 Hardware Overview

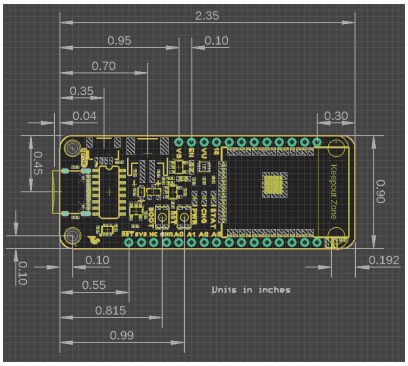
Info

All mentions of `GPIO` in this section will refer to the I/O pins of the ESP32-WROOM module as represented in the datasheets and pin numbers of the board definition in the ESP32 Arduino core. They do not correspond with the net names for the *ThingPlus Form Factor* device in the schematic. *(The device in the schematic is primarily, used internally to facilitate the board design process; just ignore the naming of the `GPIO0` - `GPIO6` nets.)*



1.2.1 Board Dimensions

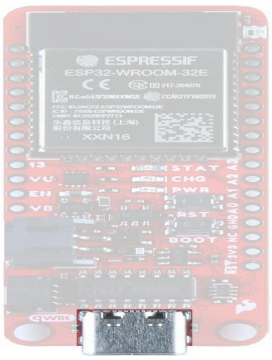
The board dimensions are illustrated in the drawing below. The listed measurements are in inches and the two mounting holes are compatible with 4-40 standoff screws.



Board dimensions (PDF) for the ESP32-WROOM Thing Plus, in inches.

1.2.2 USB-C Connector

The USB connector is provided to power and program the board. For most users, it will be the primary programing interface for the ESP32.



USB-C connector on the ESP32-WROOM Thing Plus.

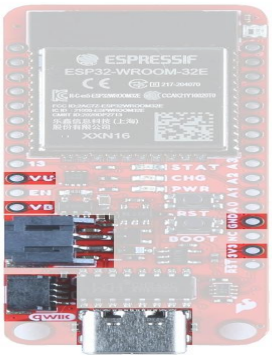
CH340 Serial-to-UART

The CH340 allows the ESP32-WROOM to communicate with a computer/host device through the board's USB-C connection. This allows the board to show up as a device on the serial (or COM) port of the computer. Users will need to install the latest drivers for the computer to recognize the board *(see **Software Overview** section)*.

1.2.3 Power

The ESP32-WROOM Thing Plus only requires **3.3V** to power the board. However, the simplest method to power the board is through the USB-C connector. Alternatively, the `3V3`, `VBAT`, and `VUSB` pins can also be used to supply power to the board.

- `VUSB`:
  - The maximum voltage for the LDOs and charge controller is **6V**.
  - The minimum voltage for the charge controller is **3.75V**.
- `VBAT`:
  - Should not be connected to anything other than a single-cell LiPo battery.
- `3V3`:
  - Requires a **regulated 3.3V**.
  - Only powers the board and not the Qwiic connector.



ESP32-WROOM Thing Plus power connections.

Below, is a general summary of the power circuitry on the board:

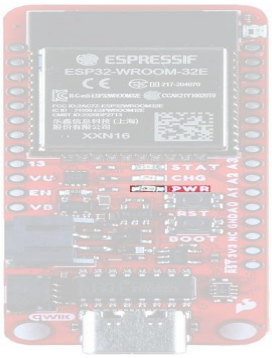
- **`3V3`** - Provides a regulated 3.3V from the USB (5V) power and/or battery connections.
  - Used to power the ESP32-WROOM module, µSD card slot, WS2812 RGB LED, CH340C Serial-to-UART bridge, and power LED.
    - The Qwiic connector is powered by its own voltage regulator, from the same power source(s).
  - The **3.3V** XC6222 LDO regulator can source up to 700mA.
    - Output is controlled by the `EN` pin on the board.
- **`VUSB`** - The voltage from the USB-C connector, usually **5V**.
  - Power source for the entire board.
    - Powers the 3.3V voltage regulators and the battery charging circuit for `VBAT`.
  - Overides power from the battery through a P-channel MOSFET, when both are connected.

    - Utilizes a BAT20J protection diode for the USB-C connection.
- `VBAT` - The voltage from the JST battery connector; meant for single cell LiPo batteries.
    - Provides power to the 3.3V voltage regulators and MAX17048 battery fuel gauge.
    - The MCP73831 linear charge management controller is powered from the USB (5V) power supply.
        - The charge controller is configured for **500mA** *(max)* rate of charge to a connected battery.
- `GND` - The common ground or the 0V reference for the voltage supplies.
- **Qwiic Connector** - Provides a regulated 3.3V voltage from the USB (5V) power and/or battery connections.
    - Operates independently from the `3V3` pin, with its own voltage regulator.
    - The **3.3V** XC6222 LDO regulator can source up to 700mA.
        - Output is controlled by `GPIO 0` of the ESP32-WROOM.

For more details, users can reference the schematic and the datasheets of the individual components in the power circuitry.

Power Status LED

The red, `PWR` LED will light up once **3.3V** is supplied to the board; however, for most users, it will light up when **5V** is supplied through the USB connection or when a LiPo battery is connected to the JST connector.



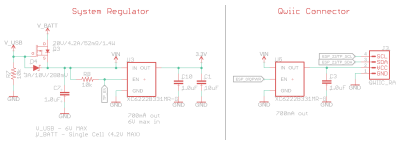
ESP32-WROOM Thing Plus `PWR` status LED indicator.

Charging Circuit

The charging circuit utilizes the MCP73831 linear charge management controller and is powered directly from the USB-C connector or `VUSB`. The controller is configured for a **500mA** charge rate and active charging is indicated by the yellow, `CHG` LED. If the charge controller is shutdown or charging is complete, the `CHG` LED will turn off. For more information, please refer to the MCP73831 datasheet and the **Indicator LEDs** section below.

Power Control

The power source for the XC6222 LDO voltage regulators is controlled by a P-channel MOSFET. In addition, the **3.3V** regulated output from the XC6222 LDOs are enabled by the control pin (`CE`).



Circuits for the 3.3V power on the ESP32-WROOM Thing Plus.

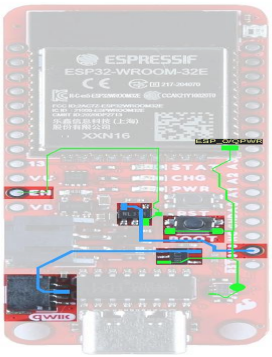
The P-channel MOSFET operates based on the voltages at the MOSFET's gate and source pins. Depending on the power supplies connected to the board, the MOSFET will switch between the battery and USB-C connection as power sources for the XC6222 voltage regulators.

Power Source	MOSFET				Power Control Description
	Gate	Source	$V_{GS} = V_{Gate} - V_{Source}$	MOSFET Operation	
USB Only	$V_{USB} = 5V$	$V_{USB} - V_f$	$V_{USB} - (V_{USB} - V_f)$ $V_{GS} = V_f$	MOSFET Off $R_{GS} = \infty$ Switch Open	**Power to the XC6222 is supplied by the USC-C connection.** Power from the USB-C connection is passed through the Schottky diode. Due to the voltage drop from the Schottky diode, the gate threshold voltage for the MOSFET is positive and equivalent to the diode's forward voltage ($\mathbf{V_f}$).Therefore, the MOSFET behaves as an open switch.
Battery Only	$V_{USB} = 0V$	Dep. Mode: $V_{Source} = 0$ Charged Cap.: $V_{Batt} = 3 - 4.2V$	Dep. Mode: $V_{GS} = 0$ Charged Cap.: $V_{USB} - V_{Batt} = -V_{Batt}$ $-3V > V_{GS} > -4.2V$	MOSFET On R_{GS} = Low Switch Closed	**Power to the XC6222 is supplied from the battery connection.** As a depletion type P-channel MOSFET, the mosfet acts as a normally closed switch when the gate threshold voltage is zero. Therefore, power from the battery is able to charge the capacitor and create a negative gate threshold voltage. The MOSFET remains behaving as a closed switch and power to the XC6222 is provided from the battery.
USB & Battery	$V_{USB} = 5V$	$V_{USB} - V_f$	$V_{GS} = V_f$	MOSFET Off $R_{GS} = \infty$ Switch Open	**Power to the XC6222 is supplied by the USC-C connection.** Power from the USB-C connection is passed through the Schottky diode. Due to the voltage drop from the Schottky diode, the gate threshold voltage for the MOSFET is positive and equivalent to the diode's forward voltage ($\mathbf{V_f}$).Therefore, the

MOSFET behaves as an open switch.

The control pin (`CE`) of the XC6222 LDOs also provides an additional amount of control for the board's power. By default, the regulated 3.3V output is enabled. To disable and shutdown the output voltage from the XC6222, the control pin needs to be pulled low (i.e. shorted to ground (`GND`)). For more information, please refer to the XC6222 datasheet.

- The 3.3V power for the board (`3V3`) is controlled by the `EN` pin, which is broken out on the board.
- The 3.3V power for the Qwiic connector is controlled by `GPIO 0` of the ESP32-WROOM.



XC6222 control pins on the ESP32-WROOM Thing Plus.

Warning

⚡ **Note:** The `BOOT` button is also connected to `GPIO 0`. Therefore, pressing the `BOOT` button will momentarily disable power to the Qwiic connector.

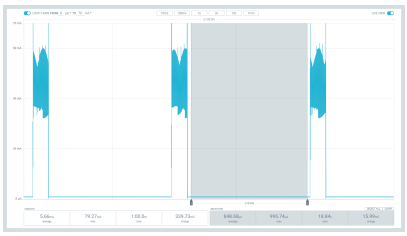
Current Consumption

According to the specifications, the ESP32-WROOM draws about **240 mA** during RF transmissions. With the WiFi example in this tutorial, have measured it to average around **140 mA** and peak at *300 mA* while actively transceiving. The table below, summarizes the approximate current draw of the ESP32-WROOM Thing Plus (USB-C) for various operational conditions. The measurements in the table below, were made with the Nordic Power Profiler Kit II.

	Avg. Current Draw			
Operation	LiPo: **3.5V** *(Low < 5%)*	LiPo: **3.7V** *(~15%)*	LiPo: **4.2V** *(~100%)*	USB-C: **5V** *(No Battery)*
Idle (Blank Code)	63 mA *86 mA (peak)*	63.5 mA *89 mA (peak)*	64 mA *88.6 mA (peak)*	67 mA *89.9 mA (peak)*
Idle: USB + Battery Power	--	--	90.5 µA *721 µA (peak)*	N/A
Idle: Battery Charging *(Current from USB-C)*	395 mA *420 mA (peak)*	590 mA *600 mA (peak)*	> 110 mA *(before shutdown)*	N/A
RGB *(White @ 100%)*	78.4 mA *105.9 mA (peak)*	79 mA *106.8 mA (peak)*	79.9 mA *105.3 mA (peak)*	82.5 mA *108.2 mA (peak)*
WiFi Example *(Transceiving)*	135 mA *295 mA (peak)*	137 mA *310 mA (peak)*	137 mA *307 mA (peak)*	140 mA *300 mA (peak)*

Deep Sleep Example *(MCU Inactive)*	2.5 mA *2.95 mA (peak)*	2.5 mA *3 mA (peak)*	2.55 mA *3 mA (peak)*	2.85 mA *3.3 mA (peak)*
Deep Sleep Example *(MCU Inactive + Jumpers Cut)*	842 µA *1.24 mA (peak)*	848 µA *1.23 mA (peak)*	866 µA *1.24 mA (peak)*	1.19 mA *1.58 mA (peak)*

It is possible for users to reach sub-mA power consumption levels with the deep sleep power modes. Using the `TimerWakeUp` Deep Sleep example code, the LED jumpers cut, and powering the board through the LiPo battery connection we measured a power consumption of **845 µA** *(990 µA peak)* @ **3.7V** while the MCU was inactive.



*The current measurement from `VBAT` at **3.7V** during deep sleep.*

1.2.4 ESP32-WROOM

This variant of the ESP32 Thing Plus is designed around the ESP32-WROOM module with 16MB of flash memory. Espressif's ESP32-WROOM module is a versitile, WiFi+BT+BLE MCU module that targets a wide variety of applications. At the core of this module is the ESP32-D0WDQ6 system on a chip (SoC) which is designed to be both scalable and adaptive. Its laundry list of features include:

- Xtensa® Dual-Core 32-bit LX6 Microprocessor *(up to 240MHz)*
  - 448KB ROM and 520KB SRAM
  - 16MB of Embedded SPI Flash Storage
- Cryptographic Hardware Accelerators
  - AES, SHA2, ECC, RSA-4096
- Integrated 802.11 b/g/n WiFi 2.4GHz Transceiver *(up to 150Mbps)*
- Integrated dual-mode Bluetooth (Bluetooth v4.2 and BLE)
- 26 GPIO *(including strapping pins)*
  - 8x Capacitive Touch Electrodes
- Operating Voltage: **3.0 to 3.6V**
  - WiFi: 380mA *(peak)*
  - Light-Sleep: 800µA
  - Deep-Sleep: 10 - 150µA

ESP32-WROOM module on the ESP32 Thing Plus (USB-C).

Warning

Users should be aware of the following nuances and details of this board

- The ESP32-WROOM is only compatible with **2.4GHz WiFi** networks; it will not work on the 5GHz bands.
- For details on the boot mode configuration, please refer to section **3.3 Strapping Pins** of the ESP32-WROOM module datasheet.

Info

The ESP32-WROOM module has various power modes:

- **Active** - The chip radio is powered on. The chip can receive, transmit, or listen.
- **Modem Sleep** - The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.
- **Light Sleep** - The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP coprocessor are running.
- **Deep Sleep** - Only the RTC memory and RTC peripherals are powered on. The ULP coprocessor is functional.
- **Hibernation** - Only one RTC timer on the slow clock and certain RTC GPIOs are active.
- **Off** - Chip is powered off

*For more information on the power management of the ESP32-WROOM module, pleaser refer to **Section 3.7** and **Tables: 8 and 17** of the ESP32 SoC Datasheet.*

Debugging

For users interested in debugging their code, the JTAG pins are broken out on the board. However, the debugging feature is only available through the ESP-IDF.

- `TMS: GPIO 14`
- `TDI: GPIO 12`
- `TCK: GPIO 13`

- **TDO :** `GPIO 15`

> **Info**
>
> Users should be aware that `GPIO 13` is connected to the `STAT` LED with a pull down resistor.

Firmware Download Mode

Users can manually force the board into the serial bootloader with the `BOOT` button. Please, refer to the **Boot Button** section below for more information.

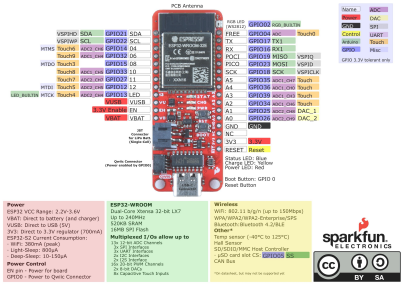
1.2.5 Peripherals and I/O

> **Warning**
>
> **Note:** Users should be aware of the following nuances of this board
>
> - ⚡ All the GPIO on the ESP32-WROOM Thing Plus are **3.3V** pins.
>   - The I/O pins are **not 5V-tolerant**! To interface with higher voltage components, a **logic level adapter** is recommended.
> - ⚡ There are electrical limitations to the amount of current that the ESP32-WROOM module can sink or source. For more details, check out the ESP32-WROOM module datasheet.
> - There are some limitations to the ADC performance, see the **Note** from the **ADC Characteristics** section of the ESP32 SoC datasheet.

The ESP32-WROOM module has *26 multifunctional GPIO*, of which, **21 I/O pins** broken out into a feather form factor layout on this board. All of the ESP32-WROOM Thing Plus (USB-C) pins have a .1" pitch spacing for headers. With the pin multiplexing capabilities of the ESP32 SoC, various pins can have several functionalities. For more technical specifications on the **I/O** pins, please refer to the ESP32 SoC datasheet.

- 13x 12-bit analog to digital converter (ADC) channels
- 3x UARTs (only two are configured by default in the Arduino IDE, one UART is used for bootloading/debug)
- 3x SPI (only one is configured by default in the Arduino IDE)
- 2x I^2C (only one is configured by default in the Arduino IDE)
- 2x I^2S Audio
- 2x digital-to-analog converter (DAC) channels
- 16x 20-bit PWM outputs
- 8x Capacitive Touch Inputs

SparkFun ESP32 Thing Plus (USB-C) (WRL-20168)



Graphical datasheet for the ESP32-WROOM Thing Plus (USB-C).

Info

Users should be aware of the following limitations for the board in the Arduino IDE.

- Not all of the features, listed above, are available in the Arduino IDE. For the full capabilities of the ESP32, the Espressif IDF should be utilized.
- Only one I^2C bus is defined.
- Only two UART interfaces are available.
  - **UART (USB):** `Serial`
  - `RX` / `TX` **Pins:** `Serial1`
- Only one SPI bus is defined.

For digital pins, users will need to declare the `pinMode()` *(link)* in the setup of their **sketch** (programs written in the Arduino IDE) for the pins used.

INPUT

When configured properly, an **input** pin will be looking for a **HIGH** or **LOW** state. **Input** pins are **High Impedance** and takes very little current to move the input pin from one state to another.

OUTPUT

When configured as an **output** the pin will be at a **HIGH** or **LOW** voltage. **Output** pins are **Low Impedance**: This means that they can provide a relatively substantial amount of current to other circuits.

> **Warning**
>
> ⚡ There are electrical limitations to the amount of current that the ESP32-WROOM module can sink or source. For more details, check out the ESP32-WROOM module datasheet.

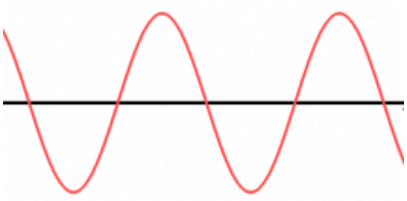
Additional Functions

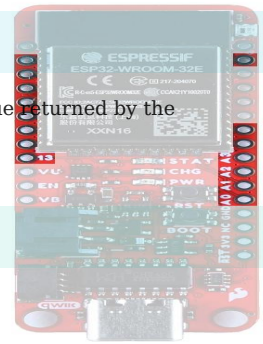
There are several pins that have special functionality in addition to general **digital I/O**. These pins and their additional functions are listed in the tabs below. For more technical specifications on the **I/O** pins, you can refer to the schematic, ESP32-WROOM module datasheet, ESP32 SoC datasheet, and documentation for the ESP32 Arduino core.

Analog Inputs | PWM & Analog (DAC) Outputs | Serial Pins | SPI Pins | I^2C Pins

The provides a **12-bit ADC** input on thirteen of its I/O pins. This functionality is accessed in the Arduino IDE using the `analogRead(pin)` function. *(*The available ADC pins are highlighted in the image below.)*

> **Info**
>
> By default, in the Arduino IDE, `analogRead()` returns a 10-bit value. To change the resolution of the value returned by the `analogRead()` function, use the `analogReadResolution(bits)` function.

> **Tip**
>
> To learn more about analog vs. digital signals, check out this great tutorial.
>
> Analog vs. Digital
>
> 



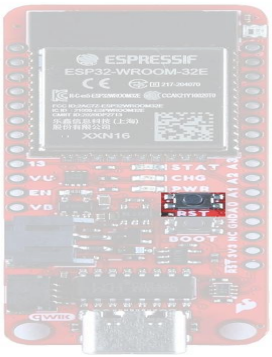
Analog input pins on the ESP32-WROOM Thing Plus.

1.2.6 Buttons

There are two buttons on ESP32-WROOM Thing Plus; a RST and BOOT button.

Reset Button

The RST *(reset)* button allows users to reset the program running on the ESP32-WROOM module without unplugging the board.

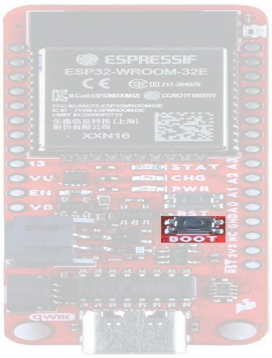


RST button on the ESP32-WROOM Thing Plus.

Boot Button

The BOOT button can be used to force the board into the serial bootloader. Holding down the BOOT button, while connecting the board to a computer through its USB-C connector or resetting the board will cause it to enter the Firmware Download mode. The board will remain in this mode until it power cycles (happens automatically after uploading new firmware) or the RST button is pressed.

1. Hold the BOOT button down.
2. Reset the MCU.
    - While unpowered, connect the board to a computer with through the USB-C connection.
    - While powered, press the RST button.
3. Release the BOOT button.
4. After programming is completed, reboot the MCU.
    - Press the RST button.
    - Power cycle the board.



BOOT button on the ESP32-WROOM Thing Plus.

> **Warning**
>
> ⚡ The BOOT button is also connected to `GPIO 0`, which controls the voltage output to the Qwiic connector. Therefore, pressing the BOOT button will momentarily disable power to the Qwiic connector.

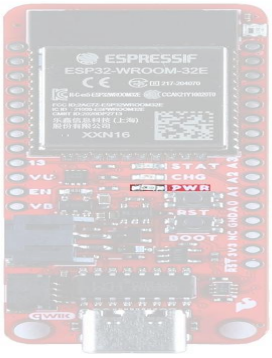
1.2.7 Indicator LEDs

There are four indicator LEDs on the ESP32-WROOM Thing Plus:

- **`PWR`:** Power (Red)
- **`CHG`:** Battery Charging (Yellow)
- **`13`:** `GPIO 13` (Blue)
- **`WS2812`:** `GPIO 02` (RGB)

Power LED

The red, power (`PWR`) LED will light up once **3.3V** is supplied to the board. For most users, it will light up when **5V** is supplied through the USB connection and/or when a LiPo battery is attached to the JST connector.

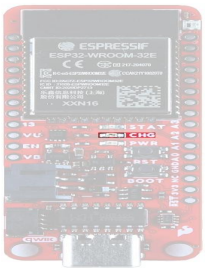


ESP32-WROOM Thing Plus `PWR` status LED indicator.

Battery Charging LED

The yellow, battery charging (`CHG`) LED indicates the status of the MCP73831 charge management controller. The LED will shut off when no battery is present, when the charge management controller is in standby (*after the battery charging has been completed*), or when the charge management controller is shutdown. The LED will illuminate when the charge management controller is in the process of charging the battery. For more information, please refer to the MCP73831 datasheet.

Charge Cycle State	LED
Shutdown • Thermal Shutdown • $V_{DD} < V_{BAT}$	**Off** (High Z)
No Battery Present	**Off** (High Z)
	Off (H)

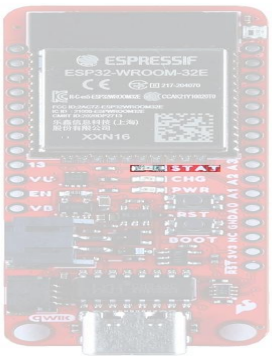


Charge Complete – Standby	
Preconditioning	**On** (L)
Constant-Current Fast Charge	**On** (L)
Constant Voltage	**On** (L)

The battery charging (`CHG`) LED indicator on the ESP32-WROOM Thing Plus. (Click to enlarge)

STAT LED

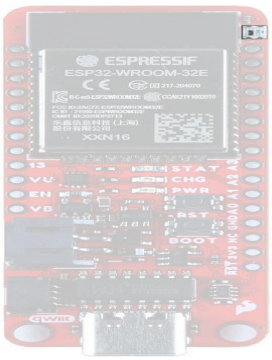
The blue, status (`STAT`) LED is typically used as a test or status LED to make sure that a board is working or for basic debugging. This indicator is connected to `GPIO 13` .



The status (`STAT`) LED indicator on the ESP32-WROOM Thing Plus.

WS2812 RGB LED

The WS2812 RGB LED is controlled with a 24-bit (GRB) data signal. This indicator is connected to `GPIO 02` and the digital output pin from the LED is available through a test point. For more information, please refer to the WS2812C datasheet.



`WS2812` LED indicator on the ESP32-WROOM Thing Plus.

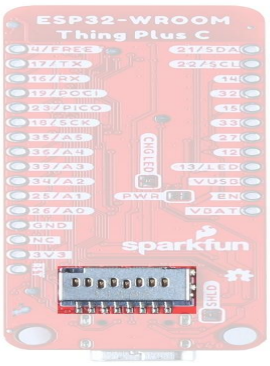
> **Info**
>
> The latest ESP32 Arduino core, now provides a basic RGB LED driver for a WS2812 *(or NeoPixel)* LED populated the board. For an example of how to utilize the RGB LED driver check out the BlinkRGB example code, which can be accessed from the File drop down menu *(i.e **File** > **Examples** > **ESP32** > **GPIO** > **BlinkRGB**)*.

1.2.8 µSD Slot

> **Info**
>
> To comply with the latest OSHW design practices, we have adopted the new SPI signal nomenclature (**SDO/SDI** and **PICO/POCI**). The terms Master and Slave are now referred to as Controller and Peripheral. Please refer to this announcement on the decision to transition to the new naming convention.

The ESP32-WROOM Thing Plus (USB-C) includes an µSD card slot. This is great for data logging applications or storing files. The µSD card slot is connected to the following dedicated GPIO:

- **`GPIO 5`**: `DATA 3` / `CS`
- **`N/A`**: `DATA 2`
- **`N/A`**: `DATA 1`
- **`GPIO 19`**: `DATA 0` / `POCI` (or Peripheral's `SDO`)
- **`GPIO 18`**: `CLK` / `SCK`
- **`GPIO 23`**: `CMD` / `PICO` (or Peripheral's `SDI`)



µSD card slot on the ESP32-WROOM Thing Plus.

1.2.9 Jumpers

There are two jumpers on the back of the board that can be used to easily modify the hardware connections on the board.

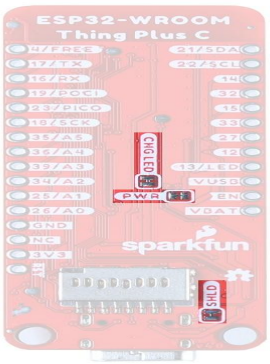
- **SHLD** - This jumper can be used to disconnect the USB shield from `GND`.
- **PWR** - This jumper can be used to remove power to the `PWR` LED.
- **CHG LED** - This jumper can be used to remove power to the `CHG` LED.
    - Avoid cutting the box's silkscreen; there are traces under it:

Traces around the `CHG LED` jumper. (Click to enlarge)

Info

Never modified a jumper before? Check out our Jumper Pads and PCB Traces tutorial for a quick introduction!

How to Work with Jumper Pads and PCB Traces



The jumpers on the back of the ESP32-WROOM Thing Plus.

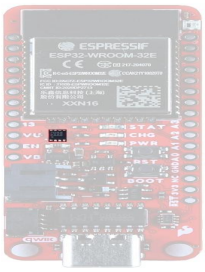
1.2.10 Primary I^{2}C Bus

The Qwiic connector and battery fuel gauge are attached to the primary I^{2}C bus. The primary I^{2}C bus for this board utilizes the pin connections, detailed in the table below:

Connection	VDD	GND	SCL	SDA
Battery Fuel Gauge *(MAX17048)*	VBAT	**GND**	GPIO 22	GPIO 21
Qwiic Connector	GPIO 0 *(Enables 3.3V)*	**GND**	GPIO 22	GPIO 21

Battery Fuel Gauge

The MAX17048 fuel gauge measures the approximate charge or discharge rate, state of charge (SOC) (based on ModelGauge algorithm), and voltage of a connected battery. Additionally, the chip is powered directly from `VBAT`, when a LiPo battery is connected. For more information, please refer to the MAX17048 datasheet.



The MAX17048 fuel gauge on the ESP32-WROOM Thing Plus. (Click to enlarge)

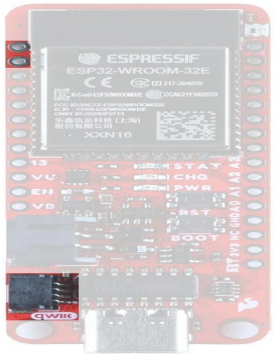
I^2C Address	**0x36** (7-bit) **0x6C** (write)/**0x6D** (read)
Voltage Measurement	Range: 2.5 - 5 V Precision: ±7.5 mV/Cell Resolution 1.25 mV/Cell
Current Consumption	Sleep: .5 - 2 µA Hibernate: 3 - 5 µA Active: 23 - 40 µA

Info

The `Alert` pin for the MAX17048 is not connected and cannot be utilized.

Qwiic Connector

A Qwiic connector is provided for users to seamlessly integrate with SparkFun's Qwiic Ecosystem.



Qwiic connector and I^{2}C pins on the ESP32-WROOM Thing Plus.

POWER CONTROL

In order to enable power for the Qwiic connector, users must toggle `GPIO 0` high. This enables the power output from the XC6222 LDO regulator to the Qwiic connector, which can sources up to **700mA** at **3.3V** . In order to conserve battery power or in low power applications, users will can toggle `GPIO 0` low, to disable the power to the Qwiic connector.

> **Warning**
>
> ⚡ `GPIO 0` is also connected to the `BOOT` button. Therefore, pressing the `BOOT` button will momentarily disable power to the Qwiic connector.

WHAT IS QWIIC?

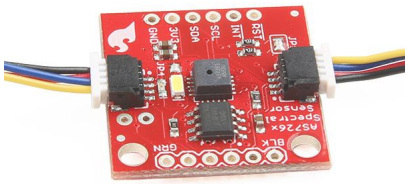
The Qwiic system is intended a quick, hassle-free cabling/connector system for I^{2}C devices. Qwiic is actually a play on words between "quick" and I^{2}C or "iic".

Features of the Qwiic System

No Soldering | Polarized Connector | Daisy Chain

Keep your soldering iron at bay

Cables plug easily between boards making quick work of setting up a new prototype. We currently offer three different lengths of Qwiic cables as well as a breadboard friendly cable to connect any Qwiic enabled board to anything else. Initially you may need to solder headers onto the shield to connect your platform to the Qwiic system but once that's done it's plug and go!



Qwiic cables connected to Spectral Sensor Breakout

2023-03-22

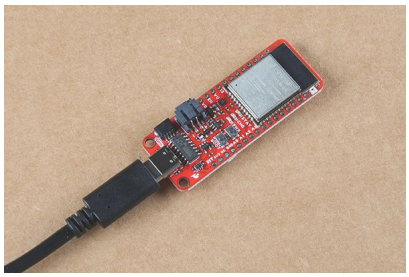
2023-03-22

santaimpersonator

1.3 Hardware Assembly

1.3.1 USB Programming

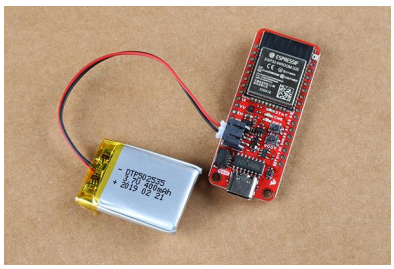
The USB connection is utilized for programming and serial communication. Users only need to plug their ESP32-WROOM Thing Plus into a computer using a USB-C cable.



The ESP32-WROOM Thing Plus with USB-C cable attached.

1.3.2 Battery

For remote IoT applications, a Li-Po battery can be connected. Additionally, users may be interested in utilizing a solar panel and USB-C cable to recharge their battery.



The ESP32-WROOM Thing Plus with a battery connected. (Click to enlarge)

Solar Panel Charger - 10W

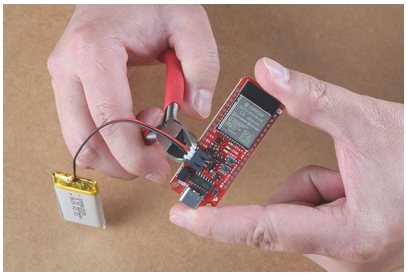
TOL-16835

USB 3.1 Cable A to C - 3 Foot

TOL-14743

Warning

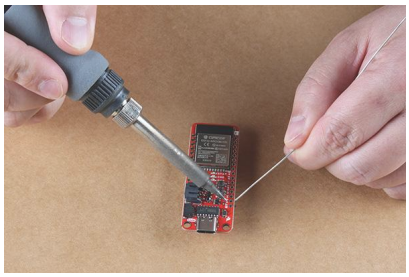
DO NOT remove batteries by pulling on their wires. Instead, it is recommended that pair of dikes (i.e. diagonal wire cutters), pliers, or tweezers be used to pull on the JST connector housing, to avoid damaging the battery wiring.



Using a pair of dikes to disconnect a battery. (Click to enlarge)

1.3.3 Headers

The pins for the ESP32-WROOM Thing Plus are broken out to 0.1"-spaced pins on the outer edges of the board. When selecting headers, be sure you are aware of the functionality you need. If you have never soldered before or need a quick refresher, check out our How to Solder: Through-Hole Soldering guide.

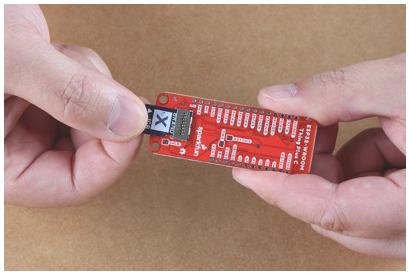


Soldering headers to the ESP32-WROOM Thing Plus.

The Feather Stackable Header Kit is a great option as it allows users to stack shields (*w/ Feather footprint*) or it can be placed on a breadboard; while the pins are still accessible from the female/male headers.

µSD Card Slot

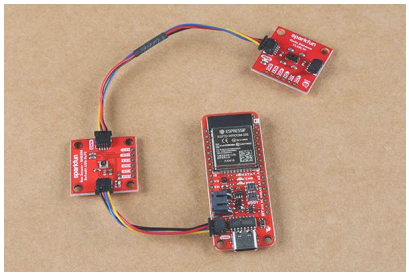
The ESP32-WROOM Thing Plus (USB-C) includes an µSD card slot on the back of the board. The cardholder functions through a push/pull operation. *(The card slot doesn't include a spring retention mechanism; cards are held in place through friction.)*



Users can slide-in or pull-out a µSD card from the cardholder.

Qwiic Devices

The Qwiic system allows users to effortlessly prototype with a Qwiic compatible I^2C device without soldering. Users can attach any Qwiic compatible sensor or board, with just a Qwiic cable. (**The example below, is for demonstration purposes and is not pertinent to the board functionality or this tutorial.*)



The BME688 environmental and VL53L1X distance Qwiic sensor boards connected to the ESP32-WROOM Thing Plus.

2023-03-22

2023-03-22

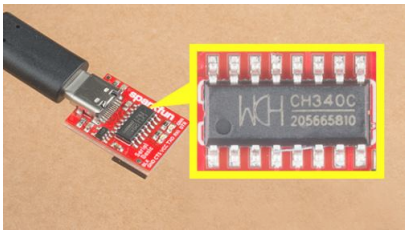
santaimpersonator

1.4 Software Overview

CH340 Driver

Users will need to install the appropriate driver for their computer to recognize the serial-to-UART chip on their board/adapter. Most of the latest operating systems will recognize CH340C chip on the board and automatically install the required driver.

To manually install the CH340 driver on their computer, users can download it from the WCH website. For more information, check out our How to Install CH340 Drivers Tutorial.



How to Install CH340 Drivers

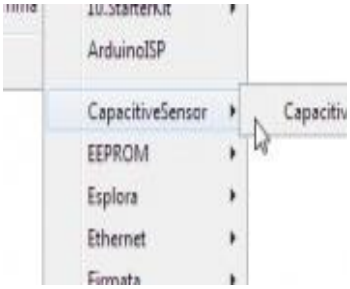
Arduino IDE

> **Info**
>
> For first-time users, who have never programmed before and are looking to use the Arduino IDE, we recommend beginning with the SparkFun Inventor's Kit (SIK), which includes a simpler board like the Arduino Uno or SparkFun RedBoard and is designed to help users get started programming with the Arduino IDE.

Most users may already be familiar with the Arduino IDE and its use. However, for those of you who have never heard the name *Arduino* before, feel free to check out the Arduino website. To get started with using the Arduino IDE, check out our tutorials below:

What is an Arduino?

Installing Arduino IDE

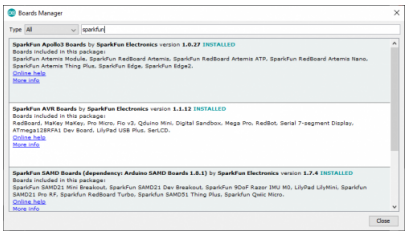


Installing an Arduino Library

Installing Board Definitions in the Arduino IDE

INSTALL BOARD DEFINITION

Install the latest **ESP32** board definitions in the Arduino IDE.



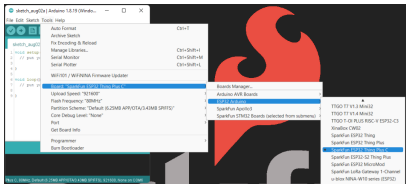
Installing Board Definitions in the Arduino IDE

Info

For more instructions, users can follow this tutorial on Installing Additional Cores provided by Arduino. Users will also need the `.json` file for the Espressif Arduino core:

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

When selecting a board to program in the Arduino IDE, users should select the **SparkFun ESP32 Thing Plus C** from the Tools drop-down menu *(_i.e.* ***Tools > Board > ESP32 Arduino > SparkFun ESP32 Thing Plus C****)*. Alternatively, users can also select the **ESP32 Dev Module**; however, they may lose some pin assignments (i.e. `LED_BUILTIN`).



Select the ***SparkFun ESP32 Thing Plus C*** *from the Tools drop-down menu in the Arduino IDE.*

2023-03-22

2023-03-22

santaimpersonator

2. Resources

2.1 Product Resources

- Product Page
- Schematic (PDF)
- Eagle Files (ZIP)
- Board Dimensions (PDF)
- Graphical Datasheet (PDF)
- SFE Product Showcase Video
- GitHub Hardware Repo

2.1.1 Additional Resources

- SparkFun Learn - Hookup Guide
- SparkFun Thing Plus Boards
- SparkFun Qwiic Connect System
- SparkFun Technical Assistance

2.2 Hardware Component Documentation

- ESP32 Module
  - ESP32-WROOM Module (PDF)
    - ESP32 SoC (PDF)
- Power Components
  - Voltage Regulator: XC6222 (PDF)
  - Battery:
    - Charge Controller: MCP73831 (PDF)
    - Fuel Gauge: MAX17048 (PDF)
- LED: WS2812 (PDF)

2.3 Manufacturer's Resources

Espressif also provides great resources for their ESP32 module:

- ESP32 Product Resource Page
- Espressif's Forum for the ESP32
- Espressif GitHub Repositories
  - ESP32 Arduino Core
    - `.json` file needed for Epressif's ESP32 Arduino Core:
      `https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json`
  - ESP-IDF -- IoT Development Framework
    - Programming Guide.

2023-03-22

2023-03-22

santaimpersonator

3. Support

3.1 Troubleshooting Tips

Need Help?

If you need technical assistance or more information on a product that is not working as you expected, we recommend heading on over to the SparkFun Technical Assistance page for some initial troubleshooting.

SparkFun Technical Assistance Page

If you can't find what you need there, the SparkFun Forums is a great place to search product forums and ask questions.

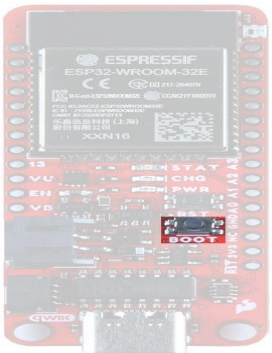
Account Registration Required

If this is your first visit to our forum, you'll need to create a Forum Account to post questions.

Upload Issues

If users have issues during the uploading process, they can try to manually force the board into the serial bootloader with the BOOT button. Holding down the BOOT button, while connecting the board to a computer through its USB-C connector or resetting the board will cause the MCU to enter the Firmware Download mode and its serial bootloader. The board will remain in this mode until it power cycles (which happens automatically after uploading new firmware) or the RST button is pressed.

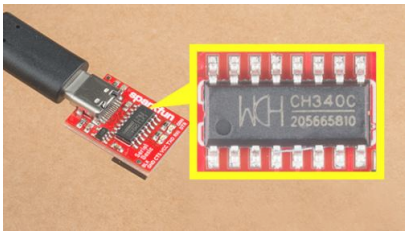
1. Hold the BOOT button down.
2. Reset the MCU.
    - While unpowered, connect the board to a computer through the USB-C connection.
    - While powered, press the RST button.
3. Release the BOOT button.
4. After programming is completed, reboot the MCU.
    - Press the RST button.
    - Power cycle the board.



BOOT *button on the ESP32-WROOM Thing Plus.*

COM Port Not Shown

If the board doesn't appear on a COM port, double check the correct driver has been installed. Unlike previous versions of the ESP32 Thing Plus, this variant requires the CH340 driver to be installed. *For more information, check out our How to Install CH340 Drivers Tutorial.*



How to Install CH340 Drivers

Users can also check their USB cable; some cables are power only. Try testing the cable with a smartphone or tablet to see if it appears as a device on the computer. If the phone/tablet doesn't appear, then the USB cable is power only.

`Serial` Stream Difficulties

We have noticed that with the ESP32 Arduino core, `Serial.available()` does not operate instantaneously. This is due to an interrupt triggered by the UART, to empty the FIFO when the `RX` pin is inactive for two-byte periods:

- At 9600 baud, `hwAvailable` takes [`number of bytes received` + 2] x 1 ms = **11 ms** before the UART indicates that data was received from: `\r\nERROR\r\n`.
- At 115200 baud, `hwAvailable` takes [`number of bytes received` +2] x .087 ms = **~1 ms** before the UART indicates that data was received from: `\r\nERROR\r\n`.

For more information, please refer to this chatroom discussion.

µSD Card

Make sure that the µSD card is compatible with the Arduino library being used for it. For example, the default SD Arduino library is only compatible with `FAT16` or `FAT32` file systems; therefore, the card capacity is limited to **16GB** or **32GB** and smaller. Another consideration is that the library was also written to only handle short 8.3 names for files.

Qwiic Connector Power

For users having issues with the power to their Qwiic devices, don't forget that `GPIO 0` controls the power output from the XC6222 LDO regulator to the Qwiic connector. Users must toggle `GPIO 0` high to enable power for the Qwiic connector. In order to conserve battery power or in low-power applications, users will toggle `GPIO 0` low, to disable the power to the Qwiic connector.

> **Note**
>
> `GPIO 0` is also connected to the BOOT button. Therefore, pressing the BOOT button will momentarily disable power to the Qwiic connector.

Current Consumption

For ultra-low power projects, these are the current consumption of the individual components, as specified in their datasheet:

- XC6222 LDO Regulator:
  - Supply Current: 100 - 220 µA

- MCP73831 Charger Controller:
  - Supply Current:
    - 510 - 1500 µA (Charging)
    - 53 - 200 µA (Charge complete; no battery)
  - Constant-Voltage Mode
    - Line/Load regulation: 100 - 50 mA
  - Fast Charge Constant-Current Mode
    - Fast Charge Current: 450 - 550 mA
  - Battery Detection Current: 6 µA
  - Leakage Current: up to 2µA
  - Status Indicator:
    - Sink Current: 25 mA
- MAX17048 Fuel Gauge:
  - Supply Current:
    - Sleep: 0.5 - 2 µA
    - Hibernate: 3 - 5 µA
    - Active: 23 - 40 µA
  - I^2C: 0.2 - 0.4 µA
- CH340C Serial-to-UART Bridge:
  - Supply Current: 4 - 12 mA
    - USB Suspended: 0.04 - 0.15 mA
- ESP32 SoC:
  - Rec Supply current: 500 mA
  - Active: 95 - 240 mA
    - w/ RF Transceiver:
      - TX: up to 380 mA
      - RX: Up to 118 mA
  - Sleep Modes:
    - Modem: 20 - 68 mA
    - Light: .8 mA
    - Deep: 10 - 150 µA
    - Hibernation: 5 µA
    - Off: 1µA
- WS2812 RGB LED:
  - Supply Current: 1µA (@5V)
  - LEDs: 5mA each (@5V)

2023-03-22

2023-03-22

santaimpersonator