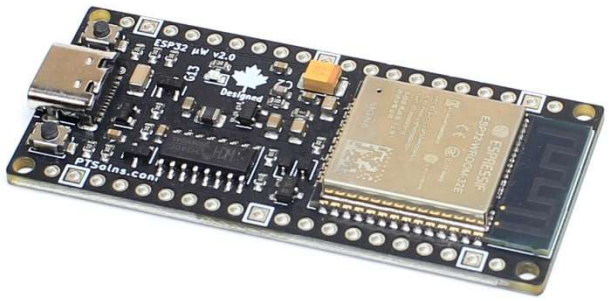
ESP32 microWatt

1 DESCRIPTION



The PTSolns *ESP32 microWatt* is a low power development board specifically designed for IoT (Internet-of-Things) and related applications. Equipped with both Wi-Fi® and Bluetooth® LE capabilities facilitated by an embedded ESP32 module. The board can be powered through either USB-C® or an external voltage input ranging from 3.5V to 6.0V, rendering it particularly suitable for single-cell 3.7V/4.2V LiPo battery applications.

The *ESP32 microWatt* has an intuitive pinout, collecting like pins in groups such as the I2C and SPI pins. Packed with many unique features, this development board is ideal for hobbyists, educators and professionals for prototyping and testing their projects.

The *microWatt Support Library* (mSL) helps novice users with various tasks and provides important product information. The mSL combined with a collection of supporting material relating to ESP32 development and reduces the barrier-to-entry into the world of IoT development.

The *ESP32 microWatt* is part of the *microWatt family*, which consists of several products designed to work together. These products, when combined offer additional features such as LiPo charging via PV solar, long distance communication, and custom prototyping.

Table of Contents

2 DOCUMENT REVISION HISTORY

Current document revision is Rev 3.

Changes to Rev 3:

- Fixed typos in URL links.

Changes to Rev 2:

- Upgraded FCC and IC compliance statement (fully compliant).

Changes to Rev 1:

- Edit of a typo on pinout and power tree diagrams.

3 PRODUCT FEATURES

This section highlights notable features of the *ESP32 microWatt.*

3.1 microWatt Support Library (mSL)

The *microWatt Support Library* (mSL) is intended to help novice users of the ESP32 module. Although it is not required to install and use the mSL, it is recommended. More details about the mSL, how to download and use it are outlined in Section 6.3.

3.2 ESP32 Module

The core of the *ESP32 microWatt* is the ESP32 module by Espressif Systems. The particular ESP32 module used onboard is:

ESP32-WROOM-32E-N8

This module has the following key features:

1) 2.4 GHz Wi-Fi®, up to 150 Mbps
2) Bluetooth® LE
3) Xtensa® dual core 32-bit LX6 microprocessor, up to 240MHz
4) 448 KB ROM, 520KB SRAM, 16 KB SRAM in RTC, 8 MB flash
5) Onboard PCB antenna
6) UART, SPI, I2C, Touch, ADC, DAC, PWM and more
7) Certifications: Bluetooth BQB, RF, REACH/RoHS

More details of this module can be found in the original Espressif Systems datasheet, which can be accessed here:

https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

3.3 Input Power Options

The *ESP32 microWatt* can be powered in multiple ways. The two most common methods to provide power are:

1) Vin pin, or
2) USB-C® connector

For full details on powering options see Section 5.1.

3.3.1 Note on USB-C Cables

From a data transfer perspective USB-C cables can be categorized into two types:

1. Data transfer capable
2. Not data transfer capable

Only the first type of USB-C cable can be used to program not just the *ESP32 microWatt*, but indeed any microcontroller. This type of cable provides power to the board, as well as facilitates data transfer between the computer and the board. **Using this type of cable is essential in programming a microcontroller**.

The second type of USB-C cable does not facilitate the transfer of data but can merely be used to power the board. Therefore, this type of USB-C cable cannot be used to program a microcontroller.

How to tell if a USB-C Cable is Data Transfer Capable?

One can easily and quickly check if a particular USB-C cable is data transfer capable by simply trying to program the *ESP32 microWatt*. If the USB-C cable is not data transfer capable, then upon plugging it into the computer with the other end into the *ESP32 microWatt* no port will appear.

3.4 Low Power

The *ESP32 microWatt* is designed such that it can powered by a single-cell 3.7V/4.2V LiPo battery and run for an extended duration before depleting the battery. With this in consideration, the board does not include the standard power LED, as this would unnecessarily drain the battery. Furthermore, the onboard CH340 IC used to communicate between the USB-C port with the ESP32 module is only powered if the USB is connected. Once the USB is disconnected the CH340 is shut off, conserving more power. Finally, in order to save as much battery charge as possible, if an external LiPo battery is connected to the *ESP32 microWatt* via the Vin pin and at the same time a USB cable is plugged in, the USB gets priority, disconnecting the battery.

While in deep sleep, with all external peripherals off, the *ESP32 microWatt* draws current as low as 60µA. This equates to a low power consumption, in the µW (“micro Watt) range, which is the reason for the product name. For more details on power consumption, see Section 5.4.

3.5 Protection

The following are embedded protection into the design of the *ESP32 microWatt*:

1) ESD protection,
2) Reverse polarity protection, and
3) Reverse current protection for USB source

Part of EMC immunity is to protect the board against electrostatic discharges (ESD). In the case of the *ESP32 microWatt*, a potential ESD may come from the USB connection. Therefore, the ESD protection circuitry is located on the USB interface on the board.

If powering the *ESP32 microWatt* from an external source via the Vin pin, there is the potential to connect the positive and negative terminals in reverse. This is particularly the case when working with LiPo batteries that have terminals connected in either positive-negative, or negative-positive orientation. For this reason, the Vin pin is protected by a P-MOSFET connected in the "drain-source" orientation. This provides reverse polarity protection to input voltages much larger than those seen on LiPo or similar batteries. For more information on this, see Section 5.2

The *ESP32 microWatt* is designed for the user to simultaneously connect a USB voltage source and an external voltage source on the Vin pin. If the Vin voltage is larger than the USB voltage, a current could flow from Vin to the USB source. To prevent this a diode is placed between the Vin and USB connector. This provides reverse current protection for current going into the USB source. For more information on this, see Section 5.2.

3.6 Onboard Buttons

There are two push buttons onboard the *ESP32 microWatt*. One is designated as a reset button (RST), while the other is used to enter boot mode (BOOT). When programming the ESP32 these buttons must be pressed with a particular timing and sequence. To simplify this task for the user, onboard components mimic the pressing of these buttons so that the user does not have to do so.

Although the RST button can be pressed by the user any time to restart the ESP32, the BOOT button does not do anything if being pressed while not programming. Therefore, the BOOT button can be used as a general-purpose input button. An example Arduino® IDE sketch demonstrating this can be found on the in the *microWatt Support Library* (mSL) package (see Section 6.3 for details on the mSL). For further details on the onboard buttons, see Section 3.9.

3.7 Onboard LEDs

There is no onboard power LED in order to conserve as much power as possible. However, there is a general-purpose LED that can be turned ON and OFF via pin G13. For further detail see Section 3.9. For a pinout diagram see Section 3.10.

3.8 Form Factor

The form factor of the *ESP32 microWatt* allows the board to be interfaced on a standard breadboard and still have one row on either side available for prototyping. Furthermore, the *ESP32 microWatt* is compatible with the PTSolns *ESP Master Key*. For physical dimensions of the *ESP32 microWatt* see Figure 4.

3.9 Block Diagram

The *ESP32 microWatt* block diagram can be found in Figure 1. This diagram gives an overview of the board hierarchy.

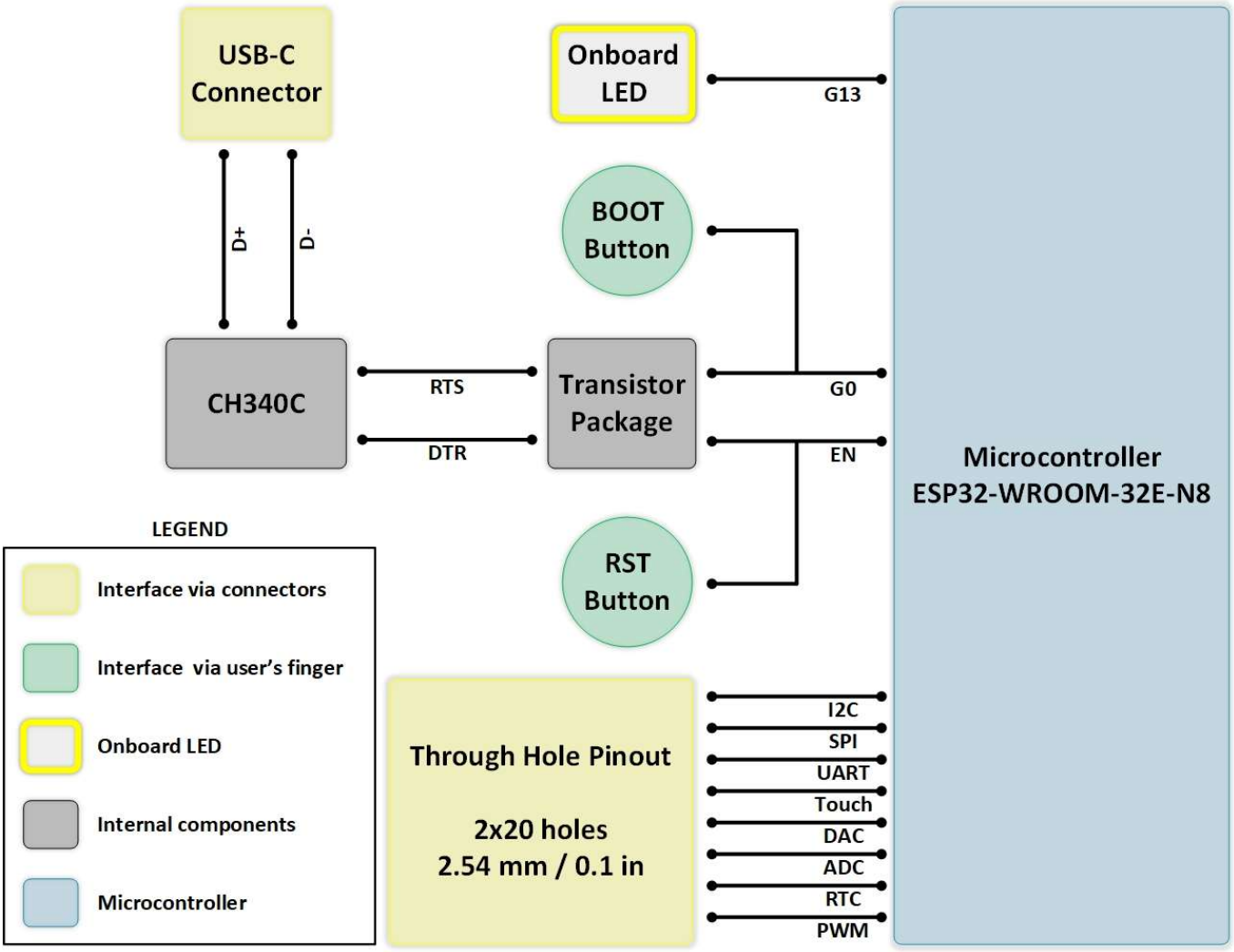
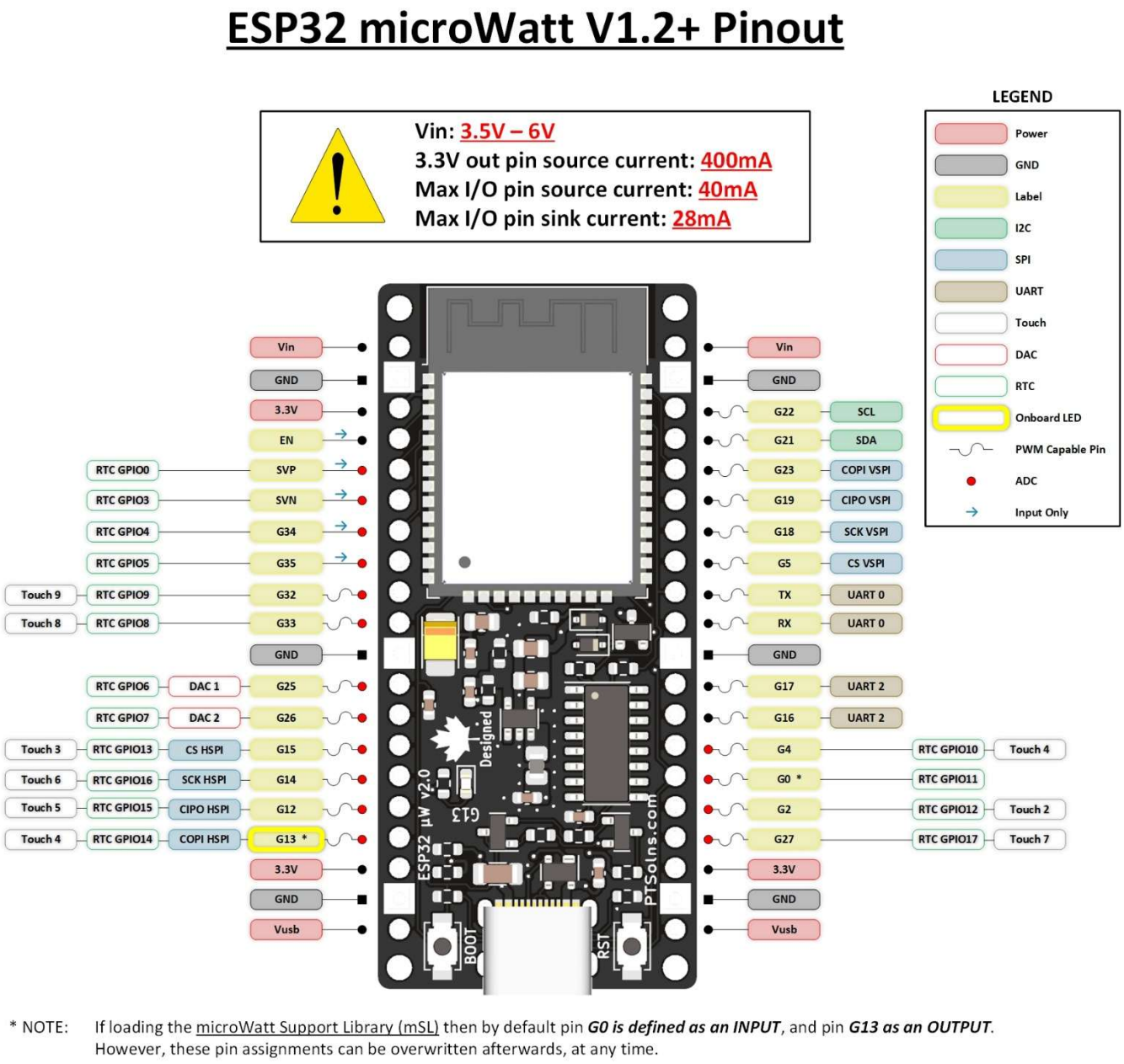


Figure 1: ESP32 microWatt Block Diagram.

3.10 Pinout Diagram

The pinout diagram of the *ESP32 microWatt* is shown in Figure 2.



* NOTE: If loading the microWatt Support Library (mSL) then by default pin ***G0 is defined as an INPUT***, and pin ***G13 as an OUTPUT***. However, these pin assignments can be overwritten afterwards, at any time.

Figure 2: ESP32 microWatt Pinout Diagram.

Not all pins of the ESP32 microcontroller are made available as input or output pins on the *ESP32 microWatt*.

Many pins have multiple applications and usages, not all of which are listed in the pinout diagram in Figure 2 However, the most common and often-used pins are shown in the pinout diagram. To see a full description of all the pins the reader is referred to the original Espressif Systems datasheet of the ESP32 module, which can be accessed here:

https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

If using the *microWatt Support Library* (mSL) (see Section 6.3), the user simply has to initiate the library by calling the *.begin()* command within the Arduino® IDE environment, and viewing the Serial Monitor on baud 115200 for a printout of the pinout diagram. The following is a sample sketch that demonstrates how to see the pinout diagram from within the Arduino® IDE environment.

```
#include <PTSolns_microWatt.h>

microWatt microWatt;

void setup() {
      microWatt.begin();
}

void loop() {
}
```

3.11 Stackability with the microWatt Family

The *ESP32 microWatt* is part of the microWatt family line of products, that is designed such that multiple products can be stacked together in order to add particular features. The term “stackability” in this sense means these products all have the same form factor and that all the pins of the *ESP32 microWatt* are available at every level of the stack. The following microWatt products can be stacked together:

- *ESP32 microWatt*
- *microWatt Charger*
- *microWatt THT Proto*
- *microWatt LoRa SX1276 915MHz*
- others to be announced.

3.11.1 Headers

The *ESP32 microWatt* does not come assembled/soldered with headers as the user may wish to install a particular set of headers as required. For example, the user can solder a set of 1x20 pin 2.54mm male headers and then use the *ESP32 microWatt* either on a breadboard, or with the *ESP Master Key*. Alternatively, the user can install 1x20 pin 2.54mm stacking female headers. This allows multiple microWatt family member boards to be stacked together. All *ESP32 microWatt* packages come with a set of male and a set of stacking female headers for the user to install as desired.

3.12 Silkscreen Printing

All the pins are labeled on the back of the *ESP32 microWatt*. A pinout diagram can be found in Section 3.10**Error! Reference source not found.**. the front of the board the two ground (GND) pins are marked with a white square around the pin.

3.13 Mark of Authenticity

Authentic PTSolns PCBs have a black solder mask color and are marked with the “PTSolns” logo in white silkscreen printing. The “Canadian Designed” symbol, consisting of the Canadian Maple Leaf with the word “Designed” underneath, can also be found on the PCB in white silkscreen printing. The “PTSolns” trademark and the “Canadian Designed” symbols are shown in Figure 3.

Figure 3: The "Canadian Designed" symbol and the "PTSolns" trademark found on authentic PTSolns PCBs.

4 PHYSICAL PROPERTIES

The physical specifications of the *ESP32 microWatt* are outlined in Table 1, as well as Figure 4.

Table 1: Physical Specifications of the *ESP32 microWatt*.

	Description	Value	Reference
PCB	Length	58.44mm	Figure 4
	Width	26.41mm	Figure 4
	Thickness	1.6mm	Figure 4
	Corner radius	1.0mm	--
	Weight (with components)	8g	--
	Color	Black	--
	Silkscreen	White	--
	Lead free HASL-RoHS surface finish		--
	FR-4 base		--
Through Holes	Number of through holes	2x20	Figure 4
	Pitch	2.54mm/0.1in	Figure 4
	Hole diameter	1.0mm	Figure 4
	Solder pad diameter	1.7mm	Figure 4
Mounting Holes	Hole diameter	2.10mm	Figure 4
	Center-to-center distance length	54.63mm	Figure 4
	Center-to-center distance width	22.86mm	Figure 4

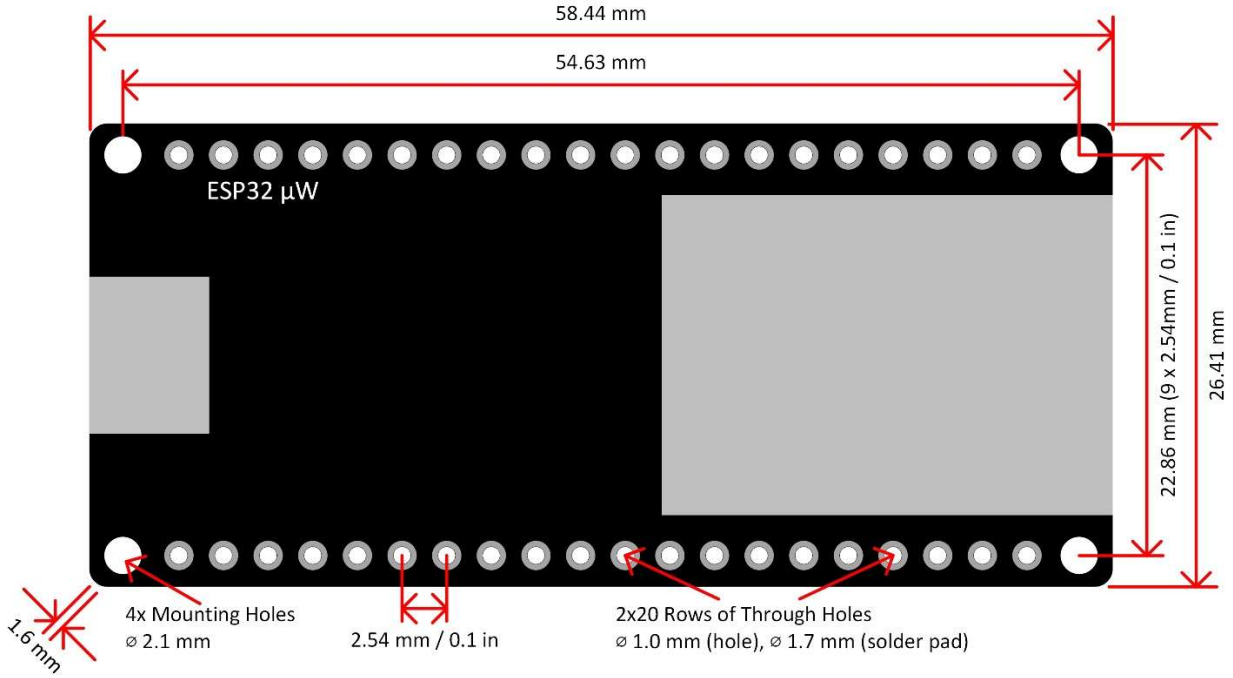


Figure 4: ESP32 microWatt Dimensions.

5 ELECTRICAL PROPERTIES

Details of the *ESP32 microWatt* as they relate to the electrical properties are outlined in the following subsections.

5.1 Power Options

The operating voltage of the board is 3.3V, and any input voltage must be conditioned to this level. The two most common methods to provide power to the *ESP32 microWatt* are:

3) Vin pin, or
4) USB-C® connector

Voltages that are supplied via either of these methods is stepped down (bucked) to the operating voltage of 3.3V using the onboard power management circuitry. **The onboard buck converter is rated for up to 6V and the user must take caution to not exceed this input rating.** See corresponding ratings in Table 3

The *ESP32 microWatt* does not provide a 5V output pin option. However, if a USB is used 5 V (typically) become available via the Vb pin.

An alternative option to power the board can be done by cutting the jumper on the back of the board, and thus isolated the buck and power management circuit. There are three 3.3V pins that can be used to power the board after the jumper is cut. More details are provided in the 4th bullet point in Section 5.2.

5.2 Power Tree

The power tree of the *ESP32 microWatt* is shown in Figure 5. The following are special points to note when considering the power management.

1. The P-MOSFET labelled Q1 in Figure 5 is connected in the “drain-source” configuration. The main advantage of this configuration is that Vin gains reverse-polarity protection (up to the breakdown voltage of the inherent diode within the P-MOSFET, see corresponding ratings in Table 3). A secondary advantage is that while the P-MOSFET is open the voltage drop across the drain to source is very small.

2. The purpose of the diode D1 is to prevent energization of the CH340C IC when no USB is being used. As the CH340C is only used during programming of the board, it can be disconnected when running the board via the Vin option (e.g. to preserve battery charge, if a battery is connected to Vin). Furthermore, the diode D1 provides reverse current protection preventing current from flowing from a Vin connected voltage source into the USB source.

3. If both Vin and USB are connected to a voltage source, the arrangement of the components Q1, D1 and D2 is such that priority will be given to the USB source. This is with consideration of any 3.7V LiPo battery that may be connected to Vin. While the user has the board connected to their computer via USB (e.g. to program it), then there is no need to waste battery power on Vin pin. This power source priority is violated if the Vin voltage exceeds approximately 5.5V, which is not a concern for 3.7V LiPo batteries.

4. Jumper JP1 is closed by default. The user may wish to open this jumper (by carefully cutting the trace with a blade or similar). Doing so separates the buck converter from the microcontroller (as well as the buttons). This then allows the user to use one of the three 3.3V pins to power the microcontroller from an external source. Note that there are several capacitors downstream the buck converter to condition the voltage. Some of these capacitors are contained on either side of the jumper JP1. By opening the jumper, only some of these filtering capacitors are isolated from the load. Also note that by opening the jumper JP1 that the input power options via the Vin pin as well as the USB-C® connecter no longer work. The user can close this jumper again by adding a solder bridge across the jumper pads.

5. The *ESP32 microWatt* does not provide voltage conditioning to output 5V. However, if a USB source is provided the Vb pin will output this voltage, which is typically 5V.

6. There are three 3.3V Pins available for the user to utilize to power external peripherals. **The user should not exceed a current draw of 0.4A on these three 3.3V pins combined.** Note that this current rating excludes any power consumed by the onboard components (e.g. the ESP32 microcontroller). Please see corresponding ratings in Table 3.

   The user should be aware that if the *ESP32 microWatt* is being powered by an external 3.7V/4.2V LiPo battery that likely they will not be able to draw up to the rated current from the 3.3V pins. As the load on any battery increases their terminal voltage decreases. For a 3.7V LiPo battery that is fully charged (typically resulting in an open-circuit terminal voltage of 4.2V), the user may only be able to draw half or less of the rated current on the 3.3V pins before the battery voltage drops below a usable range. A minimum of 3.5V input voltage is required to maintain the operating voltage to 3.3V. An input voltage below 3.5V will result in a sub-nominal 3.3V operating voltage. This will cause the ESP32 microcontroller to not operate properly. The properties that determine how much current can be drawn from the 3.3V pins while maintaining a stable operating voltage depend in large part on the characteristics of the connected battery.

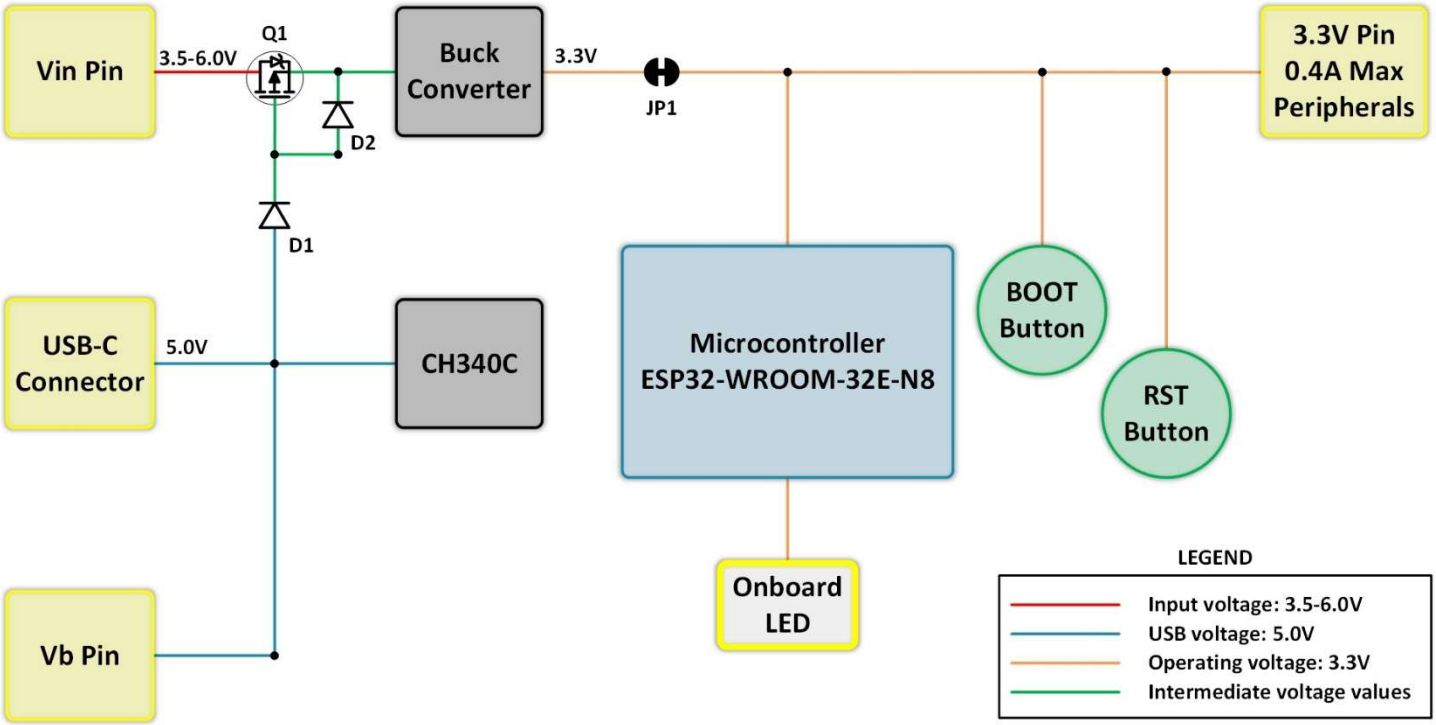


Figure 5: ESP32 microWatt Power Tree.

5.3 Peripherals on the 3.3V Pin

There is a total of three 3.3V pins available for the user to utilize. The intended purpose of these pins is to supply the operating voltage of 3.3V to externally connected peripherals (such as sensors, etc). A total current draw of all three 3.3V pins should not exceed 0.4A combined. Please see corresponding ratings in Table 3 for rating information.

5.4 Power Consumption

Power consumption of the *ESP32 microWatt* depend on various factors that include, but are not limiting to:

- Any externally connected peripherals,
- Onboard LED,
- Transmission and receiving data, or
- ESP32 CPU frequency (default 240 MHz).

With all externally connected peripherals, the onboard LED disabled, and the ESP32 microcontroller put into deep sleep, it has been measured that the *ESP32 microWatt* draws as low as 60µA of current. Table 2 tabulates the current and power consumption of the *ESP32 microWatt* when the ESP32 module is active (but not transmitting/receiving) and when it is put to deep sleep mode for a range of VIN input voltages. These experimental values were measured using specialized current measuring tools that can capture currents in the sub-µA. Furthermore, Figure 6 and Figure 7 plot this data.

Table 2: Current and Power of *ESP32 microWatt*.

	Current		Power	
Vin [V]	**Active [mA]**	**Deep Sleep [µA]**	**Active [mW]**	**Deep Sleep [µW]**
3.7	26.70	87.99	98.79	325.56
3.8	26.17	73.31	99.45	278.58
3.9	25.57	68.47	99.72	267.03
4.0	25.09	66.43	100.36	265.72
4.1	24.74	65.13	101.43	267.03
4.2	24.17	64.22	101.51	269.72
4.3	23.78	63.37	102.25	272.49
4.4	23.37	62.84	102.83	276.50
4.5	22.99	62.47	103.46	281.12
4.6	22.55	61.83	103.73	284.42
4.7	22.29	61.33	104.76	288.25
4.8	21.86	60.67	104.93	291.22
4.9	21.51	60.27	105.40	295.32
5.0	21.67	60.47	108.35	302.35

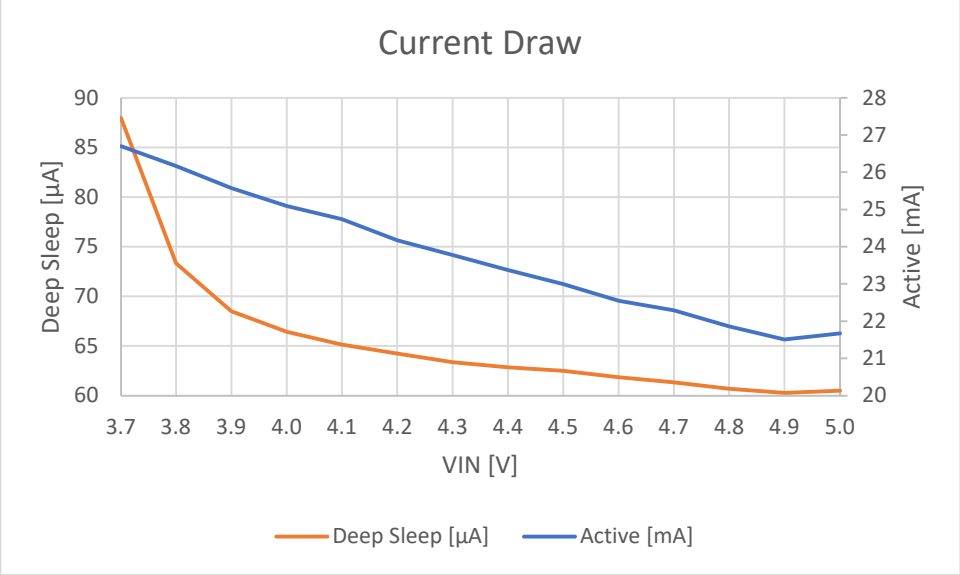


Figure 6: Current draw of the ESP32 microWatt.

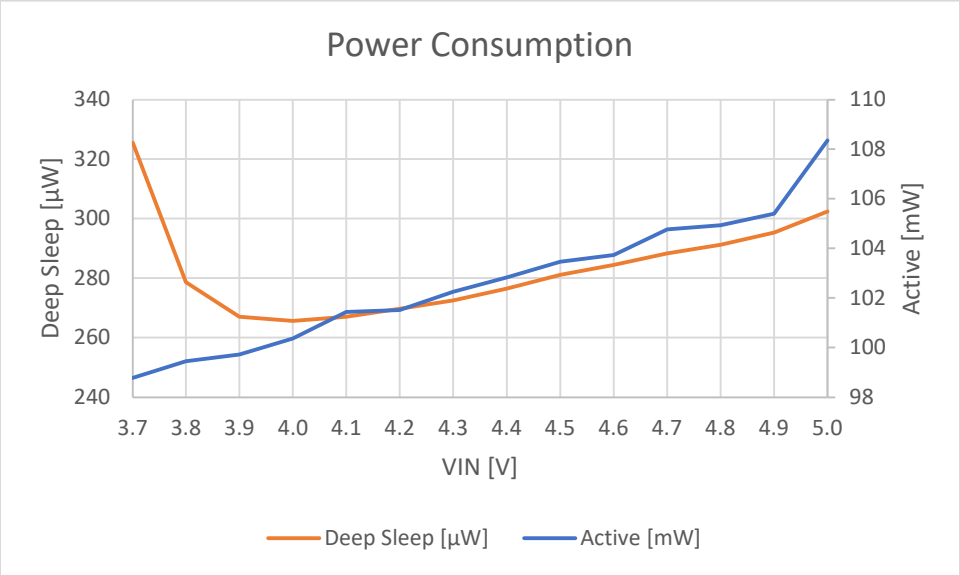


Figure 7: Power consumption of the ESP32 microWatt.

An example Arduino® IDE sketch demonstrating how to put the *ESP32 microWatt* into deep sleep can be found here:

https://github.com/PTSolns/PTSolns_microWatt/tree/main/examples

Or alternatively, by downloading the *microWatt Support Library* (mSL), called PTSolns_microWatt, within the Arduino® IDE environment and loading the corresponding example sketch.

5.5 Ratings

Ratings for the *ESP32 microWatt* are outlined in Table 3. **The user must take caution to not exceed any of these tabulated ratings, as doing otherwise may cause the board to not function properly or cause damage to one or more components.**

Table 3: Ratings for *ESP32 microWatt*.

Description	Value	Unit
Input voltage on Vin pin	3.5 - 6.0	V
Reverse polarity protection on Vin pin	20.0	V
Current draw on all three 3.3V pins combined	0.4	A
Source current on any GPIO	40	mA
Sink current on any GPIO	28	mA
With jumper JP1 open, suppling voltage on one of the 3.3 V pins. See **NOTE** below.	3.0 - 3.6	V

NOTE: If powering the board with the JP1 jumper opened using one of the three 3.3V pin, the user is reminded to take extreme caution to not supply outside the safe range. A voltage too low (less than 3.0V) will cause the ESP32 to not operate properly. A voltage too large (larger than 3.6V) may cause damage to the ESP32.

6 FIRST TIME GETTING STARTED

This section presents important information regarding the first-time usage of the *ESP32 microWatt* and on how to get started specifically within the Arduino® IDE environment.

6.1 CH340 Driver

The *ESP32 microWatt* uses the common CH340 IC to facilitate communication between what is plugged into the USB-C port (e.g. user's laptop) and the microcontroller (ESP32 module). This IC is required when programming the *ESP32 microWatt.* The driver for the CH340 typically must be installed the first time it is needed in any project. Many boards and modules make use of the CH340 so chances are that the driver is already installed. However, if the driver is not yet installed, the user must first install it in order to program the *ESP32 microWatt.* This installation typically only must be done once.

Instructions on how to install the CH340 driver (on a Windows machine) can be found in the following reference:

https://www.youtube.com/watch?v=UUQ84VKg3oM

Additionally, SparkFun Electronics has written a comprehensive tutorial on this topic, and the user is referred to their excellent documentation on this. Find the link here:

https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all

There are many other tutorials available online that can be found by searching "CH340 driver installation instructions" or similar.

6.2 Setting up Arduino® IDE

The free Arduino® IDE software must be installed, which can be downloaded from here:

https://www.Arduino.cc/en/software

Once installed, under File\Preferences in the setting "Additional boards manager URLs" the following link should be pasted (open the pop-up window and paste the link below):

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

To program the *ESP32 microWatt*, select the "ESP32 Dev Module" as the boards, similar as shown in Figure 8. More details on how to configure the Arduino® IDE to work with the ESP32 module can be found on the official Espressif Systems GitHub as follows:

https://github.com/iotechbugs/esp32-Arduino/blob/master/docs/Arduino-ide/boards_manager.md

The user is encouraged to also install the *microWatt Support Library* (mSL) within Arduino® IDE. Open the Library Manager and type "PTSolns_microWatt". This library is not required in order to realize all the features of the *ESP32 microWatt*, but it does provide support and helpful aid while programming the board.

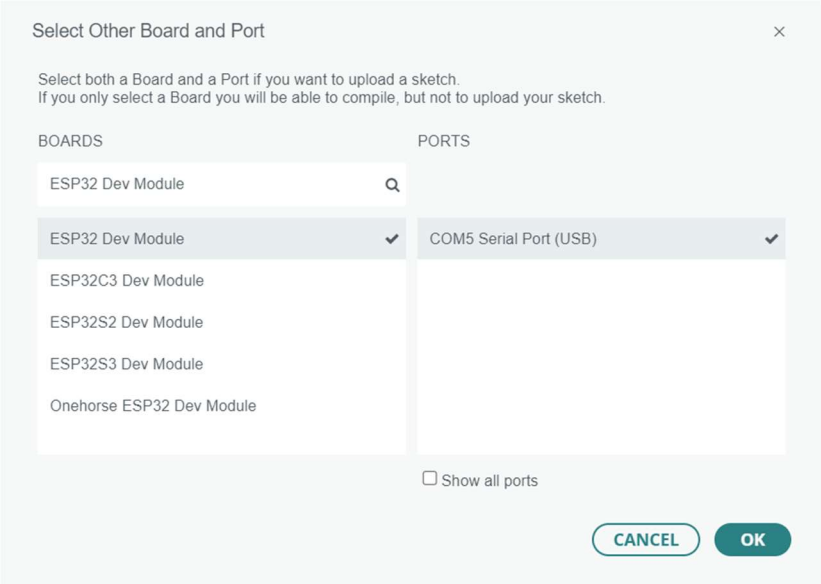


Figure 8: Board Selection within Arduino® IDE.

6.3 microWatt Support Library (mSL)

Although it is recommended that novice users install the *microWatt Support Library* (mSL), this library is not a requirement, or a must, in order to use the *ESP32 microWatt*. The mSL can be used optionally to assist with various tasks or as a general help, and is available on the GitHub repository:

https://github.com/PTSolns/PTSolns_microWatt

Or alternatively, by downloading the *microWatt Support Library* (mSL), called PTSolns_microWatt, within the Arduino® IDE environment and loading the corresponding example sketch.

Various examples are included in this package, including:

- Getting started,
- How to blink and fade,
- How to set I2C pin assignments,
- Using the BOOT button as a general-purpose input button,
- How to set the ESP32 into deep sleep,
- And more

The mSL also provides aid to the user by:

- Printing pinout diagram
- Setting I2C pins
- And more

To submit a bug or improvement relating to the mSL, please submit an issue on the GitHub repository.

7 FCC AND IC/ISED COMPLIANCE

7.1 US FCC Compliance Statement (English)

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

7.2 CA IC/ISED Compliance Statement (English and French)

This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s).

Operation is subject to the following two conditions:

1. This device may not cause interference, and
2. This device must accept any interference, including interference that may cause undesired operation of the device.

Cet appareil contient des émetteurs/récepteurs exempts de licence qui sont conformes aux CNR d'Innovation, Sciences et Développement économique Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes:

1. l'appareil ne doit pas produire de brouillage, et
2. l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

8 REFERENCES

This section lists relevant references.

- The *microWatt Support Library* (mSL), can be downloaded from GitHub:

  https://github.com/PTSolns/PTSolns_microWatt/tree/main/examples

  Or alternatively, by downloading the *microWatt Support Library* (mSL), called PTSolns_microWatt, within the Arduino® IDE environment and loading the corresponding example sketch.

- Espressif Systems ESP32 module datasheet:

  https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

- Arduino IDE software (See Section 6.1):
  https://www.arduino.cc/en/software

- Espressif Systems tutorial on how to configure Arduino® IDE:

  https://github.com/iotechbugs/esp32-Arduino®/blob/master/docs/Arduino®-ide/boards_manager.md

- PTSolns' Documentation Repository Sub-Domain:
  https://docs.PTSolns.com

- PTSolns website:
  https://PTSolns.com/

- CH340 driver installation tutorial (See Section 6.1):

  By PTSolns:
  https://www.youtube.com/watch?v=UUQ84VKg3oM

  By Sparkfun:
  https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all

- PTSolns support:
  https://ptsolns.com/pages/contact