

# ESP32-M1 Reach Out Development Board Wi-Fi Transmitter Power Setting



Perth, Western Australia

This material is confidential to Bison Science and may not be disclosed in whole or in part to any third party nor used in any manner whatsoever other than for the purposes expressly consented to by Bison Science in writing.

This material is also not be reproduced, stored in a retrieval system or transmitted in any form or by any means in whole or in part without the express written consent of Bison Science.

## Document history

Version	Reason for Revision	Date
1.0	New document	07-04-2021

## Contents

1. Introduction.....	1
1.1. Objective.....	1
1.2. Background .....	1
1.3. Related Documents .....	1
2. Wi-Fi Transmitter Power Details .....	2
2.1. Varying Transmitter Power .....	2
2.2. Power Setting Code .....	4
3. Power Measurement on ESP32-M1 .....	5
3.1. Transmitter Power Measurement with RFFE .....	5
3.2. Transmitter Power Measurement RFFE Bypass Mode.....	6

## Acronyms

AP – Access Point

API – Application Programming Interface

FW - Firmware

TX – Transmitter

PA – Power Amplifier

RF – Radio Frequency

RFFE – RF Front End

## 1. Introduction

### 1.1. Objective

This document serves as a Wi-Fi transmitter power setting for the ESP32-M1 Reach Out development board. Discussion in this document is solely for the ESP32-M1 board and the power setting functionality, including power setting API, related code for this setting, power measurement with RFFE with different power setting and RFFE bypass power measurement with different power setting.

### 1.2. Background

The ESP32-M1 board is designed to help makers, engineers, researchers, and hobbyist to develop robust communication systems using Wi-Fi protocols for a distance. The Wi-Fi transmitter design is crucial for ESP32-M1 to achieve the range needed, in relation to the hardware design the power setting is important for the transmitter to transmit (TX) the right amount of power for the transmitter to work in the linear region and not saturating the power amplifier (PA).

The ESP32 chip able to increase and reduce the power level of the transmitter. This is the in-build function in the chip and able to access via firmware (FW). The RFFE uses ESP32 to control the TX power level. The range of the ESP32-M1 can be increase and reduce based on the power setting in ESP32, this is not part of the discussion of this document.

### 1.3. Related Documents

Please refer to below document for Technical Guide for ESP32-M1,

<https://github.com/bionscience/ESP32-M1-Reach-Out/blob/main/Datasheet/BS%20ESP32-M1.pdf>

and ESP32-M1 Datasheet

<https://github.com/bionscience/ESP32-M1-Reach-Out/blob/main/Datasheet/BS%20ESP32-M1.pdf>

### 1.4. Version

The code used in this document are from release v4.1.1. All the measurement performed in this document used below code for SoftAP without Station.

[https://github.com/espressif/esp-idf/tree/release/v4.1/examples/wifi/getting\\_started/softAP](https://github.com/espressif/esp-idf/tree/release/v4.1/examples/wifi/getting_started/softAP)

## 2. Wi-Fi Transmitter Power Details

### 2.1. Varying Transmitter Power

The TX power in ESP32 is set using **max\_tx\_power** API. This API takes an `int8_t` value from the power table shown in Table 2.1.

*Table 2.1: Relationship between Set Value and Actual Value.*

Set Value	Actual Value
[8, 19]	8
[20, 27]	20
[28, 33]	28
[34, 43]	34
[44, 51]	44
[52, 55]	52
[56, 59]	56
[60, 65]	60
[66, 71]	66
[72, 77]	72
78	78

The *Set Value* is the value (#number) set in **max\_tx\_power** API, and the *Actual Value* is the reply from ESP32 for the *Set Value range*. The *Set Value* has a range, for example [8, 19] which means the range can be set from 8 to 19 and the reply from ESP32 for *Actual Value* is 8. The *Actual Value* can be obtained from ESP32 using **get\_max\_tx\_power** API, the code discussed below in Section 2.2.

The ESP32-M1 TX power using the above power table values is shown in Table 2.2. The power is measured on the ESP32-M1 board at the output for ESP32 including matching, see Fig 2.1 for the measurement area.

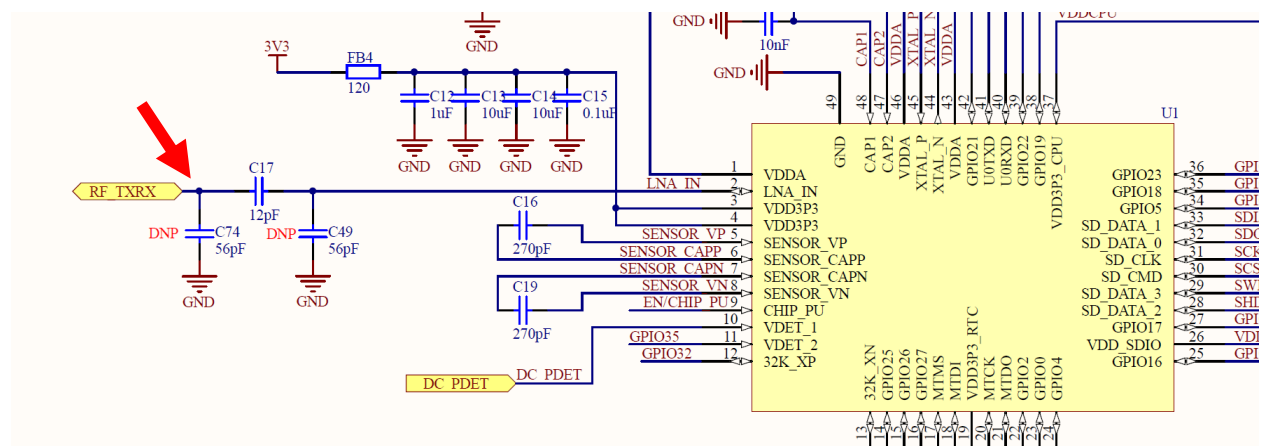


Fig 2.2: The red arrow shows the output power measured area on ESP32.

The ESP32-M1 running the SoftAP app mentioned in Section 1.4 able to transmit from 5.3dBm to 21.2dBm.

*Note: The released notes in the below link mentioned the power table Set Value as per Table 2.2 has different TX power from 2dBm to 20dBm. There is no clear indication of how these were achieved and what app has been used or this is based on the calculated value.*

[https://docs.espressif.com/projects/esp-idf/en/v4.1.1/api-reference/network/esp\\_wifi.html](https://docs.espressif.com/projects/esp-idf/en/v4.1.1/api-reference/network/esp_wifi.html)

Table 2.2: The Set Value and its corresponding measured TX Power.

Set Value	TX Power in dBm
8	5.3
20	8.2
28	10.1
34	11.4
44	13.8
52	14.8
56	15.8
60	16.7
66	18.1
72	19.6
78	21.2

## 2.2. Power Setting Code

The code to set the power (*Set Value*) is specified below,

```
ESP_ERROR_CHECK(esp_wifi_set_max_tx_power(Set Value));
```

and to read the power value (*Actual Value*) from ESP32 is specified below.

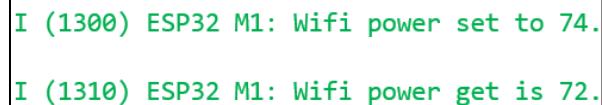
```
ESP_ERROR_CHECK(esp_wifi_get_max_tx_power(&Set Value));
```

The full working code shown below, the code set the *Set Value* to 74, this value is in range of [72, 77], print the value, get back the *Actual Value* and display the value. The output of the code is shown in Figure 2.2.

```
int8_t power = 74;
int8_t read_power;

ESP_ERROR_CHECK(esp_wifi_set_max_tx_power(power));
ESP_LOGI(TAG, "Wifi power set to %d.", a);
ESP_ERROR_CHECK(esp_wifi_get_max_tx_power(&read_power));
ESP_LOGI(TAG, "Wifi power get is %d.", j);
```

The Actual Value is 72 from the above code and match Table 2.1.



```
I (1300) ESP32 M1: Wifi power set to 74.
I (1310) ESP32 M1: Wifi power get is 72.
```

Fig 2.2: The Set Value and Actual Value from ESP32 on power setting.



### 3. Power Measurement on ESP32-M1

#### 3.1. Transmitter Power Measurement with RFFE

The power measurement for ESP32-M1 is performed with RFFE in this section, the RF power is measured from the lowest *Set Value* in the power table to the highest value. Table 3.1 shows the *Set Value* and the corresponding measured TX power at the antenna port, the setup for this measurement is illustrated in Figure 3.1.

The measurement shows the power increases from *Set Value* 8 to 60, and subsequently, the power did not increase. RFFE PA is saturated after *Set Value* 60. From these measurements, the typical value to use for *Set Value* is between 56 to 60 for 30dBm (1W) output TX conducted power.

*Table 3.1: The Set Value and its corresponding measured RF Power.*

Set Value	Measured Value in dBm
8	21.2
20	23.97
28	26.3
34	27.5
44	29.1
52	28.8
56	29.8
60	30.28
66	30.6
72	30.7
78	31.1

Figure 3.1 below shows the measurement setup for TX power measurement. The measurement performed with a 10dB attenuator using the SoftAP app without Station.

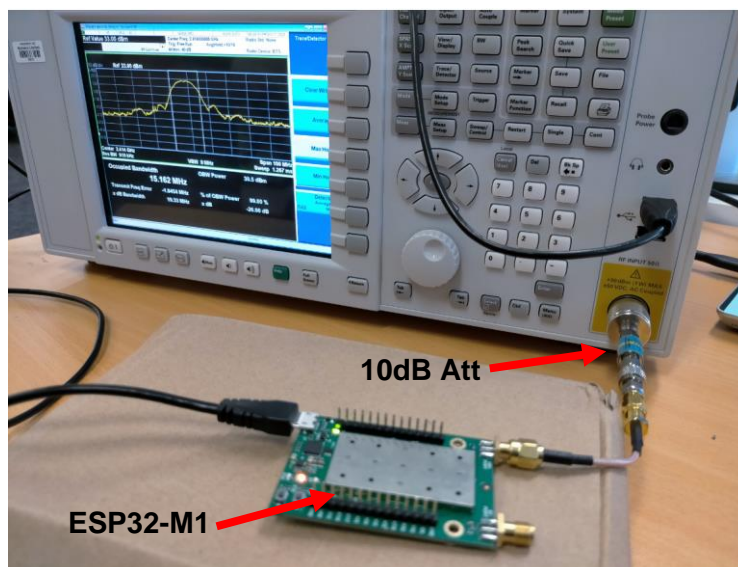


Fig 3.1: Measurement setup for TX power measurement.

### 3.2. Transmitter Power Measurement RFFE Bypass Mode

The RFFE bypass mode discussed in the Technical Guide link attached in Section 1.4. In this mode the RFFE is off, the TX power measured at BT Port. The measurement shows the TX power is from 2.4dBm to 16.4dBm.

Table 3.2: The Set Value its corresponding RF Power.

Set Value	Measured Value in dBm
8	2.4
20	5.4
28	7.5
34	9.2
44	11.4
52	10.2
56	11.1
60	12.1
66	13.6
72	14.7
78	16.4

----End of Document---