Wio-S3

LoRa Wireless Module - Powered by ESP32-S3R8

Datasheet

V1.0

Content

1 Introduction

Wio-S3 wireless module is embedded with the ESP32-S3R8 dual core Xtensa® LX7 MCU and the long range SX1262 LoRa chip. It supports LoRaWAN frequency plan on EU868, US915 and more, as well as 2.4GHz Wi-Fi and BLE 5.0 connectivity. It integrates 16MB Flash and 8MB PSRAM, making it ideal for edge AI processing. We offer two versions (with or without an onboard IPEX connector), offering developers versatile antenna integration options. Equipped with a TCXO, it ensures stable and accurate RF performance, which is suitable for IoT applications that require long range, high performance, and reliable wireless communication.

1.1 Feature

- **Based on ESP32-S3R8 dual core Xtensa® LX7 MCU (Up to 240MHz)**
- **Large Storage Capabilities:** Equipped with 16 MB external Flash and 8 MB PSRAM, Built-in 512 KB SRAM (TCM).
- **Rich Peripheral Interfaces:** Including UART, I2C, SPI, USB, ADC, PWM, I2S, SD/MMC, CAN (TWAI®), and GPIO*25.
- **Wi-Fi & Bluetooth® Connectivity:** Supports 2.4 GHz Wi-Fi (IEEE 802.11 b/g/n) and Bluetooth® 5.0 Low Energy connectivity.
- **Based on Semtech SX1262 chip**
- **Global LoRaWAN® Support:** Supports multiple LoRaWAN® frequency plans including EU868,

US915, AS923, AU915, AS923, KR920, IN865 for global deployment.

- **High performance:**
  - Sensitivity: -137dBm @SF12, BW125kHz
  - RF Output Power: Up to 20.9 dBm @868/915MHz
- **High Stability:** Integrated TCXO for stable frequency reference and improved RF communication reliability.
- **Low Power Consumption:** as low as 9.3uA sleep current
- **RF shielding:** Contain a shield shell for signal protection.
- **Flexible Hardware Design:** Available in two versions (with or without an onboard IPEX connector), offering developers versatile antenna integration options.
- **Lower Cost**
- **Compact Size: 21.6 * 16.5 * 3.3mm 38 pins SMT**
- **IoT Application:** Suitable for smart agriculture, industrial IoT, environmental monitoring, asset tracking, and edge AI applications.

2 Description

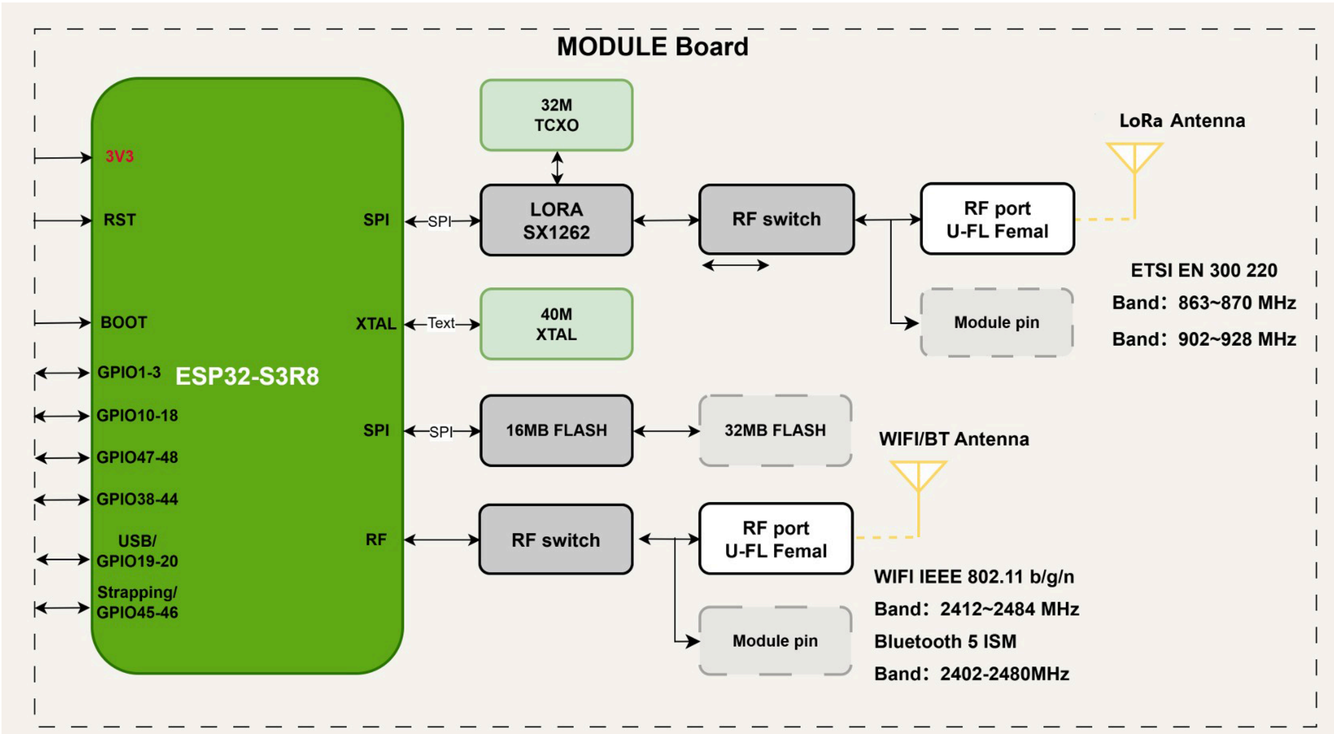

Fig. 1: Wio-S3 Schematic Diagram

2.1 Pin definition

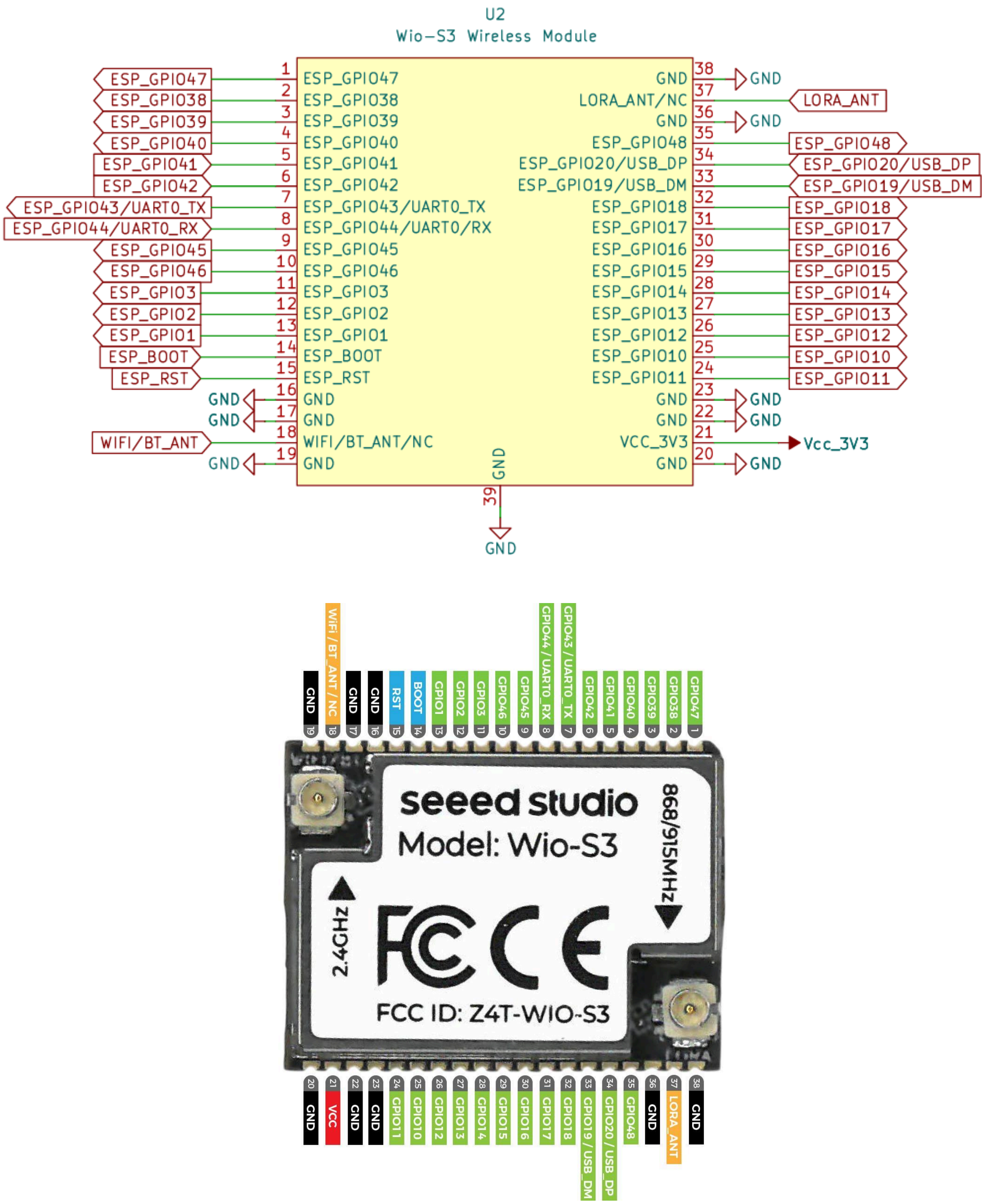

Fig. 2: Wio-S3 Pin arrangement

Table 1: Wio-S3 Pinout

Pin No.	Name	Type	Description
1	ESP_GPIO47	I/O/T	MCU GPIO
2	ESP_GPIO38	I/O/T	MCU GPIO
3	ESP_GPIO39	I/O/T	MCU GPIO
4	ESP_GPIO40	I/O/T	MCU GPIO
5	ESP_GPIO41	I/O/T	MCU GPIO
6	ESP_GPIO42	I/O/T	MCU GPIO
7	ESP_GPIO43 / UART0_TX	I/O/T	MCU GPIO; UART0_TX from MCU
8	ESP_GPIO44 / UART0_RX	I/O/T	MCU GPIO; UART0_RX from MCU
9	ESP_GPIO45	I/O/T	MCU GPIO
10	ESP_GPIO46	I/O/T	MCU GPIO
11	ESP_GPIO3	I/O/T	MCU GPIO
12	ESP_GPIO2	I/O/T	MCU GPIO
13	ESP_GPIO1	I/O/T	MCU GPIO
14	ESP_BOOT	I/O/T	MCU GPIO0, Boot Mode: Pull low to enter Download Boot Mode.
15	ESP_RST	I	CHIP_PU: High Level: Chip Enabled; Low Level: Chip Disabled; Ensure that the CHIP_PU pin is not left floating.
16	GND	-	Ground
17	GND	-	Ground
18	WIFI / BT_ANT / NC	I/O	WiFi and BT RF Input and Output from MCU
19	GND	-	Ground
20	GND	-	Ground
21	VCC 3V3	-	Supply voltage for the module
22	GND	-	Ground

23	GND	-	Ground
24	ESP_GPIO11	I/O/T	MCU GPIO
25	ESP_GPIO10	I/O/T	MCU GPIO
26	ESP_GPIO12	I/O/T	MCU GPIO
27	ESP_GPIO13	I/O/T	MCU GPIO
28	ESP_GPIO14	I/O/T	MCU GPIO
29	ESP_GPIO15	I/O/T	MCU GPIO
30	ESP_GPIO16	I/O/T	MCU GPIO
31	ESP_GPIO17	I/O/T	MCU GPIO
32	ESP_GPIO18	I/O/T	MCU GPIO
33	ESP_GPIO19 / USB_DM	I/O/T	MCU GPIO; USB_D- from MCU
34	ESP_GPIO20 / USB_DP	I/O/T	MCU GPIO; USB_D+ from MCU
35	ESP_GPIO48	I/O/T	MCU GPIO
36	GND	-	Ground
37	LORA_ANT / NC	-	LoRa RF Input and Output
38	GND	-	Ground
39	GND	-	Ground

3 Electrical characteristics

3.1 Extreme working conditions

Reaching or exceeding the maximum ratings listed in the table below can cause equipment damage.

Table 2: Absolute Maximum Ratings

Item	Description	Min	Max	Unit
Input power pins [1]	Allowed input voltage	-0.3	3.6	V
I_{output} [2]	Total IO output current	—	1500	mA
T_{STORE}	Storage temperature	-40	150	°C

[1] For more information on input power pins, please refer to Espressif_ESP32-S3R8_Datasheet: Section 2.5.1 Power Pins.

[2] The product proved to be fully functional after all its IO pins were pulled high while being connected to ground for 24 consecutive hours at ambient temperature of 25 C.

3.2 Normal working conditions

Table 3: Recommended Operating Conditions

Item[1]	Description		Min	Typ	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC[2]	Recommended input voltage		3	3.3	3.6	V
VDD_SPI (as input power supply)	—		1.8	3.3	3.6	V
VDD3P3_CPU [3]	Recommended input voltage		3	3.3	3.6	V
I_{VDD} [4]	Cumulative input current		0.5	—	—	A
T_A	Ambient temperature	ESP32-S3R8 [5]	-40	—	65	°C

[1] For more information, please refer to Espressif_ESP32-S3R8_Datasheet: Section 2.5 Power Supply.

[2] If VDD3P3_RTC is used to power VDD_SPI (see Section 2.5.2 Power Scheme), the voltage drop on R_{SPI} should be accounted for. See also Section 5.3 VDD_SPI Output Characteristics.

[3] When programming eFuse, VDD3P3_CPU should not exceed 3.3 V.

[4] When using a single power supply, the output current must reach 500 mA or above.

[5] For ESP32-S3R8 and ESP32-S3R8V chips, if the PSRAM ECC function is enabled, the maximum ambient temperature can be increased to 85 °C, but the available capacity of the PSRAM will be reduced by 1/16. For further specifications, please refer to Section 1: ESP32-S3 Series Comparison.

3.3 Module specifications

Table 4: Wio-S3 Features

Items	Parameter	Specifications	Unit
Core	MCU	Xtensa® 32-bit LX7	
	Storage	16MB Flash, 512KB SRAM	
Structure	Size	21.6 * 16.5 * 3.3mm	mm
	Package	38 pins, SMT	
Electrical Characteristics	Power Supply	3.0 - 3.6V(@3.3V typical)	V
	Minimum Sleep Current	9.3uA	uA
	Minimum Standby Current	1.43mA	mA
	Operation Current (Transmitter+MCU)	113mA @16dBm in 868MHz type	mA
		125mA @22dBm in 868MHz type	

<table>
<tr><td rowspan="12"></td><td></td><td colspan="3">127mA @22dBm in 915MHz type</td><td></td></tr>
<tr><td rowspan="2">Operation Current (Receiver+MCU)</td><td colspan="3">5.5mA @BW125kHz, 868MHz type</td><td rowspan="2">mA</td></tr>
<tr><td colspan="3">5.7mA @BW125kHz, 915MHz type</td></tr>
<tr><td rowspan="2">Output power</td><td colspan="3">20.9dBm max @868MHz LoRa</td><td rowspan="2">dBm</td></tr>
<tr><td colspan="3">20.74dBm max @915MHz LoRa</td></tr>
<tr><td rowspan="4">Sensitivity</td><td colspan="3">@SF12, BW125kHz</td><td rowspan="4">dBm</td></tr>
<tr><td>Fr(MHz)</td><td>type</td><td>max</td></tr>
<tr><td>868</td><td>-137</td><td>-137</td></tr>
<tr><td>915</td><td>-136.5</td><td>-136.9</td></tr>
<tr><td rowspan="2">Harmonics (LoRa)</td><td colspan="3">< -41dBm (2nd Harmonic)</td><td>dBm</td></tr>
<tr><td colspan="3">< -49dBm (3rd Harmonic)</td><td>dBm</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td rowspan="8">Interface</td><td>RFIO</td><td colspan="3">2 RF ports (Wi-Fi/BT & LoRa)</td><td></td></tr>
<tr><td>UART</td><td colspan="3">3 group of UART</td><td></td></tr>
<tr><td>I2C</td><td colspan="3">2 group of I2C</td><td></td></tr>
<tr><td>ADC</td><td colspan="3">2 groups of ADC, 12-bit</td><td></td></tr>
<tr><td>NRST</td><td colspan="3">1 Manual reset pin input</td><td></td></tr>
<tr><td>SPI</td><td colspan="3">1 group of SPI</td><td></td></tr>
<tr><td>USB</td><td colspan="3">1 group of USB</td><td></td></tr>
<tr><td>BOOT</td><td colspan="3">1 Manual Boot pin input</td><td></td></tr>
<tr><td>Working Temperature</td><td colspan="4">-40℃ ~ 85℃</td><td>℃</td></tr>
</table>

3.4 Power Consumption

Table 5: Wio-S3 Power Consumption Data

Mode		Power Mode	Items	Data(avg)	Note
WiFi	WiFi Tx	3.3V	802.11B	349mA	WiFi antenna connected, LoRa sleeping
			802.11G	333mA	
			802.11N	310mA	
	WiFi Rx		802.11B	104mA	
			802.11G	103mA	
			802.11N	103mA	
LoRa	LoRa Tx	3.3V	868MHz, 16dBm	113mA	LoRa antenna connected, WiFi & BLE off
			868MHz, 22dBm	125mA	
			915MHz, 22dBm	127mA	
		3V	868MHz, 16dBm	109mA	
			868MHz, 22dBm	124mA	
			915MHz, 22dBm	123mA	
	LoRa Rx	3.3V	868MHz	5.5mA	
			915MHz	5.7mA	
WiFi + LoRa		3.3V	WiFi 11B AP + LoRa 915MHz, 22dBm	201mA	
			WiFi 11B AP + LoRa 868MHz, 16dBm	189mA	
BLE + LoRa		3.3V	BLE advertising + LoRa 915MHz, 22dBm	158mA	
			BLE advertising + LoRa 868MHz, 16dBm	146mA	
Deep Sleep		3.3V	ESP32-S3 Deep Sleep, LoRa Sleep	9.3uA	ESP32-S3: all peripherals off,

				deep sleep
Standby / Light Sleep	3.3V	ESP32-S3 Light Sleep, LoRa Standby	1.43mA	ESP32-S3: light sleep, RTC/ULP active, wake on any event; LoRa standby (STDB_RC), internal RC wake; all RF off

NOTE: The above power consumption data was measured under laboratory test conditions and is provided for reference only. Actual power consumption may vary depending on firmware, peripherals, operating mode, environmental conditions, and hardware configuration.

4 Typical RF performance test

4.1 Wio-S3 Performance Testing

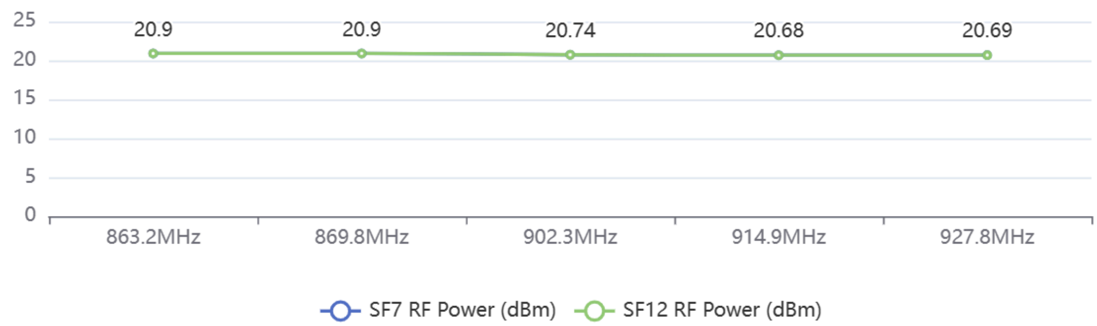

Fig. 3: Wio-S3 LoRa RF Power vs Frequency

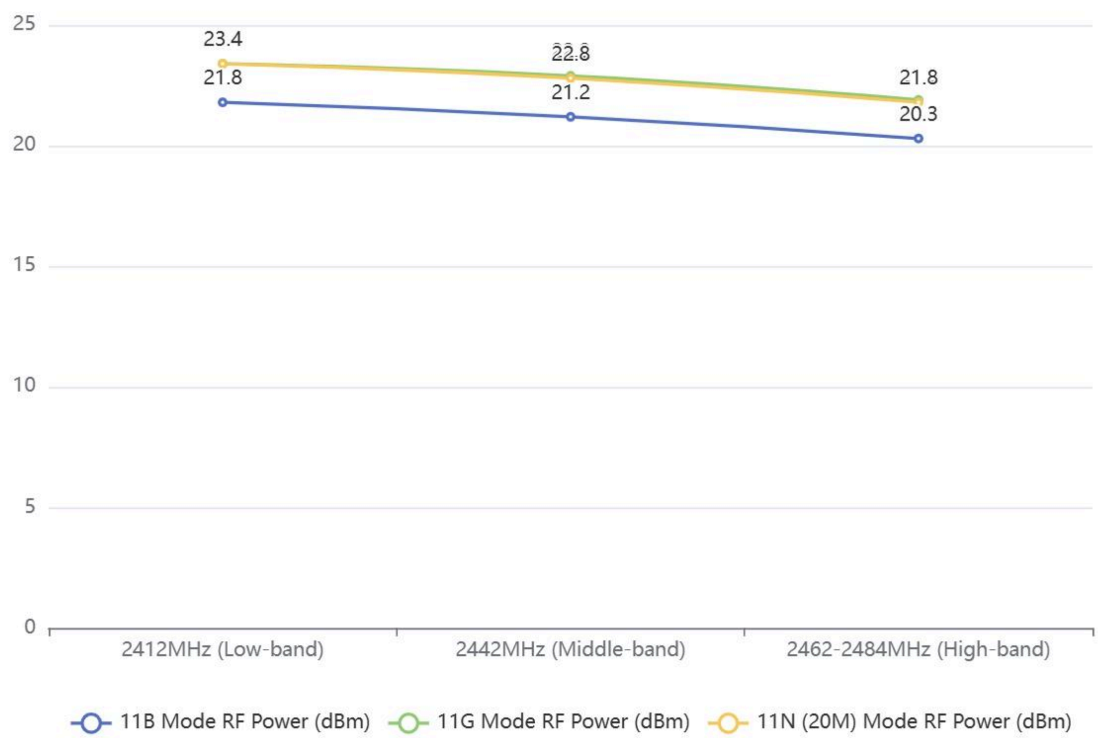

Fig. 4: Wio-S3 2.4G WiFi RF Power vs Frequency

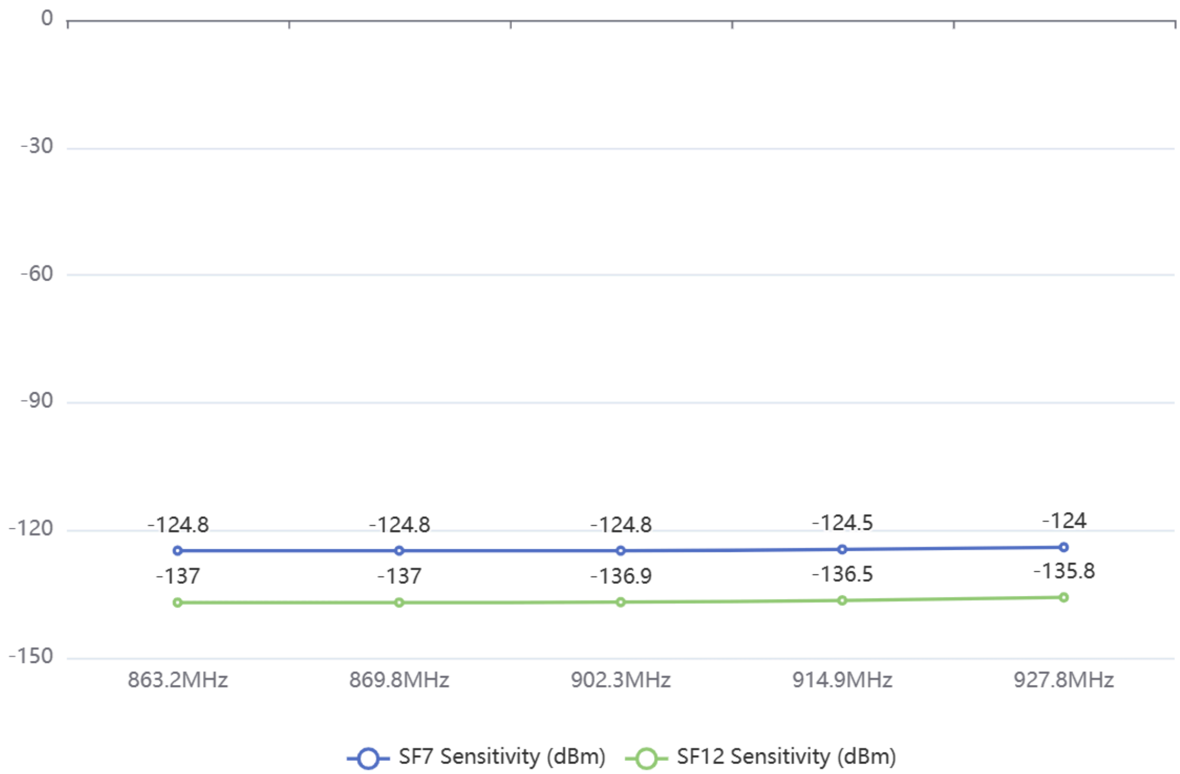

Fig. 5: Wio-S3 LoRa Sensitivity vs Frequency

5 Wio-S3 Command List

5.1 System Command

1. **help [<string>] [-v <0|1>]**

Print the summary of all registered commands if no arguments are given, otherwise print summary of given command.

Item	Description	Data
<string>	Name of command	Optional. If provided, display description for the specified command
-v, --verbose=<0|1>	If specified, list console commands with given verbose level	0 or 1

Example: `help`
Function: Show all available commands.

Example: `help lora_tx`
Function: Show detailed information for the lora_tx command.

Example: `help -v 1`
Function: Show full command list with verbose mode enabled.

5.2 LoRa Command

1. **lora_tx [-f <f>] [-s <6~12>] [-b <0|1|2>] [-c <1|2|3|4>] [-p] [--crc=<0|1>] [--iq=<0|1>] [--net=<0|1>] [-i <t>] [-d <d>] [-n <n>]**

Send LoRa data packets with configurable radio parameters.

Item	Description	Data
-f, --freq=<f>	Set the radio frequency in Hz	range: 415000000 ~ 940000000 Hz, default: 868000000
-s, --sf=<6~12>	Set Lora SF	range: 6 ~ 12, default: 7
-b, --bw=<0|1|2>	Set Lora Bandwidth	0: 125KHz; 1: 250KHz; 2: 500KHz, default: 0

-c, --cr=<1|2|3|4>	Set Lora Coding Rate	1: CR_4_5; 2: CR_4_6; 3: CR_4_7; 4: CR_4_8, default: 1
-p, --power=	Set the radio power	LPA range: -17 ~ +14 dB, HPA range: -9 ~ +22 dB, default: 10
--crc=<0|1>	Set Lora CRC	0: DISABLE; 1: ENABLE, default: 1
--iq=<0|1>	Set Lora IQ mode	0: STANDARD; 1: INVERTED, default: 0
--net=<0|1>	Set Public Network	0: Private Network, 1: Public Network, default: 0
-i, --interval=<t>	Set the tx interval (ms)	default: 0
-d, --txt=<d>	Set the text data to send	default: hello
-n, --num=<n>	Set Number of packets sent	0: Keep sending, default: 1

Example: `lora_tx -f 868000000 -s 7 -b 0 -c 1 -p 10 -d hello`
Function: Send a LoRa packet with frequency 868000000 Hz, SF7, 125 kHz bandwidth, coding rate 4/5, and send “hello” at TX power 10 dB.

2. **lora_rx [-f <f>] [-s <6~12>] [-b <0|1|2>] [-c <1|2|3|4>] [--crc=<0|1>] [--iq=<0|1>] [--net=<0|1>] [--boosted=<0|1>]**

Receive LoRa packets with configurable radio parameters.

Item	Description	Data
-f, --freq=<f>	Set the radio frequency in Hz	range: 415000000 ~ 940000000 Hz, default: 868000000
-s, --sf=<6~12>	Set Lora SF	range: 6 ~ 12, default: 7
-b, --bw=<0|1|2>	Set Lora Bandwidth	0:125KHz; 1:250KHz; 2:500KHz, default: 0
-c, --cr=<1|2|3|4>	Set Lora Coding Rate	1:CR_4_5; 2:CR_4_6; 3:CR_4_7; 4:CR_4_8, default: 1
--crc=<0|1>	Set Lora CRC	0:DISABLE; 1:ENABLE, default: 1
--iq=<0|1>	Set Lora IQ mode	0:STANDARD; 1:INVERTED, default: 0
--net=<0|1>	Set Public Network	0: Private Network, 1: Public Network,

		default: 0
--boosted=<0|1>	Set boosted RX mode	1: Boosted RX, 0: Normal RX, default: 1

Example: `lora_rx -f 868000000 -s 7 -b 0`
Function: Receive LoRa packets at frequency 868000000 Hz, SF7, 125 kHz bandwidth in normal RX mode.

3. **lora_cw [-f <f>] [-p] [-o]**

Transmit a continuous wave signal for RF testing and spectrum measurement.

Item	Description	Data
-f, --freq=<f>	Set the radio frequency in Hz	range: 415000000 ~ 940000000 Hz, default: 868000000
-p, --power=	Set the radio power	LPA: -17 ~ +14 dB, HPA: -9 ~ +22 dB, default: 10
-o, --ocp=	Set current limit step	range: 0 ~ 63, step 2.5mA, default: 24

Example: `lora_cw -f 868000000 -p 10 -o 24`
Function: Transmit continuous wave signal at 868 MHz with 10 dB TX power and 60 mA OCP current limit for RF testing.

4. **lora_fcc_fhss [-m <0|1>] [-s <6~12>] [-c <1|2|3|4>] [-p] [--crc=<0|1>] [--iq=<0|1>] [--net=<0|1>] [-i <t>] [-d <d>]**

Transmit LoRa packets using frequency hopping mode for FCC compliance and RF certification testing.

Item	Description	Data
-m, --mode=<0|1>	Set the FHSS mode	0: FHSS_125K_MODE, 1: FHSS_500K_MODE, default: 0
-s, --sf=<6~12>	Set Lora SF	range: 6 ~ 12, default: 10
-c, --cr=<1|2|3|4>	Set Lora Coding Rate	1: CR_4_5; 2: CR_4_6; 3: CR_4_7; 4: CR_4_8, default: 1
-p, --power=	Set the radio power	LPA range: -17 ~ +14 dB,

		HPA range: -9 ~ +22 dB, default: 10
--crc=<0|1>	Set Lora CRC	0: DISABLE; 1: ENABLE, default: 1
--iq=<0|1>	Set Lora IQ mode	0: STANDARD; 1: INVERTED, default: 0
--net=<0|1>	Set Public Network	0: Private Network, 1: Public Network, default: 0
-i, --interval=<t>	Set the tx interval ms	default: 0
-d, --txt=<d>	Set the text data to send	default: hello seeed! 1234567

Example: `lora_fcc_fhss -m 1 -s 10 -c 1 -p 14 --crc=1 --iq=0 --net=0 -i 1000 -d test`
Function: Transmit FHSS LoRa packets using FHSS_500K_mode, SF10, coding rate 4/5, 14 dB TX power, CRC enabled, standard IQ mode, private network mode, and send payload "test" every 1000 ms.

5.3 GPIO Command

1. **gpio [-p <0~48>] [-d <0|1>] [-v <0|1>]**

Control GPIO pins for input/output testing and hardware validation.

Item	Description	Data
-p, --pin=<0~48>	GPIO pin number	range: 0 ~ 48, default: 0
-d, --direction=<0|1>	Set the gpio direction	0: INPUT, 1: OUTPUT, default: 1
-v, --value=<0|1>	Set the gpio output value	0: LOW, 1: HIGH, default: 0

Example: `gpio -p 10 -d 1 -v 1`
Function: Set GPIO 10 as output mode and drive the pin output to high level.

5.4 Flash Command

1. **flash [-b <100~1000>]**

Test flash memory read and write performance using block operations.

Item	Description	Data
-b, --block=<100~1000>	Set the block number to run read/write test	range: 100 ~ 1000, default: 100

Example: `flash -b 500`
Function: Perform flash memory read/write test using 500 data blocks.

5.5 WiFi Command

1. **ap_set <ssid> [<pass>] [-a <authmode>] [-n <channel>] [-m <max_conn>]**

   Configure the device to operate in WiFi AP mode with custom SSID, password, authentication type, and channel settings.

Item	Description	Data
<ssid>	SSID of AP	Required
<pass>	Password of AP	Optional
-a, --authmode=<authmode>	WiFi authentication type	open / wep / wpa2 / wpa2_enterprise
-n, --channel=<channel>	WiFi channel	Optional
-m, --max_conn=<max_conn>	Max station number	default: 2

Example: `ap_set MyAP 12345678 -a wpa2 -n 6 -m 4`
Function: Create a WPA2 WiFi AP named "MyAP" on channel 6 with maximum 4 client connections.

2. **ap_query**

   Display current AP configuration and disconnect station mode connection.

3. **wifi <action> [--espnow_enc=<int>] [--storage=<str>]**

   Display current AP configuration and disconnect station mode connection.

Item	Description	Data
<action>	WiFi operation action	init / deinit / start / stop / restart / status

--espnow_enc=<int>	ESP-NOW encryption number(idf>=4.4)	only for init / restart
--storage=<str>	WiFi storage type	flash / ram

Example: `wifi init --storage=flash`
Function: Initialize WiFi subsystem and store configuration in flash memory.

4. **wifi_count [<action>]**

Display current AP configuration and disconnect station mode connection.

Item	Description	Data
<action>	Counter operation	query(default) / clear

Example: `wifi_count clear`
Function: Clear current WiFi count statistics.

5. **wifi_mode <mode>**

Display current AP configuration and disconnect station mode connection.

Item	Description	Data
<mode>	WiFi mode	ap / sta / apsta

Example: `wifi_mode apsta`
Function: Enable simultaneous AP and Station mode.

6. **wifi_protocol [<protocol>] [--2g=<2g_proto>] [--5g=<5g_proto>] [-i <interface>]**

Configure the WiFi protocol supported by the selected interface (AP or STA).

Two configuration methods are available:

- `<protocol>`: Configure WiFi protocol using `esp_wifi_set_protocol()`
- `--2g` / `--5g`: Configure protocols separately for the 2.4 GHz and 5 GHz bands using `esp_wifi_set_protocols()`

Note:

- `esp_wifi_set_protocol()` and `esp_wifi_set_protocols()` cannot be used simultaneously.
- If no parameters are provided, the current WiFi protocol configuration will be displayed.

Item	Description	Data
<protocol>	Configure WiFi protocol using esp_wifi_set_protocol()	Example: b, b/g, b/g/n g/n is invalid.
--2g=<2g_proto>	Configure 2.4 GHz protocol using esp_wifi_set_protocols()	API: esp_wifi_set_protocols. Protocol string: 'lr', 'b', 'g', 'n', 'ax' Combine with lr mode: 'lr/b', 'lr/g', etc. Protocol bitmap value: raw bitmap value with hex, eg: 0x40
--5g=<5g_proto>	Configure 5 GHz protocol using esp_wifi_set_protocols()	Same format as --2g
-i, --interface=<interface>	Select WiFi interface	ap, sta, default: sta

Example: `wifi_protocol`
Function: Display the current WiFi protocol configuration.
Example: `wifi_protocol --2g=lr/b/g/n -i sta`
Function: Configure the STA interface to support LR, 802.11b, 802.11g, and 802.11n protocols on the 2.4 GHz band.

7. wifi_bandwidth [<cbw>] [--2g=<2g_cbw>] [--5g=<5g_cbw>] [-i <interface>]

Configure the WiFi protocol supported by the selected interface (AP or STA).

Two configuration methods are available:

- `<cbw>`: Configure bandwidth using `esp_wifi_set_bandwidth()`
- `--2g` / `--5g`: Configure bandwidth separately for 2.4 GHz and 5 GHz bands using `esp_wifi_set_bandwidths()`

Note:

- `esp_wifi_set_bandwidth()` and `esp_wifi_set_bandwidths()` cannot be used simultaneously.
- If no parameters are provided, the current WiFi bandwidth configuration will be displayed.

Item	Description	Data
<cbw>	Configure WiFi bandwidth using esp_wifi_set_bandwidth()	20, 40
--2g=<2g_cbw>	Configure 2.4 GHz bandwidth using esp_wifi_set_bandwidths()	20, 40
--5g=<5g_cbw>	Configure 5 GHz bandwidth using esp_wifi_set_bandwidths()	20, 40
-i, --interface=<interface>	Select WiFi interface	ap, sta, default: sta

Example: `wifi_bandwidth`
Function: Display the current WiFi bandwidth configuration.
Example: `wifi_bandwidth --2g=20 -i sta`
Function: Configure the 2.4 GHz band to use a 20 MHz channel bandwidth.

8. **wifi_ps <type>**

Set WiFi power mode. This command configures the WiFi Power Save mode of the device. Different modes balance power consumption and network performance differently.

Item	Description	Data
<type>	WiFi mode	0: WIFI_PS_NONE, 1: WIFI_PS_MIN_MODEM, 2: WIFI_PS_MAX_MODEM

Mode	Name	Latency	Simple Description
0	WIFI_PS_NONE	Lowest	No power saving, full performance
1	WIFI_PS_MIN_MODEM	Medium	Balanced power saving and performance
2	WIFI_PS_MAX_MODEM	Highest	Maximum power saving, higher delay

Example: `wifi_ps 1`
Function: Sets WiFi power save mode to minimum modem power saving.

9. wifi_country [<code>] [-s <int>] [-n <int>] [-p <str>]

Set or get WiFi country configuration. This command configures the WiFi regulatory domain, including allowed channels and transmit rules based on country regulations.

Item	Description	Data
<code>	Country code setting (or query current country if not provided)	CN / US / JP / etc.
-s, --schan=<int>	Start WiFi channel number	Integer (e.g. 1)
-n, --nchan=<int>	Total number of supported channels	Integer (e.g. 13)
-p, --policy=<str>	Country policy mode	auto (default) / manual

Example: `wifi_country CN -s 1 -n 13 -p auto`
Function: Set WiFi country to CN, enable channels 1–13, and use automatic regulatory policy.

10. sta_connect <ssid> [<pass>] [-b <bssid>] [-n <channel>] [--no-disconnect] [--no-reconnect] [--full-scan] [--failure_retry=<int>] [--5g-offset=<rssi_5g_offset>]

Connect the device in STA mode to a specified WiFi access point with optional scan and reconnect configurations.

Item	Description	Data
<ssid>	SSID of WiFi AP	String (WiFi name)
<pass>	Password of AP	String (WiFi password)
-b, --bssid=<bssid>	Specific AP MAC address	MAC address (xx:xx:xx:xx:xx:xx)
-n, --channel=<channel>	AP WiFi channel	Integer
--no-disconnect	For test: do not run esp_wifi_disconnect before connect	/
--no-reconnect	Disable auto-reconnect in wifi disconnect handler	/
--full-scan	Enable full channel scan during sta connection	/

--failure_retry=<int>	Number of retry attempts after connection failure	Integer
--5g-offset=<rssi_5g_offset>	Offset value for 5g ssid when do full scan connect	Integer

Example: `sta_connect MyWiFi 12345678 -n 6 --failure_retry=3 --full-scan`
Function: Connect STA mode to specified WiFi access point with optional scan, retry, and connection behavior control.

11. sta_disconnect

Disconnect the device from the current WiFi AP or stop reconnect.

Example: `sta_disconnect`

12. sta_scan [-h] [<ssid>] [-b <bssid>] [-n <int>] [--max=<int>] [--min=<int>] [--passive] [--passive-time=<int>] [--dwell=<int>] [-2 <int/hex>] [-5 <int/hex>] [--count-only]

Scan available WiFi access points in station mode.

Item	Description	Data
<ssid>	SSID of WiFi AP	String (WiFi name)
-b, --bssid=<bssid>	BSSID of AP	MAC address (xx:xx:xx:xx:xx:xx)
-n, --channel=<int>	Channel of AP	Integer
-h, --show-hidden	Show hidden APs	/
--max=<int>	Maximum active scan time	Integer (ms)
--min=<int>	Minimum active scan time	Integer (ms)
--passive	Set scan type to passive	/
--passive-time	Passive scan time	Integer (ms)
--dwell=<int>	Home channel dwell time	Integer (ms)
-2, --bitmap-2g=<int/hex>	2.4GHz channel scan bitmap	Hex/int (e.g. 0x842 for ch1/6/11)
-5, --bitmap-5g=<int/hex>	5GHz channel scan bitmap	Hex/int (e.g. 0x6 for ch36/40)
--count-only	For test: do scan ap count only.	/

Example: `sta_scan -h --passive --passive-time=120 --dwell=30 -2 0x842 -5 0x6`

Function: Perform a full passive WiFi scan showing hidden APs, scanning selected 2.4GHz and 5GHz channels, and using controlled dwell and scan timing per channel.

13. wifi_txpower [-u <unit>] [--reset]

Set WiFi transmit power to a specified level or reset it to the default SDK configuration.

Item	Description	Data
-u, --unit	TX power value, each unit = 0.25 dBm (e.g. 8 = 2 dBm)	Integer
--reset	Reset TX power to default value from SDK configuration	/

Example: `wifi_txpower -u 8`
Function: Set the WiFi transmit power to 2dBm.

5.6 Network Command

1. ping [-W <timeout>] [-i <interval>] [-s <size>] [-c <count>] [-Q <n>] [host] [--abort]

Send ICMP ECHO_REQUEST packets to a network host for connectivity testing.

Item	Description	Data
-W, --timeout=<timeout>	Time to wait for response	Integer (seconds)
-i, --interval=<interval>	Interval between sending packets	Float/Integer (seconds)
-s, --packetsize=<size>	Use <size> as number of data bytes to be sent	Integer (bytes)
-c, --count=<count>	Number of packets to send, stop after <count> replies	Integer, default: 5
-Q, --tos=<n>	IP Type of Service field value	Integer
host	Target IP address or domain name	String
--abort	Stop running ping process	ping abort -> esp_ping_stop

Example: `ping -c 4 8.8.8.8`

Function: Send 4 ICMP echo requests to 8.8.8.8 to test network connectivity.

2. **iperf [-suV] [-c <host>] [-p <port>] [-l <length>] [-i <interval>] [-t <time>] [-b <bandwidth>] [-f <format>] [--abort]**

   Measure network performance (throughput, bandwidth, and latency) using TCP or UDP.

Item	Description	Data
-c, --client	Run in client mode and connect to server	IP/domain
-s, --server	Run in server mode	/
-u, --udp	Use UDP instead of TCP	/
-V, --ipv6_domain	Use IPv6 protocol	/
-p, --port=<port>	Server port number	Integer
-l, --len=<length>	Buffer length for transmission	Integer (bytes), Defaults: TCP=16384, IPv4 UDP=1470, IPv6 UDP=1450
-i, --interval=<interval>	Seconds between periodic bandwidth reports	Float/Integer (seconds)
-t, --time=<time>	Test duration	Integer (seconds), default 10s
-b, --bandwidth=<bandwidth>	Target bandwidth	Mbits/sec
-f, --format=<format>	Output format	String, k = Kbits/sec, m = Mbits/sec
--abort	Stop running iperf test	/

Example: `iperf -c 192.168.1.10 -t 10 -i 1`
Function: Run a TCP bandwidth test for 10 seconds against server 192.168.1.10, reporting results every 1 second.

5.7 BLE Command

1. **ble [-a <0|1>]**

Control BLE advertising state for Bluetooth testing and discovery.

Item	Description	Data
-a, --adv=<0|1>	BLE advertising control	0: Advertising Stop, 1: Advertising Begin, default: 0

Example: `ble -a 1`
Function: Start BLE advertising mode for device discovery.

5.8 Power Mode Command

1. **sleep [-m <0|1|2>]**

Configure device sleep mode for low power testing.

Item	Description	Data
-m, --mode=<0|1|2>	Configure device sleep mode for low power testing.	0: standby, 1: light sleep, 2: deep sleep, default: 0

Example: `sleep -m 2`
Function: Switch device into specified low power sleep mode.

6 Application information

6.1 Package information

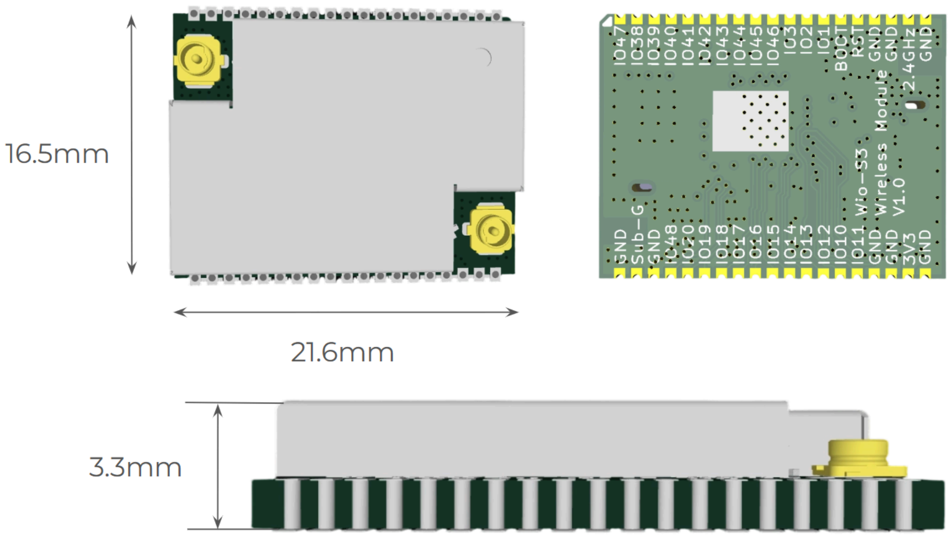

Fig. 6: Wio-S3 Module Appearance

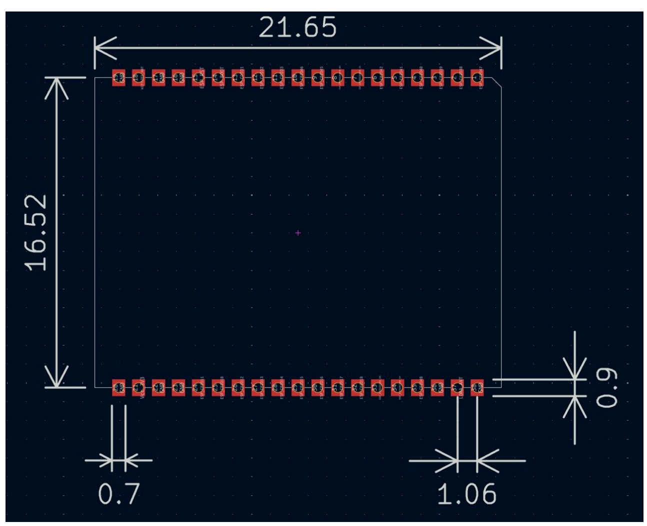

Fig. 7: Wio-S3 PCB Layout

6.2 External interface of the module

The Wio-S3 module provides 38 SMT pins for external system integration. These pins integrate a comprehensive set of digital and analog interfaces, including up to 25 general GPIOs, UART, I2C, SPI, USB (DM/DP), ADC, PWM, I2S, SD/MMC, and CAN (TWAITM). Depending on the SKU, the antenna connection is implemented in two ways: the Wio-S3 Wireless Module (with IPEX connector) provides standard IPEX connectors for both Wi-Fi/BT and LoRa antennas, allowing direct plug-and-play antenna attachment; the Wio-S3-N Wireless Module (without IPEX connector) exposes the same RF signals via bare RF pads (WiFi/BT_ANT and LORA_ANT), enabling users to solder an external antenna or connect a custom RF front-end. These rich GPIO interfaces enable flexible connection to sensors, making the module suitable for complex IoT and edge AI applications.

6.3 Reference design based on Wio-S3 module

This reference design facilitates the quick development of dual-mode LoRaWAN® and Wi-Fi/Bluetooth nodes, allowing users to achieve rapid configuration simply by sending commands over the dedicated UART interface. The system integrates dual 5V USB-C inputs with 1A fuse protection, a PMOS-controlled 3.3V BUCK regulator, and dedicated U.FL RF ports, while routing essential RST, BOOT (GPIO0), and peripheral GPIO pins for easy programming and sensor expansion.

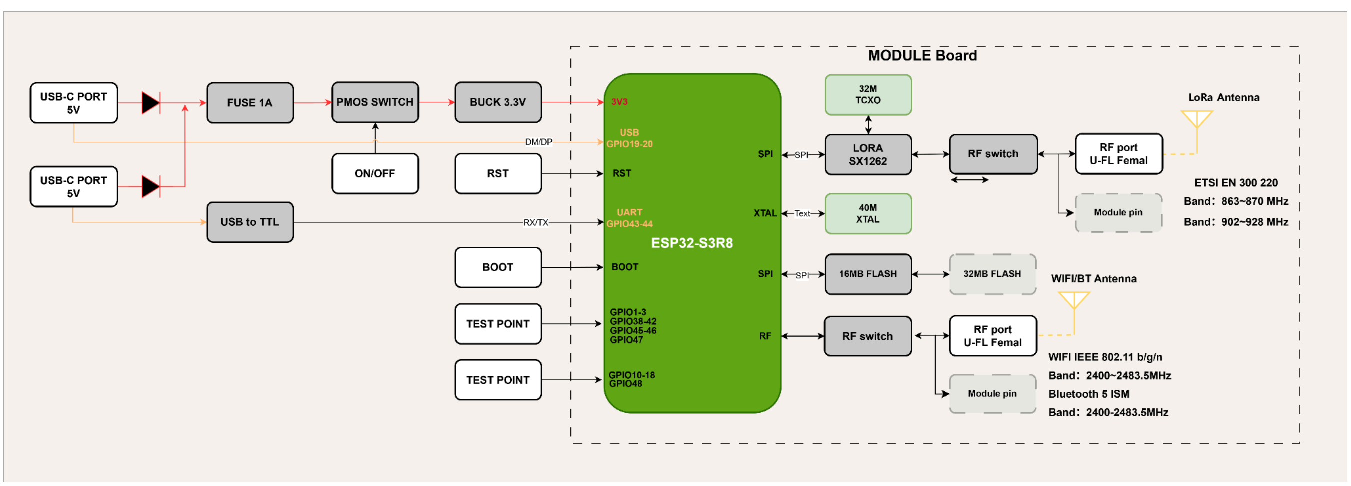

Fig. 7: Wio-S3 Module Reference Design

7 LoRaWAN® application information

7.1 LoRaWAN® application

The topology of the LoRaWAN® network is a star network, and the gateway acts as a relay between nodes and network servers. The gateway is connected to the network server through a standard IP link, and the node device uses LoRa® or FSK to communicate with one or more gateways. Communication is bidirectional, although it is mainly upstream communication from the node to the network server.

The communication between the node and the gateway uses different frequencies and rates. The choice of rate is a compromise between power consumption and distance, and different rates do not interfere with each other. According to different spreading factors and bandwidths, the rate of LoRa® can be from 300bps to 50Kbps. In order to maximize battery life and network capacity, the network server manages the node's rate and output power through rate adaptation (ADR).

The node device may transmit on a random channel at any time and at any rate, as long as the following conditions are met:

The channel currently used by the node is pseudo-random. This makes the system more resistant to interference.

The maximum transmission time (dwell time of the channel) and duty cycle of the node depends on the frequency band used and local regulations.

Wio-S3 module integrates the powerful ESP32-S3 Dual-core Xtensa LX7 MCU and a high-performance LoRa® RF chip. Benefiting from ESP32-S3's advanced power management, the module achieves ultra-low power consumption in Deep-sleep mode, making it highly suitable for various long-term battery-powered LoRaWAN® applications.

7.2 Design LoRaWAN® wireless sensor based on Wio-S3

For rapid deployment, users can easily configure and test the module by sending commands through standard serial tools. Meanwhile, Wio-S3 features a powerful, fully-open ESP32-S3 processor. It supports the global LoRaWAN® standard protocol while providing rich GPIOs and hardware interfaces (such as I2C, SPI, UART, and ADC).

Customers do not need an external host MCU; they can directly develop sensor drivers, data processing logic, and the LoRaWAN® protocol stack within the Wio-S3 module itself using ESP-IDF or Arduino IDE. This highly integrated single-module solution significantly reduces hardware costs and helps customers rapidly accelerate their sensor products to the global LoRaWAN® market.

8 Ordering information

Table 6: Version Comparison

SKU	Name	Specification
100020327	Wio-S3 Wireless Module	With IPEX connector
100079384	Wio-S3-N Wireless Module	Without IPEX connector

Technical Support: techsupport@seeed.io / sensecap@seeed.cc

Sales Team: iot@seeed.cc

9 ODM & OEM Services

With decades of ODM & OEM experience, our engineers and product experts are proficient in delivering customization service for popular open-source hardware platforms – NVIDIA®Jetson™, Raspberry Pi ®, Beagleboard ®,and more. Use the Wio-E5 module to create industrial-grade sensors or development boards for rapid AIoT implementation. We're dedicated to supporting you and streamlining your idea-to-product journey. Wearereadyto bring your product concept to the market with Seeed Studio's industrial capabilities from design, manufacturing, testing, certification, global distribution, and marketplace. Todesignwiththe Wio-E5 module, please contact iot@seeed.cc

10 Reflow Soldering Parameters

The Wio-S3 module is designed for convenient application in production, including soldering onto a PCB using reflow soldering techniques. A critical factor is that users need to select the appropriate solder paste and ensure it meets the temperature requirements during reflow.

NOTE: The module temperature must not exceed 260°C during reflow soldering. The duration in the reflow zone should not exceed 30 seconds.

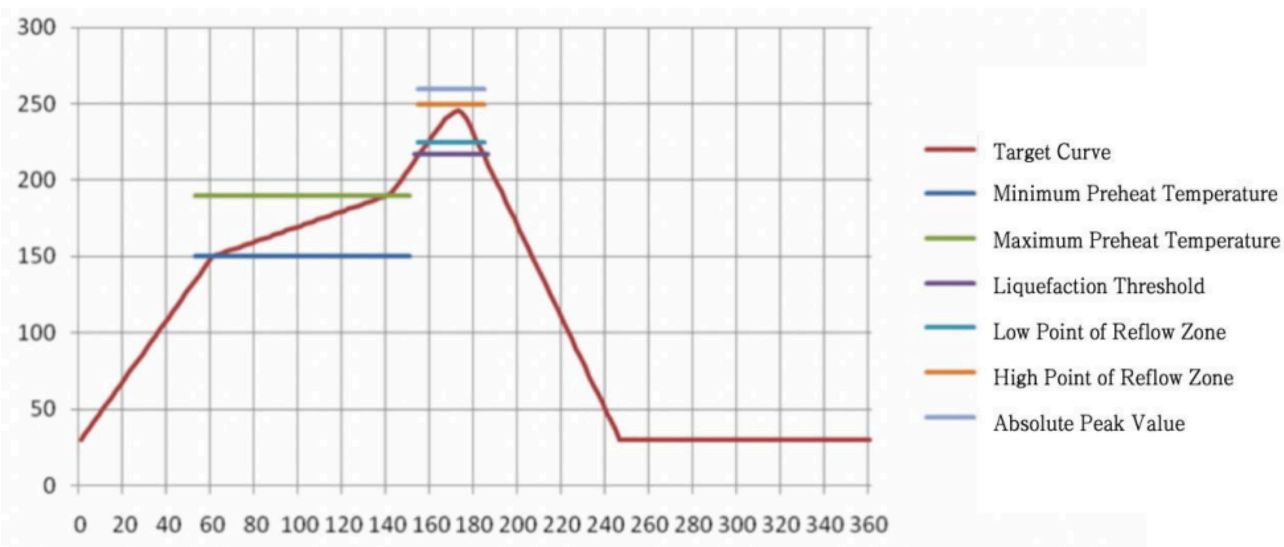

Table 7: Soldering Specification

Technical Specifications	Value	Unit
Ramp Rate	1 ~ 3	°C/Sec
Cooling Rate	2 ~ 4	°C/Sec
Preheat Ramp Rate	0.5 ~ 1	°C/Sec
Preheat Duration MIN	70	Sec
Preheat Duration MAX	120	Sec
Preheat Temperature MIN	150	°C

Preheat Temperature MAX	190	°C
Maximum time above solder paste liquefaction temperature	70	Sec
Minimum time above solder paste liquefaction temperature	50	Sec
Reflow zone dwell time	30	Sec
Peak temperature dwell time maximum	5	Sec
Recommended Liquidus Threshold	218	°C
Reflow Zone Minimum Temperature	240	°C
Reflow Zone Maximum Temperature	250	°C
Absolute Peak Temperature	260	°C

11 Version

Version	Description	Date	Author
v1.0	First release	2026-05-28	Janet Zhang