ESP32-S2 Series

Datasheet Version 1.8

Xtensa® single-core 32-bit LX7 microprocessor
2.4 GHz Wi-Fi (802.11 b/g/n)
Optional flash and PSRAM in the chip's package
43 GPIOs, rich set of peripherals
QFN56 (7×7 mm) package

Including:

ESP32-S2

ESP32-S2FH2

ESP32-S2FH4

ESP32-S2FN4R2

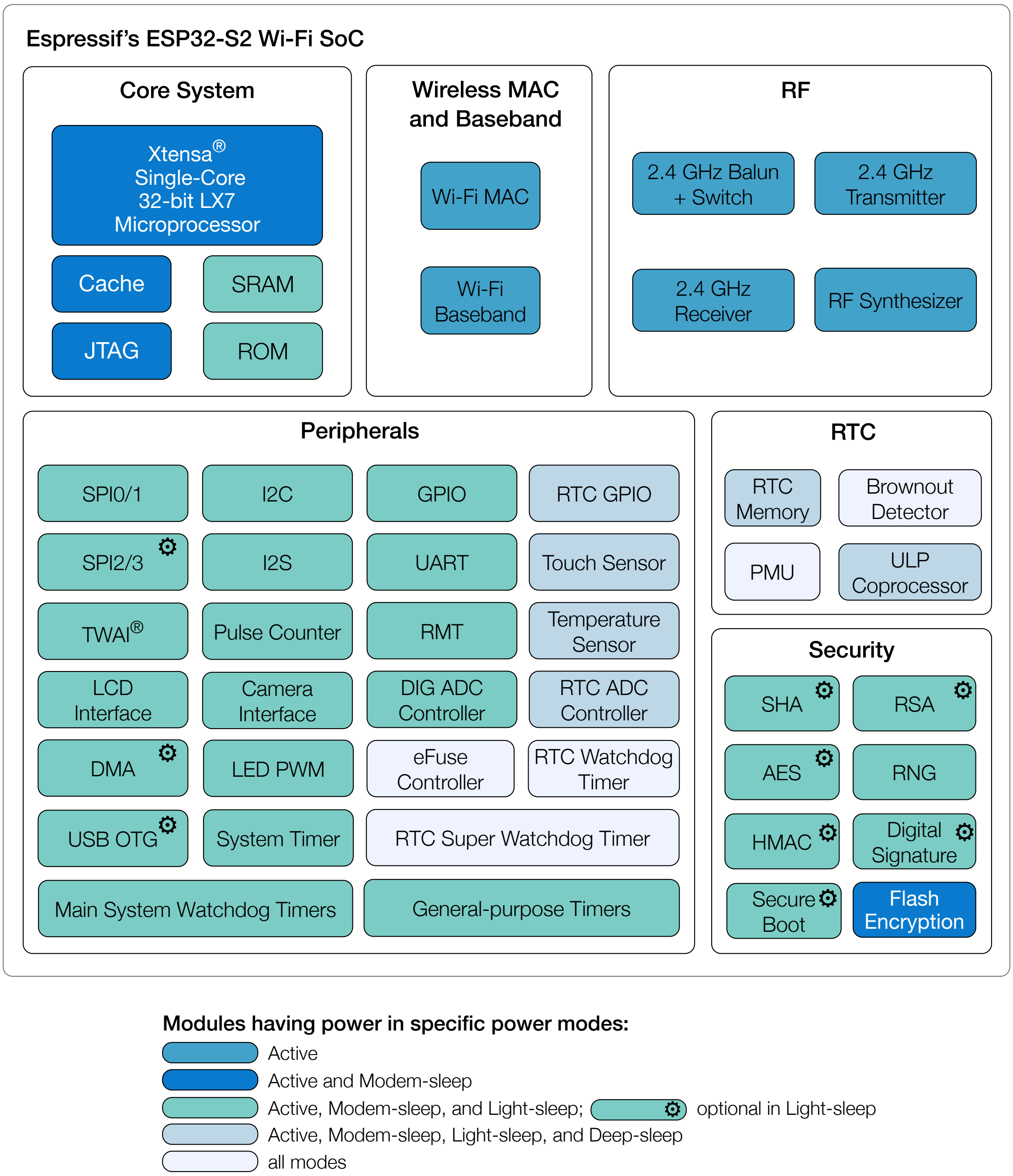
ESP32-S2R2

ESPRESSIF

Product Overview

ESP32-S2 is a highly-integrated low-power MCU-based system on a chip (SoC) with 2.4 GHz Wi-Fi. It consists of a single-core microprocessor (Xtensa® 32-bit LX7), a ultra-low-power coprocessor, a Wi-Fi baseband, RF module, and numerous peripherals.

The functional block diagram of the SoC is shown below.



ESP32-S2 Functional Block Diagram

For more information on power consumption, see Section 4.1.3.3 *Power Management Unit*.

Features

Wi-Fi

- IEEE 802.11 b/g/n-compliant
- Supports 20 MHz, 40 MHz bandwidth in 2.4 GHz band
- Single-band 1T1R mode with data rate up to 150 Mbps
- WMM
- TX/RX A-MPDU, RX A-MSDU
- Immediate Block ACK
- Fragmentation and defragmentation
- Automatic Beacon monitoring (hardware TSF)
- Four virtual Wi-Fi interfaces
- Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes
  Note that when ESP32-S2 is in Station mode and performs a scan, the SoftAP channel will change along with the Station channel.
- Antenna diversity
- 802.11mc FTM

CPU and Memory

- Xtensa® single-core 32-bit LX7 microprocessor, up to 240 MHz
- CoreMark® score:
  - 1 core at 240 MHz: 542.80 CoreMark; 2.26 CoreMark/MHz
- ROM: 128 KB
- SRAM: 320 KB
- SRAM in RTC: 16 KB
- In-package flash and PSRAM (see details in Chapter 1 *ESP32-S2 Series Comparison*)
- SPI/QSPI/OSPI supports multiple flash and external RAM chips
- Access to flash accelerated by cache
- Supports flash in-Circuit Programming (ICP)

Advanced Peripheral Interfaces

- 43 programmable GPIOs
- Digital interfaces:
  - Four SPIs
  - I2S

  - Two I2Cs
  - Two UARTs
  - RMT (TX/RX)
  - LED PWM controller, up to 8 channels
  - Four pulse counters
  - Full-speed USB OTG
  - DVP 8/16 camera interface, implemented using the hardware resources of I2S
  - LCD interface (8-bit serial RGB/8080/6800), implemented using the hardware resources of SPI2
  - LCD interface (8/16/24-bit parallel), implemented using the hardware resources of I2S
  - DMA controller
  - TWAI® controller compatible with ISO 11898-1 (CAN Specification 2.0)
- Analog interfaces:
  - Two 12-bit SAR ADCs, up to 20 channels
  - Two 8-bit DACs
  - 14 touch sensing GPIOs
  - Temperature sensor
- Timers:
  - 64-bit general-purpose timer
  - 64-bit system timer
  - Three watchdog timers
  - Super watchdog timer
  - XTAL32K watchdog timer

Power Management

- Fine-resolution power control through a selection of clock frequency, duty cycle, Wi-Fi operating modes, and individual power control of internal components
- Five power modes designed for typical scenarios: Active, Modem-sleep, Light-sleep, Deep-sleep, Hibernation
- Ultra-Low-Power (ULP) coprocessors:
  - ULP-RISC-V coprocessor
  - ULP-FSM coprocessor
- RTC memory remains powered on in Deep-sleep mode

Security

- Secure boot - permission control on accessing internal and external memory
- Flash encryption - memory encryption and decryption
- 4096-bit OTP, up to 1792 bits for users
- Cryptographic hardware acceleration:
  - AES-128/192/256 (FIPS PUB 197)
  - Hash (FIPS PUB 180-4)
  - RSA
  - Random Number Generator (RNG)
  - HMAC
  - Digital signature

RF Module

- Antenna switches, RF balun, power amplifier, low-noise receive amplifier
- Up to +19.5 dBm of power for an 802.11b transmission
- Up to +18 dBm of power for an 802.11n transmission

Applications

With low power consumption, ESP32-S2 is an ideal choice for IoT devices in the following areas:

- Smart Home
- Industrial Automation
- Health Care
- Consumer Electronics
- Smart Agriculture
- POS Machines
- Service Robot
- Audio Devices
- Generic Low-power IoT Sensor Hubs
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- USB Devices
- Speech Recognition
- Image Recognition
- Wi-Fi + Bluetooth Networking Card
- Touch and Proximity Sensing

Note:

Check the link or the QR code to make sure that you use the latest version of this document:
https://www.espressif.com/documentation/esp32-s2_datasheet_en.pdf

Contents

List of Tables

List of Figures

1 ESP32-S2 Series Comparison

1.1 Nomenclature

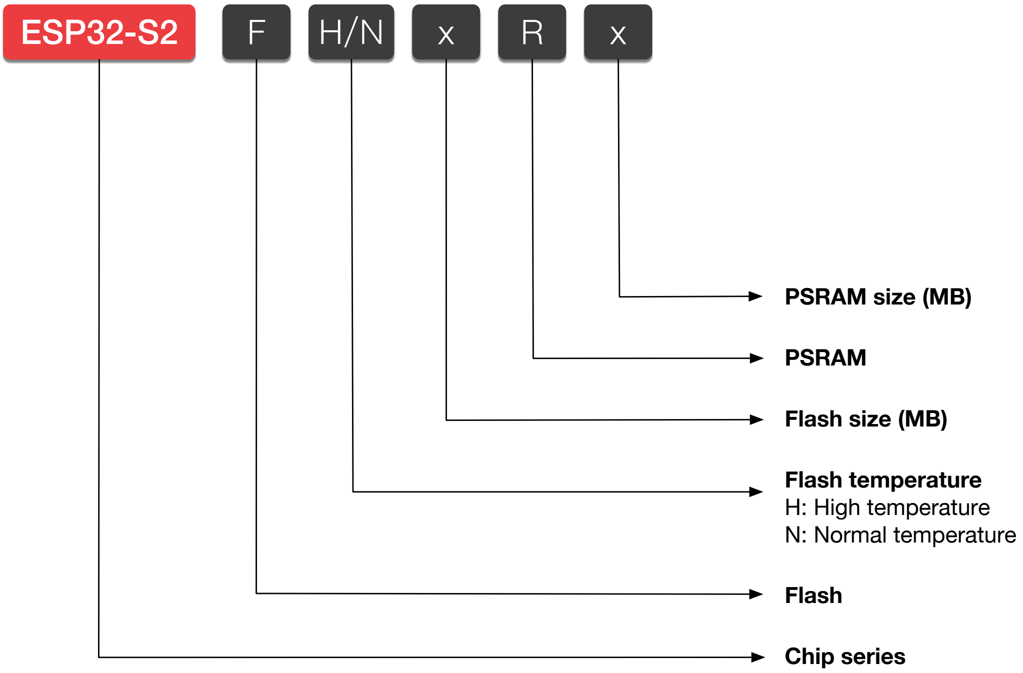


Figure 1-1. ESP32-S2 Series Nomenclature

1.2 Comparison

Table 1-1. ESP32-S2 Series Comparison

Part Number [1]	In-Package Flash[5]	In-Package PSRAM	Ambient Temp. [2]	VDD_SPI Voltage [3]
ESP32-S2	—	—	–40 ~ 105 °C [3]	3.3 V/1.8 V
ESP32-S2FH2	2 MB (Quad SPI)[4]	—	–40 ~ 105 °C	3.3 V
ESP32-S2FH4	4 MB (Quad SPI)	—	–40 ~ 105 °C	3.3 V
ESP32-S2FN4R2	4 MB (Quad SPI)	2 MB (Quad SPI)	–40 ~ 85 °C	3.3 V
ESP32-S2R2	—	2 MB (Quad SPI)	–40 ~ 85 °C	3.3 V

[1] For details on chip marking and packing, see Section 7 *Packaging*.

[2] Ambient temperature specifies the recommended temperature range of the environment immediately outside an Espressif chip.

[3] For more information on VDD_SPI, see Section 2.5 *Power Supply*.

[4] For details about SPI modes, see Section 2.6 *Pin Mapping Between Chip and Flash/PSRAM*.

2 Pins

2.1 Pin Layout

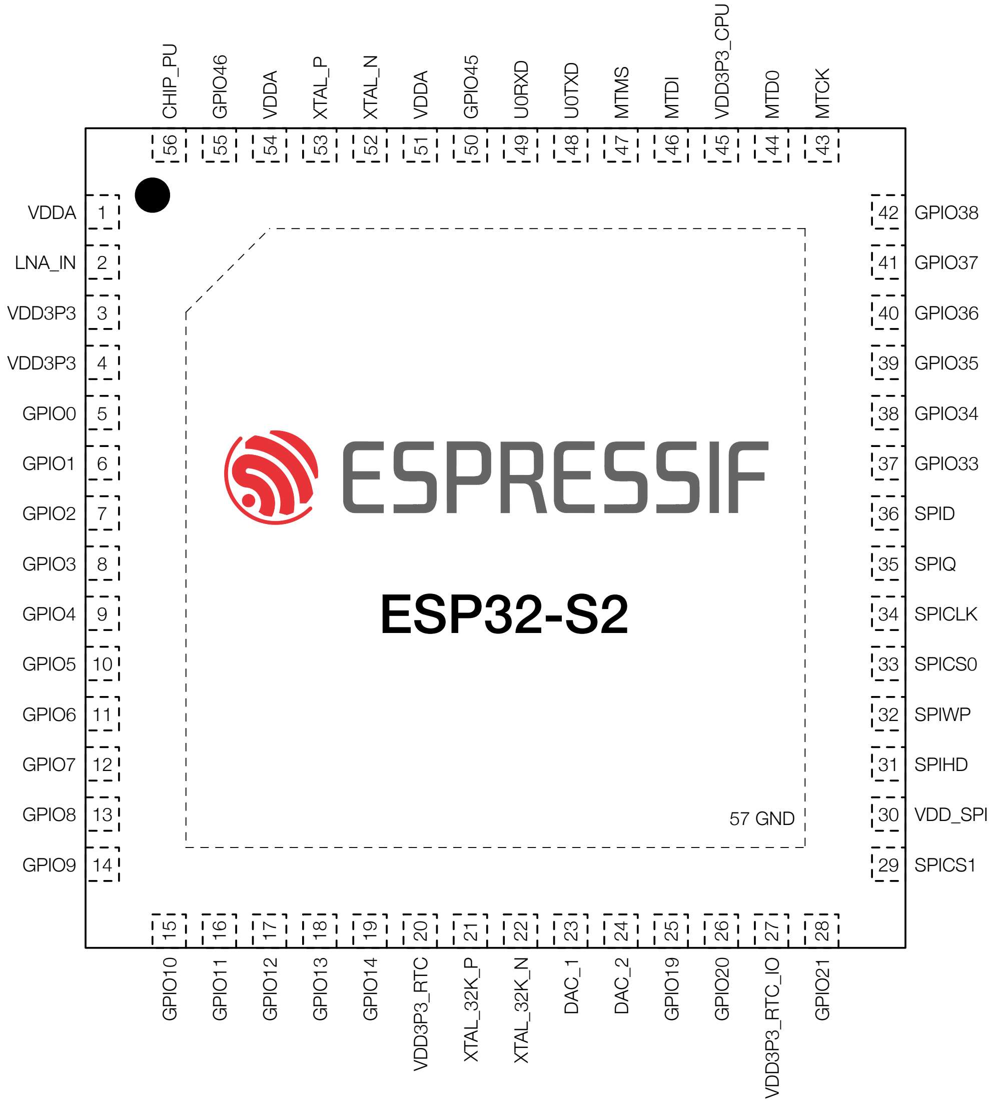


Figure 2-1. ESP32-S2 Pin Layout (Top View)

2.2 Pin Overview

The ESP32-S2 chip integrates multiple peripherals that require communication with the outside world. To keep the chip package size reasonably small, the number of available pins has to be limited. So the only way to route all the incoming and outgoing signals is through pin multiplexing. Pin muxing is controlled via software programmable registers (see *ESP32-S2 Technical Reference Manual* > Chapter *IO MUX and GPIO Matrix*).

All in all, the ESP32-S2 chip has the following types of pins:

- **IO pins** with the following predefined sets of functions to choose from:
  - **Each** IO pin has predefined **IO MUX functions** – see Table 2-3 *IO MUX Functions*
  - **Some** IO pins have predefined **RTC functions** – see Table **?? ??**
  - **Some** IO pins have predefined **analog functions** – see Table 2-8 *Analog Functions*

  Predefined functions means that each IO pin has a set of direct connections to certain signals from on-chip peripherals. During run-time, the user can configure which peripheral signal from a predefined set to connect to a certain pin at a certain time via memory mapped registers.
- **Analog pins** that have exclusively-dedicated **analog functions** – see Table 2-10 *Analog Pins*
- **Power pins** that supply power to the chip components and non-power pins – see Table 2-11 *Power Pins*

Table 2-1 *Pin Overview* gives an overview of all the pins. For more information, see the respective sections for each pin type below, or ESP32-S2 Consolidated Pin Overview.

Table 2-1. Pin Overview

				Pin Settings[5]		Pin Function Sets[1]		
Pin No.	**Pin Name**	**Pin Type**	**Pin Providing Power**[2 3 4]	**At Reset**	**After Reset**	**IO MUX**	**RTC IO MUX**	**Analog**
1	VDDA	Power						
2	LNA_IN	Analog						
3	VDD3P3	Power						
4	VDD3P3	Power						
5	GPIO0	IO	VDD3P3_RTC_IO	WPU, IE	WPU, IE	**IO MUX**	RTC IO MUX	
6	GPIO1	IO	VDD3P3_RTC_IO	IE	IE	**IO MUX**	RTC IO MUX	Analog
7	GPIO2	IO	VDD3P3_RTC_IO	IE	IE	**IO MUX**	RTC IO MUX	Analog
8	GPIO3	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
9	GPIO4	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
10	GPIO5	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
11	GPIO6	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
12	GPIO7	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
13	GPIO8	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
14	GPIO9	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
15	GPIO10	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
16	GPIO11	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
17	GPIO12	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
18	GPIO13	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
19	GPIO14	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog

Cont'd on next page

Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power[2] [3] [4]	Pin Settings[5]		Pin Function Sets[1]		
				At Reset	After Reset	IO MUX	RTC IO MUX	Analog
20	VDD3P3_RTC	Power						
21	XTAL_32K_P	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
22	XTAL_32K_N	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
23	DAC_1	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
24	DAC_2	IO	VDD3P3_RTC_IO		IE	**IO MUX**	RTC IO MUX	Analog
25	GPIO19	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
26	GPIO20	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	Analog
27	VDD3P3_RTC_IO	Power						
28	GPIO21	IO	VDD3P3_RTC_IO			**IO MUX**	RTC IO MUX	
29	SPICS1	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
30	VDD_SPI	Power						
31	SPIHD	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
32	SPIWP	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
33	SPICS0	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
34	SPICLK	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
35	SPIQ	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
36	SPID	IO	VDD_SPI	WPU, IE	WPU, IE	**IO MUX**		
37	GPIO33	IO	VDD_SPI/VDD3P3_CPU		IE	**IO MUX**		
38	GPIO34	IO	VDD_SPI/VDD3P3_CPU		IE	**IO MUX**		
39	GPIO35	IO	VDD_SPI/VDD3P3_CPU		IE	**IO MUX**		
40	GPIO36	IO	VDD_SPI/VDD3P3_CPU		IE	**IO MUX**		
41	GPIO37	IO	VDD_SPI/VDD3P3_CPU		IE	**IO MUX**		
42	GPIO38	IO	VDD3P3_CPU		IE	**IO MUX**		
43	MTCK	IO	VDD3P3_CPU		IE[6]	**IO MUX**		
44	MTDO	IO	VDD3P3_CPU		IE	**IO MUX**		
45	VDD3P3_CPU	Power						
46	MTDI	IO	VDD3P3_CPU		IE	**IO MUX**		
47	MTMS	IO	VDD3P3_CPU		IE	**IO MUX**		
48	U0TXD	IO	VDD3P3_CPU	WPU, IE	WPU, IE	**IO MUX**		
49	U0RXD	IO	VDD3P3_CPU	WPU, IE	WPU, IE	**IO MUX**		
50	GPIO45	IO	VDD3P3_CPU	WPD, IE	WPD, IE	**IO MUX**		
51	VDDA	Power						
52	XTAL_N	Analog						
53	XTAL_P	Analog						
54	VDDA	Power						
55	GPIO46	IO	VDD3P3_CPU	WPD, IE	WPD, IE	**IO MUX**		
56	CHIP_PU	Analog	VDD3P3_RTC_IO					

1. **Bold** marks the pin function set in which a pin has its default function in the default boot mode. See Section 3.1 *Chip Boot Mode Control*.

2. In column **Pin Providing Power**, regarding pins powered by VDD_SPI:
   - Power actually comes from the internal power rail supplying power to VDD_SPI. For details, see Section 2.5.2 *Power Scheme*.

3. In column **Pin Providing Power**, regarding pins powered by VDD3P3_CPU / VDD_SPI:
   - Pin Providing Power (either VDD3P3_CPU or VDD_SPI) can be configured via a register, see *ESP32-S2 Technical Reference Manual* > Chapter *IO MUX and GPIO Matrix*.

4. Default drive strength for all IO pins is 20 mA.

5. Column **Pin Settings** shows predefined settings at reset and after reset with the following abbreviations:
   - IE – input enabled
   - WPU – internal weak pull-up resistor enabled
   - WPD – internal weak pull-down resistor enabled
6. Depends on the value of EFUSE_DIS_PAD_JTAG
   - `0` - WPU is enabled
   - `1` - pin floating

2.3 IO Pins

2.3.1 IO MUX Functions

The IO MUX allows multiple input/output signals to be connected to a single input/output pin. Each IO pin of ESP32-S2 can be connected to one of the five signals (IO MUX functions, i.e., F0-F4), as listed in Table 2-3 *IO MUX Functions*.

Among the five sets of signals:

- Some are routed via the GPIO Matrix (**GPIO0, GPIO1, etc.**), which incorporates internal signal routing circuitry for mapping signals programmatically. It gives the pin access to almost any peripheral signals. However, the flexibility of programmatic mapping comes at a cost as it might affect the latency of routed signals. For details about connecting to peripheral signals via GPIO Matrix, see *ESP32-S2 Technical Reference Manual* > Chapter *IO MUX and GPIO Matrix*.
- Some are directly routed from certain peripherals (**U0TXD, MTCK, etc.**), including UART0/1, JTAG, SPI0/1, and SPI2 - see Table 2-2 *Peripheral Signals Routed via IO MUX*.

Table 2-2. Peripheral Signals Routed via IO MUX

Pin Function	Signal	Description
U...TXD U...RXD U...RTS U...CTS	Transmit data Receive data Request to send Clear to send	UART0/1 interface
MTCK MTDO MTDI MTMS	Test clock Test Data Out Test Data In Test Mode Select	JTAG interface for debugging
SPIQ SPID SPIHD SPIWP SPICLK SPICS...	Master in, slave out Master out, slave in Hold Write protect Clock Chip select	3.3 V SPI0/1 interface for connection to in-package or off-package flash/PSRAM via the SPI bus. It supports 1-, 2-, 4-line SPI modes. See also Section 2.6 *Pin Mapping Between Chip and Flash/PSRAM*
SPIIO... SPIDQS	Data Data strobe/data mask	The higher 4 bits data line interface and DQS interface for 3.3 V SPI0/1 interface in 8-line SPI mode
SUBSPIQ SUBSPID SUBSPIHD SUBSPIWP SUBSPICLK SUBSPICS...	Master in, slave out Master out, slave in Hold Write protect Clock Chip select	1.8 V SPI0/1 interface for connection to in-package or off-package flash/PSRAM via the SUBSPI bus. It supports 1-, 2-, 4-line SPI modes
FSPIQ FSPID FSPIHD FSPIWP FSPICLK FSPICS0	Master in, slave out Master out, slave in Hold Write protect Clock Chip select	SPI2 interface for fast SPI connection. It supports 1-, 2-, 4-line SPI modes
FSPIIO...	Data	The higher 4 bits data line interface and DQS interface for SPI2 interface in 8-line SPI mode

Cont'd on next page

Table 2-2 – cont’d from previous page

Pin Function	Signal	Description
FSPIDQS	Data strobe/data mask	
CLK_OUT...	Clock output	Output clock signals generated by the chip’s internal components

Table 2-3 *IO MUX Functions* shows the IO MUX functions of IO pins.

Table 2-3. IO MUX Functions

		IO MUX Function									
Pin No.	GPIO	F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
5	GPIO0	**GPIO0**	I/O/T	GPIO0	I/O/T						
6	GPIO1	**GPIO1**	I/O/T	GPIO1	I/O/T						
7	GPIO2	**GPIO2**	I/O/T	GPIO2	I/O/T						
8	GPIO3	**GPIO3**	I/O/T	GPIO3	I/O/T						
9	GPIO4	**GPIO4**	I/O/T	GPIO4	I/O/T						
10	GPIO5	**GPIO5**	I/O/T	GPIO5	I/O/T						
11	GPIO6	**GPIO6**	I/O/T	GPIO6	I/O/T						
12	GPIO7	**GPIO7**	I/O/T	GPIO7	I/O/T						
13	GPIO8	**GPIO8**	I/O/T	GPIO8	I/O/T			SUBSPICS1	O/T		
14	GPIO9	**GPIO9**	I/O/T	GPIO9	I/O/T			SUBSPIHD	I1/O/T	FSPIHD	I1/O/T
15	GPIO10	**GPIO10**	I/O/T	GPIO10	I/O/T	FSPIIO4	I1/O/T	SUBSPICS0	O/T	FSPICS0	I1/O/T
16	GPIO11	**GPIO11**	I/O/T	GPIO11	I/O/T	FSPIIO5	I1/O/T	SUBSPID	I1/O/T	FSPID	I1/O/T
17	GPIO12	**GPIO12**	I/O/T	GPIO12	I/O/T	FSPIIO6	I1/O/T	SUBSPICLK	O/T	FSPICLK	I1/O/T
18	GPIO13	**GPIO13**	I/O/T	GPIO13	I/O/T	FSPIIO7	I1/O/T	SUBSPIQ	I1/O/T	FSPIQ	I1/O/T
19	GPIO14	**GPIO14**	I/O/T	GPIO14	I/O/T	FSPIDQS	O/T	SUBSPIWP	I1/O/T	FSPIWP	I1/O/T
21	GPIO15	**GPIO15**	I/O/T	GPIO15	I/O/T	U0RTS	O				
22	GPIO16	**GPIO16**	I/O/T	GPIO16	I/O/T	U0CTS	I1				
23	GPIO17	**GPIO17**	I/O/T	GPIO17	I/O/T	U1TXD	O				
24	GPIO18	**GPIO18**	I/O/T	GPIO18	I/O/T	U1RXD	I1	CLK_OUT3	O		
25	GPIO19	**GPIO19**	I/O/T	GPIO19	I/O/T	U1RTS	O	CLK_OUT2	O		
26	GPIO20	**GPIO20**	I/O/T	GPIO20	I/O/T	U1CTS	I1	CLK_OUT1	O		
28	GPIO21	**GPIO21**	I/O/T	GPIO21	I/O/T						
29	GPIO26	**SPICS1**	O/T	GPIO26	I/O/T						
31	GPIO27	**SPIHD**	I1/O/T	GPIO27	I/O/T						
32	GPIO28	**SPIWP**	I1/O/T	GPIO28	I/O/T						
33	GPIO29	**SPICS0**	O/T	GPIO29	I/O/T						
34	GPIO30	**SPICLK**	O/T	GPIO30	I/O/T						
35	GPIO31	**SPIQ**	I1/O/T	GPIO31	I/O/T						
36	GPIO32	**SPID**	I1/O/T	GPIO32	I/O/T						
37	GPIO33	**GPIO33**	I/O/T	GPIO33	I/O/T	FSPIHD	I1/O/T	SUBSPIHD	I1/O/T	SPIIO4	I1/O/T
38	GPIO34	**GPIO34**	I/O/T	GPIO34	I/O/T	FSPICS0	I1/O/T	SUBSPICS0	O/T	SPIIO5	I1/O/T
39	GPIO35	**GPIO35**	I/O/T	GPIO35	I/O/T	FSPID	I1/O/T	SUBSPID	I1/O/T	SPIIO6	I1/O/T
40	GPIO36	**GPIO36**	I/O/T	GPIO36	I/O/T	FSPICLK	I1/O/T	SUBSPICLK	O/T	SPIIO7	I1/O/T
41	GPIO37	**GPIO37**	I/O/T	GPIO37	I/O/T	FSPIQ	I1/O/T	SUBSPIQ	I1/O/T	SPIDQS	I0/O/T

Cont’d on next page

Cont'd from previous page

		IO MUX Function									
Pin No.	GPIO	F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
42	GPIO38	**GPIO38**	I/O/T	GPIO38	I/O/T	FSPIWP	I1/O/T	SUBSPIWP	I1/O/T		
43	GPIO39	**MTCK**	I1	GPIO39	I/O/T	CLK_OUT3	O	SUBSPICS1	O/T		
44	GPIO40	**MTDO**	O/T	GPIO40	I/O/T	CLK_OUT2	O				
46	GPIO41	**MTDI**	I1	GPIO41	I/O/T	CLK_OUT1	O				
47	GPIO42	**MTMS**	I0	GPIO42	I/O/T						
48	GPIO43	**U0TXD**	O	GPIO43	I/O/T	CLK_OUT1	O				
49	GPIO44	**U0RXD**	I1	GPIO44	I/O/T	CLK_OUT2	O				
50	GPIO45	**GPIO45**	I/O/T	GPIO45	I/O/T						
55	GPIO46	**GPIO46**	I	GPIO46	I						

[1] **Bold** marks the default pin functions in the default boot mode. See Section 3.1 *Chip Boot Mode Control*.

[2] Regarding highlighted cells, see Section 2.3.5 *Restrictions for GPIOs and RTC_GPIOs*.

[3] Each IO MUX function (F*n*, *n* = 0 ~ 4) is associated with a *type*. The description of *type* is as follows:

- I – input. O – output. T – high impedance.
- I1 – input; if the pin is assigned a function other than F*n*, the input signal of F*n* is always `1`.
- I0 – input; if the pin is assigned a function other than F*n*, the input signal of F*n* is always `0`.

2.3.2 RTC Functions

When the chip is in Deep-sleep mode, the IO MUX described in Section 2.3.1 *IO MUX Functions* will not work. That is where the RTC IO MUX comes in. It allows multiple input/output signals to be a single input/output pin in Deep-sleep mode, as the pin is connected to the RTC system and powered by VDD3P3_RTC.

RTC IO pins can be assigned to **RTC functions**. They can

- Either work as RTC GPIOs (**RTC_GPIO0, RTC_GPIO1, etc.**), connected to the ULP coprocessor
- Or connect to RTC peripheral signals (**sar_i2c_scl_0, sar_i2c_sda_0, etc.**) - see Table 2-5 *RTC Peripheral Signals Routed via RTC IO MUX*

Table 2-4. RTC Peripheral Signals Routed via RTC IO MUX

Pin Function	Signal	Description
sar_i2c_scl...	Serial clock	RTC I2C0/1 interface
sar_i2c_sda...	Serial data	

Table **?? ??** shows the RTC functions of RTC IO pins.

2.3.3 RTC Functions

When the chip is in Deep-sleep mode, the IO MUX described in Section 2.3.1 *IO MUX Functions* will not work. That is where the RTC IO MUX comes in. It allows multiple input/output signals to be a single input/output pin in Deep-sleep mode, as the pin is connected to the RTC system and powered by VDD3P3_RTC.

RTC IO pins can be assigned to **RTC functions**. They can

- Either work as RTC GPIOs (**RTC_GPIO0, RTC_GPIO1, etc.**), connected to the ULP coprocessor
- Or connect to RTC peripheral signals (**sar_i2c_scl_0, sar_i2c_sda_0, etc.**) - see Table 2-5 *RTC Peripheral Signals Routed via RTC IO MUX*

Table 2-5. RTC Peripheral Signals Routed via RTC IO MUX

Pin Function	Signal	Description
sar_i2c_scl...	Serial clock	RTC I2C0/1 interface
sar_i2c_sda...	Serial data	

Table 2-6 *RTC Functions* shows the RTC functions of RTC IO pins.

Table 2-6. RTC Functions

Pin	RTC	RTC Function [2]			
No.	IO Name [1]	F0	F1	F2	F3
5	RTC_GPIO0	RTC_GPIO0			sar_i2c_scl_0
6	RTC_GPIO1	RTC_GPIO1			sar_i2c_sda_0
7	RTC_GPIO2	RTC_GPIO2			sar_i2c_scl_1
8	RTC_GPIO3	RTC_GPIO3			sar_i2c_sda_1
9	RTC_GPIO4	RTC_GPIO4			
10	RTC_GPIO5	RTC_GPIO5			

Cont'd on next page

Table 2-6 – cont'd from previous page

Pin	RTC	RTC Function[2]			
No.	IO Name[1]	F0	F1	F2	F3
11	RTC_GPIO6	RTC_GPIO6			
12	RTC_GPIO7	RTC_GPIO7			
13	RTC_GPIO8	RTC_GPIO8			
14	RTC_GPIO9	RTC_GPIO9			
15	RTC_GPIO10	RTC_GPIO10			
16	RTC_GPIO11	RTC_GPIO11			
17	RTC_GPIO12	RTC_GPIO12			
18	RTC_GPIO13	RTC_GPIO13			
19	RTC_GPIO14	RTC_GPIO14			
21	RTC_GPIO15	RTC_GPIO15			
22	RTC_GPIO16	RTC_GPIO16			
23	RTC_GPIO17	RTC_GPIO17			
24	RTC_GPIO18	RTC_GPIO18			
25	RTC_GPIO19	RTC_GPIO19			
26	RTC_GPIO20	RTC_GPIO20			
28	RTC_GPIO21	RTC_GPIO21			

[1] This column lists the RTC GPIO names, since RTC functions are configured with RTC GPIO registers that use RTC GPIO numbering.

[2] Regarding highlighted cells, see Section 2.3.5 *Restrictions for GPIOs and RTC_GPIOs*.

2.3.4 Analog Functions

Some IO pins also have **analog functions**, for analog peripherals (such as ADC) in any power mode. Internal analog signals are routed to these analog functions, see Table *2-7 Analog Signals Routed to Analog Functions*.

Table 2-7. Analog Signals Routed to Analog Functions

Pin Function	Signal	Description
TOUCH...	Touch sensor channel ... signal	Touch sensor interface
ADC..._CH...	ADC1/2 channel ... signal	ADC1/2 interface
XTAL_32K_N XTAL_32K_P	Negative clock signal Positive clock signal	32 kHz external clock input/output connected to ESP32-S2's crystal or oscillator
USB_D- USB_D+	Data - Data +	USB OTG
DAC_...	DAC_...	Digital-to-Analog converter

Table *2-8 Analog Functions* shows the analog functions of IO pins.

Table 2-8. Analog Functions

		Analog Function	
Pin No.	RTC IO[1]	F0	F1
6	RTC_GPIO1	TOUCH1	ADC1_CH0
7	RTC_GPIO2	TOUCH2	ADC1_CH1
8	RTC_GPIO3	TOUCH3	ADC1_CH2
9	RTC_GPIO4	TOUCH4	ADC1_CH3
10	RTC_GPIO5	TOUCH5	ADC1_CH4
11	RTC_GPIO6	TOUCH6	ADC1_CH5
12	RTC_GPIO7	TOUCH7	ADC1_CH6
13	RTC_GPIO8	TOUCH8	ADC1_CH7
14	RTC_GPIO9	TOUCH9	ADC1_CH8
15	RTC_GPIO10	TOUCH10	ADC1_CH9
16	RTC_GPIO11	TOUCH11	ADC2_CH0
17	RTC_GPIO12	TOUCH12	ADC2_CH1
18	RTC_GPIO13	TOUCH13	ADC2_CH2
19	RTC_GPIO14	TOUCH14	ADC2_CH3
21	RTC_GPIO15	XTAL_32K_P	ADC2_CH4
22	RTC_GPIO16	XTAL_32K_N	ADC2_CH5
23	RTC_GPIO17	DAC_1	ADC2_CH6
24	RTC_GPIO18	DAC_2	ADC2_CH7
25	RTC_GPIO19	USB_D-	ADC2_CH8
26	RTC_GPIO20	USB_D+	ADC2_CH9

[1] This column lists the RTC GPIO names, since analog functions are configured with RTC GPIO registers that use RTC GPIO numbering.

[2] Regarding highlighted cells, see Section 2.3.5 *Restrictions for GPIOs and RTC_GPIOs*.

2.3.5 Restrictions for GPIOs and RTC_GPIOs

All IO pins of ESP32-S2 have GPIO and some have RTC_GPIO pin functions. However, the IO pins are multiplexed and can be configured for different purposes based on the requirements. Some IOs have restrictions for usage. It is essential to consider the multiplexed nature and the limitations when using these IO pins.

In tables of this chapter, some pin functions are in red or yellow. These functions indicate pins that require extra caution when used as GPIO / GPIO :

- IO Pins – allocated for communication with in-package flash/PSRAM and NOT recommended for other uses. For details, see Section 2.6 *Pin Mapping Between Chip and Flash/PSRAM*.
- IO Pins – have one of the following important functions:
  - **Strapping pins** – need to be at certain logic levels at startup. See Section 3 *Boot Configurations*.

    > **Note:**
    > Strapping pins are highlighted by *Pin Name* or configurations *At Reset*, instead of the pin functions.

  - **USB_D+/-** – by default, connected to the USB OTG. To function as GPIOs, these pins need to be reconfigured.
  - **JTAG interface** – often used for debugging. See Table 2-2 *Peripheral Signals Routed via IO MUX*.
  - **UART0 interface** – often used for debugging. See Table 2-2 *Peripheral Signals Routed via IO MUX*.
  - **8-line SPI interface** – no restrictions, unless the chip is connected to flash/PSRAM using 8-line SPI mode.

For more information about assigning pins, please see Section 2.3.6 *Peripheral Pin Assignment* and ESP32-S2 Consolidated Pin Overview.

2.3.6 Peripheral Pin Assignment

Table 2-9 *Peripheral Pin Assignment* highlights which pins can be assigned to each peripheral interface according to the following priorities:

- Priority 1 (P1) : Fixed pins connected directly to peripheral signals via IO MUX or RTC IO MUX.
  If a peripheral interface does not have priority 1 pins, such as UART2, it can be assigned to any GPIO pins from priority 2 to priority 4.
- Any GPIO pins mapping to peripheral signals via GPIO Matrix, can be priority 2, 3, or 4.
  - Priority 2 (P2) : GPIO pins can be freely used without restrictions.
  - Priority 3 (P3) : GPIO pins should be used with caution, as they may conflict with the following important functions described in Section 2.3.5 *Restrictions for GPIOs and RTC_GPIOs*:
    - GPIO0, GPIO45, GPIO46 : Strapping pins.
    - GPIO39, GPIO40, GPIO41, GPIO42 : JTAG interface.
    - GPIO43, GPIO44 : UART0 interface.
    - GPIO33, GPIO34, GPIO35, GPIO36, GPIO37 : The higher 4 bits data line interface and DQS interface for the SPI0/1 interface in 8-line SPI mode, and can be GPIO pins if the chip is not connected to flash or PSRAM in 8-line SPI mode.
  - Priority 4 (P4) : GPIO pins already allocated or not recommended for use, as described in Section 2.3.5 *Restrictions for GPIOs and RTC_GPIOs*:
    - GPIO26, GPIO27, GPIO28, GPIO29, GPIO30, GPIO31, GPIO32 : SPI0/1 interface connected to the in-package flash and PSRAM, or recommended for the off-package flash and PSRAM.

  If a peripheral interface does not have priority 2 to 4 pins, such as USB Serial/JTAG, it means it can be assigned only to priority 1 pins.

Note:

- For details about which peripheral signals are connected to IO MUX or RTC IO MUX pins, please refer to Section 2.3.1 *IO MUX Functions* or Section 2.3.3 *RTC Functions*.
- For details about which peripheral signals can be assigned to GPIO pins, please refer to *ESP32-S2 Technical Reference Manual* > Chapter IO MUX and GPIO Matrix > Section Peripheral Signal List.

Table 2-9. Peripheral Pin Assignment

Pin No.	Pin Name	ADC1	ADC2	DAC	Touch Sensor	JTAG	UART0	UART1	SPI0/SPI1 (recommended)	SPI0/SPI1 (alternative)	SPI2 (recommended)	SPI2 (alternative)	SPI3	USB OTG	I2S	I2C	PCNT	RMT	TWAI	LEDC
1	VDDA																			
2	LNA_IN																			
3	VDD3P3																			
4	VDD3P3																			
5	GPIO0						GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)	GPIO0 (P3)
6	GPIO1	ADC1_CH0 (P1)			TOUCH1 (P1)		GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)	GPIO1 (P2)
7	GPIO2	ADC1_CH1 (P1)			TOUCH2 (P1)		GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)	GPIO2 (P2)
8	GPIO3	ADC1_CH2 (P1)			TOUCH3 (P1)		GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)	GPIO3 (P2)
9	GPIO4	ADC1_CH3 (P1)			TOUCH4 (P1)		GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)	GPIO4 (P2)
10	GPIO5	ADC1_CH4 (P1)			TOUCH5 (P1)		GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)	GPIO5 (P2)
11	GPIO6	ADC1_CH5 (P1)			TOUCH6 (P1)		GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)	GPIO6 (P2)
12	GPIO7	ADC1_CH6 (P1)			TOUCH7 (P1)		GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)	GPIO7 (P2)
13	GPIO8	ADC1_CH7 (P1)			TOUCH8 (P1)		GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	SUBSPICS1 (P1)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)	GPIO8 (P2)
14	GPIO9	ADC1_CH8 (P1)			TOUCH9 (P1)		GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	SUBSPIHD (P1)	FSPIHD (P1)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)	GPIO9 (P2)
15	GPIO10	ADC1_CH9 (P1)			TOUCH10 (P1)		GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)	SUBSPICS0 (P1)	FSPCS0 (P1)	FSPIIO4 (P1)	GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)	GPIO10 (P2)
16	GPIO11		ADC2_CH0 (P1)		TOUCH11 (P1)		GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)	SUBSPID (P1)	FSPID (P1)	FSPIIO5 (P1)	GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)	GPIO11 (P2)
17	GPIO12		ADC2_CH1 (P1)		TOUCH12 (P1)		GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)	SUBSPICLK (P1)	FSPICLK (P1)	FSPIIO6 (P1)	GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)	GPIO12 (P2)
18	GPIO13		ADC2_CH2 (P1)		TOUCH13 (P1)		GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)	SUBSPIQ (P1)	FSPIQ (P1)	FSPIIO7 (P1)	GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)	GPIO13 (P2)
19	GPIO14		ADC2_CH3 (P1)		TOUCH14 (P1)		GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)	SUBSPIWP (P1)	FSPIWP (P1)	FSPIDQS (P1)	GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)	GPIO14 (P2)
20	VDD3P3_RTC																			
21	XTAL_32K_P		ADC2_CH4 (P1)				U0RTS (P1)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)	GPIO15 (P2)
22	XTAL_32K_N		ADC2_CH5 (P1)				U0CTS (P1)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)	GPIO16 (P2)
23	DAC_1		ADC2_CH6 (P1)	DAC_1 (P1)			GPIO17 (P2)	U1TXD (P1)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)	GPIO17 (P2)
24	DAC_2		ADC2_CH7 (P1)	DAC_2 (P1)			GPIO18 (P2)	U1RXD (P1)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)	GPIO18 (P2)
25	GPIO19		ADC2_CH8 (P1)				GPIO19 (P3)	U1RTS (P1)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)	USB_D- (P1)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)	GPIO19 (P3)
26	GPIO20		ADC2_CH9 (P1)				GPIO20 (P3)	U1CTS (P1)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)	USB_D+ (P1)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)	GPIO20 (P3)
27	VDD3P3_RTC_IO																			
28	GPIO21						GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)	GPIO21 (P2)
29	SPICS1						GPIO26 (P4)	GPIO26 (P4)	SPICS1 (P1)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)	GPIO26 (P4)
30	VDD_SPI																			
31	SPIHD						GPIO27 (P4)	GPIO27 (P4)	SPIHD (P1)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)	GPIO27 (P4)
32	SPIWP						GPIO28 (P4)	GPIO28 (P4)	SPIWP (P1)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)	GPIO28 (P4)
33	SPICS0						GPIO29 (P4)	GPIO29 (P4)	SPICS0 (P1)	GPIO29 (P4)	GPIO29 (P4)	FSPIHD (P1)	GPIO29 (P4)	GPIO29 (P4)	GPIO29 (P4)	GPIO29 (P4)	GPIO29 (P4)	GPIO29 (P4)	GPIO29 (P4)	GPIO29 (P4)
34	SPICLK						GPIO30 (P4)	GPIO30 (P4)	SPICLK (P1)	GPIO30 (P4)	GPIO30 (P4)	FSPICS0 (P1)	GPIO30 (P4)	GPIO30 (P4)	GPIO30 (P4)	GPIO30 (P4)	GPIO30 (P4)	GPIO30 (P4)	GPIO30 (P4)	GPIO30 (P4)
35	SPIQ						GPIO31 (P4)	GPIO31 (P4)	SPIQ (P1)	GPIO31 (P4)	GPIO31 (P4)	FSPID (P1)	GPIO31 (P4)	GPIO31 (P4)	GPIO31 (P4)	GPIO31 (P4)	GPIO31 (P4)	GPIO31 (P4)	GPIO31 (P4)	GPIO31 (P4)
36	SPID						GPIO32 (P4)	GPIO32 (P4)	SPID (P1)	GPIO32 (P4)	GPIO32 (P4)	FSPICLK (P1)	GPIO32 (P4)	GPIO32 (P4)	GPIO32 (P4)	GPIO32 (P4)	GPIO32 (P4)	GPIO32 (P4)	GPIO32 (P4)	GPIO32 (P4)
37	GPIO33						GPIO33 (P3)	GPIO33 (P3)	SPIIO4 (P1)	SUBSPIHD (P1)	GPIO33 (P3)	FSPIQ (P1)	GPIO33 (P3)	GPIO33 (P3)	GPIO33 (P3)	GPIO33 (P3)	GPIO33 (P3)	GPIO33 (P3)	GPIO33 (P3)	GPIO33 (P3)
38	GPIO34						GPIO34 (P3)	GPIO34 (P3)	SPIIO5 (P1)	SUBSPICS0 (P1)	GPIO34 (P3)	FSPIWP (P1)	GPIO34 (P3)	GPIO34 (P3)	GPIO34 (P3)	GPIO34 (P3)	GPIO34 (P3)	GPIO34 (P3)	GPIO34 (P3)	GPIO34 (P3)
39	GPIO35						GPIO35 (P3)	GPIO35 (P3)	SPIIO6 (P1)	SUBSPID (P1)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)	GPIO35 (P3)
40	GPIO36						GPIO36 (P3)	GPIO36 (P3)	SPIIO7 (P1)	SUBSPICLK (P1)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)	GPIO36 (P3)
41	GPIO37						GPIO37 (P3)	GPIO37 (P3)	SPIDQS (P1)	SUBSPIQ (P1)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)	GPIO37 (P3)
42	GPIO38						GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	SUBSPIWP (P1)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)	GPIO38 (P2)
43	MTCK					MTCK (P1)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	SUBSPICS1 (P1)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)	GPIO39 (P3)
44	MTDO					MTDO (P1)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)	GPIO40 (P3)
45	VDD3P3_CPU																			
46	MTDI					MTDI (P1)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)	GPIO41 (P3)
47	MTMS					MTMS (P1)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)	GPIO42 (P3)
48	U0TXD						U0TXD (P1)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)	GPIO43 (P3)
49	U0RXD						U0RXD (P1)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)	GPIO44 (P3)
50	GPIO45						GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)	GPIO45 (P3)
51	VDDA																			
52	XTAL_N																			
53	XTAL_P																			
54	VDDA																			
55	GPIO46						GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)	GPIO46 (P3)
56	CHIP_PU																			

[1] For USB OTG, use USB_D- and USB_D+ when on internal PHY, and the USB_D- and USB_D+ can be swapped by configuring the EFUSE_USB_EXCHG_PINS bit according to *ESP32-S2 Technical Reference Manual*; use any GPIOs via GPIO Matrix when on external PHY.

[2] Signals of UART0, UART1, SPI0/1, SPI2 and USB OTG interfaces can be mapped to any GPIO pins through the GPIO Matrix, regardless of whether they are directly routed to fixed pins via IO MUX.

2.4 Analog Pins

Table 2-10. Analog Pins

Pin No.	Pin Name	Pin Type	Pin Function
2	LNA_IN	I/O	Low Noise Amplifier (RF LNA) input / output signals
52	XTAL_N	—	External clock input/output connected to ESP32-S2's oscillator. P/N means differential clock positive/negative.
53	XTAL_P		
56	CHIP_PU	I	High: on, the chip is started up. Low: off, the chip is shut down. Note: Do not leave the CHIP_PU pin floating.

2.5 Power Supply

2.5.1 Power Pins

The chip is powered via the power pins described in Table 2-11 *Power Pins*.

Table 2-11. Power Pins

Pin No.	Pin Name	Direction	Power Supply	
			Power Domain/Other	IO Pins
1	VDDA	Input	Analog power domain	
3	VDD3P3	Input	Analog power domain	
4	VDD3P3	Input	Analog power domain	
20	VDD3P3_RTC	Input	RTC domain	
27	VDD3P3_RTC_IO	Input	RTC and part of Digital power domains	RTC IO
30	VDD_SPI [3,4]	Input	In-package memory (backup power line)	
		Output	In-package and off-package flash/PSRAM	SPI IO
45	VDD3P3_CPU	Input	Digital power domain	Digital IO
51	VDDA	Input	Analog power domain	
54	VDDA	Input	Analog power domain	
57	GND	—	External ground connection	

[1] See in conjunction with Section 2.5.2 *Power Scheme*.

[2] For recommended and maximum voltage and current, see Section 5.1 *Absolute Maximum Ratings* and Section 5.2 *Recommended Operating Conditions*.

[3] To configure VDD_SPI as input or output, see *ESP32-S2 Technical Reference Manual* > Chapter *Low-power Management*.

[4] To configure output voltage, see Section 3.2 *VDD_SPI Voltage Control* and Section 5.3 *VDD_SPI Output Characteristics*.

[5] RTC IO pins are those powered by VDD3P3_RTC_IO and so on, as shown in Figure 2-2 *ESP32-S2 Power Scheme*. See also Table 2-1 *Pin Overview* > Column *Pin Providing Power*.

2.5.2 Power Scheme

The power scheme is shown in Figure 2-2 *ESP32-S2 Power Scheme*.

The components on the chip are powered via voltage regulators.

Table 2-12. Voltage Regulators

Voltage Regulator	Output	Power Supply
Digital	1.1 V	Digital power domain
Low-power	1.1 V	RTC power domain
Flash	1.8 V	Can be configured to power in-package flash/PSRAM or off-package memory

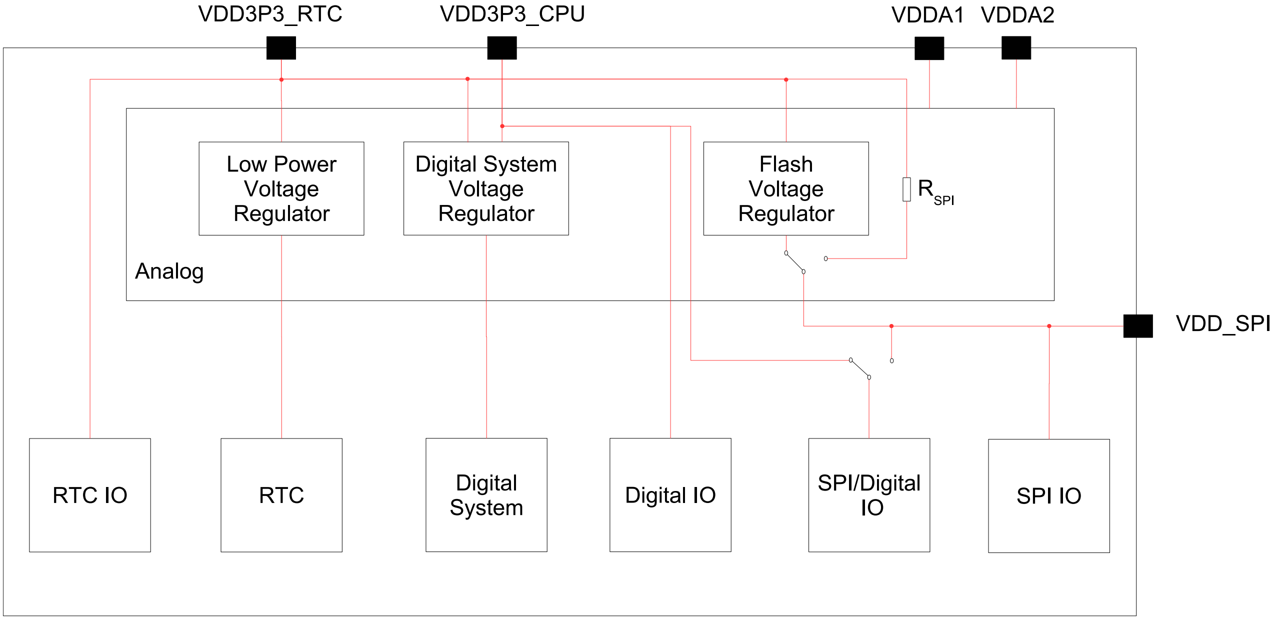


Figure 2-2. ESP32-S2 Power Scheme

2.5.3 Chip Power-up and Reset

Once the power is supplied to the chip, its power rails need a short time to stabilize. After that, CHIP_PU – the pin used for power-up and reset – is pulled high to activate the chip. For information on CHIP_PU as well as power-up and reset timing, see Figure 2-3 and Table 2-13.

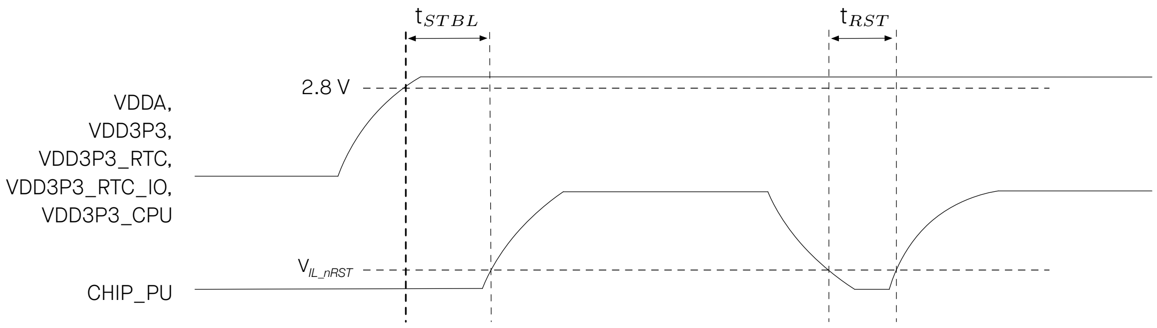


Figure 2-3. Visualization of Timing Parameters for Power-up and Reset

Table 2-13. Description of Timing Parameters for Power-up and Reset

Parameter	Description	Min (μs)
t_{STBL}	Time reserved for the power rails of VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_RTC_IO, and VDD3P3_CPU to stabilize before the CHIP_PU pin is pulled high to activate the chip	50
t_{RST}	Time reserved for CHIP_PU to stay below V_{IL_nRST} to reset the chip (see Table 5-4)	50

2.6 Pin Mapping Between Chip and Flash/PSRAM

Table 2-14 lists the pin mapping between the chip and flash/PSRAM for all SPI modes.

For chip variants with in-package flash/PSRAM (see Table 1-1 *ESP32-S2 Series Comparison*), the pins allocated for communication with in-package flash/PSRAM can be identified depending on the SPI mode used.

For off-package flash/PSRAM, these are the recommended pin mappings.

For more information on SPI controllers, see also Section 4.2.1.2 *SPI Controller*.

Notice:

Do not use the pins connected to in-package flash/PSRAM for any other purposes.

Table 2-14. Pin Mapping Between Chip and Flash or PSRAM

		Single SPI		Dual SPI		Quad SPI/QPI		Octal SPI/OPI	
Pin No.	Pin Name	Flash	PSRAM	Flash	PSRAM	Flash	PSRAM	Flash	PSRAM
29	SPICS1		CE#		CE#		CE#		CE#
31	SPIHD	HOLD#	SIO3	HOLD#	SIO3	HOLD#	SIO3	DQ3	SIO3
32	SPIWP	WP#	SIO2	WP#	SIO2	WP#	SIO2	DQ2	DQ3
33	SPICS0	CS#		CS#		CS#		CS#	
35	SPIQ	DO	SO/SIO1	DO	SO/SIO1	DO	SO/SIO1	DQ1	DQ2
36	SPID	DI	SI/SIO0	DI	SI/SIO0	DI	SI/SIO0	DQ0	DQ1
37	GPIO33							DQ4	DQ4
38	GPIO34							DQ5	DQ5
39	GPIO35							DQ6	DQ6
40	GPIO36							DQ7	DQ7
41	GPIO37							DQS/DM	DQS/DM

[1] CS0 is for in-package flash

[2] CS1 is for in-package PSRAM

3 Boot Configurations

The chip allows for configuring the following boot parameters through strapping pins and eFuse parameters at power-up or a hardware reset, without microcontroller interaction.

- **Chip boot mode**
  - Strapping pins: GPIO0 and GPIO46
- **VDD_SPI voltage**
  - Strapping pin: GPIO45
  - eFuse parameters: EFUSE_VDD_SPI_FORCE and EFUSE_VDD_SPI_TIEH
- **ROM message printing**
  - Strapping pin: GPIO46
  - eFuse parameters: EFUSE_UART_PRINT_CONTROL and EFUSE_UART_PRINT_CHANNEL

The default values of all the above eFuse parameters are 0, which means that they are not burnt. Given that eFuse is one-time programmable, once programmed to 1, it can never be reverted to 0. For how to program eFuse parameters, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *eFuse Controller*.

The default values of the strapping pins, namely the logic levels, are determined by pins' internal weak pull-up/pull-down resistors at reset if the pins are not connected to any circuit, or connected to an external high-impedance circuit.

Table 3-1. Default Configuration of Strapping Pins

Strapping Pin	Default Configuration	Bit Value
GPIO0	Weak pull-up	1
GPIO45	Weak pull-down	0
GPIO46	Weak pull-down	0

To change the bit values, the strapping pins should be connected to external pull-down/pull-up resistances. If the ESP32-S2 is used as a device by a host MCU, the strapping pin voltage levels can also be controlled by the host MCU.

All strapping pins have latches. At system reset, the latches sample the bit values of their respective strapping pins and store them until the chip is powered down or shut down. The states of latches cannot be changed in any other way. It makes the strapping pin values available during the entire chip operation, and the pins are freed up to be used as regular IO pins after reset.

The timing of signals connected to the strapping pins should adhere to the *setup time* and *hold time* specifications in Table 3-2 and Figure 3-1.

Table 3-2. Description of Timing Parameters for the Strapping Pins

Parameter	Description	Min (ms)
t_{SU}	*Setup time* is the time reserved for the power rails to stabilize before the CHIP_PU pin is pulled high to activate the chip.	0
t_H	*Hold time* is the time reserved for the chip to read the strapping pin values after CHIP_PU is already high and before these pins start operating as regular IO pins.	3

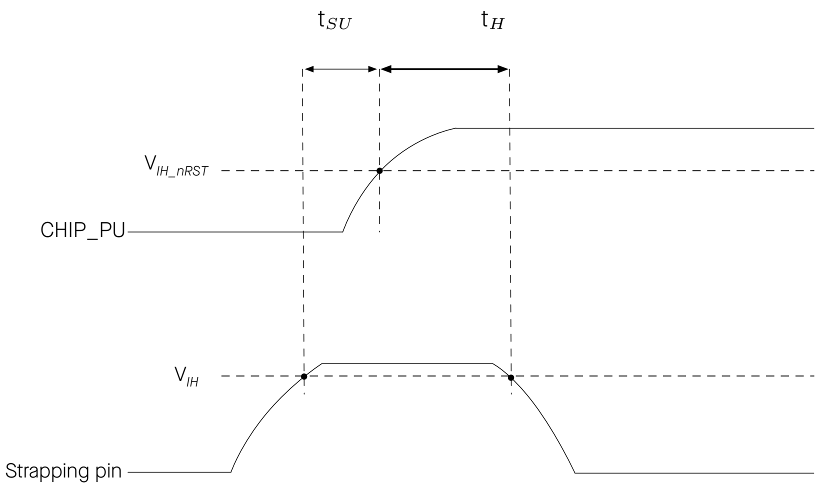


Figure 3-1. Visualization of Timing Parameters for the Strapping Pins

3.1 Chip Boot Mode Control

GPIO0 and GPIO46 control the boot mode after the reset is released. See Table 3-3 *Chip Boot Mode Control*.

Table 3-3. Chip Boot Mode Control

Boot Mode	GPIO0	GPIO46
SPI boot mode	**1**	Any value
Joint download boot mode [2]	0	0

[1] **Bold** marks the default value and configuration.

[2] Joint Download Boot mode supports the following download methods:

- USB-OTG Download Boot
- UART Download Boot
- SPI Download Boot

3.2 VDD_SPI Voltage Control

The required VDD_SPI voltage for the chips of the ESP32-S2 Series can be found in Table 1-1 *ESP32-S2 Series Comparison*.

Depending on the value of EFUSE_VDD_SPI_FORCE, the voltage can be controlled in two ways.

Table 3-4. VDD_SPI Voltage Control

VDD_SPI power source [2]	Voltage	EFUSE_VDD_SPI_FORCE	GPIO45	EFUSE_VDD_SPI_TIEH
VDD3P3_RTC via R_{SPI}	**3.3 V**	**0**	**0**	Ignored
		1	Ignored	1
Flash Voltage Regulator	1.8 V	0	1	Ignored
		1	Ignored	0

[1] **Bold** marks the default value and configuration.

[2] See Section *2.5.2 Power Scheme*.

3.3 ROM Messages Printing Control

During the boot process, the messages by the ROM code can be printed to:

- **(Default) UART0**
- UART1

EFUSE_UART_PRINT_CONTROL and GPIO46 control ROM messages printing to **UART** as shown in Table 3-5 *UART ROM Message Printing Control*.

EFUSE_UART_PRINT_CHANNEL controls if the ROM messages will be printed to UART0 or UART1.

- 0: UART0
- 1: UART1

Table 3-5. UART ROM Message Printing Control

UART ROM Code Printing	EFUSE_UART_PRINT_CONTROL	GPIO46
Enabled	**0**	Ignored
	1	0
	2	1
Disabled	1	1
	2	0
	3	Ignored

[1] **Bold** marks the default value and configuration.

4 Functional Description

4.1 System

This section describes the core of the chip's operation, covering its microprocessor, memory organization, system components, and security features.

4.1.1 Microprocessor and Master

This subsection describes the core processing units within the chip and their capabilities.

4.1.1.1 CPU

ESP32-S2 contains one low-power Xtensa® 32-bit LX7 microprocessor with the following features:

- 7-stage pipeline that supports the clock frequency of up to 240 MHz
- 16/24-bit Instruction Set providing high code-density
- support for 32-bit multiplier and 32-bit divider
- unbuffered GPIO instructions
- support for 32 interrupts at six levels
- support for windowed ABI with 64 physical general registers
- support for trace function with TRAX compressor, up to 16 KB trace memory
- JTAG for debugging

For information about the Xtensa® Instruction Set Architecture, please refer to
Xtensa® Instruction Set Architecture (ISA) Summary

4.1.1.2 ULP Coprocessor

The ULP co-processor is designed as a simplified, low-power replacement of CPU in sleep modes. It can be also used to supplement the functions of the CPU in normal working mode. The ULP co-processor and RTC memory remain powered on during the Deep-sleep mode. Hence, the developer can store a program for the ULP co-processor in the RTC slow memory to access RTC GPIO, RTC peripheral devices, RTC timers and internal sensors during the Deep-sleep mode.

ESP32-S2 has two ULP co-processors, with one based on RISC-V instruction set architecture (ULP-RISC-V) and the other on finite state machine (ULP-FSM). The clock of the co-processor is the internal 8 MHz oscillator.

ULP-RISC-V has the following features:

- support for RV32IMC instruction set
- thirty-two 32-bit general-purpose registers
- 32-bit multiplier and divider
- support for interrupts

- boot by the CPU, its dedicated timer, or RTC GPIO

ULP-FSM has the following features:

- support for common instructions including arithmetic, jump, and program control instructions
- support for on-board sensor measurement instructions
- boot by the CPU, its dedicated timer, or RTC GPIO

Note that these two co-processors cannot work simultaneously.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *ULP Coprocessor (ULP)*.

4.1.1.3 DMA Controller

ESP32-S2 includes a DMA controller that allows peripheral-to-memory and memory-to-memory data transfer at a high speed. It has the following features:

- AHB bus architecture
- Half-duplex and full-duplex mode
- Programmable length of data to be transferred in bytes
- INCR burst transfer when accessing internal RAM
- Access to an address space of 320 KB at most in internal RAM
- Access to an address space of 10.5 MB at most in external RAM
- High-speed data transfer using DMA

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *DMA Controller (DMA)*.

4.1.2 Memory Organization

This subsection describes the memory arrangement to explain how data is stored, accessed, and managed for efficient operation.

Figure 4-1 illustrates the address mapping structure of ESP32-S2.

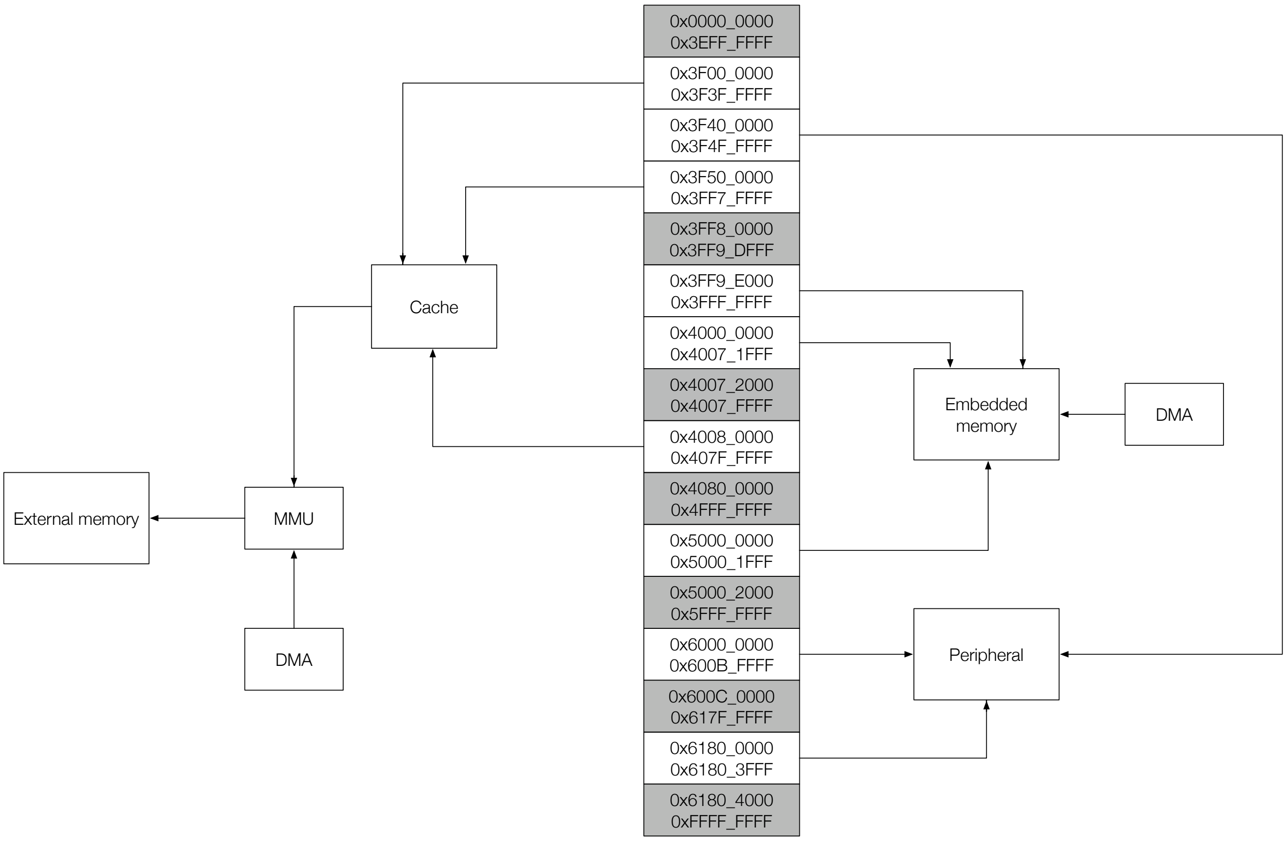


Figure 4-1. Address Mapping Structure

> **Note:**
> The memory space with gray background is not available for use.

4.1.2.1 Internal Memory

ESP32-S2's internal memory includes:

- **128 KB of ROM**: for booting and core functions
- **320 KB of on-chip SRAM**: for data and instructions, running at a configurable frequency of up to 240 MHz.
- **RTC FAST Memory**: 8 KB of SRAM in RTC. It can be accessed by the main CPU. It can retain data in Deep-sleep mode.
- **RTC SLOW Memory**: 8 KB of SRAM in RTC. It can be accessed by the main CPU or the co-processor. It can retain data in Deep-sleep mode.
- **4 Kbit of eFuse**: 1792 bits are reserved for user data, such as encryption key and device ID.
- **In-package flash and PSRAM**: see details in Chapter 1 *ESP32-S2 Series Comparison*

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *System and Memory*.

4.1.2.2 External Memory

ESP32-S2 supports multiple external QSPI/OSPI flash and RAM chips. It also supports hardware encryption/decryption based on XTS-AES to protect developers' programs and data in flash and RAM.

The external flash and RAM can be mapped into the CPU instruction memory space and read-only data memory space. The RAM can also be mapped into the CPU data memory space. Up to 1 GB of external flash and RAM can be supported.

Through high-speed caches, ESP32-S2 can support the following mappings at the same time.

- Up to 7.5 MB of instruction memory space can be mapped at a time into flash and RAM. If more than 3.5 MB are mapped, cache performance may be slightly reduced due to the CPU's pipeline characteristics.
- Up to 4 MB of read-only data memory space can be mapped into flash or RAM as individual 64 KB blocks. 8-bit, 16-bit and 32-bit reads are supported.
- Up to 10.5 MB of read-write data memory space can be mapped into RAM as individual 64 KB blocks. 8-bit, 16-bit and 32-bit reads and writes are supported. Blocks from this 10.5 MB space can also be mapped into flash, for read operations only.

Note:
After ESP32-S2 is initialized, firmware can customize the mapping of external RAM or flash into the CPU address space.

For more information, please refer to Chapter *ESP32-S2 Technical Reference Manual* > Chapter *System and Memory*.

4.1.2.3 Cache

ESP32-S2 has independent instruction Cache and data Cache that have the following features:

- configurable size of 8 KB or 16 KB
- 4-way set associative
- block size of 16 bytes or 32 bytes
- pre-load function
- lock function
- support for critical word first and early restart

4.1.3 System Components

This subsection describes the essential components that contribute to the overall functionality and control of the system.

4.1.3.1 Clock

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *Reset and Clock*.

CPU Clock

The CPU clock has four possible sources:

- external 40 MHz crystal clock
- internal 8 MHz oscillator
- PLL clock
- audio PLL clock

The application can select the clock source from the external crystal clock source, the PLL clock, the audio PLL clock, or the internal 8 MHz oscillator. The selected clock source drives the CPU clock directly, or after division, depending on the application.

Note:
ESP32-S2 is unable to operate without an external crystal clock.

RTC Clock

The RTC slow clock has three possible sources:

- external low-speed (32 kHz) crystal clock
- internal RC oscillator (typically about 90 kHz, and adjustable)
- internal 31.25 kHz clock (derived from the internal 8 MHz oscillator divided by 256)

The RTC fast clock has two possible sources:

- external divide-by-4 crystal clock
- internal divide-by-N oscillator of 8 MHz

The RTC slow clock is used for RTC counter, RTC watchdog and low-power controller; while the RTC fast clock for RTC peripherals and sensing controllers.

Audio PLL Clock

The audio clock is generated by the low-noise fractional-N PLL.

4.1.3.2 System Timer

64-bit Timers

There are four general-purpose timers embedded in ESP32-S2. They are all 64-bit generic timers which are based on 16-bit prescalers and 64-bit auto-reload-capable up/down-timers.

The timers' features are summarized as follows:

- a 16-bit clock prescaler, from 1 to 65536
- a 64-bit time-base counter programmable to be incrementing or decrementing
- able to read real-time value of the time-base counter

- halting and resuming the time-base counter
- programmable alarm generation
- timer value reload (Auto-reload at alarm or software-controlled instant reload)
- level and edge interrupt generation

For more information, please refer to ESP32-S2 Technical Reference Manual > Chapter *Timer Group (TIMG)*.

Watchdog Timers

The ESP32-S2 contains three watchdog timers: one in each of the two timer groups (called Main System Watchdog Timers, or MWDT) and one in the RTC Module (called the RTC Watchdog Timer, or RWDT). Each watchdog timer allows for four separately configurable stages and each stage can be programmed to take one of three (or four for RWDT) actions upon expiry, unless the watchdog is fed or disabled. The actions upon expiry are: interrupt, CPU reset, core reset and system reset. Only RWDT can trigger a system reset that will reset the entire digital circuits, which is the main system including the RTC itself. A timeout value can be set for each stage individually.

During the flash boot process, RWDT and the first MWDT are enabled automatically in order to detect and recover from booting errors.

Watchdog timers have the following features:

- four stages, each with a programmable timeout value. Each stage can be configured and enabled/disabled separately
- one of three/four (for MWDTs/ RWDT) possible actions (interrupt, CPU reset, core reset and system reset) available upon expiry of each stage
- 32-bit expiry counter
- write protection, to prevent RWDT and MWDT configuration from being altered inadvertently
- flash boot protection
  If the boot process from an SPI flash does not complete within a predetermined period of time, the watchdog will reboot the entire main system.

For more information, please refer to ESP32-S2 Technical Reference Manual > Chapter *Watchdog Timers (WDT)*.

4.1.3.3 Power Management Unit

With the use of advanced power-management technologies, ESP32-S2 can switch between different power modes.

- Active mode: CPU and chip radio are powered on. The chip can receive, transmit, or listen.
- Modem-sleep mode: The CPU is operational and the clock speed can be reduced. The Wi-Fi baseband and radio are disabled, but Wi-Fi connection can remain active.
- Light-sleep mode: The CPU is paused. The RTC peripherals, as well as the ULP co-processor are running. Any wake-up events (MAC, RTC timer, or external interrupts) will wake up the chip. Wi-Fi connection can remain active.

- Deep-sleep mode: Only the RTC memory and RTC peripherals are powered on. Wi-Fi connection data are stored in the RTC memory. The ULP co-processor is functional.
- Hibernation mode: The internal 8-MHz oscillator and ULP co-processor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and certain RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

For power consumption in different power modes, please refer to Chapter 5.6 *Current Consumption*.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *Low-Power Management (RTC_CNTL)*.

4.1.4 Cryptography and Security Component

This subsection describes the security features incorporated into the chip, which safeguard data and operations.

4.1.4.1 Cryptographic Hardware Accelerators

ESP32-S2 is equipped with hardware accelerators of general algorithms, such as AES (FIPS PUB 197), ECB/CBC/OFB/CFB/CTR (NIST SP 800-38A), GCM (NIST SP 800-38D), SHA (FIPS PUB 180-4), and RSA, which support independent arithmetic, such as large-number multiplication and large-number modular multiplication. The maximum operation length for RSA and large-number modular multiplication is 4096 bits. The maximum operand length for large-number multiplication is 2048 bits.

4.1.4.2 Physical Security Features

- Transparent external flash and RAM encryption (AES-XTS) with software inaccessible key prevents unauthorized readout of user application code or data.
- Secure Boot feature uses a hardware root of trust to ensure only signed firmware (with RSA-PSS signature) can be booted.
- HMAC module can use a software inaccessible MAC key to generate SHA-HMAC signatures for identity verification, as well as other uses.
- Digital Signature module can use a software inaccessible secure key to generate MAC signatures for identity verification.

4.2 Peripherals

This section describes the chip's peripheral capabilities, covering connectivity interfaces and on-chip sensors that extend its functionality.

4.2.1 Connectivity Interface

This subsection describes the connectivity interfaces on the chip that enable communication and interaction with external devices and networks.

4.2.1.1 General Purpose Input / Output Interface (GPIO)

ESP32-S2 has 43 GPIO pins which can be assigned various functions by programming the appropriate registers. Some GPIOs can be used both for digital signals but also for analog functions, such as ADC, DAC and touch sensing.

All GPIOs can be configured as internal pull-up or pull-down, or set to high impedance, except for GPIO46, which is fixed to pull-down. When configured as an input, the input value can be read by software through the register. The input can also be set to edge-trigger or level-trigger to generate CPU interrupts. Except for GPIO46 (input only), all digital IO pins are bi-directional, non-inverting and tristate, including input and output buffers with tristate control. These pins can be multiplexed with other functions, such as the UART, SPI, etc. For low-power operations, the GPIOs can be set to hold their states.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *IO MUX and GPIO Matrix (GPIO, IO_MUX)*.

4.2.1.2 SPI Controller

ESP32-S2 features four SPI interfaces (SPI0, SPI1, SPI2 and SPI3). SPI0 and SPI1 can only be configured to operate in SPI memory mode; SPI2 can be configured to operate in SPI memory and general-purpose SPI modes; SPI3 can only be configured to operate in general-purpose SPI mode.

- **SPI Memory mode** In SPI memory mode, SPI0, SPI1 and SPI2 interface with external SPI memory. Data are transferred in unit of byte. Up to 8-line STR/DDR reads and writes are supported. The clock frequency is configurable to a maximum of 80 MHz in STR mode and a maximum of 40 MHz in DDR mode.
- **SPI2 General-purpose SPI (GP-SPI) mode**

  When SPI2 acts as a general-purpose SPI, it can operate in master and slave modes. The master mode supports 2-line full-duplex communication and 1-/2-/4-/8-line half-duplex communication. The slave mode supports 2-line full-duplex communication and 1-/2-/4-line half-duplex communication. The host's clock frequency is configurable. Data are transferred in unit of byte. The clock polarity (CPOL) and phase (CPHA) are also configurable. The SPI2 interface supports DMA.

  - In 2-line full-duplex communication mode, the host's clock frequency is configurable to 80 MHz at most, and the slave's clock frequency to 40 MHz at most. Four modes of SPI transfer format are supported.
  - In 1-/2-/4-/8-line half-duplex communication mode, the host's clock frequency is configurable to 80 MHz at most and the four modes of SPI transfer format are supported.

  - In 1-/2-/4-line half-duplex communication mode, the slave's clock frequency is configurable to 40 MHz at most, and the four modes of SPI transfer format are also supported.

- **SPI3 General-purpose SPI (GP-SPI) mode**

  As a general-purpose SPI interface, SPI3 can operate in master and slave modes, in 2-line full-duplex and 1-line half-duplex communication modes. The host's clock frequency is configurable. Data are transferred in unit of byte. The clock polarity (CPOL) and phase (CPHA) are also configurable. The SPI3 interface supports DMA.

  - In 2-line full-duplex communication mode, the host's clock frequency is configurable to a maximum of 80 MHz, and the slave's clock frequency to 40 MHz at most. Four modes of SPI transfer format are supported.
  - In 1-line half-duplex communication mode, the host's clock frequency is configurable to a maximum of 80 MHz, and the slave's clock frequency to 40 MHz at most. The four modes of SPI transfer format are supported.

The mapping between SPI bus signals and GPIO pins is shown in Table 4-1 *Mapping of SPI Signal Buses and Chip Pins*:

Table 4-1. Mapping of SPI Signal Buses and Chip Pins

Standard SPI		Extended SPI			
Full-Duplex	Half-Duplex	Chip Pad Signals			
SPI Signal Bus	SPI Signal Bus	Pin Functions	SPI Signal Bus	FSPI Signal Bus	SPI3 Signal Bus
MOSI	MOSI	D	SPID	FSPID	SPI3_D
MISO	(MISO)	Q	SPIQ	FSPIQ	SPI3_Q
CS	CS	CS	SPICS0 ~ 1	FSPICS0 ~ 5	SPI3_CS0 ~ 2
CLK	CLK	CLK	SPICLK	FSPICLK	SPI3_CLK
-	-	WP	SPIWP	FSPIWP	-
-	-	HD	SPIHD	FSPIHD	SPI3_HD
-	-	CD	-	FSPICD	SPI3_CD
-	-	DQS	SPIDQS	FSPIDQS	SPI3_DQS
-	-	IO4 ~ 7	SPIIO4 ~ 7	FSPIIO4 ~ 7	-
-	-	VSYNC	-	FSPI_VSYNC	-
-	-	HSYNC	-	FSPI_HSYNC	-
-	-	DE	-	FSPI_DE	-

In most cases, the data port connection between ESP32-S2 and external flash is as follows:

SPI 8-line mode:

- SPID (SPID) = IO0
- SPIQ (SPIQ) = IO1
- SPIWP (SPIWP) = IO2
- SPIHD (SPIHD) = IO3
- GPIO33 = IO4
- GPIO34 = IO5

- GPIO35 = IO6
- GPIO36 = IO7
- GPIO37 = DQS

SPI 4-line mode:

- SPID (SPID) = IO0
- SPIQ (SPIQ) = IO1
- SPIWP (SPIWP) = IO2
- SPIHD (SPIHD) = IO3

SPI 2-line mode:

- SPID (SPID) = IO0
- SPIQ (SPIQ) = IO1

SPI 1-line mode:

- SPID (SPID) = DI
- SPIQ (SPIQ) = DO
- SPIWP (SPIWP) = WP#
- SPIHD (SPIHD) = HOLD#

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *SPI Controller (SPI)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.3 LCD Controller

SPI2 supports parallel 8-bit RGB, I8080 and Moto6800 interfaces. I2S supports 8/16/24-bit parallel interface (8080).

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *SPI Controller (SPI)* and *ESP32-S2 Technical Reference Manual* > Chapter *I2S Controller (I2S)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.4 UART Controller

ESP32-S2 has two UART interfaces, i.e., UART0, UART1, which provide asynchronous communication (RS232 and RS485) and IrDA support, communicating at a speed of up to 5 Mbps. UART provides hardware management of the CTS and RTS signals and software flow control (XON and XOFF). All of the interfaces can be accessed by the DMA controller or directly by the CPU.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *UART Controller (UART)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.5 I2C Controller

ESP32-S2 has two I2C bus interfaces which can serve as I2C master or slave, depending on the user's configuration. The I2C interfaces support:

- standard mode (100 Kbit/s)
- fast mode (400 Kbit/s)
- up to 5 MHz (constrained by SDA pull-up strength)
- 7-bit/10-bit addressing mode
- dual addressing mode

Users can program command registers to control I2C interfaces, so that they have more flexibility.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *I2C Controller (I2C)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.6 I2S Controller

ESP32-S2 includes a standard I2S interface. It can operate in master or slave mode, in full-duplex and half-duplex communication modes, and can be configured to operate with an 8-/16-/24-/32-bit resolution as an input or output channel. BCK clock frequency, from 10 kHz up to 40 MHz, is supported.

The I2S interface has a dedicated DMA controller. PCM interface is supported.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *I2S Controller (I2S)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.7 Camera Interface

ESP32-S2 supports one 8 or 16-bit DVP image sensor, with clock frequency of up to 40 MHz. The camera interface is implemented by using the hardware resources of I2S.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *I2S Controller (I2S)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.8 Remote Control Peripheral

The infrared remote controller supports four channels of infrared remote transmission and reception. By programming the pulse waveform, it supports various infrared and other single wire protocols. Four channels share a 256 × 32-bit block of memory to store the transmitting or receiving waveform.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *Remote Control Peripheral (RMT)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.9 Pulse Count Controller

The pulse counter captures pulse and counts pulse edges through multiple modes. It has four channels, each of which captures four signals at a time. The four input signals include two pulse signals and two control signals.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *Pulse Count Controller (PCNT)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.10 LED PWM Controller

The LED PWM controller can generate eight independent channels. The LED PWM controller:

- can generate digital waveforms with configurable periods and duties. The accuracy of duty can be up to 18 bits within a 1 ms period.
- has multiple clock sources, including APB clock and external crystal clock.
- can operate when the CPU is in Light-sleep mode.
- supports gradual increase or decrease of duty cycle, which is useful for the LED RGB color-gradient generator.

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *LED PWM Controller (LEDC)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.11 USB OTG

ESP32-S2 features a full-speed USB OTG interface which is compliant with the USB 2.0 specification (Note that it does not support the faster 480 Mbit/s high-speed transfer mode). It has the following features:

- software-configurable endpoint settings and suspend/resume
- support for dynamic FIFO sizing
- support for session request protocol (SRP) and host negotiation protocol (HNP)
- a full-speed USB PHY integrated in the chip

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *USB On-The-Go (USB)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.1.12 Two-wire Automotive Interface

ESP32-S2 has a TWAI® controller with the following features:

- compatible with ISO 11898-1 protocol (CAN Specification 2.0)
- standard frame format (11-bit ID) and extended frame format (29-bit ID)
- bit rates from 1 Kbit/s to 1 Mbit/s
- multiple modes of operation: Normal, Listen Only, and Self-Test
- 64-byte receive FIFO
- special transmissions: single-shot transmissions and self reception
- acceptance filter (single and dual filter modes)
- error detection and handling: error counters, configurable error interrupt threshold, error code capture, arbitration lost capture

For more information, please refer to *ESP32-S2 Technical Reference Manual* > Chapter *Two-wire Automotive Interface (TWAI)*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.2 Analog Signal Processing

This subsection describes components on the chip that sense and process real-world data.

4.2.2.1 ADC

ESP32-S2 integrates two 12-bit SAR ADCs and supports measurements on 20 channels (analog-enabled pins). The ULP-coprocessor in ESP32-S2 is also designed to measure voltage. The ULP can operate while the main

CPU is in Deep-sleep mode, which lowers the total power consumption. By using threshold settings, and / or via other triggers or events, we can interrupt the CPU from the sleep state.

The ADCs can be configured to measure voltage on up to 20 pins.

For ADC characteristics, please refer to Table 5.5 *ADC Characteristics*.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.2.2 DAC

ESP32-S2 has two 8-bit DAC channels that convert two digital signals into two analog voltage signal outputs. The two DAC channels support independent conversions. The design structure is composed of integrated resistor strings and a buffer. This dual DAC supports VDD3P3_RTC_IO power supply as input voltage reference.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.2.2.3 Temperature Sensor

The temperature sensor generates a voltage that varies with temperature. The voltage is internally converted via an ADC into a digital value.

The temperature sensor has a range of –20 °C to 110 °C. It is designed primarily to sense the temperature changes inside the chip. The temperature value depends on factors like microcontroller clock frequency or I/O load. Generally, the chip's internal temperature is higher than the ambient operating temperature.

4.2.2.4 Touch Sensor

ESP32-S2 has 14 capacitive-sensing GPIOs, which detect variations induced by touching or approaching the GPIOs with a finger or other objects. The low-noise nature of the design and the high sensitivity of the circuit allow relatively small pads to be used. Arrays of pads can also be used, so that a larger area or more points can be detected. The touch sensing performance can be further enhanced by the waterproof design and digital filtering feature.

> **Note:**
> ESP32-S2 Touch Sensor has not passed the Conducted Susceptibility (CS) test for now, and thus has limited application scenarios.

Pin Assignment

For details, see Section 2.3.6 *Peripheral Pin Assignment*.

4.3 Wireless Communication

This section describes the chip's wireless communication capabilities, spanning radio technology and Wi-Fi.

4.3.1 Radio

The ESP32-S2 radio consists of the following blocks:

- 2.4 GHz receiver
- 2.4 GHz transmitter
- Bias and regulators
- Balun and transmit-receive switch
- Clock generator

4.3.1.1 2.4 GHz Receiver

The 2.4 GHz receiver demodulates the 2.4 GHz RF signal to quadrature baseband signals and converts them to the digital domain with two high-resolution, high-speed ADCs. To adapt to varying signal channel conditions, RF filters, Automatic Gain Control (AGC), DC offset cancelation circuits and baseband filters are integrated with ESP32-S2.

4.3.1.2 2.4 GHz Transmitter

The 2.4 GHz transmitter modulates the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high-powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. The use of digital calibration further improves the linearity of the power amplifier.

Additional calibrations are integrated to cancel any radio imperfections, such as:

- carrier leakage
- I/Q amplitude/phase matching
- baseband nonlinearities
- RF nonlinearities
- antenna matching

These built-in calibration routines reduce the cost, time, and specialized equipment required for product testing, and certification.

4.3.1.3 Clock Generator

The clock generator produces quadrature clock signals of 2.4 GHz for both the receiver and the transmitter. All components of the clock generator are integrated into the chip, including all inductors, varactors, filters, regulators and dividers.

The clock generator has built-in calibration and self-test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms which ensure the best performance of the receiver and the transmitter.

4.3.2 Wi-Fi

The ESP32-S2 Wi-Fi radio and baseband support the following features:

- 802.11b/g/n
- 802.11n MCS0-7 that supports 20 MHz and 40 MHz bandwidth
- 802.11n MCS32
- 802.11n 0.4 μs guard-interval
- single stream, data rate up to 150 Mbps
- STBC RX (Single spatial stream)
- adjustable transmitting power
- antenna diversity
  ESP32-S2 supports antenna diversity with an external RF switch. One or more GPIOs control the RF switch and select the best antenna to minimize the effects of channel imperfections.

4.3.2.1 Wi-Fi MAC

ESP32-S2 implements the full 802.11 b/g/n Wi-Fi MAC protocol. It supports the Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF). Power management is handled automatically with minimal host interaction to minimize the active-duty period.

The ESP32-S2 Wi-Fi MAC applies low-level protocol functions automatically. They are as follows:

- 4 × virtual Wi-Fi interfaces
- simultaneous Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- RTS protection, CTS protection, Immediate Block ACK
- fragmentation and defragmentation
- TX/RX A-MPDU, RX A-MSDU
- TXOP
- WMM
- CCMP, TKIP, WAPI, WEP, BIP
- automatic beacon monitoring (hardware TSF)
- 802.11mc FTM

4.3.3 Networking Features

Users are provided with libraries for TCP/IP networking, ESP-MESH networking, and other networking protocols over Wi-Fi. TLS 1.0, 1.1 and 1.2 support is also provided.

5 Electrical Characteristics

5.1 Absolute Maximum Ratings

Stresses above those listed in Table 5-1 *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only and normal operation of the device at these or any other conditions beyond those indicated in Section 5.2 *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 5-1. Absolute Maximum Ratings

Parameter	Description	Min	Max	Unit
Input power pins [1]	Allowed input voltage	–0.3	3.6	V
I_{output} [2]	Cumulative IO output current	—	1800	mA
T_{STORE}	Storage temperature	–40	150	°C

[1] For more information on input power pins, see 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.

5.2 Recommended Operating Conditions

For recommended ambient temperature, see Section 1 *ESP32-S2 Series Comparison*.

Table 5-2. Recommended Operating Conditions

Parameter [1]	Description	Min	Typ	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC	Recommended input voltage	2.8	3.3	3.6	V
VDD3P3_RTC_IO [2]	Recommended input voltage	3.0	3.3	3.6	V
VDD_SPI (as input)	—	1.8	3.3	3.6	V
VDD3P3_CPU [3]	Recommended input voltage	2.8	3.3	3.6	V
I_{VDD}	Cumulative input current	0.5	—	—	A
T_J	Junction temperature	–40	—	125	°C

[1] See in conjunction with Section 2.5 *Power Supply*.

[2] If VDD3P3_RTC_IO 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] If writing to eFuses, the voltage on VDD3P3_CPU should not exceed 3.3 V as the circuits responsible for burning eFuses are sensitive to higher voltages.

5.3 VDD_SPI Output Characteristics

Table 5-3. VDD_SPI Internal and Output Characteristics

Parameter	Description [1]	Typ	Unit
R_{SPI}	VDD_SPI powered by VDD3P3_RTC via R_{SPI} for 3.3 V flash/PSRAM [2]	5	Ω
I_{SPI}	Output current when VDD_SPI is powered by Flash Voltage Regulator for 1.8 V flash/PSRAM	40	mA

[1] See in conjunction with Section 2.5.2 *Power Scheme*.

[2] VDD3P3_RTC must be more than *VDD_flash_min* + *I_flash_max* * R_{SPI};

where

- *VDD_flash_min* – minimum operating voltage of flash/PSRAM
- *I_flash_max* – maximum operating current of flash/PSRAM

5.4 DC Characteristics (3.3 V, 25 °C)

Table 5-4. DC Characteristics (3.3 V, 25 °C)

Parameter	Description	Min	Typ	Max	Unit
C_{IN}	Pin capacitance	—	2	—	pF
V_{IH}	High-level input voltage	0.75 × VDD [1]	—	VDD [1] + 0.3	V
V_{IL}	Low-level input voltage	–0.3	—	0.25 × VDD [1]	V
I_{IH}	High-level input current	—	—	50	nA
I_{IL}	Low-level input current	—	—	50	nA
V_{OH} [2]	High-level output voltage	0.8 × VDD [1]	—	—	V
V_{OL} [2]	Low-level output voltage	—	—	0.1 × VDD [1]	V
I_{OH}	High-level source current (VDD [1] = 3.3 V, V_{OH} >= 2.64 V, PAD_DRIVER = 3)	—	40	—	mA
I_{OL}	Low-level sink current (VDD [1] = 3.3 V, V_{OL} = 0.495 V, PAD_DRIVER = 3)	—	28	—	mA
R_{PU}	Internal weak pull-up resistor	—	45	—	kΩ
R_{PD}	Internal weak pull-down resistor	—	45	—	kΩ
V_{IH_nRST}	Chip reset release voltage (CHIP_PU voltage is within the specified range)	0.75 × VDD [1]	—	VDD [1] + 0.3	V
V_{IL_nRST}	Chip reset voltage (CHIP_PU voltage is within the specified range)	–0.3	—	0.25 × VDD [1]	V

[1] VDD – voltage from a power pin of a respective power domain.

[2] V_{OH} and V_{OL} are measured using high-impedance load.

5.5 ADC Characteristics

The measurements in this section are taken with an external 100 nF capacitor connected to the ADC, using DC signals as input, and at an ambient temperature of 25 °C with disabled Wi-Fi.

Table 5-5. ADC Characteristics

Symbol	Parameter	Min	Max	Unit
DNL (Differential nonlinearity) [1]	ADC connected to an external 100 nF capacitor; DC signal input; Ambient temperature at 25 °C; Wi-Fi off	–7	7	LSB
INL (Integral nonlinearity)		–12	12	LSB
Sampling rate	—	—	100	kSPS [2]

[1] To get better DNL results, you can sample multiple times and apply a filter, or calculate the average value.

[2] kSPS means kilo samples-per-second.

Note:
When reading voltages greater than 2450 mV, ADC accuracy will be worse than that in the table above.

The calibrated ADC results after hardware calibration and software calibration are shown in Table 5-6. For higher accuracy, you may implement your own calibration methods.

Table 5-6. ADC Calibration Results

Parameter	Description	Min	Max	Unit
Total error	ATTEN0, effective measurement range of 0 ~ 700	–10	10	mV
	ATTEN1, effective measurement range of 0 ~ 950	–11	11	mV
	ATTEN2, effective measurement range of 0 ~ 1200	–13	13	mV
	ATTEN3, effective measurement range of 0 ~ 2300	–17	17	mV

Note:
The above ADC measurement range and accuracy are applicable to chips manufactured on and after the Date Code **212023** on shielding cases, or assembled on and after the D/C 1 and D/C 2 **2321** on bar-code labels. For chips manufactured or assembled earlier than these date codes, please ask our sales team to provide the actual range and accuracy according to batch.

For details of Date Code and D/C, please refer to *Espressif Chip Packaging Information*.

5.6 Current Consumption

5.6.1 RF Current Consumption in Active Mode

The current consumption measurements are taken with a 3.3 V supply at 25 °C of ambient temperature at the RF port. All transmitters' measurements are based on a 100% duty cycle.

Table 5-7. Wi-Fi Current Consumption Depending on RF Modes

Work Mode [1]	Description		Peak (mA)
Active (RF working)	TX	802.11b, 20 MHz, 1 Mbps, @19.5 dBm	310
		802.11g, 20 MHz, 54 Mbps, @15 dBm	220
		802.11n, 20 MHz, MCS7, @13 dBm	200
		802.11n, 40 MHz, MCS7, @13 dBm	160
	RX[1]	802.11b/g/n, 20 MHz	63
		802.11n, 40 MHz	68

[1] The consumption in RX mode is measured when peripherals are disabled and the CPU is idle.

5.6.2 Current Consumption in Other Modes

The measurements below are applicable to ESP32-S2, ESP32-S2FH2, and ESP32-S2FH4. Since ESP32-S2FN4R2 and ESP32-S2R2 come with in-package PSRAM, their current consumption might be higher.

Table 5-8. Current Consumption in Modem-sleep Mode

Mode	CPU Frequency (MHz)	Description	Typ	
			All Peripherals Clocks Disabled (mA)	All Peripherals Clocks Enabled (mA)[1]
Modem-sleep[2,3]	240	CPU is idle	20.0	28.0
		CPU is running	23.0	32.0
	160	CPU is idle	14.0	21.0
		CPU is running	16.0	24.0
	80	CPU is idle	10.5	18.4
		CPU is running	12.0	20.0

[1] In practice, the current consumption might be different depending on which peripherals are enabled.

[2] In Modem-sleep mode, Wi-Fi is clock gated.

[3] In Modem-sleep mode, the consumption might be higher when accessing flash. For a flash rated at 80 Mbit/s, in SPI 2-line mode the consumption is 10 mA.

Table 5-9. Current Consumption in Low-Power Modes

Work mode	Description		Typ (μA)
Light-sleep[1]	VDD_SPI and Wi-Fi are powered down, and all GPIOs are high-impedance		750
Deep-sleep	The ULP co-processor is powered on[2]	ULP-FSM	170
		ULP-RISC-V	190
	ULP sensor-monitored pattern[3]		22
	RTC timer + RTC memory		25
	RTC timer only		20
Power off	CHIP_PU is set to low level, the chip is powered off		1

[1] In Light-sleep mode, with all related SPI pins pulled up, the current consumption of the embedded PSRAM is 140 μA. Chip variants with in-package PSRAM include ESP32-S2FN4R2 and ESP32-S2R2.

[2] During Deep-sleep, when the ULP co-processor is powered on, peripherals such as GPIO and I2C are able to operate.

[3] The "ULP sensor-monitored pattern" refers to the mode where the ULP coprocessor or the sensor works periodically. When touch sensors work with a duty cycle of 1%, the typical current consumption is 22 μA.

5.7 Memory Specifications

The data below is sourced from the memory vendor datasheet. These values are guaranteed through design and/or characterization but are not fully tested in production. Devices are shipped with the memory erased.

Table 5-10. Flash Specifications

Parameter	Description	Min	Typ	Max	Unit
VCC	Power supply voltage (1.8 V)	1.65	1.80	2.00	V
	Power supply voltage (3.3 V)	2.7	3.3	3.6	V
F$_C$	Maximum clock frequency	80	—	—	MHz
—	Program/erase cycles	100,000	—	—	cycles
T$_{RET}$	Data retention time	20	—	—	years
T$_{PP}$	Page program time	—	0.8	5	ms
T$_{SE}$	Sector erase time (4 KB)	—	70	500	ms
T$_{BE1}$	Block erase time (32 KB)	—	0.2	2	s
T$_{BE2}$	Block erase time (64 KB)	—	0.3	3	s
T$_{CE}$	Chip erase time (16 Mb)	—	7	20	s
	Chip erase time (32 Mb)	—	20	60	s
	Chip erase time (64 Mb)	—	25	100	s
	Chip erase time (128 Mb)	—	60	200	s
	Chip erase time (256 Mb)	—	70	300	s

5.8 Reliability

Table 5-11. Reliability Qualifications

Test Item	Test Conditions	Test Standard
HTOL (High Temperature Operating Life)	125 °C, 1000 hours	JESD22-A108
ESD (Electro-Static Discharge Sensitivity)	HBM (Human Body Mode) [1] ± 2000 V	JS-001
	CDM (Charge Device Mode) [2] ± 1000 V	JS-002
Latch up	Current trigger ± 200 mA	JESD78
	Voltage trigger 1.5 × VDD$_{max}$	

Cont'd on next page

Table 5-11 – cont'd from previous page

Test Item	Test Conditions	Test Standard
Preconditioning	Bake 24 hours @125 °C Moisture soak (level 3: 192 hours @30 °C, 60% RH) IR reflow solder: 260 + 0 °C, 20 seconds, three times	J-STD-020, JESD47, JESD22-A113
TCT (Temperature Cycling Test)	–65 °C / 150 °C, 500 cycles	JESD22-A104
uHAST (Highly Accelerated Stress Test, unbiased)	130 °C, 85% RH, 96 hours	JESD22-A118
HTSL (High Temperature Storage Life)	150 °C, 1000 hours	JESD22-A103
LTSL (Low Temperature Storage Life)	–40 °C, 1000 hours	JESD22-A119

[1] JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process.

[2] JEDEC document JEP157 states that 250 V CDM allows safe manufacturing with a standard ESD control process.

6 RF Characteristics

This section contains tables with RF characteristics of the Espressif product.

The RF data is measured at the antenna port, where RF cable is connected, including the front-end loss. The front-end circuit is a 0 Ω resistor.

Devices should operate in the center frequency range allocated by regional regulatory authorities. The target center frequency range and the target transmit power are configurable by software. See ESP RF Test Tool and Test Guide for instructions.

Unless otherwise stated, the RF tests are conducted with a 3.3 V (±5%) supply at 25 °C ambient temperature.

6.1 Wi-Fi Radio

Table 6-1. Wi-Fi Frequency

Parameter	Min (MHz)	Typ (MHz)	Max (MHz)
Center frequency of operating channel	2412	—	2484

6.1.1 Wi-Fi RF Transmitter (TX) Characteristics

Table 6-2. TX Power with Spectral Mask and EVM Meeting 802.11 Standards

Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11b, 1 Mbps	—	19.5	—
802.11b, 11 Mbps	—	19.5	—
802.11g, 6 Mbps	—	18.0	—
802.11g, 54 Mbps	—	18.0	—
802.11n, HT20, MCS0	—	18.0	—
802.11n, HT20, MCS7	—	17.0	—
802.11n, HT40, MCS0	—	18.0	—
802.11n, HT40, MCS7	—	16.5	—

Table 6-3. TX EVM Test

Rate	Min (dB)	Typ (dB)	SL[1] (dB)
802.11b, 1 Mbps, @19.5 dBm	—	–25	–10
802.11b, 11 Mbps, @19.5 dBm	—	–25	–10
802.11g, 6 Mbps, @18 dBm	—	–28	–5
802.11g, 54 Mbps, @18 dBm	—	–28	–25
802.11n, HT20, MCS0, @18 dBm	—	–26	–5

Cont'd on next page

Table 6-3 – cont'd from previous page

Rate	Min (dB)	Typ (dB)	SL[1] (dB)
802.11n, HT20, MCS7, @17 dBm	—	–30	–27
802.11n, HT40, MCS0, @18 dBm	—	–28	–5
802.11n, HT40, MCS7, @16.5 dBm	—	–30	–27

[1] SL stands for standard limit value.

6.1.2 Wi-Fi RF Receiver (RX) Characteristics

Table 6-4. RX Sensitivity

Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11b, 1 Mbps	—	–97	—
802.11b, 2 Mbps	—	–95	—
802.11b, 5.5 Mbps	—	–93	—
802.11b, 11 Mbps	—	–88	—
802.11g, 6 Mbps	—	–92	—
802.11g, 9 Mbps	—	–91	—
802.11g, 12 Mbps	—	–89	—
802.11g, 18 Mbps	—	–87	—
802.11g, 24 Mbps	—	–84	—
802.11g, 36 Mbps	—	–80	—
802.11g, 48 Mbps	—	–76	—
802.11g, 54 Mbps	—	–75	—
802.11n, HT20, MCS0	—	–92	—
802.11n, HT20, MCS1	—	–88	—
802.11n, HT20, MCS2	—	–85	—
802.11n, HT20, MCS3	—	–83	—
802.11n, HT20, MCS4	—	–79	—
802.11n, HT20, MCS5	—	–75	—
802.11n, HT20, MCS6	—	–74	—
802.11n, HT20, MCS7	—	–72	—
802.11n, HT40, MCS0	—	–89	—
802.11n, HT40, MCS1	—	–86	—
802.11n, HT40, MCS2	—	–83	—
802.11n, HT40, MCS3	—	–80	—
802.11n, HT40, MCS4	—	–76	—
802.11n, HT40, MCS5	—	–72	—
802.11n, HT40, MCS6	—	–71	—
802.11n, HT40, MCS7	—	–69	—

Table 6-5. Maximum RX Level

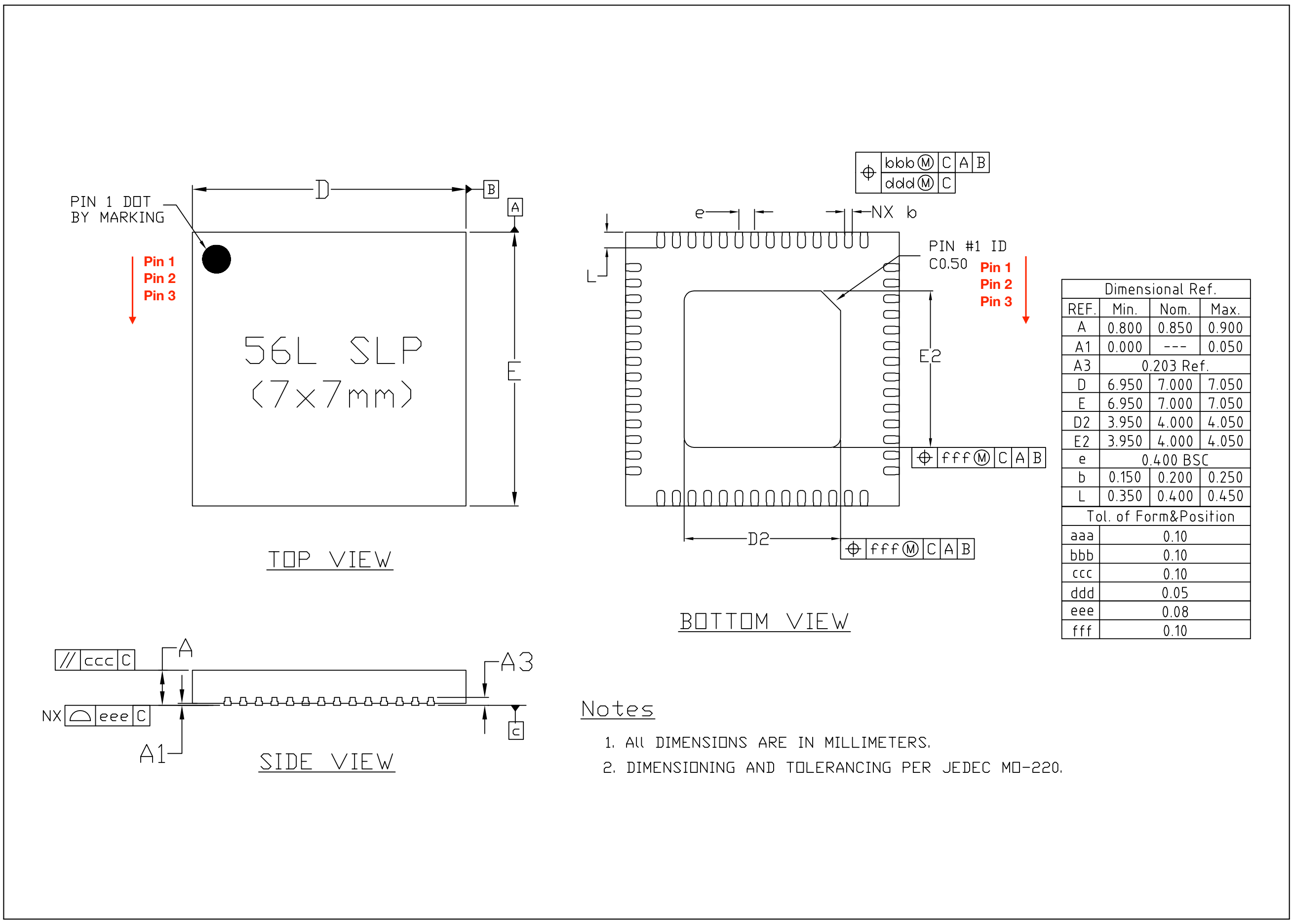
Rate	Min (dBm)	Typ (dBm)	Max (dBm)
802.11b, 1 Mbps	—	5	—
802.11b, 11 Mbps	—	5	—
802.11g, 6 Mbps	—	5	—
802.11g, 54 Mbps	—	0	—
802.11n, HT20, MCS0	—	5	—
802.11n, HT20, MCS7	—	0	—
802.11n, HT40, MCS0	—	5	—
802.11n, HT40, MCS7	—	0	—

Table 6-6. RX Adjacent Channel Rejection

Rate	Min (dB)	Typ (dB)	Max (dB)
802.11b, 1 Mbps	—	35	—
802.11b, 11 Mbps	—	35	—
802.11g, 6 Mbps	—	31	—
802.11g, 54 Mbps	—	14	—
802.11n, HT20, MCS0	—	31	—
802.11n, HT20, MCS7	—	13	—
802.11n, HT40, MCS0	—	19	—
802.11n, HT40, MCS7	—	8	—

7 Packaging

- The two packages only vary in EPAD size.
- For information about tape, reel, and chip marking, please refer to *ESP32-S2 Chip Packaging Information*.
- The pins of the chip are numbered in anti-clockwise order starting from Pin 1 in the top view. For pin numbers and pin names, see also Figure 2-1 *ESP32-S2 Pin Layout (Top View)*.
- The recommended land pattern source file (dxf) is available for download. You can view the file with Autodesk Viewer.
  - ESP32-S2
  - ESP32-S2F

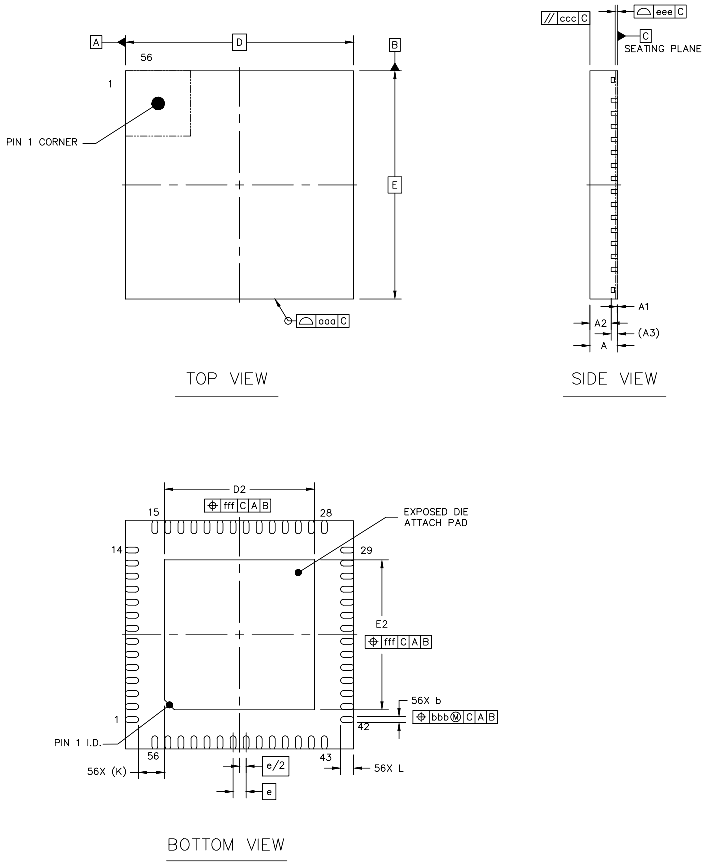


Dimensional Ref.			
REF.	Min.	Nom.	Max.
A	0.800	0.850	0.900
A1	0.000	---	0.050
A3	0.203 Ref.		
D	6.950	7.000	7.050
E	6.950	7.000	7.050
D2	3.950	4.000	4.050
E2	3.950	4.000	4.050
e	0.400 BSC		
b	0.150	0.200	0.250
L	0.350	0.400	0.450
Tol. of Form&Position			
aaa	0.10		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

Notes

1. All DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER JEDEC MO-220.

Figure 7-1. QFN56 (7×7 mm) Package, for Variants Excluding ESP32-S2FN4R2



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.02	0.05
MOLD THICKNESS		A2	---	0.65	---
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.13	0.18	0.23
BODY SIZE	X	D	7 BSC		
	Y	E	7 BSC		
LEAD PITCH		e	0.4 BSC		
EP SIZE	X	D2	4.5	4.6	4.7
	Y	E2	4.5	4.6	4.7
LEAD LENGTH		L	0.3	0.4	0.5
LEAD TIP TO EXPOSED PAD EDGE		K	0.8 REF		
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		ccc	0.1		
COPLANARITY		eee	0.08		
LEAD OFFSET		bbb	0.07		
EXPOSED PAD OFFSET		fff	0.1		

NOTES
1.REFER TO JEDEC MO-220;
2.COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH PAD;
3.BAN TO USE THE LEVEL 1 ENVIRONMENT-RELATED SUBSTANCES OF JCET PRESCRIBING;
4.FINISH: Cu/EP • Sn8~20s

Figure 7-2. QFN56 (7×7 mm) Package, for ESP32-S2FN4R2

ESP32-S2 Consolidated Pin Overview

Table 7-1. Consolidated Pin Overview

				Pin Settings		RTC IO MUX Function		Analog Function		IO MUX Function									
Pin No.	Pin Name	Pin Type	Pin Providing Power	At Reset	After Reset	F0	F3	F0	F1	F0	Type	F1	Type	F2	Type	F3	Type	F4	Type
1	VDDA	Power																	
2	LNA_IN	Analog																	
3	VDD3P3	Power																	
4	VDD3P3	Power																	
5	GPIO0	IO	VDD3P3_RTC_IO	WPU, IE	WPU, IE	RTC_GPIO0	sar_i2c_scl_0			**GPIO0**	I/O/T	GPIO0	I/O/T						
6	GPIO1	IO	VDD3P3_RTC_IO	IE	IE	RTC_GPIO1	sar_i2c_sda_0	TOUCH1	ADC1_CH0	**GPIO1**	I/O/T	GPIO1	I/O/T						
7	GPIO2	IO	VDD3P3_RTC_IO	IE	IE	RTC_GPIO2	sar_i2c_scl_1	TOUCH2	ADC1_CH1	**GPIO2**	I/O/T	GPIO2	I/O/T						
8	GPIO3	IO	VDD3P3_RTC_IO			RTC_GPIO3	sar_i2c_sda_1	TOUCH3	ADC1_CH2	**GPIO3**	I/O/T	GPIO3	I/O/T						
9	GPIO4	IO	VDD3P3_RTC_IO			RTC_GPIO4		TOUCH4	ADC1_CH3	**GPIO4**	I/O/T	GPIO4	I/O/T						
10	GPIO5	IO	VDD3P3_RTC_IO			RTC_GPIO5		TOUCH5	ADC1_CH4	**GPIO5**	I/O/T	GPIO5	I/O/T						
11	GPIO6	IO	VDD3P3_RTC_IO			RTC_GPIO6		TOUCH6	ADC1_CH5	**GPIO6**	I/O/T	GPIO6	I/O/T						
12	GPIO7	IO	VDD3P3_RTC_IO			RTC_GPIO7		TOUCH7	ADC1_CH6	**GPIO7**	I/O/T	GPIO7	I/O/T						
13	GPIO8	IO	VDD3P3_RTC_IO			RTC_GPIO8		TOUCH8	ADC1_CH7	**GPIO8**	I/O/T	GPIO8	I/O/T			SUBSPICS1	O/T		
14	GPIO9	IO	VDD3P3_RTC_IO		IE	RTC_GPIO9		TOUCH9	ADC1_CH8	**GPIO9**	I/O/T	GPIO9	I/O/T			SUBSPIHD	I1/O/T	FSPIHD	I1/O/T
15	GPIO10	IO	VDD3P3_RTC_IO		IE	RTC_GPIO10		TOUCH10	ADC1_CH9	**GPIO10**	I/O/T	GPIO10	I/O/T	FSPIIO4	I1/O/T	SUBSPICS0	O/T	FSPICS0	I1/O/T
16	GPIO11	IO	VDD3P3_RTC_IO		IE	RTC_GPIO11		TOUCH11	ADC2_CH0	**GPIO11**	I/O/T	GPIO11	I/O/T	FSPIIO5	I1/O/T	SUBSPID	I1/O/T	FSPID	I1/O/T
17	GPIO12	IO	VDD3P3_RTC_IO		IE	RTC_GPIO12		TOUCH12	ADC2_CH1	**GPIO12**	I/O/T	GPIO12	I/O/T	FSPIIO6	I1/O/T	SUBSPICLK	O/T	FSPICLK	I1/O/T
18	GPIO13	IO	VDD3P3_RTC_IO		IE	RTC_GPIO13		TOUCH13	ADC2_CH2	**GPIO13**	I/O/T	GPIO13	I/O/T	FSPIIO7	I1/O/T	SUBSPIQ	I1/O/T	FSPIQ	I1/O/T
19	GPIO14	IO	VDD3P3_RTC_IO		IE	RTC_GPIO14		TOUCH14	ADC2_CH3	**GPIO14**	I/O/T	GPIO14	I/O/T	FSPIDQS	O/T	SUBSPIWP	I1/O/T	FSPIWP	I1/O/T
20	VDD3P3_RTC	Power																	
21	XTAL_32K_P	IO	VDD3P3_RTC_IO			RTC_GPIO15		XTAL_32K_P	ADC2_CH4	**GPIO15**	I/O/T	GPIO15	I/O/T	U0RTS	O				
22	XTAL_32K_N	IO	VDD3P3_RTC_IO			RTC_GPIO16		XTAL_32K_N	ADC2_CH5	**GPIO16**	I/O/T	GPIO16	I/O/T	U0CTS	I1				
23	DAC_1	IO	VDD3P3_RTC_IO		IE	RTC_GPIO17		DAC_1	ADC2_CH6	**GPIO17**	I/O/T	GPIO17	I/O/T	U1TXD	O				
24	DAC_2	IO	VDD3P3_RTC_IO		IE	RTC_GPIO18		DAC_2	ADC2_CH7	**GPIO18**	I/O/T	GPIO18	I/O/T	U1RXD	I1	CLK_OUT3	O		
25	GPIO19	IO	VDD3P3_RTC_IO			RTC_GPIO19		USB_D-	ADC2_CH8	**GPIO19**	I/O/T	GPIO19	I/O/T	U1RTS	O	CLK_OUT2	O		
26	GPIO20	IO	VDD3P3_RTC_IO			RTC_GPIO20		USB_D+	ADC2_CH9	**GPIO20**	I/O/T	GPIO20	I/O/T	U1CTS	I1	CLK_OUT1	O		
27	VDD3P3_RTC_IO	Power																	
28	GPIO21	IO	VDD3P3_RTC_IO			RTC_GPIO21				**GPIO21**	I/O/T	GPIO21	I/O/T						
29	SPICS1	IO	VDD_SPI	WPU, IE	WPU, IE					**SPICS1**	O/T	GPIO26	I/O/T						
30	VDD_SPI	Power																	
31	SPIHD	IO	VDD_SPI	WPU, IE	WPU, IE					**SPIHD**	I1/O/T	GPIO27	I/O/T						
32	SPIWP	IO	VDD_SPI	WPU, IE	WPU, IE					**SPIWP**	I1/O/T	GPIO28	I/O/T						
33	SPICS0	IO	VDD_SPI	WPU, IE	WPU, IE					**SPICS0**	O/T	GPIO29	I/O/T						
34	SPICLK	IO	VDD_SPI	WPU, IE	WPU, IE					**SPICLK**	O/T	GPIO30	I/O/T						
35	SPIQ	IO	VDD_SPI	WPU, IE	WPU, IE					**SPIQ**	I1/O/T	GPIO31	I/O/T						
36	SPID	IO	VDD_SPI	WPU, IE	WPU, IE					**SPID**	I1/O/T	GPIO32	I/O/T						
37	GPIO33	IO	VDD_SPI/VDD3P3_CPU		IE					**GPIO33**	I/O/T	GPIO33	I/O/T	FSPIHD	I1/O/T	SUBSPIHD	I1/O/T	SPIIO4	I1/O/T
38	GPIO34	IO	VDD_SPI/VDD3P3_CPU		IE					**GPIO34**	I/O/T	GPIO34	I/O/T	FSPICS0	I1/O/T	SUBSPICS0	O/T	SPIIO5	I1/O/T
39	GPIO35	IO	VDD_SPI/VDD3P3_CPU		IE					**GPIO35**	I/O/T	GPIO35	I/O/T	FSPID	I1/O/T	SUBSPID	I1/O/T	SPIIO6	I1/O/T
40	GPIO36	IO	VDD_SPI/VDD3P3_CPU		IE					**GPIO36**	I/O/T	GPIO36	I/O/T	FSPICLK	I1/O/T	SUBSPICLK	O/T	SPIIO7	I1/O/T
41	GPIO37	IO	VDD_SPI/VDD3P3_CPU		IE					**GPIO37**	I/O/T	GPIO37	I/O/T	FSPIQ	I1/O/T	SUBSPIQ	I1/O/T	SPIDQS	I0/O/T
42	GPIO38	IO	VDD3P3_CPU		IE					**GPIO38**	I/O/T	GPIO38	I/O/T	FSPIWP	I1/O/T	SUBSPIWP	I1/O/T		
43	MTCK	IO	VDD3P3_CPU		IE					**MTCK**	I1	GPIO39	I/O/T	CLK_OUT3	O	SUBSPICS1	O/T		
44	MTDO	IO	VDD3P3_CPU		IE					**MTDO**	O/T	GPIO40	I/O/T	CLK_OUT2	O				
45	VDD3P3_CPU	Power																	
46	MTDI	IO	VDD3P3_CPU		IE					**MTDI**	I1	GPIO41	I/O/T	CLK_OUT1	O				
47	MTMS	IO	VDD3P3_CPU		IE					**MTMS**	I0	GPIO42	I/O/T						
48	U0TXD	IO	VDD3P3_CPU	WPU, IE	WPU, IE					**U0TXD**	O	GPIO43	I/O/T	CLK_OUT1	O				
49	U0RXD	IO	VDD3P3_CPU	WPU, IE	WPU, IE					**U0RXD**	I1	GPIO44	I/O/T	CLK_OUT2	O				
50	GPIO45	IO	VDD3P3_CPU	WPD, IE	WPD, IE					**GPIO45**	I/O/T	GPIO45	I/O/T						
51	VDDA	Power																	
52	XTAL_N	Analog																	
53	XTAL_P	Analog																	
54	VDDA	Power																	
55	GPIO46	IO	VDD3P3_CPU	WPD, IE	WPD, IE					**GPIO46**	I	GPIO46	I						
56	CHIP_PU	Analog	VDD3P3_RTC_IO																

* For details, see Section 2 *Pins*. Regarding highlighted cells, see Section 2.3.5 *Restrictions for GPIOs and RTC_GPIOs*.

Datasheet Versioning

Datasheet Version	Status	Watermark	Definition
v0.1 ~ v0.5 (excluding v0.5)	Draft	Confidential	This datasheet is under development for products in the design stage. Specifications may change without prior notice.
v0.5 ~ v1.0 (excluding v1.0)	Preliminary release	Preliminary	This datasheet is actively updated for products in the verification stage. Specifications may change before mass production, and the changes will be documentation in the datasheet's Revision History.
v1.0 and higher	Official release	—	This datasheet is publicly released for products in mass production. Specifications are finalized, and major changes will be communicated via Product Change Notifications (PCN).
Any version	—	Not Recommended for New Design (NRND)[1]	This datasheet is updated less frequently for products not recommended for new designs.
Any version	—	End of Life (EOL)[2]	This datasheet is no longer mtained for products that have reached end of life.

[1] Watermark will be added to the datasheet title page only when all the product variants covered by this datasheet are not recommended for new designs.

[2] Watermark will be added to the datasheet title page only when all the product variants covered by this datasheet have reached end of life.

Glossary

module

A self-contained unit integrated within the chip to extend its capabilities, such as cryptographic modules, RF modules 2

peripheral

A hardware component or subsystem within the chip to interface with the outside world 2

in-package flash

Flash integrated directly into the chip's package, and external to the chip die 3

off-package flash

Flash external to the chip's package 16

strapping pin

A type of GPIO pin used to configure certain operational settings during the chip's power-up, and can be reconfigured as normal GPIO after the chip's reset 29

eFuse parameter

A parameter stored in an electrically programmable fuse (eFuse) memory within a chip. The parameter can be set by programming EFUSE_PGM_DATA*n*_REG registers, and read by reading a register field named after the parameter 29

SPI boot mode

A boot mode in which users load and execute the existing code from SPI flash 30

joint download boot mode

A boot mode in which users can download code into flash via the UART or other interfaces (see Table 3-3 *Chip Boot Mode Control* > Note), and load and execute the downloaded code from the flash or SRAM 30

Related Documentation and Resources

Related Documentation

- *ESP32-S2 Technical Reference Manual* – Detailed information on how to use the ESP32-S2 memory and peripherals.
- *ESP32-S2 Hardware Design Guidelines* – Guidelines on how to integrate the ESP32-S2 into your hardware product.
- *ESP32-S2 Series SoC Errata* – Descriptions of known errors in ESP32-S2 series of SoCs.
- *Certificates*
  https://espressif.com/en/support/documents/certificates
- *ESP32-S2 Product/Process Change Notifications (PCN)*
  https://espressif.com/en/support/documents/pcns?keys=ESP32-S2
- *ESP32-S2 Advisories* – Information on security, bugs, compatibility, component reliability.
  https://espressif.com/en/support/documents/advisories?keys=ESP32-S2
- *Documentation Updates and Update Notification Subscription*
  https://espressif.com/en/support/download/documents

Developer Zone

- ESP-IDF Programming Guide for ESP32-S2 – Extensive documentation for the ESP-IDF development framework.
- *ESP-IDF* and other development frameworks on GitHub.
  https://github.com/espressif
- *ESP32 BBS Forum* – Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
  https://esp32.com/
- *ESP-FAQ* – A summary document of frequently asked questions released by Espressif.
  https://espressif.com/projects/esp-faq/en/latest/index.html
- *The ESP Journal* – Best Practices, Articles, and Notes from Espressif folks.
  https://blog.espressif.com/
- See the tabs *SDKs and Demos*, *Apps*, *Tools*, *AT Firmware*.
  https://espressif.com/en/support/download/sdks-demos

Products

- *ESP32-S2 Series SoCs* – Browse through all ESP32-S2 SoCs.
  https://espressif.com/en/products/socs?id=ESP32-S2
- *ESP32-S2 Series Modules* – Browse through all ESP32-S2-based modules.
  https://espressif.com/en/products/modules?id=ESP32-S2
- *ESP32-S2 Series DevKits* – Browse through all ESP32-S2-based devkits.
  https://espressif.com/en/products/devkits?id=ESP32-S2
- *ESP Product Selector* – Find an Espressif hardware product suitable for your needs by comparing or applying filters.
  https://products.espressif.com/#/product-selector?language=en

Contact Us

- See the tabs *Sales Questions*, *Technical Enquiries*, *Circuit Schematic & PCB Design Review*, *Get Samples* (Online stores), *Become Our Supplier*, *Comments & Suggestions*.
  https://espressif.com/en/contact-us/sales-questions

Revision History

Date	Version	Release notes
2025.11	v1.8	• Upgraded the document template; • Updated the coremark result.
2024.07	v1.7	• According to PCN20240602, upgraded from chip revision v0.0 to chip revision v1.0; • Updated the resolution of SAR ADC in Features and Section 4.2.2.1 *ADC*; • Updated Section 6.1 *Wi-Fi Radio*; • Updated the configuration of pin DAC_2 after reset in Appendix A.1. IO MUX.
2023.02	v1.6	• Added Table 3-2 *Description of Timing Parameters for the Strapping Pins* and Figure 3-1 *Visualization of Timing Parameters for the Strapping Pins*; • Added references to the Technical Reference Manual in Chapter 4 *Functional Description*; • Added cumulative IO current to Table 5-1 *Absolute Maximum Ratings*.
2022.12	v1.5	• Delete feature ”Supports external power amplifier”; • Added a note about Xtensa® Instruction Set Architecture (ISA) Summary in Section 4.1.1.1 *CPU*; • Added the package diagram for ESP32-S2FN4R2 in Chapter 7 *Packaging*.
2022.09	v1.4	• Updated Figure ESP32-S2 Functional Block Diagram to show power modes; • Added CoreMark score in Features; • Added a note about external crystal clock in Section 4.1.3.1 *CPU Clock*; • Added Table *Mapping of SPI Signal Buses and Chip Pins*; • Added Section 4.1.1.3 *DMA Controller*; • Added the clock of ULP coprocessor in Section 4.1.1.2 *ULP Coprocessor*; • Added note 3 to Table 5.2 *Recommended Operating Conditions*; • Updated Section 5.6 *Current Consumption*; • Replaced ”chip family” with ”chip series” following Espressif's taxonomy; • Updated Section ”Learning Resources” and renamed to ”Related Documentation and Resources”; • Other updates to wording.

Cont'd on next page

Cont'd from previous page

Date	Version	Release notes
2021.06	v1.3	• Added chip variant ESP32-S2R2; • Updated Table 5-11 *Reliability Qualifications*; • Added the link to recommended PCB land pattern in Chapter 7 *Packaging*; • Added Chapter 7 ; • Other minor updates.
2021.02	v1.2	• Added chip variant ESP32-S2FN4R2; • Added information about Section 4.2.1.12 *Two-wire Automotive Interface*; • Updated operating temperature to ambient temperature in Table 1-1 *ESP32-S2 Series Comparison*; • Updated Table 5-7 *Wi-Fi Current Consumption Depending on RF Modes*; • Updated current consumption drawn by ULP-FSM and ULP-RISC-V respectively in Table 5-9 *Current Consumption in Low-Power Modes*.
2020.09	v1.1	• Added chip variant ESP32-S2FH2、ESP32-S2FH4; • Added Chapter 1 *ESP32-S2 Series Comparison*.
2020.06	v1.0	• Modified the second note under Table Strapping Pins; • Modified the frequency of internal RC oscillator in Section 4.1.3.1 *RTC Clock* from 150 kHz to 90 kHz; • Renamed RISCV to RISC-V and ULP-RISCV to ULP-RISC-V in Section 4.1.1.2 *ULP Coprocessor*; • Modified a few figures in Table 5-9 *Current Consumption in Low-Power Modes*; • Added a note about V$_{OH}$ and V$_{OL}$ under Table 5-4 *DC Characteristics (3.3 V, 25 °C)*; • Added Table 5-11 *Reliability Qualifications*; • Other small changes.
2019.11	v0.4	• Updated Section 4.1.1.2 *ULP Coprocessor*; • Updated Section 4.1.3.2 *System Timer*; • Updated Table GPIO Matrix; • Added documentation feedback hyperlink; • Fixed formatting issues; • Other small changes.
2019.08	v0.3	Overall update.
2019.06	v0.2	• Updated Figure 2-2 *ESP32-S2 Power Scheme*; • Updated Section 3 *Boot Configurations*; • Updated Figure 4-1 *Address Mapping Structure*; • Updated Chapter 5 *Electrical Characteristics*.
2019.04	v0.1	Preliminary release.

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

ALL THIRD PARTY'S INFORMATION IN THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES TO ITS AUTHENTICITY AND ACCURACY.

NO WARRANTY IS PROVIDED TO THIS DOCUMENT FOR ITS MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, NOR DOES ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

www.espressif.com