ESP32-C5 Series

Datasheet

Ultra-low-power SoC with 32-bit RISC-V single-core microprocessor
2.4 and 5 GHz dual-band Wi-Fi 6 (802.11ax), Bluetooth® 5 (LE), Zigbee and Thread (802.15.4)
Optional 4 MB flash in the chip's package
27 or 19 GPIOs, rich set of peripherals
QFN48 (6×6 mm) or QFN40 (5×5 mm) package

Including:

ESP32-C5

ESP32-C5FH4

NOTE:

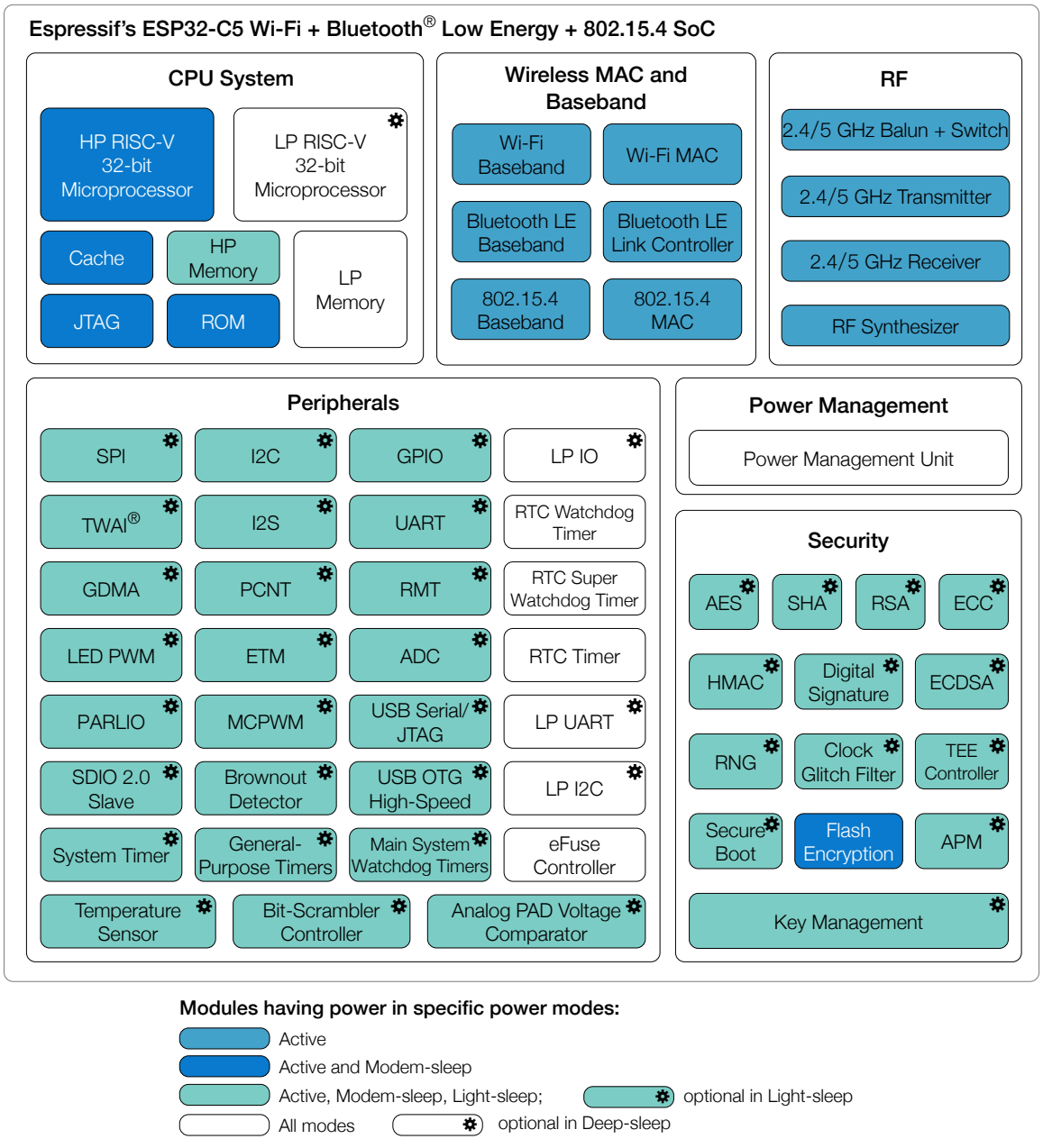
Specifications in this datasheet are very close to the final MP version, but can still be different. Notes for beta chips' specifications will be provided.

Version 0.1
Espressif Systems

Product Overview

The ESP32-C5 SoC (System on Chip) supports 2.4 and 5 GHz dual-band Wi-Fi 6, Bluetooth LE 5, Zigbee 3.0 and Thread 1.3.

The functional block diagram of the SoC is shown below.



ESP32-C5 Functional Block Diagram

For more information on power consumption, see Section 3.9 Low Power Management.

Features

Wi-Fi

- 1T1R in 2.4 and 5 GHz dual band
- Operating frequency: 2412 ~ 2484 MHz, 5160 ~ 5885 MHz
- IEEE 802.11ax-compliant
  - 20 MHz-only non-AP mode
  - Uplink and downlink OFDMA, especially suitable for simultaneous connections in high-density environments
  - Downlink MU-MIMO (multi-user, multiple input, multiple output) to increase network capacity
  - Beamformee that improves signal quality
  - Extended range (ER) mode to cover a wider range
  - Spatial reuse to maximize parallel transmissions
  - Target wake time (TWT) that optimizes power saving mechanisms
- IEEE 802.11ac-compliant
  - 20 MHz bandwidth
- Fully compatible with IEEE 802.11b/g/n protocol
  - 20 MHz and 40 MHz bandwidth
  - Data rate up to 150 Mbps
  - Wi-Fi Multimedia (WMM)
  - TX/RX A-MPDU, TX/RX A-MSDU
  - Immediate Block ACK
  - Fragmentation and defragmentation
  - Transmit opportunity (TXOP)
  - Automatic Beacon monitoring (hardware TSF)
  - 4 × virtual Wi-Fi interfaces
  - Simultaneous support for Infrastructure BSS in Station mode, SoftAP mode, Station + SoftAP mode, and promiscuous mode
    Note that when ESP32-C5 scans in Station mode, the SoftAP channel will change along with the Station channel
  - Antenna diversity
  - 802.11mc FTM

Bluetooth®

- Bluetooth LE: Bluetooth 5 certified
- Bluetooth Mesh 1.1
- Speed: 125 Kbps, 500 Kbps, 1 Mbps, 2 Mbps
- LE advertising extensions
- Multiple connections (central and peripheral)
- Multiple advertisement sets
- LE secure connections
- LE Privacy 1.2
- Channel Selection Algorithm #2

IEEE 802.15.4

- Compliant with IEEE 802.15.4-2015 protocol
- OQPSK PHY in 2.4 GHz band
- Data rate: 250 Kbps
- Thread 1.3
- Zigbee 3.0

CPU and Memory

- High-performance (HP) RISC-V processor:
  - Clock speed: up to 240 MHz
  - Five stage pipeline
- Low-power (LP) RISC-V processor:
  - Clock speed: up to 40 MHz
  - Two stage pipeline

- ROM: 320 KB
- HP SRAM: 512 KB
- LP SRAM: 16 KB
- Flash controller with cache is supported
- Flash in-Circuit Programming (ICP) is supported

Advanced Peripheral Interfaces

- 27 × GPIOs (QFN48), or 19 × GPIOs (QFN40)
- Analog interfaces:
  - 1 × 12-bit SAR ADC, up to 6 channels
  - 1 × Temperature sensor
  - 1 × Brownout detector
  - 1 × Analog PAD voltage comparator
- Digital interfaces:
  - 2 × UART
  - 1 × Low-power (LP) UART
  - 2 × SPI ports for communication with flash
  - 1 × General purpose SPI port
  - 1 × I2C
  - 1 × Low-power (LP) I2C
  - 1 × I2S
  - 1 × Pulse count controller
  - 1 × USB 2.0 OTG High-Speed
  - 1 × USB Serial/JTAG controller
  - 2 × TWAI® controller, compatible with ISO 11898-1 (CAN Specification 2.0)
  - 1 × SDIO 2.0 slave controller
  - 1 × LED PWM controller, up to 6 channels
  - 1 × Motor Control PWM (MCPWM), up to 6 channels
  - 1 × Remote control peripheral (RMT) (TX/RX)
  - 1 × Parallel IO interface (PARLIO)
  - 1 × General DMA controller (GDMA), with 3 transmit channels and 3 receive channels
  - 1 × Bit-scrambler controller
  - 1 × Event task matrix (ETM)
- Timers:
  - 1 × 52-bit system timer
  - 2 × 54-bit general-purpose timers
  - 1 × 48-bit RTC timer
  - 3 × digital watchdog timers
  - 1 × analog watchdog timer

Applications

With low power consumption, ESP32-C5 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
- Wi-Fi Dongle

Contents

List of Tables

List of Figures

1 ESP32-C5 Series Comparison

1.1 Nomenclature

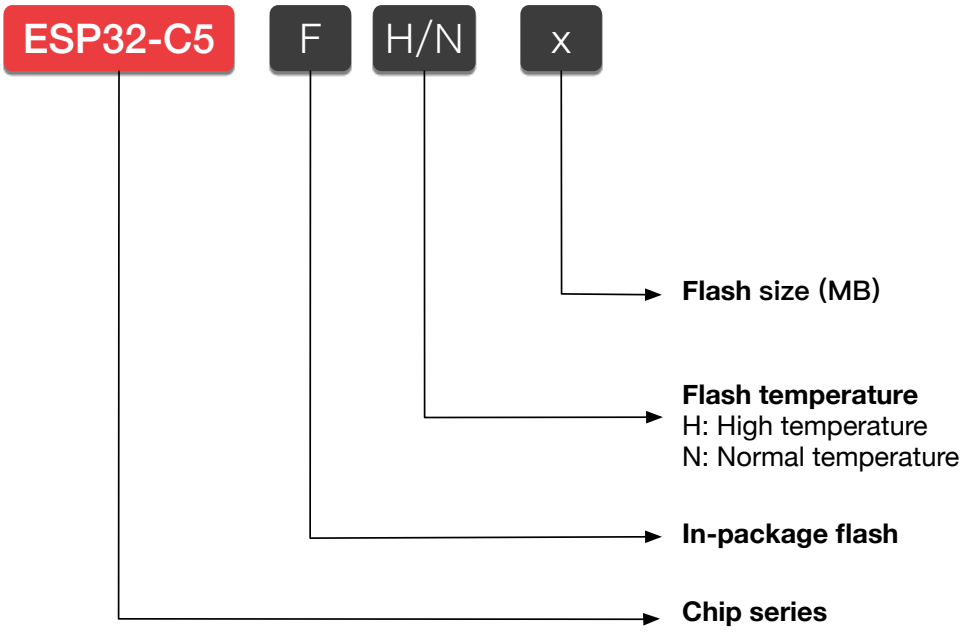


Figure 1-1. ESP32-C5 Series Nomenclature

1.2 Comparison

Table 1-1. ESP32-C5 Series Comparison

Ordering Code	In-Package Flash	Ambient Temp.[1] (°C)	Package[2]
ESP32-C5	—[3]	–40 ~ 105	QFN48 (6×6 mm)
ESP32-C5FH4	4 MB (Quad SPI)[4]	–40 ~ 105	QFN40 (5×5 mm)

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

[2] Mass-produced chips will be available in the QFN48 and QFN40 packages, while beta chips come in the QFN68 package.

[3] Can connect a flash outside the chip package. For details, see Section 3.1.4 Off-Package Flash.

[4] For details about SPI modes, see Section 2.7 Pin Mapping Between Chip and Flash.

2 Pins

2.1 Pin Layout

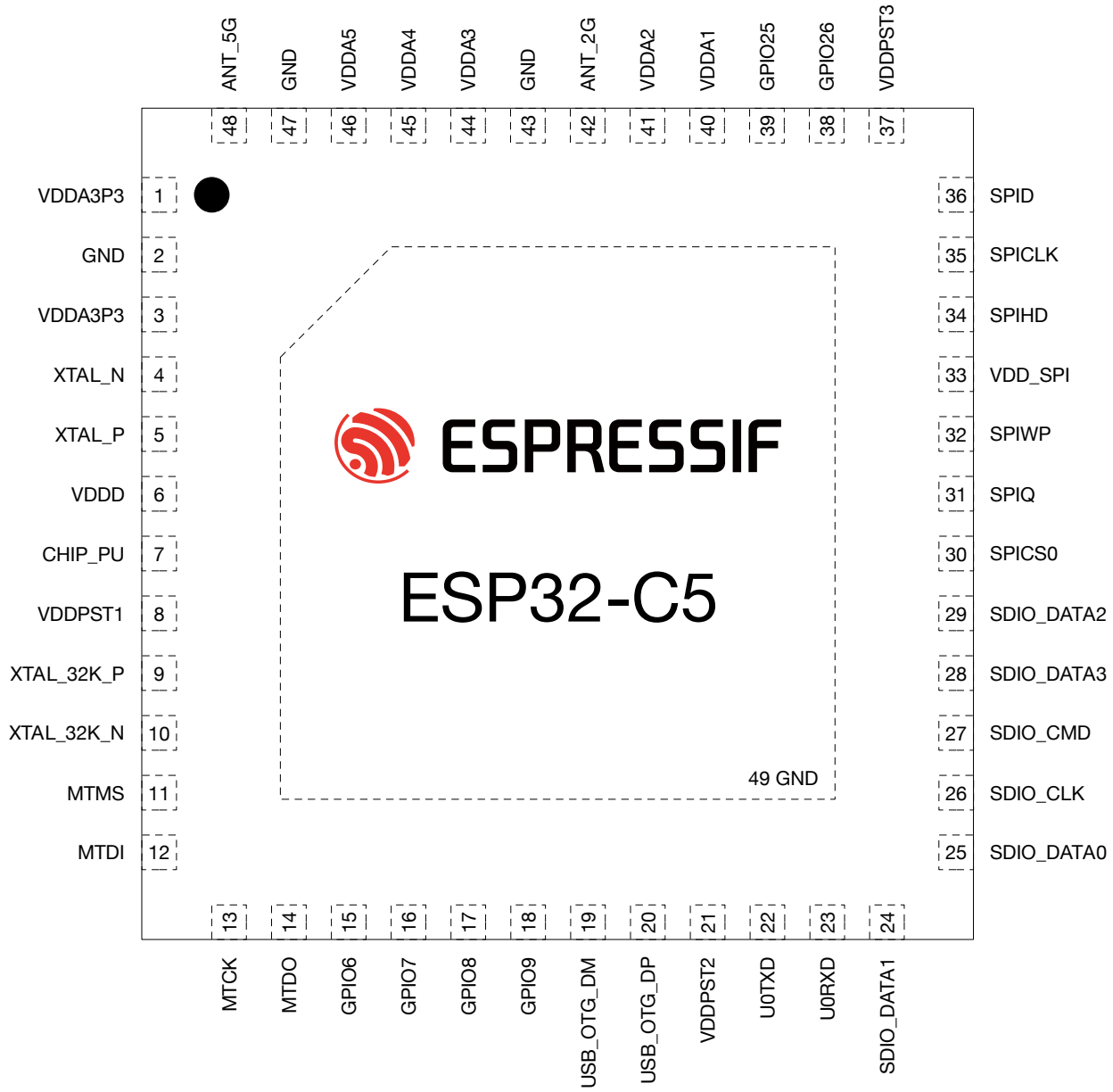


Figure 2-1. ESP32-C5 Pin Layout (QFN48, Top View)

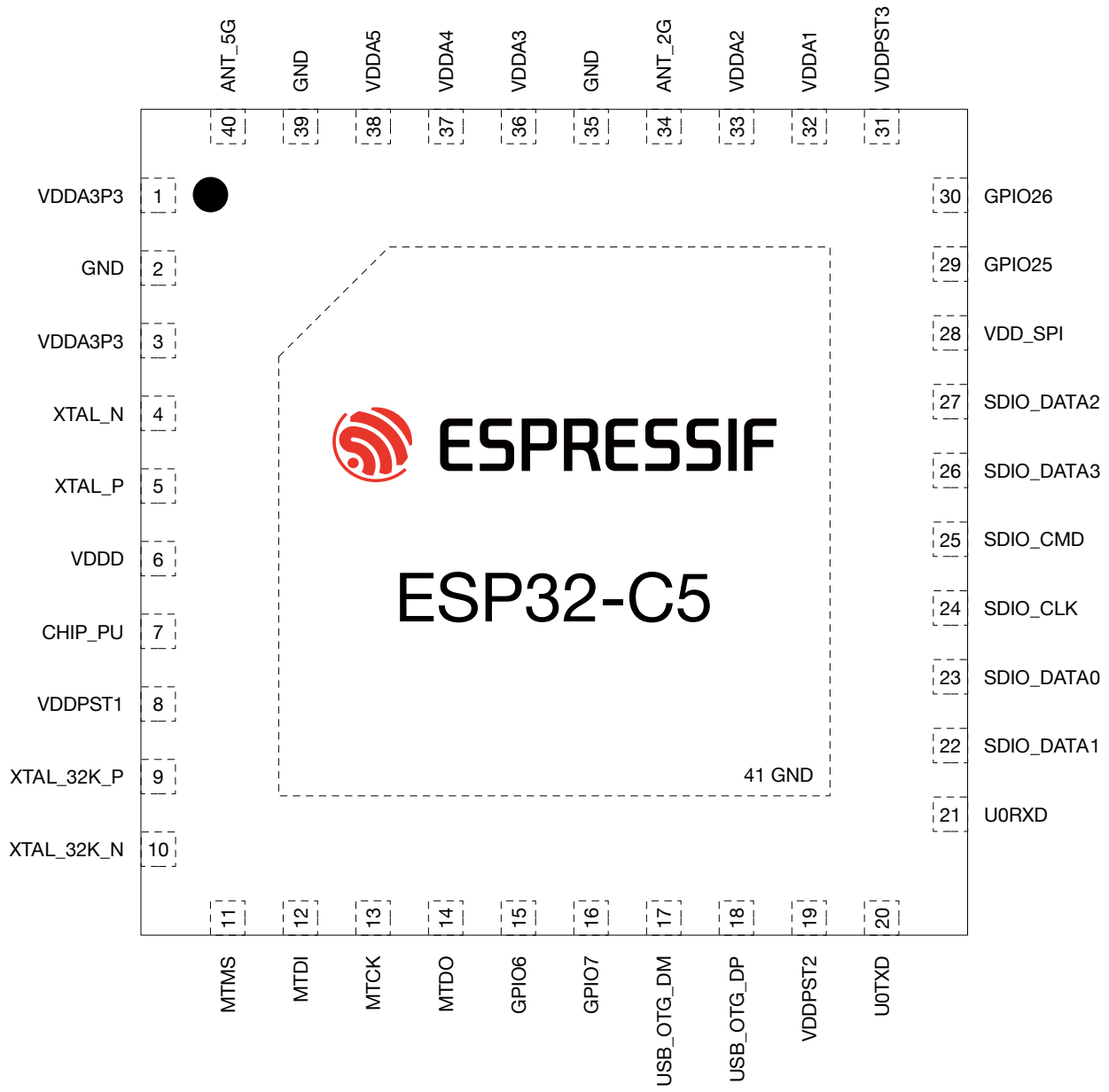


Figure 2-2. ESP32-C5 Pin Layout (QFN40, Top View)

2.2 Pin Overview

The ESP32-C5 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.

All in all, the ESP32-C5 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 and GPIO functions** – see Table 2-3 *QFN48 IO MUX Pin Functions* or Table 2-4 *QFN40 IO MUX Pin Functions*
  - **Some** IO pins have predefined **LP IO MUX functions** – see Table 2-5 *LP IO MUX Functions*
  - **Some** IO pins have predefined **analog functions** – see Table 2-6 *Analog Functions*

  Predefined functions means that each IO pin has a set of direct connections to certain on-chip components. During run-time, the user can configure which component 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 2-7 *Analog Pins*
- **Power pins** supply power to the chip components and non-power pins – see Table 2-8 *Power Pins*

GPIO Input Mode

The input function of GPIO can be configured as hysteresis or normal mode:

- **Hysteresis mode** - In the hysteresis mode, the threshold voltage for flipping between high and low levels of GPIO input depends on the direction of level flipping. Specifically, the voltage threshold for flipping from high to low level is slightly lower than the voltage threshold for flipping from low to high level.
- **Normal mode** - The threshold voltage for flipping between high and low levels of GPIO input is independent of the direction of level flipping. In other words, the voltage threshold for flipping from high to low level is the same as the voltage threshold for flipping from low to high level.

Notes for Table 2-1 QFN48 Pin Overview or Table 2-2 QFN40 Pin Overview (see below):

1. For more information, see respective sections below. Alternatively, see Appendix A – ESP32-C5 Consolidated Pin Overview.
2. **Bold** marks the pin function set in which a pin has its default function in the default boot mode. See Section 2.6.1 Chip Boot Mode Control.
3. 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.
4. Except for GPIO25 and GPIO26 whose default drive strength is 40 mA, the default drive strength for all the other pins is 20 mA.
5. Column **Pin Settings** shows predefined settings at reset and after reset with the following abbreviations:
   - IE – input enabled

- OE – output enabled
- WPU – internal weak pull-up resistor enabled
- WPD – internal weak pull-down resistor enabled
- USB_PU – USB pull-up resistor enabled
  - By default, the USB function is enabled for USB pins (i.e., GPIO25 and GPIO26), and the pin pull-up is decided by the USB pull-up. The USB pull-up resistor is controlled by USB_SERIAL_JTAG_DP/DM_PULLUP and its pull-up value is controlled by USB_SERIAL_JTAG_PULLUP_VALUE.
  - When the USB function is disabled, USB pins are used as regular GPIOs. In this case, the internal weak pull-up and pull-down resistors of the USB pins are disabled by default, but are configurable by IO_MUX_GPIO_FUN_WPU/WPD.

6. Depends on the value of EFUSE_DIS_PAD_JTAG
   - `0` - default value. Input enabled, and internal weak pull-up resistor enabled (IE & WPU)
   - `1` - input enabled (IE)
7. Pins not available in the QFN40 package compared to the QFN48

Table 2-1. QFN48 Pin Overview

Pin No.	Pin Name [7]	Pin Type [1]	Pin Providing Power [3-5]	Pin Settings [5,6]		Pin Function Sets [1,2]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
1	VDDA3P3	Power						
2	GND	Power						
3	VDDA3P3	Power						
4	XTAL_N	Analog						
5	XTAL_P	Analog						
6	VDDD	Power						
7	CHIP_PU	Analog						
8	VDDPST1	Power						
9	XTAL_32K_P	IO	VDDPST1			**IO MUX**	LP IO MUX	Analog
10	XTAL_32K_N	IO	VDDPST1			**IO MUX**	LP IO MUX	Analog
11	MTMS	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
12	MTDI	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
13	MTCK	IO	VDDPST1		IE, WPU [6]	**IO MUX**	LP IO MUX	Analog
14	MTDO	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
15	GPIO6	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
16	GPIO7	IO	VDDPST1	IE, WPU	IE, WPU	**IO MUX**	LP IO MUX	
17	GPIO8 [7]	IO	VDDPST1			**IO MUX**		Analog
18	GPIO9 [7]	IO	VDDPST1			**IO MUX**		Analog
19	USB_OTG_DM	Analog	VDDPST2					
20	USB_OTG_DP	Analog	VDDPST2					
21	VDDPST2	Power						
22	U0TXD	IO	VDDPST2		OE, WPU [5]	**IO MUX**		
23	U0RXD	IO	VDDPST2		IE, WPU	**IO MUX**		
24	SDIO_DATA1	IO	VDDPST2		IE	**IO MUX**		

Cont'd on next page

Table 2-1 – cont'd from previous page

Pin No.	Pin Name [7]	Pin Type [1]	Pin Providing Power [3-5]	Pin Settings [5,6]		Pin Function Sets [1,2]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
25	SDIO_DATA0	IO	VDDPST2		IE	**IO MUX**		
26	SDIO_CLK	IO	VDDPST2		IE	**IO MUX**		
27	SDIO_CMD	IO	VDDPST2		IE	**IO MUX**		
28	SDIO_DATA3	IO	VDDPST2		IE	**IO MUX**		
29	SDIO_DATA2	IO	VDDPST2		IE	**IO MUX**		
30	SPICS0 [7]	IO	VDD_SPI	WPU	IE, WPU	**IO MUX**		
31	SPIQ [7]	IO	VDD_SPI	WPU	IE, WPU	**IO MUX**		
32	SPIWP [7]	IO	VDD_SPI	WPU	IE, WPU	**IO MUX**		
33	VDD_SPI	Power/IO	—			IO MUX		**Analog**
34	SPIHD [7]	IO	VDD_SPI	WPU	IE, WPU	**IO MUX**		
35	SPICLK [7]	IO	VDD_SPI	WPU	IE, WPU	**IO MUX**		
36	SPID [7]	IO	VDD_SPI	WPU	IE, WPU	**IO MUX**		
37	VDDPST3	Power						
38	GPIO26	IO	VDDPST3		IE, USB_PU	IO MUX		**Analog**
39	GPIO25	IO	VDDPST3		IE	IO MUX		**Analog**
40	VDDA1	Power						
41	VDDA2	Power						
42	ANT_2G	Analog						
43	GND	Power						
44	VDDA3	Power						
45	VDDA4	Power						
46	VDDA5	Power						
47	GND	Power						
48	ANT_5G	Analog						

Table 2-2. QFN40 Pin Overview

Pin No.	Pin Name	Pin Type [1]	Pin Providing Power [3-5]	Pin Settings [5,6]		Pin Function Sets [1,2]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
1	VDDA3P3	Power						
2	GND	Power						
3	VDDA3P3	Power						
4	XTAL_N	Analog						
5	XTAL_P	Analog						
6	VDDD	Power						
7	CHIP_PU	Analog						
8	VDDPST1	Power						
9	XTAL_32K_P	IO	VDDPST1			**IO MUX**	LP IO MUX	Analog
10	XTAL_32K_N	IO	VDDPST1			**IO MUX**	LP IO MUX	Analog
11	MTMS	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
12	MTDI	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
13	MTCK	IO	VDDPST1		IE, WPU [6]	**IO MUX**	LP IO MUX	Analog
14	MTDO	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog
15	GPIO6	IO	VDDPST1	IE	IE	**IO MUX**	LP IO MUX	Analog

Cont'd on next page

Table 2-2 – cont'd from previous page

Pin No.	Pin Name	Pin Type [1]	Pin Providing Power [3-5]	Pin Settings [5,6]		Pin Function Sets [1,2]		
				At Reset	After Reset	IO MUX	LP IO MUX	Analog
16	GPIO7	IO	VDDPST1	IE, WPU	IE, WPU	**IO MUX**	LP IO MUX	
17	USB_OTG_DM	Analog	VDDPST2					
18	USB_OTG_DP	Analog	VDDPST2					
19	VDDPST2	Power						
20	U0TXD	IO	VDDPST2		OE, WPU [5]	**IO MUX**		
21	U0RXD	IO	VDDPST2		IE, WPU	**IO MUX**		
22	SDIO_DATA1	IO	VDDPST2		IE	**IO MUX**		
23	SDIO_DATA0	IO	VDDPST2		IE	**IO MUX**		
24	SDIO_CLK	IO	VDDPST2		IE	**IO MUX**		
25	SDIO_CMD	IO	VDDPST2		IE	**IO MUX**		
26	SDIO_DATA3	IO	VDDPST2		IE	**IO MUX**		
27	SDIO_DATA2	IO	VDDPST2		IE	**IO MUX**		
28	VDD_SPI	Power/IO	—			IO MUX		**Analog**
29	GPIO25	IO	VDDPST3		IE	IO MUX		**Analog**
30	GPIO26	IO	VDDPST3		IE, USB_PU	IO MUX		**Analog**
31	VDDPST3	Power						
32	VDDA1	Power						
33	VDDA2	Power						
34	ANT_2G	Analog						
35	GND	Power						
36	VDDA3	Power						
37	VDDA4	Power						
38	VDDA5	Power						
39	GND	Power						
40	ANT_5G	Analog						

2.3 IO Pins

2.3.1 IO MUX and GPIO Pin Functions

The pins of ESP32-C5 can be assigned any function (F0-F2) from their respective sets of IO MUX functions as listed in Table 2-3 *QFN48 IO MUX Pin Functions* or Table 2-4 *QFN40 IO MUX Pin Functions*.

Each set of the IO MUX functions has a general purpose input/output (**GPIO0, GPIO1, etc.**) function. If a pin is assigned a GPIO function, this pin's signal is routed via the GPIO matrix, which incorporates internal signal routing circuitry for mapping signals programmatically. It gives the pin access to almost any IO MUX function. However, the flexibility of programmatic mapping comes at a cost as it might affect speed and latency of routed signals.

Notes for 2-3 QFN48 IO MUX Pin Functions or Table 2-4 QFN40 IO MUX Pin Functions:

1. **Bold** marks the default pin functions in the default boot mode. See Section 2.6.1 Chip Boot Mode Control.
2. Regarding highlighted cells, see Section 2.3.4 Restrictions for GPIOs and LP GPIOs.
3. Each IO MUX function (F*n*, *n* = 0 ~ 2) 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`.
4. Function names:

GPIO…	General-purpose input/output with signals routed via the GPIO matrix.
U…RXD U…TXD	UART0/1 receive/transmit signals.
SDIO…	SDIO interface signals.

5. Groups of functions (see the markings in the table):
   a. JTAG interface for debugging.
   b. UART interface for debugging.
   c. SPI0/1 interface for connection to in-package or off-package flash via SPI bus. See also Section 2.7 Pin Mapping Between Chip and Flash.
   d. SPI2 main interface for fast SPI connection. Among these pins, FSPICS0 is for input or output signals in master or slave mode, whereas FSPICS1 ~ FSPICS5 are for output signals in master mode.

Table 2-3. QFN48 IO MUX Pin Functions

Pin No.	IO MUX / GPIO Name	IO MUX Function					
		0	Type	1	Type	2	Type
9	GPIO0	**GPIO0**	I/O/T	GPIO0	I/O/T		
10	GPIO1	**GPIO1** 5a	I/O/T	GPIO1	I/O/T	5d	
11	GPIO2	**MTMS**	I1	GPIO2	I/O/T	FSPIQ	I1/O/T
12	GPIO3	**MTDI**	I1	GPIO3	I/O/T		I1/O/T
13	GPIO4	**MTCK**	I1	GPIO4	I/O/T	FSPIHD	I1/O/T
14	GPIO5	**MTDO**	O/T	GPIO5	I/O/T	FSPIWP	I1/O/T
15	GPIO6	**GPIO6**	I/O/T	GPIO6	I/O/T	FSPICLK	I1/O/T
16	GPIO7	**GPIO7**	I/O/T	GPIO7	I/O/T	FSPID	I1/O/T
17	GPIO8	**GPIO8**	I/O/T	GPIO8	I/O/T		
18	GPIO9	**GPIO9** 5b	I/O/T	GPIO9	I/O/T		
22	GPIO10	**U0TXD**	O	GPIO10	I/O/T		
23	GPIO11	**U0RXD**	I1	GPIO11	I/O/T	5d	
24	GPIO12	**SDIO_DATA1**	I1/O/T	GPIO12	I/O/T	FSPICS0	I1/O/T
25	GPIO13	**SDIO_DATA0**	I1/O/T	GPIO13	I/O/T	FSPICS1	O/T
26	GPIO14	**SDIO_CLK**	I1	GPIO14	I/O/T	FSPICS2	O/T
27	GPIO15	**SDIO_CMD**	I1/O/T	GPIO15	I/O/T	FSPICS3	O/T
28	GPIO16	**SDIO_DATA3**	I1/O/T	GPIO16	I/O/T	FSPICS4	O/T
29	GPIO17	**SDIO_DATA2** 5c	I1/O/T	GPIO17	I/O/T	FSPICS5	O/T
30	GPIO18	**SPICS0**	O/T	GPIO18	I/O/T		
31	GPIO19	**SPIQ**	I1/O/T	GPIO19	I/O/T		
32	GPIO20	SPIWP	I1/O/T	**GPIO20**	I/O/T		
33	GPIO21	GPIO21 5c	I/O/T	GPIO21	I/O/T		
34	GPIO22	SPIHD	I1/O/T	**GPIO22**	I/O/T		
35	GPIO23	**SPICLK**	O/T	GPIO23	I/O/T		
36	GPIO24	**SPID**	I1/O/T	GPIO24	I/O/T		
38	GPIO26	GPIO26	I/O/T	GPIO26	I/O/T		
39	GPIO25	GPIO25	I/O/T	GPIO25	I/O/T		

Table 2-4. QFN40 IO MUX Pin Functions

Pin No.	IO MUX / GPIO Name	IO MUX Function					
		0	Type	1	Type	2	Type
9	GPIO0	**GPIO0**	I/O/T	GPIO0	I/O/T		
10	GPIO1	**GPIO1** 5a	I/O/T	GPIO1	I/O/T	5d	
11	GPIO2	**MTMS**	I1	GPIO2	I/O/T	FSPIQ	I1/O/T
12	GPIO3	**MTDI**	I1	GPIO3	I/O/T		I1/O/T
13	GPIO4	**MTCK**	I1	GPIO4	I/O/T	FSPIHD	I1/O/T
14	GPIO5	**MTDO**	O/T	GPIO5	I/O/T	FSPIWP	I1/O/T
15	GPIO6	**GPIO6**	I/O/T	GPIO6	I/O/T	FSPICLK	I1/O/T
16	GPIO7	**GPIO7** 5b	I/O/T	GPIO7	I/O/T	FSPID	I1/O/T
20	GPIO10	**U0TXD**	O	GPIO10	I/O/T		
21	GPIO11	**U0RXD**	I1	GPIO11	I/O/T	5d	
22	GPIO12	**SDIO_DATA1**	I1/O/T	GPIO12	I/O/T	FSPICS0	I1/O/T
23	GPIO13	**SDIO_DATA0**	I1/O/T	GPIO13	I/O/T	FSPICS1	O/T
24	GPIO14	**SDIO_CLK**	I1	GPIO14	I/O/T	FSPICS2	O/T
25	GPIO15	**SDIO_CMD**	I1/O/T	GPIO15	I/O/T	FSPICS3	O/T
26	GPIO16	**SDIO_DATA3**	I1/O/T	GPIO16	I/O/T	FSPICS4	O/T
27	GPIO17	**SDIO_DATA2**	I1/O/T	GPIO17	I/O/T	FSPICS5	O/T
28	GPIO21	GPIO21	I/O/T	GPIO21	I/O/T		
29	GPIO25	GPIO25	I/O/T	GPIO25	I/O/T		
30	GPIO26	GPIO26	I/O/T	GPIO26	I/O/T		

2.3.2 LP IO MUX Functions

LP IO MUX functions are mainly used for the LP CPU and LP peripherals of the LP digital system. LP IO MUX functions and data input/output are configured by the LP CPU. When the HP digital system is shut down, the LP digital system can still operate independently.

Notes for Table 2-5 LP IO MUX Functions:

1. **Bold** marks the default pin functions in the default boot mode. See Section 2.6.1 Chip Boot Mode Control.
2. Regarding highlighted cells, see Section 2.3.4 Restrictions for GPIOs and LP GPIOs.
3. Function names:

LP_GPIO…	General-purpose input/output configured by LP CPU.
LP_UART…	LP UART functions.
LP_I2C…	LP I2C functions.

Table 2-5. LP IO MUX Functions

Pin No.	LP IO Name	LP IO MUX Function	
		0	1
9	LP_GPIO0	LP_GPIO0	LP_UART_DTRN
10	LP_GPIO1	LP_GPIO1	LP_UART_DSRN
11	LP_GPIO2	LP_GPIO2	LP_UART_RTSN
12	LP_GPIO3	LP_GPIO3	LP_UART_CTSN
13	LP_GPIO4	LP_GPIO4	LP_UART_RXD
14	LP_GPIO5	LP_GPIO5	LP_UART_TXD
15	LP_GPIO6	LP_GPIO6	LP_I2C_SDA
16	LP_GPIO7	LP_GPIO7	LP_I2C_SCL

2.3.3 Analog Functions

Notes for Table 2-6 Analog Functions:

1. **Bold** marks the default pin functions in SPI Boot mode.
2. Regarding highlighted cells, see Section 2.3.4 Restrictions for GPIOs and LP GPIOs.
3. Function names:

XTAL_32K_P XTAL_32K_N	32 kHz external clock input/output connected to ESP32-C5's oscillator. P/N means differential clock positive/negative.
ADC1_CH…	Analog to digital conversion channel for ADC1.
USB_D- USB_D+	USB Serial/JTAG function. USB signal is a differential signal transmitted over a pair of D+ and D- wires.
PAD_COMP0 PAD_COMP1	Analog PAD voltage comparator.

Table 2-6. Analog Functions

QFN48 Pin No.	QFN40 Pin No.	Analog IO Name	Analog Function 0	Analog Function 1
9	9	GPIO0	XTAL_32K_P	
10	10	GPIO1	XTAL_32K_N	ADC1_CH0
11	11	GPIO2		ADC1_CH1
12	12	GPIO3		ADC1_CH2
13	13	GPIO4		ADC1_CH3
14	14	GPIO5		ADC1_CH4
15	15	GPIO6		ADC1_CH5
17	—	GPIO8	PAD_COMP0	
18	—	GPIO9	PAD_COMP1	
33	28	GPIO21	**VDD_SPI**	
39	29	GPIO25	**USB_D-**	
38	30	GPIO26	**USB_D+**	

2.3.4 Restrictions for GPIOs and LP GPIOs

All IO pins of the ESP32-C5 have GPIO and some have LP GPIO pin functions. However, the IO pins are multiplexed and have other important pin functions. This should be taken into account while certain pins are chosen for general purpose input output.

In tables in Section 2.3 *IO Pins*, some pin functions are highlighted. The non-highlighted GPIO or LP GPIO pins are recommended for use first. If more pins are needed, the highlighted GPIOs or LP GPIOs should be chosen carefully to avoid conflicts with important pin functions.

The highlighted IO pins have the following important pin functions:

- GPIO – allocated for communication with flash and NOT recommended for other uses. For details, see Section 2.7 Pin Mapping Between Chip and Flash.
- GPIO – have one of the following important functions:
  - **Strapping pins** – need to be at certain logic levels at startup. See Section 2.6 Strapping Pins.
  - **USB_D+/-** – by default, connected to the USB Serial/JTAG Controller. To function as GPIOs, these pins need to be reconfigured.
  - **JTAG interface** – often used for debugging. See Table 2-3 *QFN48 IO MUX Pin Functions* or Table 2-4 *QFN40 IO MUX Pin Functions*, note 5a. To free these pins up, the pin functions USB_D+/- of the USB Serial/JTAG Controller can be used instead. See also Section 2.6.4 JTAG Signal Source Control.
  - **UART interface** – often used for debugging. See Table 2-3 *QFN48 IO MUX Pin Functions* or Table 2-4 *QFN40 IO MUX Pin Functions*, note 5b.

See also Appendix A – ESP32-C5 Consolidated Pin Overview.

2.4 Analog Pins

Table 2-7. Analog Pins

QFN48 Pin No.	QFN40 Pin No.	Pin Name	Pin Type	Pin Function
4	4	XTAL_N	—	External clock input/output connected to chip's crystal or oscillator. P/N means differential clock positive/negative.
5	5	XTAL_P	—	
7	7	CHIP_PU	—	High: On, enables the chip (powered up). Low: Off, the chip powers off (powered down). Note: Do not leave the CHIP_PU pin floating.
19	17	USB_OTG_DM	I/O	USB_OTG input/output
20	18	USB_OTG_DP	I/O	USB_OTG input/output
42	34	ANT_2G	I/O	RF input/output
48	40	ANT_5G	I/O	RF input/output

2.5 Power Supply

2.5.1 Power Pins

The chip is powered via the power pins described in Table 2-8 *Power Pins*.

Table 2-8. Power Pins

QFN48 Pin No.	QFN40 Pin No.	Pin Name	Direction	Power Supply [1,2]	
				Power Domain / Other	IO Pins [4]
1	1	VDDA3P3	Input	Analog power domain	
2	2	GND	—	External ground connection	
3	3	VDDA3P3	Input	Analog power domain	
6	6	VDDD	Input	Analog power domain	
8	8	VDDPST1	Input	LP digital power domain	LP IO
21	19	VDDPST2	Input	HP digital and part of analog pin power domains	HP IO
33	28	VDD_SPI [3]	Input	In-package flash (backup power line)	
			Output	In-package flash and off-package flash	
37	31	VDDPST3	Input	HP digital power domain	HP IO
40	32	VDDA1	Input	Analog power domain	
41	33	VDDA2	Input	Analog power domain	
43	35	GND	—	External ground connection	
44	36	VDDA3	Input	Analog power domain	
45	37	VDDA4	Input	Analog power domain	
46	38	VDDA5	Input	Analog power domain	
47	39	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 4.1 Absolute Maximum Ratings and Section 4.2 Recommended Power Supply Characteristics.

[3] To configure VDD_SPI as input or output.

[4] LP IO pins are those powered by VDDPST1 and so on, as shown in Figure 2-3 ESP32-C5 Power Scheme. See also Table 2-1 *QFN48 Pin Overview* or Table 2-2 *QFN40 Pin Overview* > Column *Pin Providing Power*.

2.5.2 Power Scheme

The power scheme is shown in Figure 2-3 ESP32-C5 Power Scheme.

The components on the chip are powered via voltage regulators.

Table 2-9. Voltage Regulators

Voltage Regulator	Output	Power Supply
HP	1.1 V	HP power domain
LP	1.1 V	LP power domain

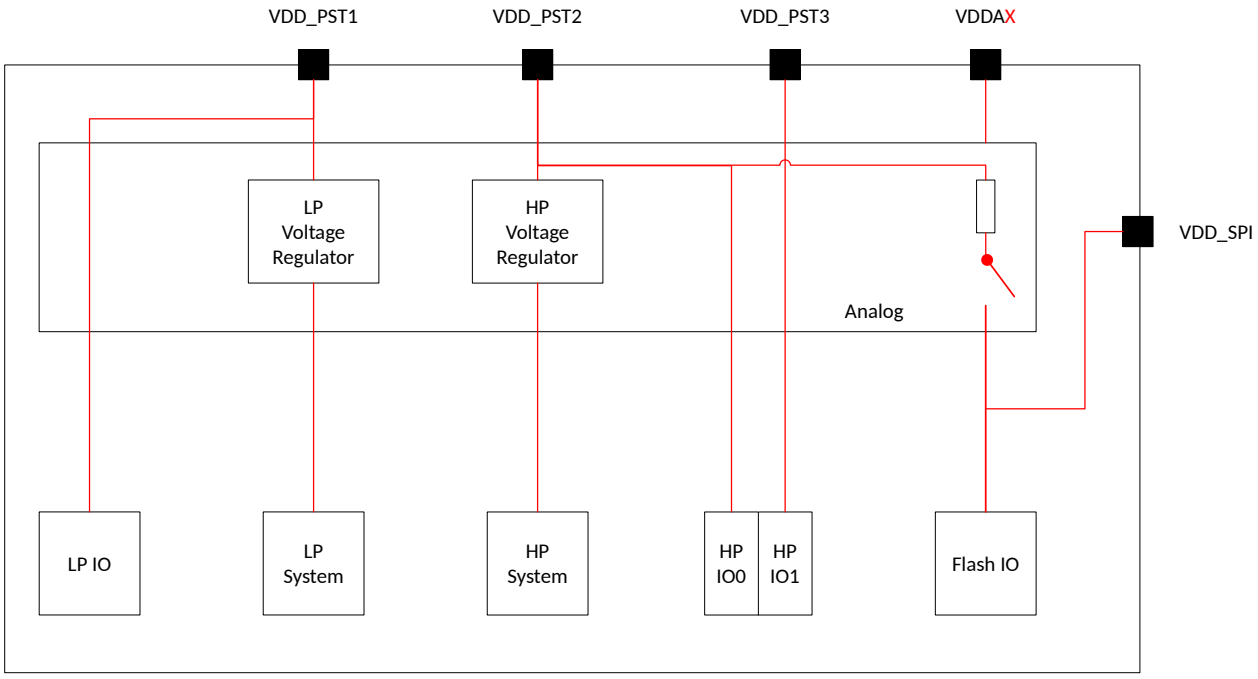


Figure 2-3. ESP32-C5 Power Scheme

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 2-4 and Table 2-10.

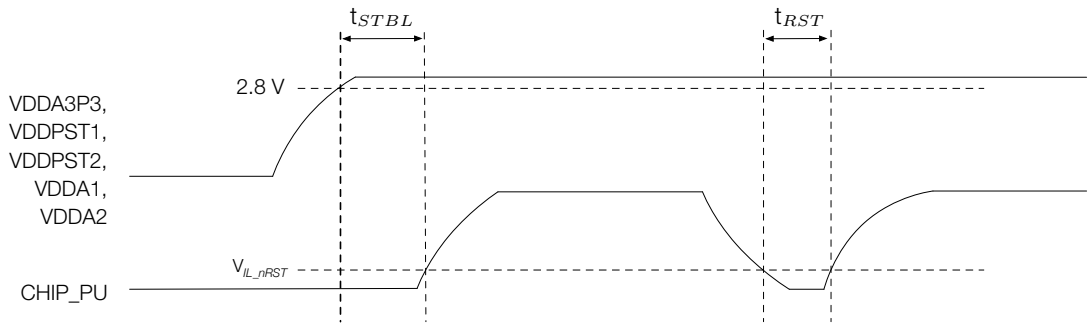


Figure 2-4. Visualization of Timing Parameters for Power-up and Reset

Table 2-10. Description of Timing Parameters for Power-up and Reset

Parameter	Description	Min (μs)
t_{STBL}	Time reserved for the power rails of VDDA3P3, VDDPST1, VDDPST2, VDDA1 and VDDA2 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 4-4)	50

2.6 Strapping Pins

At each startup or reset, a chip requires some initial configuration parameters, such as in which boot mode to load the chip, etc. These parameters are passed over via the strapping pins. After reset, the strapping pins operate as regular IO pins.

The parameters controlled by the given strapping pins at chip reset are as follows:

- **Chip boot mode** – GPIO6 and GPIO7
- **SDIO sampling and driving clock edge** – MTMS and MTDI
- **ROM code printing to UART** – GPIO6
- **JTAG signal source** – MTDO

GPIO7 is connected to the chip's internal weak pull-up resistor at chip reset. This resistor determines the default bit value of GPIO7. Also, the resistor determines the bit value if GPIO7 is connected to an external high-impedance circuit. For the resistor value, See Section 4.4 *DC Characteristics (3.3 V, 25 °C)*.

Table 2-11. Default Configuration of Strapping Pins

Strapping Pin	Default Configuration	Bit Value
GPIO6	Floating	–
GPIO7	Pull-up	1
MTMS	Floating	–
MTDI	Floating	–
MTDO	Floating	–

To change the bit values, the strapping pins should be connected to external pull-down/pull-up resistances. If the ESP32-C5 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.

Regarding the timing requirements for the strapping pins, there are such parameters as *setup time* and *hold time*. For more information, see Table 2-12 and Figure 2-5.

Table 2-12. 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

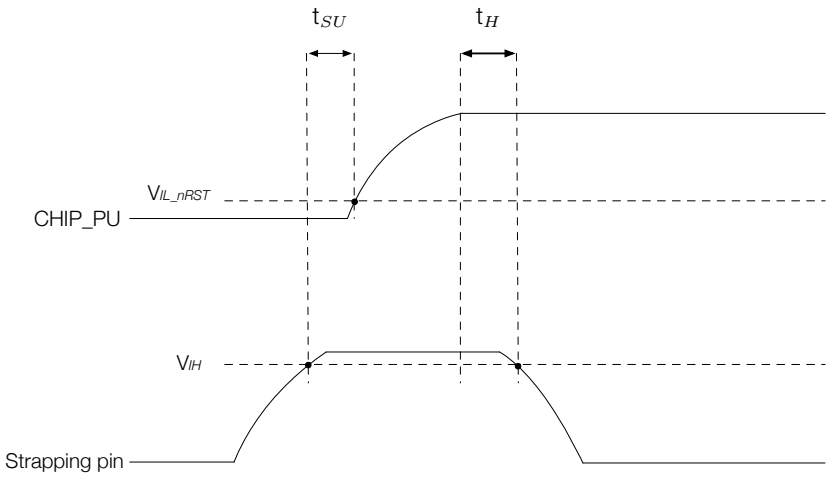


Figure 2-5. Visualization of Timing Parameters for the Strapping Pins

2.6.1 Chip Boot Mode Control

GPIO6 and GPIO7 control the boot mode after the reset is released. See Table 2-13 Boot Mode Control Boot Mode Control.

Table 2-13. Boot Mode Control

Boot Mode	GPIO6	GPIO7
Default Configuration	– (Floating)	1 (Pull-up)
SPI Boot (default)	Any value	1
Joint Download Boot	1	0

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

- USB-Serial-JTAG Download Boot
- UART Download Boot
- SDIO Download Boot

In SPI Boot mode, the ROM bootloader loads and executes the program from SPI flash to boot the system.

In Joint Download Boot mode, users can download binary files into flash using UART0, USB, or SDIO interfaces. It is also possible to download binary files into SRAM and execute it from SRAM.

In addition to SPI Boot and Joint Download Boot modes, ESP32-C5 also supports SPI Download Boot mode.

2.6.2 SDIO Sampling and Driving Clock Edge Control

The strapping pin MTMS and MTDI can be used to decide on which clock edge to sample signals and drive output lines. See Table 2-14 SDIO Input Sampling Edge/Output Driving Edge Control.

Table 2-14. SDIO Input Sampling Edge/Output Driving Edge Control

MTMS	MTDI	Edge behavior
– (Floating)	– (Floating)	Default Configuration
0	0	Falling edge sampling, falling edge output
0	1	Falling edge sampling, rising edge output
1	0	Rising edge sampling, falling edge output
1	1	Rising edge sampling, rising edge output

2.6.3 ROM Messages Printing Control

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

- **(Default) UART0 and USB Serial/JTAG controller**
- **UART0**
- **USB Serial/JTAG controller**

EFUSE_UART_PRINT_CONTROL and GPIO6 control ROM messages printing to **UART0** as shown in Table 2-15 ROM Message Printing Control.

Table 2-15. ROM Message Printing Control

Register[1]	eFuse[2]	GPIO6	ROM Message Printing
0	0 (0b00)	x[3]	ROM messages are always printed to UART0 during boot
	1 (0b01)	0	Print is enabled during boot
		1	Print is disabled during boot
	2 (0b10)	0	Print is disabled during boot
		1	Print is enabled during boot
	3 (0b11)	x	Print is disabled during boot
1	x	x	Print is disabled during boot

[1] Register: LP_AON_STORE4_REG[0]

[2] eFuse: EFUSE_UART_PRINT_CONTROL

[3] x: x indicates that the value has no effect on the result and can be ignored.

EFUSE_DIS_USB_SERIAL_JTAG_ROM_PRINT controls the printing to **USB Serial/JTAG controller**. When this bit is 1, printing to USB Serial/JTAG controller is disabled. When this bit is 0, and USB Serial/JTAG controller is enabled via EFUSE_DIS_USB_SERIAL_JTAG, ROM messages can be printed to USB Serial/JTAG controller.

2.6.4 JTAG Signal Source Control

The strapping pin MTDO can be used to control the source of JTAG signals during the early boot process. This pin does not have any internal pull resistors and the strapping value must be controlled by the external circuit that cannot be in a high impedance state.

As Table 2-16 shows, MTDO is used in combination with EFUSE_DIS_PAD_JTAG, EFUSE_DIS_USB_JTAG, and EFUSE_JTAG_SEL_ENABLE.

Table 2-16. JTAG Signal Source Control

eFuse 1[1]	eFuse 2[2]	eFuse 3[3]	MTDO	JTAG Signal Source
0	0	0	x[4]	USB Serial/JTAG Controller
		1	0	JTAG pins MTDI, MTCK, MTMS, and MTDO
			1	USB Serial/JTAG Controller
0	1	x	x	JTAG pins MTDI, MTCK, MTMS, and MTDO
1	0	x	x	USB Serial/JTAG Controller
1	1	x	x	JTAG is disabled

[1] eFuse 1: EFUSE_DIS_PAD_JTAG

[2] eFuse 2: EFUSE_DIS_USB_JTAG

[3] eFuse 3: EFUSE_JTAG_SEL_ENABLE

[4] x: x indicates that the value has no effect on the result and can be ignored.

2.7 Pin Mapping Between Chip and Flash

Table 2-17 lists the pin mapping between the chip and off-package flash for all SPI modes.

For chip variants with in-package flash (namely variants in QFN40 package, see Table 1-1 ESP32-C5 Series Comparison), the pins allocated for communication with in-package flash are not routed out, but you can take Table 2-17 as a reference.

For more information on SPI controllers, see also Section 3.4.2 Serial Peripheral Interface (SPI).

Notice:

It is not recommended to use the pins connected to flash for any other purposes.

Table 2-17. Pin Mapping Between QFN48 Chip and Off-package Flash

QFN40 Pin No.	Pin Name	Single SPI Flash	Dual SPI Flash	Quad SPI Flash
35	SPICLK	CLK	CLK	CLK
30	SPICS0	CS#	CS#	CS#
36	SPID	MOSI	SIO0	SIO0
31	SPIQ	MISO	SIO1	SIO1
32	SPIWP	WP#		SIO2
34	SPIHD	HOLD#		SIO3

[1] SIO: Serial Data Input and Output

3 Functional Description

This chapter describes the functions of ESP32-C5.

3.1 CPU and Memory

3.1.1 HP CPU

ESP32-C5 has a high-performance (HP) 32-bit RISC-V single-core processor with the following features:

- five-stage pipeline that supports clock frequency of up to 240 MHz
- RV32IMAC ISA (instruction set architecture)
- two-cycle pipelined multiplier and radix-4 SRT divider
- Zc extensions (Zcb, Zcmp, and Zcmt)
- custom hardware loop instructions (Xhwlp)
- compliant with RISC-V Core Local Interrupt (CLINT)
- compliant with RISC-V Core-Local Interrupt Controller (CLIC)
- branch predictor BHT, BTB, and RAS
- up to 3 hardware breakpoints/watchpoints
- up to 16 PMP/PMA regions
- Machine and User privilege modes
- USB/JTAG for debugging
- compliant with RISC-V debug specification v0.13
- offline trace debug that is compliant with RISC-V Trace Specification v2.0

3.1.2 LP CPU

ESP32-C5 integrates a low-power (LP) 32-bit RISC-V processor. This LP CPU is designed as a simplified, low-power replacement of HP CPU in sleep modes. It can be also used to supplement the functions of the HP CPU in normal working mode. The LP CPU and LP memory remain powered on in Deep-sleep mode. Hence, the developer can store a program for the LP CPU in the LP memory to access LP IO, LP peripherals, and real-time timers in Deep-sleep mode.

LP CPU has the following features:

- two-stage pipeline that supports a clock frequency of up to 40 MHz
- RV32IMAC ISA (instruction set architecture)
- 3-4 cycle multiplier and iterative divider
- support for custom vectored interrupts
- up to 2 hardware breakpoints/watchpoints
- JTAG for debugging

- compliant with RISC-V debug specification v0.13
- boot by the CPU, its dedicated timer, or LP IO

3.1.3 Internal Memory

ESP32-C5's internal memory includes:

- **320 KB ROM**: for booting and core functions
- **HP memory**: 512 KB of SRAM for data and instructions
- **LP memory**: 16 KB of SRAM that can be accessed by HP CPU or LP CPU. It can retain data in Deep-sleep mode
- **4 Kbit of eFuse**: 1792 bits are reserved for your data, such as encryption key and device ID
- **In-package flash**: See details in Chapter 1 *ESP32-C5 Series Comparison*

3.1.4 Off-Package Flash

ESP32-C5 supports SPI, Dual SPI, Quad SPI, and QPI interfaces that allow connection to multiple flash outside the chip's package.

CPU's instruction memory space and read-only data memory space can map into off-package flash of ESP32-C5, whose size can be 16 MB at most. ESP32-C5 supports hardware encryption/decryption based on XTS-AES to protect developers' programs and data in flash.

Through high-speed caches, ESP32-C5 can support at a time up to:

- 16 MB of instruction memory space which can map into flash as individual blocks of 64 KB. 32-bit fetch is supported
- 16 MB of data memory space which can map into flash as individual blocks of 64 KB. 8-bit, 16-bit and 32-bit reads are supported

Note:
After ESP32-C5 is initialized, software can customize the mapping of off-package flash into the CPU address space.

3.1.5 Address Mapping Structure

Figure 3-1 shows the address mapping structure of ESP32-C5.

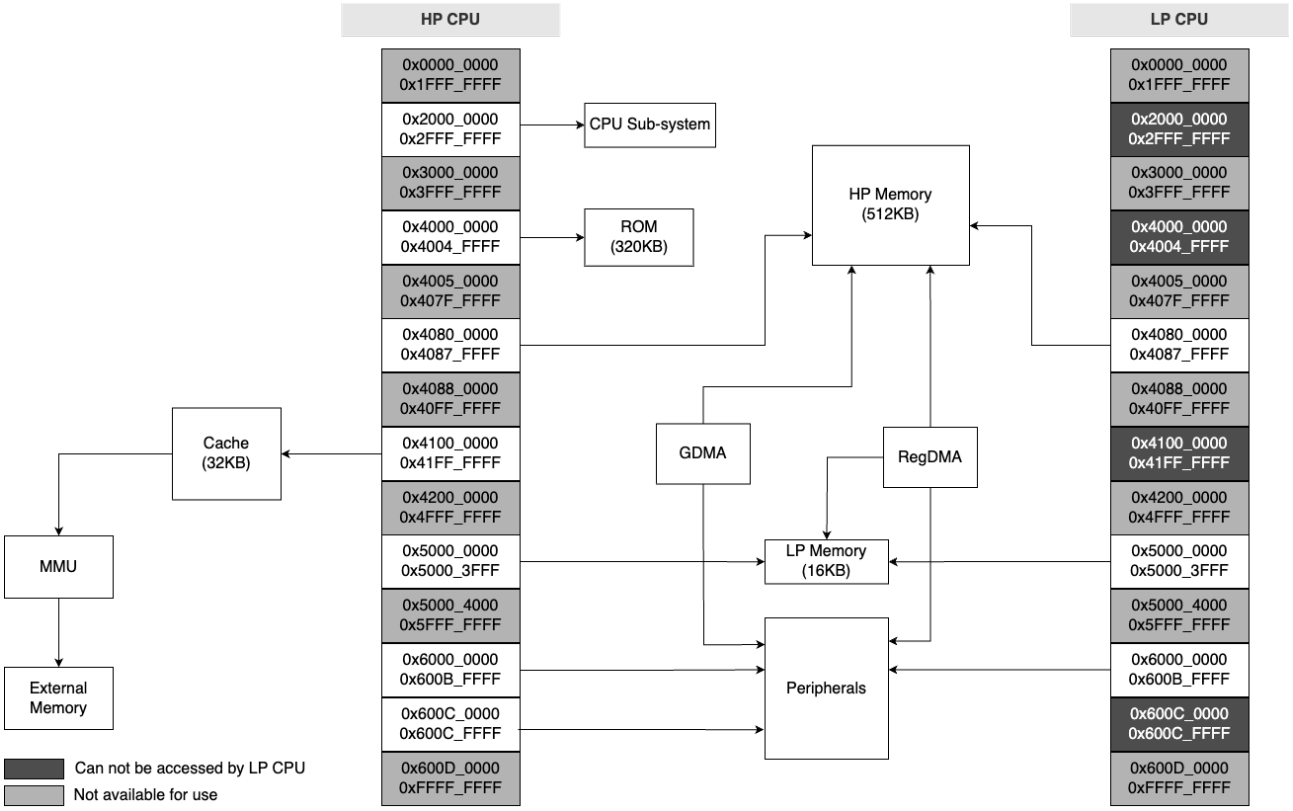


Figure 3-1. Address Mapping Structure

3.1.6 Cache

ESP32-C5 has an eight-way set associative cache. This cache is read-only and has the following features:

- size: 32 KB
- pre-load function
- lock function
- critical word first and early restart

3.1.7 eFuse Controller

ESP32-C5 contains a 4-Kbit eFuse to store parameters, which are burned and read by an eFuse controller. The eFuse controller has the following features:

- 4 Kbits in total, with 1792 bits reserved for users, e.g., encryption key and device ID
- one-time programmable storage
- configurable write protection
- configurable read protection
- various hardware encoding schemes to protect against data corruption

3.1.8 Timeout Protection

ESP32-C5 integrates 3 timeout protection modules in CPU peripherals, HP peripherals, and LP peripherals, respectively, against bus being stuck. These modules have the following features:

- integrate a 16-bit timeout counter for each timeout period
- support for interrupts
- exception records

3.2 System Clocks

3.2.1 CPU Clock

The CPU clock has three possible sources:

- external main crystal clock
- internal fast RC oscillator clock (typically about 20 MHz, and adjustable)
- PLL clock

The application can select the clock source from the three clocks above. The selected clock source drives the CPU clock directly, or after division, depending on the application. Once the CPU is reset, the default clock source would be the external main crystal clock divided by 1.

Note:
ESP32-C5 is unable to operate without an external main crystal clock.

3.2.2 Low-Power Clocks

The LP slow clock is used for RTC counter, RTC watchdog and the power management unit (PMU). It has four possible sources:

- internal low-speed RC oscillator (typically about 32 kHz, and adjustable)
- internal slow RC oscillator (typically about 150 kHz, and adjustable)
- external low-speed (32 kHz) crystal clock
- external IO clock (external clock source connected with digital IO)

The LP fast clock is used for low-power peripherals and sensor controllers. It has two possible sources:

- external main crystal clock divided by 2
- internal fast RC oscillator clock (typically about 20 MHz, and adjustable)

3.3 Analog Peripherals

3.3.1 Analog-to-Digital Converter (ADC)

ESP32-C5 integrates a 12-bit SAR ADC and supports measurements on 6 channels (analog-enabled pins).

For GPIOs assigned to ADC, please refer to Table 3-1.

3.3.2 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 –40 °C to 125 °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 temperature.

3.3.3 Analog PAD Voltage Comparator

ESP32-C5 provides a group of analog PAD voltage comparators which contain two PADs. This peripheral can be used to compare the voltages of the two PADs or compare the voltage of one PAD with a stable internal voltage that can be adjustable.

For GPIOs assigned to the analog PAD voltage comparators, please refer to Table 3-1.

3.3.4 Brownout Detector

With the brownout detector, ESP32-C5 can monitor voltages of power supply pins and trigger an interrupt or reset when voltages are abnormal.

3.4 Digital Peripherals

3.4.1 Universal Asynchronous Receiver Transmitter (UART)

ESP32-C5 has three UART interfaces, i.e. UART0, UART1 and LP UART. All the three interfaces provide hardware flow control (CTS and RTS signals) and software flow control (XON and XOFF).

UART0 and UART 1 support IrDA and asynchronous communication (RS232 and RS485) at a speed of up to 5 Mbps. UART0 and UART1 connect to GDMA via UHCI0 interface (i.e. Universal Host Controller Interface), and can be accessed by the GDMA controller or directly by the CPU.

LP UART only supports asynchronous communication (RS232) at a speed of up to 1.25 Mbps. LP UART can only by accessed by the CPU.

For GPIOs assigned to UART, please refer to Table 3-1.

3.4.2 Serial Peripheral Interface (SPI)

ESP32-C5 features three SPI interfaces (SPI0, SPI1, and SPI2). SPI0 and SPI1 can be configured to operate in SPI memory mode, while SPI2 can be configured to operate in general-purpose SPI mode.

- **SPI Memory mode**

  In SPI memory mode, SPI0 and SPI1 interface with external SPI memory. Data are transferred in unit of byte. Up to four-line STR reads and writes are supported. The clock frequency is configurable to a maximum of 120 MHz.

- **SPI2 General-purpose SPI (GP-SPI) mode**

SPI2 can operate in master and slave modes. SPI2 supports two-line full-duplex communication and single-/two-/four-line half-duplex communication in both master and slave 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 SPI2 interface can connect to GDMA.

  - In master mode, the clock frequency is 80 MHz at most, and the four modes of SPI transfer format are supported.
  - In slave mode, the clock frequency is 60 MHz at most, and the four modes of SPI transfer format are also supported.

For the recommended pin mapping between ESP32-C5 and off-package flash, please see Table 2-17 *Pin Mapping Between QFN48 Chip and Off-package Flash*.

For GPIOs assigned to SPI, please refer to Table 3-1.

3.4.3 I2C Interface

ESP32-C5 has an I2C and a LP I2C bus interfaces. I2C is used for I2C master mode or slave mode, depending on your configuration, while LP I2C is always in master mode. Both interfaces support:

- standard mode (100 Kbit/s)
- fast mode (400 Kbit/s)
- up to 800 Kbit/s (constrained by SCL and SDA pull-up strength)
- 7-bit and 10-bit addressing mode
- double addressing mode
- 7-bit broadcast address

You can configure instruction registers to control the I2C interface for more flexibility.

For GPIOs assigned to I2C, please refer to Table 3-1.

3.4.4 I2S Interface

ESP32-C5 includes a standard I2S interface. This interface can operate as a master or a slave in full-duplex mode or half-duplex mode, and supports 8-bit, 16-bit, 24-bit, or 32-bit serial communication. BCK clock frequency, from 10 kHz up to 40 MHz, is supported.

The I2S interface supports TDM Philips, TDM MSB alignment, TDM PCM standard, PDM standard, and PCM-to-PDM TX interface. It connects to the GDMA controller.

For GPIOs assigned to I2S, please refer to Table 3-1.

3.4.5 Pulse Count Controller (PCNT)

The pulse count controller (PCNT) in ESP32-C5 captures pulses and counts pulse edges in seven modes. It has the following features:

- four independent pulse counters (units) that count from 1 to 65535
- each unit consists of two independent channels sharing one pulse counter

- all channels have input pulse signals (e.g. sig_ch0_u*n*) with their corresponding control signals (e.g. ctrl_ch0_u*n*)
- independently filter glitches of input pulse signals (sig_ch0_u*n* and sig_ch1_u*n*) and control signals (ctrl_ch0_u*n* and ctrl_ch1_u*n*) on each unit
- each channel has the following parameters:
  1. selection between counting on positive or negative edges of the input pulse signal
  2. configuration to Increment, Decrement, or Disable counter mode for control signal's high and low states
- Maximum frequency of pulses: 40 MHz

For GPIOs assigned to PCNT, please refer to Table 3-1.

3.4.6 USB Serial/JTAG Controller

ESP32-C5 integrates a USB Serial/JTAG controller. This controller has the following features:

- USB 2.0 full speed compliant, capable of up to 12 Mbit/s transfer speed (note that this controller does not support the faster 480 Mbit/s high-speed transfer mode)
- CDC-ACM virtual serial port and JTAG adapter functionality
- Programming the chip's flash
- CPU debugging with compact JTAG instructions
- A full-speed USB PHY integrated in the chip

For GPIOs assigned to USB Serial/JTAG, please refer to Table 3-1.

3.4.7 USB 2.0 OTG High-Speed Interface

ESP32-C5 features a high-speed USB OTG interface along with an integrated transceiver. The USB OTG interface complies with the USB 2.0 specification. It has the following features:

General Features

- compatible with USB 2.0, OTG 1.3, and OTG 2.0
- high-speed and full-speed data rates
- host negotiation protocol (HNP) and session request protocol (SRP) as A-device or B-device
- dynamic FIFO (DFIFO) sizing up to 4 KB
- multiple memory access modes
  - Scatter/Gather DMA mode
  - Buffer DMA mode
- integrated UTMI high-speed transceiver
- remote wake-up

Device Mode Features

- Endpoint 0 always existing (bi-directional, consisting of EP0 IN and EP0 OUT)

- 15 additional endpoints (Endpoint 1 ~ 15), configurable as IN or OUT
- maximum of eight IN endpoints concurrently active at any time (including EP0 IN)
- all OUT endpoints share a single RX FIFO
- each IN endpoint has a dedicated TX FIFO

Host Mode Features

- 16 channels
  - a control channel consists of two channels (IN and OUT), as IN and OUT transactions must be handled separately. Only the Control transfer type is supported.
  - each of the other 15 channels is dynamically configurable to be IN or OUT, and supports Bulk, Isochronous, and Interrupt transfer types.
- all channels share an RX FIFO, non-periodic TX FIFO, and periodic TX FIFO. The size of each FIFO is configurable.

For GPIOs assigned to USB 2.0 OTG High-Speed Interface, please refer to Table 3-1.

3.4.8 TWAI® Controller

ESP32-C5 has two TWAI® controllers 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 (no acknowledgment required)
- 64-byte receive FIFO
- acceptance filter (single and dual filter modes)
- error detection and handling: error counters, configurable error warning limit, error code capture, arbitration lost capture, automatic transceiver standby

For GPIOs assigned to TWAI®, please refer to Table 3-1.

3.4.9 SDIO 2.0 Slave Controller

ESP32-C5 integrates an SD device interface that conforms to the industry-standard SDIO Specification Version 2.0, and allows a host controller to access the SoC, using the SDIO bus interface and protocol. The host can access the registers of the SDIO interface directly and the shared memory via a DMA engine, thus maximizing performance without engaging the processor cores.

The SDIO 2.0 Slave Controller supports the following features:

- clock range: 0 to 50 MHz
- SPI, 1-bit SDIO, and 4-bit SDIO transfer modes
- configurable sampling and driving clock edges
- special registers for direct access by host

- interrupting host to initiate data transfer
- automatic loading of SDIO bus data and automatic discarding of padding data
- block size of up to 512 bytes
- interrupt vectors between the host and the slave, allowing both to interrupt each other
- supports DMA for data transfer

For GPIOs assigned to SDIO, please refer to Table 3-1.

3.4.10 LED PWM Controller

The LED PWM controller can generate independent digital waveform on six channels. The LED PWM controller:

- can generate digital waveform with configurable periods and duty cycle. The resolution of duty cycle can be up to 20 bits
- has multiple clock sources, including 80 MHz PLL clock, external main crystal clock, and internal fast RC oscillator
- can operate when the CPU is in low-power mode (Light-sleep mode)
- supports gradual increase or decrease of duty cycle, which is useful for the LED RGB color-gradient generator
- up to 16 duty cycle ranges for each PWM generator to generate gamma curve signals - each range can be independently configured in terms of fading direction (increase or decrease), fading amount (the amount by which the duty cycle increases or decreases each time), the number of fades (how many times the duty cycle fades in one range), and fading frequency

For GPIOs assigned to LED PWM, please refer to Table 3-1.

3.4.11 Motor Control PWM (MCPWM)

ESP32-C5 integrates a MCPWM that can be used to drive digital motors and smart light. This controller has a clock divider (prescaler), three PWM timers, three PWM operators, and a dedicated capture submodule.

PWM timers are used to generate timing references. The PWM operators generate desired waveform based on the timing references. By configuration, a PWM operator can use the timing reference of any PWM timer, and use the same timing reference with other PWM operators. PWM operators can also use different PWM timers' values to produce independent PWM signals. PWM timers can be synchronized.

For GPIOs assigned to MCPWM, please refer to Table 3-1.

3.4.12 Remote Control Peripheral (RMT)

The Remote Control Peripheral (RMT) supports two channels of infrared remote transmission and two channels of infrared remote reception. By controlling pulse waveform through software, it supports various infrared and other single wire protocols. All four channels share a 192 × 32-bit memory block to store transmit or receive waveform.

For GPIOs assigned to RMT, please refer to Table 3-1.

3.4.13 Parallel IO (PARLIO) Controller

ESP32-C5 integrates a PARLIO controller for parallel data transfer. It has a transmitter and a receiver, connected with the GDMA controller. In full-duplex mode the PARLIO controller supports up to 4-bit parallel data transfer, while in half-duplex mode it supports up to 8-bit parallel data transfer.

The PARLIO controller has the following features:

- multiple clock sources and clock division, with clock frequency up to 40 MHz
- clock edge sampling
- 1/2/4/8-bit data transfer
- changeable sample sequence for data to be transmitted and received in 1-bit, 2-bit, and 4-bit mode
- support for multiple data sampling mode by the receiver
- support for multiple EOF signal generation modes by the receiver
- support for transmitter clock gating

For GPIOs assigned to PARLIO, please refer to Table 3-1.

3.4.14 General DMA Controller (GDMA)

ESP32-C5 has a general DMA controller (GDMA) with three transmit channels and three receive channels. These six channels are shared by peripherals with DMA feature. The GDMA controller implements a fixed-priority scheme among these channels.

The GDMA controller controls data transfer using linked lists. It allows peripheral-to-memory and memory-to-memory data transfer at a high speed. All channels can access internal RAM.

Peripherals on ESP32-C5 with DMA feature are SPI2, UHCI0, I2S, AES, SHA, ADC, and PARLIO.

3.4.15 Bit-Scrambler Controller

The bit-scrambler controller includes transmit and receive channels, located in the transmit and receive paths of GDMA, respectively. The bit-scrambler controller can be used for endianness conversion, adding/deleting data, formatting data stream, etc. In addition, lookup tables can be used to implement functionalities such as generating complex waveforms, ADC curve correction, etc.

Its transmit and receive channels can work independently. Each channel supports six instructions and contains the instruction cache with a depth of eight and a 2048-byte lookup table. The six instructions can be programmable by users, and the controller processes the data stream in bits according to a user-defined instruction sequence.

3.4.16 Event Task Matrix (ETM)

ESP32-C5 integrates a SOC ETM with multiple channels. Each input event on channels is mapped to an output task. Events are generated by peripherals, while tasks are received by peripherals. The SOC ETM has the following features:

- up to 50 mapping channels, each connected to an event and a task and controlled independently

- an event or a task can be mapped to any tasks or events in the matrix. That is to say, one event can be mapped to different tasks via multiple channels, or different events can be mapped to the same task via their individual channels
- peripherals supporting ETM include GPIO, LED PWM, general-purpose timers, RTC Timer, system timer, MCPWM, temperature sensor, ADC, I2S, LP CPU, GDMA, and PMU

3.5 Radio

The ESP32-C5 radio consists of the following blocks:

- 2.4 & 5 GHz receiver
- 2.4 & 5 GHz transmitter
- bias and regulators
- balun and transmit-receive switch
- clock generator

3.5.1 2.4 & 5 GHz Receiver

The 2.4 & 5 GHz receiver demodulates the 2.4 & 5 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, ESP32-C5 integrates RF filters, Automatic Gain Control (AGC), DC offset cancelation circuits, and baseband filters.

3.5.2 2.4 & 5 GHz Transmitter

The 2.4 & 5 GHz transmitter modulates the quadrature baseband signals to the 2.4 & 5 GHz RF signal, and drives the antenna with a high-powered 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.

3.5.3 Clock Generator

The clock generator produces quadrature clock signals of 2.4 & 5 GHz for both the receiver and the transmitter. All components of the clock generator are integrated into the chip, including 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.

3.6 Wi-Fi

3.6.1 Wi-Fi Radio and Baseband

The ESP32-C5 Wi-Fi radio and baseband support the following features:

- compliant with IEEE 802.11b/g/n/ac/ax
- 1T1R in 2.4 GHz and 5 GHz dual band
- 802.11ax
  - 20 MHz-only non-AP mode
  - MCS0 ~ MCS9
  - uplink and downlink OFDMA
  - downlink MU-MIMO (multi-user, multiple input, multiple output)
  - longer OFDM symbol, with 0.8, 1.6, 3.2 μs guard interval
  - DCM (dual carrier modulation), up to 16-QAM
  - single-user/multi-user beamformee
  - channel quality indication (CQI)
  - RX STBC (single spatial stream)
- 802.11ac
  - MCS0 ~ MCS7 that support 20 MHz bandwidth
- 802.11b/g/n
  - MCS0 ~ MCS7 that support 20 MHz and 40 MHz bandwidth
  - MCS32
  - data rate up to 150 Mbps
  - 0.4 μs guard interval
- adjustable transmitting power
- antenna diversity
  ESP32-C5 supports antenna diversity with an external RF switch. This switch is controlled by one or more GPIOs, and used to select the best antenna to minimize the effects of channel imperfections.

3.6.2 Wi-Fi MAC

ESP32-C5 implements the full IEEE 802.11 b/g/n/ac/ax Wi-Fi MAC protocol. It supports the Basic Service Set (BSS) STA and SoftAP operations under the Enhanced Distributed Channel Access (EDCA). Power management is handled automatically with minimal host interaction to minimize the active duty period.

The ESP32-C5 Wi-Fi MAC applies the following low-level protocol functions automatically:

- 4 × virtual Wi-Fi interfaces
- infrastructure BSS in Station mode, SoftAP mode, Station + SoftAP mode, and promiscuous mode
- RTS protection, CTS-to-Self protection, Immediate Block ACK
- fragmentation and defragmentation
- TX/RX A-MPDU, TX/RX A-MSDU
- transmit opportunity (TXOP)
- Wi-Fi multimedia (WMM)
- GCMP, CCMP, TKIP, WAPI, WEP, BIP, WPA2-PSK, and WPA3-PSK
- automatic beacon monitoring (hardware TSF)
- 802.11mc FTM
- 802.11ax supports:
  - target wake time (TWT) requester
  - multiple BSSIDs
  - triggered response scheduling
  - uplink power headroom
  - operating mode
  - buffer status report
  - Multi-user Request-to-Send (MU-RTS), Multi-user Block ACK Request (MU-BAR), and Multi-STA Block ACK (M-BA) frame
  - intra-PPDU power saving mechanism
  - two network allocation vectors (NAV)
  - BSS coloring
  - spatial reuse
  - uplink power headroom
  - operating mode control
  - buffer status report
  - TXOP duration RTS threshold
  - UL-OFDMA random access (UORA)

3.6.3 Networking Features

Espressif provides libraries for TCP/IP networking, ESP-WIFI-MESH networking, and other networking protocols over Wi-Fi. TLS 1.0, 1.1, and 1.2 is also supported.

3.7 Bluetooth LE

ESP32-C5 includes a Bluetooth Low Energy subsystem that integrates a hardware link controller, an RF/modem block and a feature-rich software protocol stack. It supports the core features of Bluetooth 5 and Bluetooth mesh.

3.7.1 Bluetooth LE PHY

Bluetooth Low Energy PHY in ESP32-C5 supports:

- 1 Mbps PHY
- 2 Mbps PHY for higher data rates
- coded PHY for longer range (125 Kbps and 500 Kbps)
- HW listen before talk (LBT)

3.7.2 Bluetooth LE Link Controller

Bluetooth Low Energy Link Controller in ESP32-C5 supports:

- LE advertising extensions, to enhance broadcasting capacity and broadcast more intelligent data
- multiple advertisement sets
- simultaneous advertising and scanning
- multiple connections in simultaneous central and peripheral roles
- adaptive frequency hopping and channel assessment
- LE channel selection algorithm #2
- LE power control
- connection parameter update
- high duty cycle non-connectable advertising
- LE privacy 1.2
- LE data packet length extension
- link layer extended scanner filter policies
- low duty cycle directed advertising
- link layer encryption
- LE Ping

3.8 802.15.4

ESP32-C5 includes an IEEE Standard 802.15.4 subsystem that integrates PHY and MAC layer. It supports various software stacks including Thread, Zigbee, Matter, HomeKit, and MQTT.

3.8.1 802.15.4 PHY

ESP32-C5's 802.15.4 PHY supports:

- O-QPSK PHY in 2.4 GHz
- 250 Kbps data rate
- RSSI and LQI supported

3.8.2 802.15.4 MAC

ESP32-C5 supports most key features defined in IEEE Standard 802.15.4-2015, including:

- CSMA/CA
- active scan and energy detect
- HW frame filter
- HW auto acknowledge
- HW auto frame pending
- coordinated sampled listening (CSL)

3.9 Low Power Management

With the use of advanced power-management technologies, ESP32-C5 can switch between different power modes. ESP32-C5 supports:

- 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 frequency can be reduced. Wi-Fi base band and radio are disabled, but Wi-Fi connection can remain active.
- Light-sleep mode: The CPU is paused. Any wake-up events (wireless power management module, SDIO host, RTC timer, or external interrupts) will wake up the chip. Wi-Fi base band and radio are disabled, but Wi-Fi connection can remain active. Users can disable the CPU and most peripherals except SRAM and wireless power management module (as shown in ESP32-C5 Functional Block Diagram) to further reduce current consumption.
- Deep-sleep mode: CPU, SRAM, and most peripherals are powered down. Only the LP memory is powered on. LP peripheral states can be configured. Wi-Fi connection data are stored in the LP memory. The LP CPU is operational.

3.10 Timers

3.10.1 System Timer

ESP32-C5 integrates a 52-bit system timer, which has two 52-bit counters and three comparators. The system timer has the following features:

- counters with an average clock frequency of 16 MHz
- three types of independent interrupts generated according to alarm value

- two alarm modes: target mode and period mode
- 52-bit alarm values and 26-bit alarm periods
- automatic reload of counter value
- counters can be stalled if the CPU is stalled or in OCD mode
- real-time alarm events

3.10.2 General Purpose Timers

ESP32-C5 is embedded with two 54-bit general-purpose timers, which are based on 16-bit prescalers and 54-bit auto-reload-capable up/down-timers.

The timers' features are summarized as follows:

- a 16-bit clock prescaler, from 2 to 65536
- a 54-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
- level interrupt generation
- real-time alarm events
- tasks in response to ETM, including enable and disable timers, enable alarms, read the timer's real-time values, reload the timer's values

3.10.3 Watchdog Timers

Digital Watchdog Timers

The ESP32-C5 contains three digital watchdog timers: one in each of the two timer groups (called Main System Watchdog Timers, or MWDT) and one in the low-power system (called the RTC Watchdog Timer, or RWDT).

During the flash boot process, RWDT and the MWDT in timer group 0 (TIMG0) 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, enabled and disabled separately
- interrupt, CPU reset, or core reset for MWDT upon expiry of each stage; interrupt, CPU reset, core reset, or system reset for RWDT 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.

Analog Watchdog Timer

The ESP32-C5 also has one analog watchdog timer: RTC super watchdog timer (SWD). SWD contains a watchdog circuit that needs to be fed for at least once during its timeout period, which is slightly less than one second. About 100 ms before watchdog timeout, it will also send out a WD_INTR signal as a request to remind the system to feed the watchdog.

If the system does not respond to SWD feed request and watchdog finally times out, SWD will generate a system level signal SWD_RSTB to reset whole digital circuits on the chip (system reset).

The source of the clock for SWD is constant and can not be selected.

SWD has the following features:

- ultra-low power
- interrupt to indicate that the SWD is about to time out
- various dedicated methods for software to feed SWD, which enables SWD to monitor the working state of the whole operating system

3.10.4 RTC Timer

ESP32-C5 incorporates a 48-bit always-on RTC timer, which offers support for two sets of programmable timeout values and can send a timeout interrupt or a wake-up signal.

3.11 Cryptography/Security Components

3.11.1 AES Accelerator (AES)

ESP32-C5 integrates an Advanced Encryption Standard (AES) accelerator, which is a hardware device that speeds up computation using AES algorithm significantly, compared to AES algorithms implemented solely in software. The AES accelerator integrated in ESP32-C5 has two working modes, which are Typical AES and DMA-AES.

Main features are as follows:

- typical AES working mode
  - AES-128/AES-256 encryption and decryption, compliant with NIST FIPS 197
- DMA-AES working mode
  - AES-128/AES-256 encryption and decryption, compliant with NIST FIPS 197
  - Block cipher mode, compliant with NIST SP 800-38A
    - ECB (Electronic Codebook)
    - CBC (Cipher Block Chaining)
    - OFB (Output Feedback)
    - CTR (Counter)
    - CFB8 (8-bit Cipher Feedback)
    - CFB128 (128-bit Cipher Feedback)

  - interrupt generation

3.11.2 SHA Accelerator (SHA)

ESP32-C5 integrates an SHA accelerator, which is a hardware device that speeds up the SHA algorithm significantly, compared to SHA algorithms implemented solely in software. The SHA accelerator integrated in ESP32-C5 has two working modes, which are Typical SHA and DMA-SHA.

Main features are as follows:

- the following hash algorithms introduced in FIPS PUB 180-4
  - SHA-1
  - SHA-224
  - SHA-256
- two working modes
  - typical SHA
  - DMA-SHA
- interleaved function in Typical SHA working mode
- interrupt function in DMA-SHA working mode

3.11.3 RSA Accelerator (RSA)

The RSA accelerator provides hardware support for high-precision computation used in various RSA asymmetric cipher algorithms, significantly improving their run time and reducing their software complexity. Compared with RSA algorithms implemented solely in software, this hardware accelerator can speed up RSA algorithms significantly. The RSA accelerator also supports operands of different lengths, which provides more flexibility during the computation.

Main features are as follows:

- large-number modular exponentiation with two optional acceleration options
- large-number modular multiplication, up to 3072 bits
- large-number multiplication, with operands up to 1536 bits
- operands of different lengths
- interrupt on completion of computation

3.11.4 ECC Accelerator (ECC)

Elliptic Curve Cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves. ECC allows smaller keys compared to RSA cryptography while providing equivalent security.

ESP32-C5's ECC Accelerator can complete various calculations based on different elliptic curves, thus accelerating the ECC algorithm and ECC-derived algorithms (such as ECDSA).

Main features are as follows:

- two elliptic curves, namely P-192 and P-256 defined in FIPS 186-3
- two coordinate systems, namely Affine Coordinates and Jacobian Coordinates
- different point operations, including point addition, point multiplication, and point verification
- different modular operations based on the order or mod base of the curve, including mod addition, mod subtraction, mod multiplication, and mod division
- interrupt upon completion of calculation

3.11.5 HMAC Accelerator (HMAC)

The Hash-based Message Authentication Code (HMAC) module computes Message Authentication Codes (MACs) using Hash algorithm SHA-256 and keys as described in RFC 2104. The 256-bit HMAC key is stored in an eFuse key block and can be set as read-protected, i. e., the key is not accessible from outside the HMAC accelerator.

Main features are as follows:

- standard HMAC-SHA-256 algorithm
- Hash result only accessible by configurable hardware peripheral (in downstream mode)
- compatibility with challenge-response authentication algorithm
- required keys for the Digital Signature (DS) peripheral (in downstream mode)
- re-enabled soft-disabled JTAG (in downstream mode)

3.11.6 Digital Signature (DS)

A Digital Signature (DS) is used to verify the authenticity and integrity of a message using a cryptographic algorithm. This can be used to validate a device's identity to a server, or to check the integrity of a message.

ESP32-C5 includes a Digital Signature (DS) module providing hardware acceleration of messages' signatures based on RSA. HMAC is used as the key derivation function to output the DS_KEY key using eFuse as the input key. Subsequently, the DS module uses DS_KEY to decrypt the pre-encrypted parameters and calculate the signature. The whole process happens in hardware so that neither the decryption key for the RSA parameters nor the input key for the HMAC key derivation function can be seen by users while calculating the signature.

Main features are as follows:

- RSA digital signatures with key length up to 3072 bits
- encrypted private key data, only decryptable by DS module
- SHA-256 digest to protect private key data against tampering by an attacker

3.11.7 Elliptic Curve Digital Signature Algorithm (ECDSA)

In cryptography, the Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA) which uses elliptic-curve cryptography.

ESP32-C5's ECDSA accelerator provides a secure and efficient environment for computing ECDSA signatures. It offers fast computations while ensuring the confidentiality of the signing process to prevent information leakage.

This makes it a valuable tool for applications that require high-speed cryptographic operations with strong security guarantees. By using the ECDSA accelerator, users can be confident that their data is being protected without sacrificing performance.

Main features are as follows:

- digital signature generation and verification
- two elliptic curves, namely P-192 and P-256 defined in FIPS 186-3
- two hash algorithms for message hash in the ECDSA operation, namely SHA-224 and SHA-256 defined in FIPS PUB 180-4
- dynamic access permission in different operation statuses to ensure information security, preventing key leakage due to intermediate data leakage

3.11.8 External Memory Encryption and Decryption (XTS_AES)

The ESP32-C5 integrates an External Memory Encryption and Decryption module that complies with the XTS-AES standard algorithm specified in IEEE Std 1619-2007, providing security for users' application code and data stored in the external memory (flash). Users can store proprietary firmware and sensitive data (e.g., credentials for gaining access to a private network) to the off-package flash.

Main features are as follows:

- general XTS-AES algorithm, compliant with IEEE Std 1619-2007
- software-based manual encryption
- high-speed auto decryption without software
- encryption and decryption functions jointly enabled/disabled by registers configuration, eFuse parameters, and boot mode
- configurable Anti-DPA

3.11.9 Secure Boot

ESP32-C5 integrates a hardware Secure Boot (V2) verification scheme based on either RSA-PSS or ECDSA algorithm. Secure Boot feature ensures that only authenticated software signed by a trusted entity can execute on ESP32-C5.

Main features are as follows:

- RSA-PSS (3072-bit key) or ECDSA P192/P256 curve-based signing scheme
- up to 3 secure signing keys are supported with independent digest slots in the eFuse
- key revocation through eFuse

3.11.10 Random Number Generator (RNG)

The ESP32-C5 contains a true random number generator, which generates 32-bit random numbers that can be used for cryptographical operations, among other things.

The random number generator in ESP32-C5 generates true random numbers, which means random numbers generated from a physical process, rather than by means of an algorithm. No number generated within the specified range is more or less likely to appear than any other number.

3.11.11 Key Manager

ESP32-C5 stores and deploys keys with the Key Manager as the security core. Key Manager uses the unique physically unclonable function (PUF) of each chip to generate the hardware unique key (HUK) which is unique to the chip and serves as the root of trust (RoT) for the chip. HUK is automatically generated each time the chip is powered on and disappears when the chip is powered off. In this way, Key Manager secures key storage and deployment.

Key Manager of ESP32-C5 stores key information (non-plaintext information for recovering the key) in the external memory, realizing flexible key management functionalities such as unlimited key storage and dynamic key switching.

3.11.12 Access Permission Management (APM)

ESP32-C5 integrates an APM module to manage access permissions.

Main features are as follows:

- DMA APM supporting 32 regions with configurable addresses
- APB APM supporting 2 regions with configurable addresses
- APB access assigning independent APM for each peripheral address range based on region control
- managing APB access permissions for HP CORE0, HP CORE1 and LP CORE independently
- managing APB access permissions for User Mode and Machine Mode independently
- exception records

3.11.13 Trusted Execution Environment (TEE) Controller

The TEE controller facilitates four security modes for masters within the system. These modes grant varying access privileges to hardware resources. Collaborating with the APM controller, the TEE determines access entitlements, ensuring security across different modes.

3.12 Peripheral Pin Configurations

Table 3-1. Peripheral Pin Configurations

Interface	Signal	Pin	Function
ADC	ADC1_CH0	XTAL_32K_N	12-bit SAR ADC
	ADC1_CH1	MTMS	
	ADC1_CH2	MTDI	
	ADC1_CH3	MTCK	
	ADC1_CH4	MTDO	
	ADC1_CH5	GPIO6	
JTAG	MTDI	MTDI	JTAG for software debugging
	MTCK	MTCK	
	MTMS	MTMS	
	MTDO	MTDO	
UART	U0RXD_in	Any GPIO pins	Two UART channels with hardware flow control and GDMA
	U0CTS_in		
	U0DSR_in		
	U0TXD_out		
	U0RTS_out		
	U0DTR_out		
	U1RXD_in		
	U1CTS_in		
	U1DSR_in		
	U1TXD_out		
	U1RTS_out		
	U1DTR_out		
LP UART	LP_UART_DTRN	XTAL_32K_P	One LP UART channel with hardware flow control and GDMA
	LP_UART_DSRN	XTAL_32K_N	
	LP_UART_RTSN	MTMS	
	LP_UART_CTSN	MTDI	
	LP_UART_RXD	MTCK	
	LP_UART_TXD	MTDO	
I2C	I2CEXT0_SCL_in	Any GPIO pins	One I2C channel in slave or master mode
	I2CEXT0_SDA_in		
	I2CEXT0_SCL_out		
	I2CEXT0_SDA_out		
LP I2C	LP_I2C_SDA	GPIO6	One LP I2C channel in slave or master mode
	LP_I2C_SCL	GPIO7	
LED PWM	ledc_ls_sig_out0~5	Any GPIO pins	Six independent PWM channels
I2S	I2S0O_BCK_in	Any GPIO pins	Stereo input and output from/to the audiocodec that support full-duplex/half-duplex TDM and PDM input/output and PDM-PCM input
	I2S_MCLK_in		
	I2SO_WS_in		
	I2SI_SD_in		
	I2SI_BCK_in		
	I2SI_WS_in		

Cont'd on next page

Table 3-1 – cont'd from previous page

Interface	Signal	Pin	Function
	I2SO_BCK_out		
	I2S_MCLK_out		
	I2SO_WS_out		
	I2SO_SD_out		
	I2SI_BCK_out		
	I2SI_WS_out		
	I2SO_SD1_out		
Remote Control Peripheral	RMT_SIG_IN0~1	Any GPIO pins	Two channels for an IR transceiver of various waveforms
	RMT_SIG_OUT0~1		
SPI0/1	SPICLK_out_mux	SPICLK	Support Standard SPI, Dual SPI, Quad SPI, and QPI that allow connection to off-package flash
	SPICS0_out	SPICS0	
	SPICS1_out	Any GPIO pins	
	SPID_in/_out	SPID	
	SPIQ_in/_out	SPIQ	
	SPIWP_in/_out	SPIWP	
	SPIHD_in/_out	SPIHD	
SPI2	FSPICLK_in/_out_mux	Any GPIO pins	The following functionality is supported: • Master mode and slave mode of SPI, Dual SPI, Quad SPI, and QPI • Connection to off-package flash, RAM and other SPI devices • Four modes of SPI transfer format • Configurable SPI frequency • 64-byte FIFO or GDMA buffer
	FSPICS0_in/_out		
	FSPICS1~5_out		
	FSPID_in/_out		
	FSPIQ_in/_out		
	FSPIWP_in/_out		
	FSPIHD_in/_out		
USB Serial/JTAG	USB_D+	GPIO26	USB-to-serial converter, and USB-to-JTAG converter
	USB_D-	GPIO25	
USB 2.0 OTG high-speed	USB_OTG_DM	USB_OTG_DM	USB 2.0 OTG high-speed interface
	USB_OTG_DP	USB_OTG_DP	
	ADPPRB_in	Any GPIO pins	
	ADPSNS_in		
	DRVBUS_out		
	ADPCHRG_out		
	ADPDISCHRG_out		
	ADPPRBEN_out		
	ADPSNSEN_out		
	PHY_REFCLK_in		

Cont'd on next page

Table 3-1 – cont'd from previous page

Interface	Signal	Pin	Function
TWAI®	TWAI0_RX	Any GPIO pins	Compatible with ISO 11898-1 protocol
	TWAI0_TX		
	TWAI0_BUS_OFF_ON		
	TWAI0_CLKOUT		
	TWAI0_STANDBY		
	TWAI1_RX		
	TWAI1_TX		
	TWAI1_BUS_OFF_ON		
	TWAI1_CLKOUT		
	TWAI1_STANDBY		
Pulse Count Controller	PCNT_SIG_CH0_in0~3	Any GPIO pins	Captures pulses and counts pulse edges in seven modes
	PCNT_SIG_CH1_in0~3		
	PCNT_CTRL_CH0_in0~3		
	PCNT_CTRL_CH1_in0~3		
MCPWM	PWM0_SYNC0~2_in	Any GPIO pins	One MCPWM to generate: • differential PWM output signals • fault input signals to be detected • input signals to be captured • external synchronization signals for PWM timers
	PWM0_out0a		
	PWM0_out0b		
	PWM0_out1a		
	PWM0_F0~2_in		
	PWM0_out1b		
	PWM0_out2a		
	PWM0_out2b		
	PWM0_CAP0~2_in		
PARLIO	PARL_RX_DATA0~7	Any GPIO pins	A module for parallel data transfer, with • 8 pins to receive parallel data • 8 pins to transmit parallel data • 2 receiver clock pins (clock input and output) • 2 transmitter clock pins (clock input and output)
	PARL_TX_DATA0~7		
	PARL_RX_CLK_in/_out		
	PARL_TX_CLK_in/_out		
SDIO	SDIO_CMD	SDIO_CMD	SDIO interface, conforming to the industry standard SDIO Specification Version 2.0
	SDIO_CLK	SDIO_CLK	
	SDIO_DATA0	SDIO_DATA0	

Cont'd on next page

Table 3-1 – cont'd from previous page

Interface	Signal	Pin	Function
	SDIO_DATA1	SDIO_DATA1	
	SDIO_DATA2	SDIO_DATA2	
	SDIO_DATA3	SDIO_DATA3	
Analog PAD voltage comparator	PAD_COMP0	GPIO8	Analog PAD voltage comparator
	PAD_COMP1	GPIO9	

4 Electrical Characteristics

Note:
The values presented in this section are **preliminary** and may change with the final release of this datasheet.

4.1 Absolute Maximum Ratings

Stresses above those listed in Table 4-1 *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 4.2 Recommended Power Supply Characteristics is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Table 4-1. Absolute Maximum Ratings

Parameter	Description	Min	Max	Unit
Input power pins[1]	Allowed input voltage	–0.3	3.6	V
T_{STORE}	Storage temperature	–40	150	°C

[1] For more information on input power pins, see Section 2.5.1 Power Pins.

4.2 Recommended Power Supply Characteristics

For recommended ambient temperature, see Section 1 ESP32-C5 Series Comparison.

Table 4-2. Recommended Power Characteristics

Parameter [1]	Description	Min	Typ	Max	Unit
VDDA1, VDDA2, VDDA3, VDDA4, VDDA5, VDDA3P3	Recommended input voltage	3.0	3.3	3.6	V
VDDPST1, VDDPST3	Recommended input voltage	3.0	3.3	3.6	V
VDD_SPI (as input)	—	3.0	3.3	3.6	V
VDDPST2 [2, 3]	Recommended input voltage	3.0	3.3	3.6	V
I_{VDD}	Cumulative input current	0.5	—	—	A

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

[2] If VDDPST2 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 4.3 VDD_SPI Output Characteristics.

[3] If writing to eFuses, the voltage on VDDPST2 should not exceed 3.3 V as the circuits responsible for burning eFuses are sensitive to higher voltages.

4.3 VDD_SPI Output Characteristics

Table 4-3. VDD_SPI Internal and Output Characteristics

Parameter	Description [1]	Typ	Unit
R_{SPI}	VDD_SPI powered by VDDPST2 via R_{SPI} for 3.3 V flash [2]	7.5	Ω

[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
- *I_flash_max* – maximum operating current of flash

4.4 DC Characteristics (3.3 V, 25 °C)

Table 4-4. 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.

Appendix A – ESP32-C5 Consolidated Pin Overview

Table 4-5. QFN48 Pin Overview

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		Analog Function		LP IO MUX Function		IO MUX Function					
				At Reset	After Reset	0	1	0	1	0	Type	1	Type	2	Type
1	VDDA3P3	Power													
2	GND	Power													
3	VDDA3P3	Power													
4	XTAL_N	Analog													
5	XTAL_P	Analog													
6	VDDD	Power													
7	CHIP_PU	Analog													
8	VDDPST1	Power													
9	XTAL_32K_P	IO	VDDPST1			XTAL_32K_P		LP_GPIO0	LP_UART_DTRN	**GPIO0**	I/O/T	GPIO0	I/O/T		
10	XTAL_32K_N	IO	VDDPST1			XTAL_32K_N	ADC1_CH0	LP_GPIO1	LP_UART_DSRN	**GPIO1**	I/O/T	GPIO1	I/O/T		
11	MTMS	IO	VDDPST1	IE	IE		ADC1_CH1	LP_GPIO2	LP_UART_RTSN	**MTMS**	I1	GPIO2	I/O/T	FSPIQ	I1/O/T
12	MTDI	IO	VDDPST1	IE	IE		ADC1_CH2	LP_GPIO3	LP_UART_CTSN	**MTDI**	I1	GPIO3	I/O/T		
13	MTCK	IO	VDDPST1		IE, WPU		ADC1_CH3	LP_GPIO4	LP_UART_RXD	**MTCK**	I1	GPIO4	I/O/T	FSPIHD	I1/O/T
14	MTDO	IO	VDDPST1	IE	IE		ADC1_CH4	LP_GPIO5	LP_UART_TXD	**MTDO**	O/T	GPIO5	I/O/T	FSPIWP	I1/O/T
15	GPIO6	IO	VDDPST1	IE	IE		ADC1_CH5	LP_GPIO6	LP_I2C_SDA	**GPIO6**	I/O/T	GPIO6	I/O/T	FSPICLK	I1/O/T
16	GPIO7	IO	VDDPST1	IE, WPU	IE, WPU			LP_GPIO7	LP_I2C_SCL	**GPIO7**	I/O/T	GPIO7	I/O/T	FSPID	I1/O/T
17	GPIO8	IO	VDDPST1			PAD_COMP0				**GPIO8**	I/O/T	GPIO8	I/O/T		
18	GPIO9	IO	VDDPST1			PAD_COMP1				**GPIO9**	I/O/T	GPIO9	I/O/T		
19	USB_OTG_DM	Analog	VDDPST2												
20	USB_OTG_DP	Analog	VDDPST2												
21	VDDPST2	Power													
22	U0TXD	IO	VDDPST2		WPU					**U0TXD**	O	GPIO10	I/O/T		
23	U0RXD	IO	VDDPST2		IE, WPU					**U0RXD**	I1	GPIO11	I/O/T		
24	SDIO_DATA1	IO	VDDPST2		IE					**SDIO_DATA1**	I1/O/T	GPIO12	I/O/T	FSPICS0	I1/O/T
25	SDIO_DATA0	IO	VDDPST2		IE					**SDIO_DATA0**	I1/O/T	GPIO13	I/O/T	FSPICS1	O/T
26	SDIO_CLK	IO	VDDPST2		IE					**SDIO_CLK**	I1	GPIO14	I/O/T	FSPICS2	O/T
27	SDIO_CMD	IO	VDDPST2		IE					**SDIO_CMD**	I1/O/T	GPIO15	I/O/T	FSPICS3	O/T
28	SDIO_DATA3	IO	VDDPST2		IE					**SDIO_DATA3**	I1/O/T	GPIO16	I/O/T	FSPICS4	O/T
29	SDIO_DATA2	IO	VDDPST2		IE					**SDIO_DATA2**	I1/O/T	GPIO17	I/O/T	FSPICS5	O/T
30	SPICS0	IO	VDD_SPI	WPU	IE, WPU					**SPICS0**	O/T	GPIO18	I/O/T		
31	SPIQ	IO	VDD_SPI	WPU	IE, WPU					**SPIQ**	I1/O/T	GPIO19	I/O/T		
32	SPIWP	IO	VDD_SPI	WPU	IE, WPU					SPIWP	I1/O/T	**GPIO20**	I/O/T		
33	VDD_SPI	Power/IO	—			**VDD_SPI**				GPIO21	I/O/T	GPIO21	I/O/T		
34	SPIHD	IO	VDD_SPI	WPU	IE, WPU					SPIHD	I1/O/T	**GPIO22**	I/O/T		
35	SPICLK	IO	VDD_SPI	WPU	IE, WPU					**SPICLK**	O/T	GPIO23	I/O/T		
36	SPID	IO	VDD_SPI	WPU	IE, WPU					**SPID**	I1/O/T	GPIO24	I/O/T		
37	VDDPST3	Power													
38	GPIO26	IO	VDDPST3		IE, WPU	**USB_D+**				GPIO26	I/O/T	GPIO26	I/O/T		
39	GPIO25	IO	VDDPST3		IE	**USB_D-**				GPIO25	I/O/T	GPIO25	I/O/T		
40	VDDA1	Power													
41	VDDA2	Power													

Cont'd on next page

Cont'd from previous page

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		Analog Function		LP IO MUX Function		IO MUX Function					
				At Reset	After Reset	0	1	0	1	0	Type	1	Type	2	Type
42	ANT_2G	Analog													
43	GND	Power													
44	VDDA3	Power													
45	VDDA4	Power													
46	VDDA5	Power													
47	GND	Power													
48	ANT_5G	Analog													

* For details, see Section 2 Pins. Regarding highlighted cells, see Section 2.3.4 Restrictions for GPIOs and LP GPIOs.

Table 4-6. QFN40 Pin Overview

Pin No.	Pin Name	Pin Type	Pin Providing Power	Pin Settings		Analog Function		LP IO MUX Function		IO MUX Function					
				At Reset	After Reset	0	1	0	1	0	Type	1	Type	2	Type
1	VDDA3P3	Power													
2	GND	Power													
3	VDDA3P3	Power													
4	XTAL_N	Analog													
5	XTAL_P	Analog													
6	VDDD	Power													
7	CHIP_PU	Analog													
8	VDDPST1	Power													
9	XTAL_32K_P	IO	VDDPST1			XTAL_32K_P		LP_GPIO0	LP_UART_DTRN	**GPIO0**	I/O/T	GPIO0	I/O/T		
10	XTAL_32K_N	IO	VDDPST1			XTAL_32K_N	ADC1_CH0	LP_GPIO1	LP_UART_DSRN	**GPIO1**	I/O/T	GPIO1	I/O/T		
11	MTMS	IO	VDDPST1	IE	IE		ADC1_CH1	LP_GPIO2	LP_UART_RTSN	**MTMS**	I1	GPIO2	I/O/T	FSPIQ	I1/O/T
12	MTDI	IO	VDDPST1	IE	IE		ADC1_CH2	LP_GPIO3	LP_UART_CTSN	**MTDI**	I1	GPIO3	I/O/T		
13	MTCK	IO	VDDPST1		IE, WPU		ADC1_CH3	LP_GPIO4	LP_UART_RXD	**MTCK**	I1	GPIO4	I/O/T	FSPIHD	I1/O/T
14	MTDO	IO	VDDPST1	IE	IE		ADC1_CH4	LP_GPIO5	LP_UART_TXD	**MTDO**	O/T	GPIO5	I/O/T	FSPIWP	I1/O/T
15	GPIO6	IO	VDDPST1	IE	IE		ADC1_CH5	LP_GPIO6	LP_I2C_SDA	**GPIO6**	I/O/T	GPIO6	I/O/T	FSPICLK	I1/O/T
16	GPIO7	IO	VDDPST1	IE, WPU	IE, WPU			LP_GPIO7	LP_I2C_SCL	**GPIO7**	I/O/T	GPIO7	I/O/T	FSPID	I1/O/T
17	USB_OTG_DM	Analog	VDDPST2												
18	USB_OTG_DP	Analog	VDDPST2												
19	VDDPST2	Power													
20	U0TXD	IO	VDDPST2		WPU					**U0TXD**	O	GPIO10	I/O/T		
21	U0RXD	IO	VDDPST2		IE, WPU					**U0RXD**	I1	GPIO11	I/O/T		
22	SDIO_DATA1	IO	VDDPST2		IE					**SDIO_DATA1**	I1/O/T	GPIO12	I/O/T	FSPICS0	I1/O/T
23	SDIO_DATA0	IO	VDDPST2		IE					**SDIO_DATA0**	I1/O/T	GPIO13	I/O/T	FSPICS1	O/T
24	SDIO_CLK	IO	VDDPST2		IE					**SDIO_CLK**	I1	GPIO14	I/O/T	FSPICS2	O/T
25	SDIO_CMD	IO	VDDPST2		IE					**SDIO_CMD**	I1/O/T	GPIO15	I/O/T	FSPICS3	O/T
26	SDIO_DATA3	IO	VDDPST2		IE					**SDIO_DATA3**	I1/O/T	GPIO16	I/O/T	FSPICS4	O/T
27	SDIO_DATA2	IO	VDDPST2		IE					**SDIO_DATA2**	I1/O/T	GPIO17	I/O/T	FSPICS5	O/T
28	VDD_SPI	Power/IO	—			**VDD_SPI**				GPIO21	I/O/T	GPIO21	I/O/T		
29	GPIO25	IO	VDDPST3		IE	**USB_D-**				GPIO25	I/O/T	GPIO25	I/O/T		
30	GPIO26	IO	VDDPST3		IE, WPU	**USB_D+**				GPIO26	I/O/T	GPIO26	I/O/T		
31	VDDPST3	Power													
32	VDDA1	Power													
33	VDDA2	Power													
34	ANT_2G	Analog													
35	GND	Power													
36	VDDA3	Power													
37	VDDA4	Power													
38	VDDA5	Power													
39	GND	Power													
40	ANT_5G	Analog													

* For details, see Section 2 Pins. Regarding highlighted cells, see Section 2.3.4 Restrictions for GPIOs and LP GPIOs.

Revision History

Date	Version	Release notes
2023-08-22	v0.1	Draft

www.espressif.com

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.