Xpert Automatix

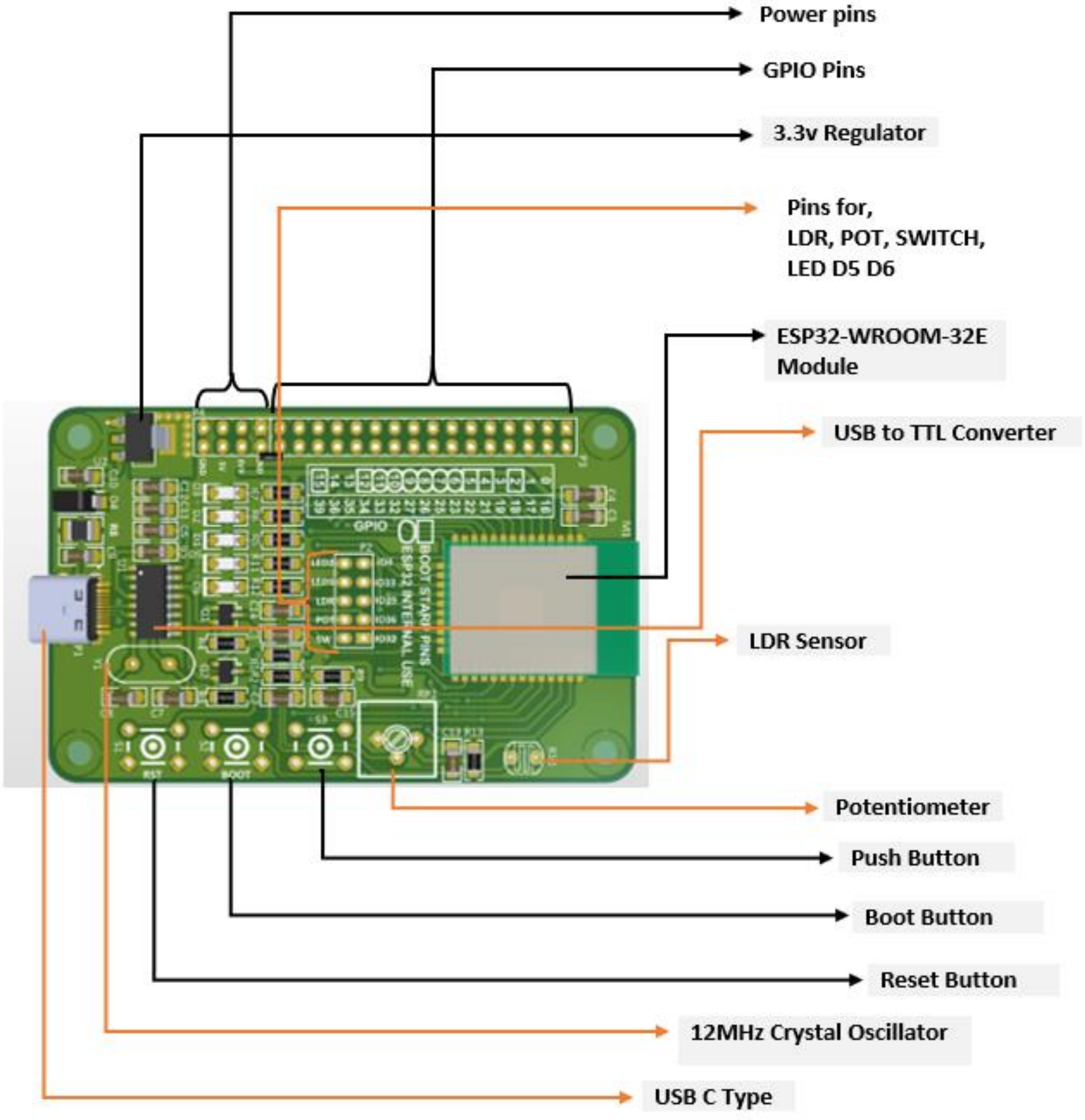
ESP32-WROOM-32E

Datasheet

Contents

1. Overview

The ESP32 development board is designed around the ESP32-WROOM-32E module, which integrates the ESP32 dual-core microcontroller. This board is tailored for IoT applications, leveraging the capabilities of the ESP32 chip from Espressif Systems.



ESP32-WROOM-32E is powerful, generic Wi-Fi + Bluetooth + BLE MCU modules that target a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

1.1. Features

CPU and On-Chip Memory

- ESP32-D0WD-V3 or ESP32-D0WDR2-V3 embedded, Xtensa dual-core 32-bit LX6 microprocessor, up to 240 MHz
- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC
- ESP32-D0WDR2-V3 also provides 2 MB PSRAM

Wi-Fi

- 802.11b/g/n
- Bit rate: 802.11n up to 150 Mbps
- A-MPDU and A-MSDU aggregation
- 0.4 μs guard interval support
- Center frequency range of operating channel: 2412 ~ 2484 MHz

Bluetooth

- Bluetooth V4.2 BR/EDR and Bluetooth LE specification
- Class-1, class-2 and class-3 transmitter
- AFH
- CVSD and SBC

Peripherals

- SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC, TWAI.
  (Compatible with ISO 11898-1, i.e., CAN
  Specification 2.0)

Integrated Components on Module

- 40 MHz crystal oscillator
- 4/8/16 MB SPI flash, This Device Consist 4 MB flash.

Antenna Options

- ESP32-WROOM-32E: On-board PCB antenna
- ESP32-WROOM-32UE: external antenna via a connector

Operating Conditions

- Operating voltage/Power supply: 3.0 ~ 3.6 V
- Operating ambient temperature:
  - 85 °C version: –40 ~ 85 °C
  - 105 °C version: –40 ~ 105 °C. Note that only the modules embedded with a 4/8 MB flash support this version.

Reliability Test

- HTOL/HTSL/uHAST/TCT/ESD

Certification

- Bluetooth certification: BQB
- RF certification: See certificates for ESP32-WROOM-32E
- Green certification: REACH/RoHS

1.2. Description

At the core of the module is the ESP32-D0WD-V3 chip or ESP32-D0WDR2-V3 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The chip also has a low-power coprocessor that can be used instead of the CPU to save power while performing tasks that do not require much computing power, such as monitoring of peripherals. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, SD card interface, Ethernet, high-speed SPI, UART, I2S, and I2C.

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is all-around: using Wi-Fi allows a large physical range and direct connection to the Internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μA, making it suitable for battery powered and wearable electronics applications. The module supports a data rate of up to 150 Mbps, and 20 dBm output power at the antenna to ensure the widest physical range. As such the module does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that users can upgrade their products even after their release, at minimum cost and effort.

Ordering Code	Flash	PSRAM	Ambient Temp (°C)	Size (mm)
ESP32-WROOM-32E-N4	4 MB (Quad SPI)	—	–40 ~ 85	18.0 × 25.5 × 3.1
ESP32-WROOM-32E-N8	8 MB (Quad SPI)	—	–40 ~ 85	
ESP32-WROOM-32E-N16	16 MB (Quad SPI)	—	–40 ~ 85	
ESP32-WROOM-32E-H4	4 MB (Quad SPI)	—	–40 ~ 105	
ESP32-WROOM-32E-H8	8 MB (Quad SPI)	—	–40 ~ 105	
ESP32-WROOM-32E-N4R2	4 MB (Quad SPI)	2 MB (Quad SPI)	–40 ~ 85	
ESP32-WROOM-32E-N8R2	8 MB (Quad SPI)	2 MB (Quad SPI)	–40 ~ 85	
ESP32-WROOM-32E-N16R2	16 MB (Quad SPI)	2 MB (Quad SPI)	–40 ~ 85	

Table 1: ESP32-WROOM-32E Series Comparison

1.3. Applications

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

- Generic Low-power IoT Sensor Hub
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- Over-the-top (OTT) Devices
- Speech Recognition
- Image Recognition
- Mesh Network
- Home Automation
- Smart Building
- Industrial Automation
- Smart Agriculture
- Audio Applications
- Health Care Applications
- Wi-Fi-enabled Toys
- Wearable Electronics
- Retail & Catering Applications

2. Block Diagram

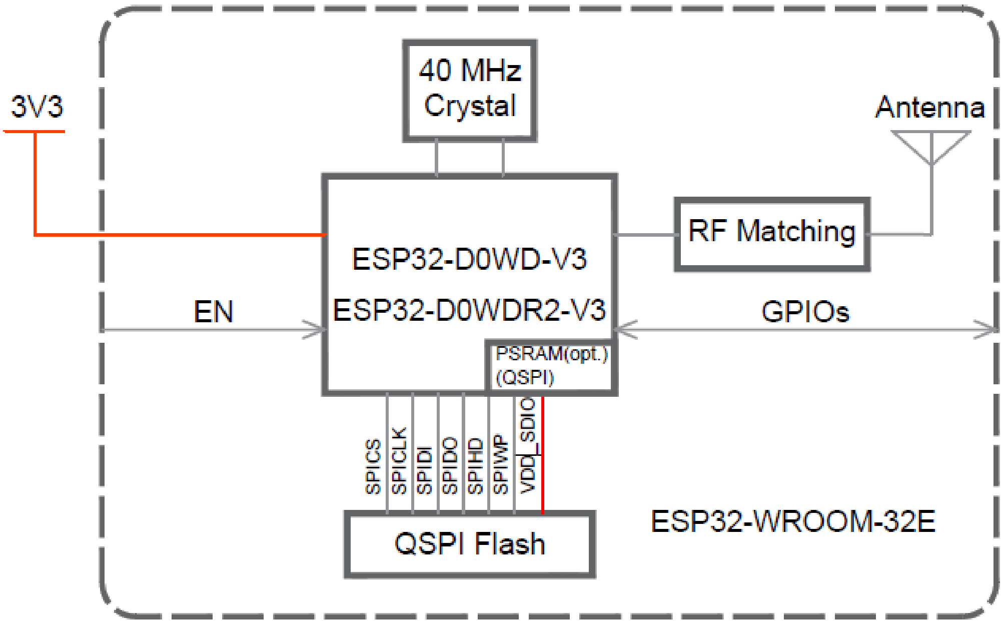
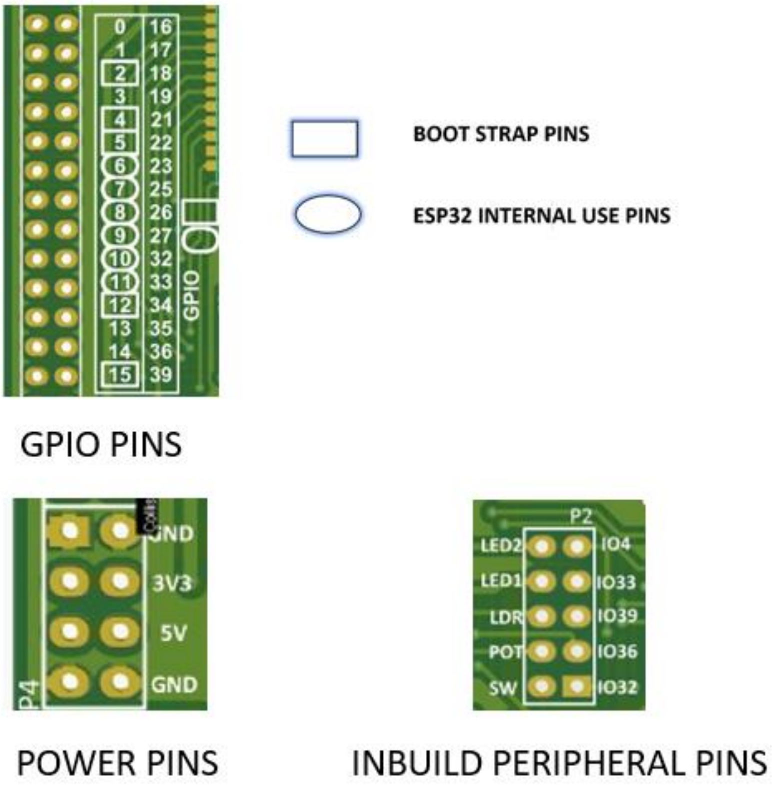


Figure 2: ESP32-WROOM-32E Block Diagram

3. Pin Definition

3.1. Pin Layout



GPIO PINS

POWER PINS

INBUILD PERIPHERAL PINS

Figure 3: ESP32-WROOM-32E DEV MODULE PINOUT

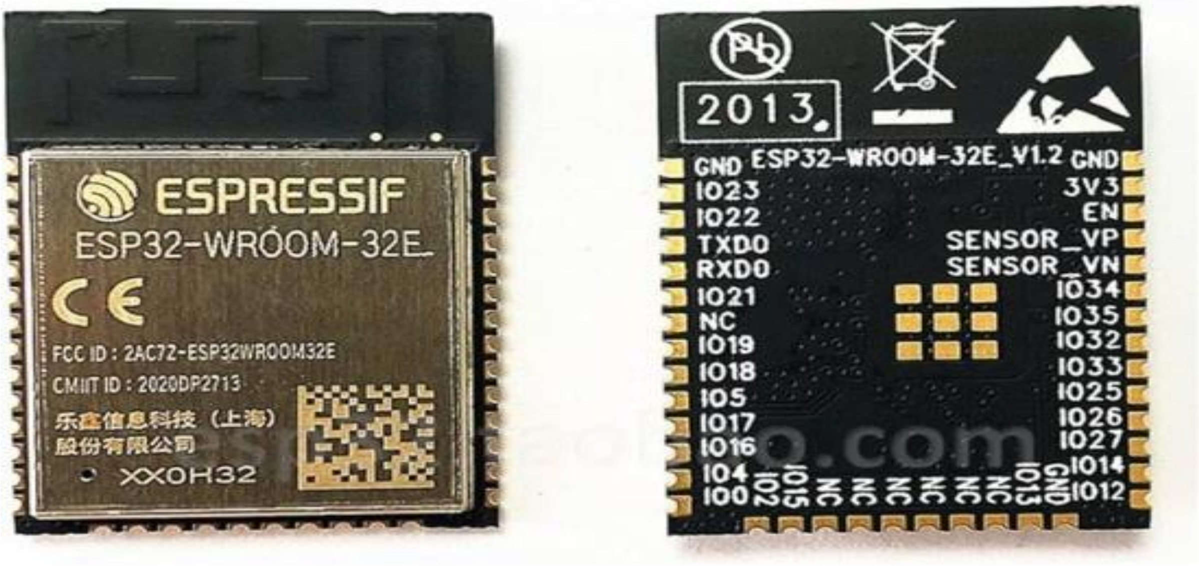


Figure 4: ESP32-WROOM-32E CONTROLLER PINOUT

3.2. Pin Description

0	GPIO0	ADC2_CH1	RTC_GPIO11	TOUCH1	CLK_OUT1	EMAC_TX_CLK			
1	GPIO1	UART0_TxD			CLK_OUT3	EMAC_RXD2			
2	GPIO2	ADC2_CH2	RTC_GPIO12	TOUCH2	HSPIWP		SD_DATA0	HS2_DATA0	
3	GPIO3	UART0_RxD			CLK_OUT2				
4	GPIO4	ADC2_CH0	RTC_GPIO10	TOUCH0	HSPIHD	EMAC_TX_ER	SD_DATA1	HS2_DATA1	
5	GPIO5	VSPI_CS0				EMAC_RX_CLK		HS1_DATA6	
6	GPIO6	Pins GPIO6 to GPIO11 on the ESP32-D0WD-V3/ESP32-D0WDR2-V3 chip are connected to the SPI flash integrated on the module and are not led out.							
7	GPIO7								
8	GPIO8								
9	GPIO9								
10	GPIO10								
11	GPIO11								
12	GPIO12	ADC2_CH5	RTC_GPIO15	TOUCH5	HSPIQ	EMAC_TXD3	SD_DATA2	HS2_DATA2	MTDI
13	GPIO13	ADC2_CH4	RTC_GPIO14	TOUCH4	HSPID	EMAC_RX_ER	SD_DATA3	HS2_DATA3	MTCK
14	GPIO14	ADC2_CH6	RTC_GPIO16	TOUCH6	HSPICLK	EMAC_TXD2	SD_CLK	HS2_CLK	MTMS
15	GPIO15	ADC2_CH3	RTC_GPIO13	TOUCH3	HSPICS0	EMAC_RXD3	SD_CMD	HS2_CMD	MTDO
16	GPIO16	U2RXD				EMAC_CLK_OUT		HS1_DATA4	
17	GPIO17	U2TXD				EMAC_CLK_OUT_180		HS1_DATA5	
18	GPIO18				VSPICLK			HS1_DATA7	
19	GPIO19			U0CTS	VSPIQ	EMAC_TXD0			
21	GPIO21	SDA			VSPIHD	EMAC_TX_EN			
22	GPIO22	SCL		U0RTS	VSPIWP	EMAC_TXD1			
23	GPIO23				VSPID			HS1_STROBE	
25	GPIO25	ADC2_CH8	RTC_GPIO6	DAC_1		EMAC_RXD0			
26	GPIO26	ADC2_CH9	RTC_GPIO7	DAC_2		EMAC_RXD1			
27	GPIO27	ADC2_CH7	RTC_GPIO17	TOUCH7		EMAC_RX_DV			
32	GPIO32	ADC1_CH4	RTC_GPIO9	TOUCH9	XTAL_32K_P (32.768 kHz crystal oscillator input)				
33	GPIO33	ADC1_CH5	RTC_GPIO8	TOUCH8	XTAL_32K_N (32.768 kHz crystal oscillator output)				
34	GPIO34	ADC1_CH6	RTC_GPIO4	INPUT ONLY					
35	GPIO35	ADC1_CH7	RTC_GPIO5	INPUT ONLY					
36	GPIO36	ADC1_CH0	RTC_GPIO0	INPUT ONLY					
39	GPIO39	ADC1_CH3	RTC_GPIO3	INPUT ONLY					

Table 2: ESP32-WROOM-32E Dev Module Pin Description

4. Electrical Characteristics

4.1. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDD33	Power supply voltage	–0.3	3.6	V
TSTORE	Storage temperature	–40	105	°C

Table 3: Absolute Maximum Ratings

4.2. Recommended Operating Conditions

Symbol	Parameter		Min	Typ	Max	Unit
VDD33	Power supply voltage		3.0	3.3	3.6	V
IV DD	Current delivered by external power supply		0.5	—	—	A
T	Operating ambient temperature	85 °C version	–40	—	85	°C
		105 °C version			105	

Table 4: Recommended Operating Conditions

4.3. DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter		Min	Typ	Max	Unit
CIN	Pin capacitance		—	2	—	pF
VIH	High-level input voltage		0.75 × VDD	—	VDD+ 0.3	V
VIL	Low-level input voltage		–0.3	—	0.25 × VDD	V
IIH	High-level input current		—	—	50	nA
IIL	Low-level input current		—	—	50	nA
VOH	High-level output voltage		0.8 × VDD	—	—	V
VOL	Low-level output voltage		—	—	0.1 × VDD	V
IOH	High-level source current (VDD1= 3.3 V, *VOH* >= 2.64 V, output drive strength set to the maximum)	VDD3P3_CPU power domain 1, 2	—	40	—	mA
		VDD3P3_RTC power domain 1, 2	—	40	—	mA
		VDD_SDIO power domain 1, 3	—	20	—	mA
IOL	Low-level sink current (VDD1= 3.3 V, *VOL* = 0.495 V, output drive strength set to the maximum)		—	28	—	mA
R*PU*	Resistance of internal pull-up resistor		—	45	—	kΩ
R*PD*	Resistance of internal pull-down resistor		—	45	—	kΩ

Symbol	Parameter	Min	Typ	Max	Unit
VIL_nRST	Low-level input voltage of CHIP_PU to shut down the chip	—	—	0.6	V

1 Please see Appendix IO MUX of *ESP32 Series Datasheet* for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2 For VDD3P3_CPU and VDD3P3_RTC power domain, per-pin current sourced in the same domain is gradually reduced from around 40 mA to around 29 mA, V*OH*>=2.64 V, as the number of current sources pins increase.
3 Pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.

Table 5: Recommended DC Characteristics (3.3 V, 25 °C

4.4. Current Consumption Characteristics

Work mode	Description		Average (mA)	Peak (mA)
Active (RF working)	TX	802.11b, 20 MHz, 1 Mbps, @19.5 dBm	239	379
		802.11g, 20 MHz, 54 Mbps, @15 dBm	190	276
		802.11n, 20 MHz, MCS7, @13 dBm	183	258
		802.11n, 40 MHz, MCS7, @13 dBm	165	211
	RX	802.11b/g/n, 20 MHz	112	112
		802.11n, 40 MHz	118	118

Table 6: Current Consumption Depending on RF Modes

4.5. Wi-Fi RF Characteristics

4.5.1. Wi-Fi RF Standards

Name		Description
Center frequency range of operating channel		2412 ~ 2484 MHz
Wi-Fi wireless standard		IEEE 802.11b/g/n
Data rate	20 MHz	11b: 1, 2, 5.5, 11 Mbps 11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps 11n: MCS0-7, 72.2 Mbps (Max)
	40 MHz	11n: MCS0-7, 150 Mbps (Max)
Antenna type		PCB antenna, external antenna

Table 7: Wi-Fi RF Characteristics

4.5.2. Transmitter Characteristics

Rate	Typ (dBm)
11b, 1 Mbps	19.5
11b, 11 Mbps	19.5
11g, 6 Mbps	18
11g, 54 Mbps	14
11n, HT20, MCS0	18
11n, HT20, MCS7	13
11n, HT40, MCS0	18
11n, HT40, MCS7	13

Table 8: TX Power Characteristics

4.5.3. Receiver Characteristics

Rate	Typ (dBm)
1 Mbps	–97
2 Mbps	–94
5.5 Mbps	–92
11 Mbps	–88
6 Mbps	–93
9 Mbps	–91
12 Mbps	–89
18 Mbps	–87
24 Mbps	–84
36 Mbps	–80
48 Mbps	–77
54 Mbps	–75
11n, HT20, MCS0	–92
11n, HT20, MCS1	–88
11n, HT20, MCS2	–86
11n, HT20, MCS3	–83
11n, HT20, MCS4	–80
11n, HT20, MCS5	–76
11n, HT20, MCS6	–74
11n, HT20, MCS7	–72
11n, HT40, MCS0	–89
11n, HT40, MCS1	–85

11n, HT40, MCS2	–83
11n, HT40, MCS3	–80
11n, HT40, MCS4	–76
11n, HT40, MCS5	–72
11n, HT40, MCS6	–71
11n, HT40, MCS7	–69

Table 9: RX Sensitivity Characteristics

Rate	Typ (dBm)
11b, 1 Mbps	5
11b, 11 Mbps	5
11g, 6 Mbps	0
11g, 54 Mbps	–8
11n, HT20, MCS0	0
11n, HT20, MCS7	–8
11n, HT40, MCS0	0
11n, HT40, MCS7	–8

Table 10: RX Maximum Input Level

Rate	Typ (dBm)
11b, 11 Mbps	35
11g, 6 Mbps	27
11g, 54 Mbps	13
11n, HT20, MCS0	27
11n, HT20, MCS7	12
11n, HT40, MCS0	16
11n, HT40, MCS7	7

Table 11: Adjacent Channel Rejection

4.6. Bluetooth Radio

4.6.1. Receiver – Basic Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @0.1% BER	—	–90	–89	–88	dBm
Maximum received signal @0.1% BER	—	0	—	—	dBm
Co-channel C/I	—	—	+7	—	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	—	—	–6	dB
	F = F0 – 1 MHz	—	—	–6	dB

	F = F0 + 2 MHz	—	—	–25	dB
	F = F0 – 2 MHz	—	—	–33	dB
	F = F0 + 3 MHz	—	—	–25	dB
	F = F0 – 3 MHz	—	—	–45	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	–10	—	—	dBm
	2000 MHz ~ 2400 MHz	–27	—	—	dBm
	2500 MHz ~ 3000 MHz	–27	—	—	dBm
	3000 MHz ~ 12.5 GHz	–10	—	—	dBm
Intermodulation	—	–36	—	—	dBm

Table 12: Receiver Characteristics – Basic Data Rate

4.6.2. Transmitter – Basic Data Rate

Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
RF transmit power*	—	—	0	—	dBm
Gain control step	—	—	3	—	dB
RF power control range	—	–12	—	+9	dBm
+20 dB bandwidth	—	—	0.9	—	MHz
Adjacent channel transmit power	F = F0 ± 2 MHz	—	–55	—	dBm
	F = F0 ± 3 MHz	—	–55	—	dBm
	F = F0 ± > 3 MHz	—	–59	—	dBm
Δ f1avg	—	—	—	155	kHz
Δ f2max	—	127	—	—	kHz
Δ f2avg/Δ f1avg	—	—	0.92	—	—
ICFT	—	—	–7	—	kHz
Drift rate	—	—	0.7	—	kHz/50 μs
Drift (DH1)	—	—	6	—	kHz
Drift (DH5)	—	—	6	—	kHz

Table 13: Transmitter Characteristics – Basic Data Rate

4.6.3. Receiver – Enhanced Data Rate

Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Unit**
$\pi/4$ DQPSK					
Sensitivity @0.01% BER	—	–90	–89	–88	dBm
Maximum received signal @0.01% BER	—	—	0	—	dBm
Co-channel C/I	—	—	11	—	dB

Adjacent channel selectivity C/I	F = F0 + 1 MHz	—	–7	—	dB
	F = F0 – 1 MHz	—	–7	—	dB
	F = F0 + 2 MHz	—	–25	—	dB
	F = F0 – 2 MHz	—	–35	—	dB
	F = F0 + 3 MHz	—	–25	—	dB
	F = F0 – 3 MHz	—	–45	—	dB
8DPSK					
Sensitivity @0.01% BER	—	–84	–83	–82	dBm
Maximum received signal @0.01% BER	—	—	–5	—	dBm
C/I c-channel	—	—	18	—	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	—	2	—	dB
	F = F0 – 1 MHz	—	2	—	dB
	F = F0 + 2 MHz	—	–25	—	dB
	F = F0 – 2 MHz	—	–25	—	dB
	F = F0 + 3 MHz	—	–25	—	dB
	F = F0 – 3 MHz	—	–38	—	dB

Table 14: Receiver Characteristics – Enhanced Data Rate

4.6.4. Transmitter – Enhanced Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 13)	—	—	0	—	dBm
Gain control step	—	—	3	—	dB
RF power control range	—	–12	—	+9	dBm
π/4 DQPSK max w0	—	—	–0.72	—	kHz
π/4 DQPSK max wi	—	—	–6	—	kHz
π/4 DQPSK max |wi + w0|	—	—	–7.42	—	kHz
8DPSK max w0	—	—	0.7	—	kHz
8DPSK max wi	—	—	–9.6	—	kHz
8DPSK max |wi + w0|	—	—	–10	—	kHz

Parameter	Conditions	Min	Typ	Max	Unit
π/4 DQPSK modulation accuracy	RMS DEVM	—	4.28	—	%
	99% DEVM	—	100	—	%
	Peak DEVM	—	13.3	—	%
8 DPSK modulation accuracy	RMS DEVM	—	5.8	—	%
	99% DEVM	—	100	—	%
	Peak DEVM	—	14	—	%

Parameter	Conditions	Min	Typ	Max	Unit
In-band spurious emissions	F = F0 ± 1 MHz	—	–46	—	dBm
	F = F0 ± 2 MHz	—	–44	—	dBm
	F = F0 ± 3 MHz	—	–49	—	dBm
	F = F0 +/– > 3 MHz	—	—	–53	dBm
EDR differential phase coding	—	—	100	—	%

Table 15: Transmitter Characteristics – Enhanced Data Rate

4.7. BLE Ratio

4.7.1. Receiver Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	—	–94	–93	–92	dBm
Maximum received signal @30.8% PER	—	0	—	—	dBm
Adjacent channel selectivity C/I	F = F0 + 1 MHz	—	–5	—	dB
	F = F0 – 1 MHz	—	–5	—	dB
	F = F0 + 2 MHz	—	–25	—	dB
	F = F0 – 2 MHz	—	–35	—	dB
	F = F0 + 3 MHz	—	–25	—	dB
	F = F0 – 3 MHz	—	–45	—	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	–10	—	—	dBm
	2000 MHz ~ 2400 MHz	–27	—	—	dBm
	2500 MHz ~ 3000 MHz	–27	—	—	dBm
	3000 MHz ~ 12.5 GHz	–10	—	—	dBm
Intermodulation	—	–36	—	—	dBm

Table 16: Receiver Characteristics – Bluetooth LE

4.7.2. Transmitter Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 13)	—	—	0	—	dBm
Gain control step	—	—	3	—	dB
RF power control range	—	–12	—	+9	dBm
Adjacent channel transmit power	F = F0 ± 2 MHz	—	–55	—	dBm
	F = F0 ± 3 MHz	—	–57	—	dBm
	F = F0 ± > 3 MHz	—	–59	—	dBm
Δ *f*1avg	—	—	—	265	kHz

Parameter	Conditions	Min	Typ	Max	Unit
Δ *f*2max	—	210	—	—	kHz
Δ *f*2avg/Δ *f*1avg	—	—	+0.92	—	—
ICFT	—	—	−10	—	kHz
Drift rate	—	—	0.7	—	kHz/50 μs
Drift	—	—	2	—	kHz

Table 17: Transmitter Characteristics – Bluetooth LE

5. Functional Description

5.1. CPU and Memory

5.1.1. CPU

ESP32 contains one or two low-power Xtensa® 32-bit LX6 microprocessor(s) with the following features:

- 7-stage pipeline to support the clock frequency of up to 240 MHz (160 MHz for ESP32-S0WD
- 16/24-bit Instruction Set provides high code-density
- Support for Floating Point Unit
- Support for DSP instructions, such as a 32-bit multiplier, a 32-bit divider, and a 40-bit MAC
- Support for 32 interrupt vectors from about 70 interrupt sources

The single-/dual-CPU interfaces include:

- Xtensa RAM/ROM Interface for instructions and data
- Xtensa Local Memory Interface for fast peripheral register access
- External and internal interrupt sources
- JTAG for debugging

5.1.2. Internal Memory

ESP32's internal memory includes:

- 448 KB of ROM for booting and core functions
- 520 KB of on-chip SRAM for data and instructions
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed
  - by the main CPU during RTC Boot from the Deep-sleep mode.
  - 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the ULP coprocessor
  - during the Deep-sleep mode.
  - 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the
  - remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.
  - In-package flash or PSRAM

5.1.3. External Flash and RAM

ESP32 supports multiple external QSPI flash and external RAM (SRAM) chips. More details can be found in *ESP32 Technical Reference Manual* > Chapter *SPI Controller*. ESP32 also supports hardware encryption/decryption based on AES to protect developers' programs and data in flash.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously.
  - When external flash is mapped into CPU instruction memory space, up to 11 MB + 248 KB can be mapped at a time. Note that if more than 3 MB + 248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.
  - When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.
- External RAM can be mapped into CPU data memory space. SRAM up to 8 MB is supported and up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

5.1.4. Address Mapping Structure

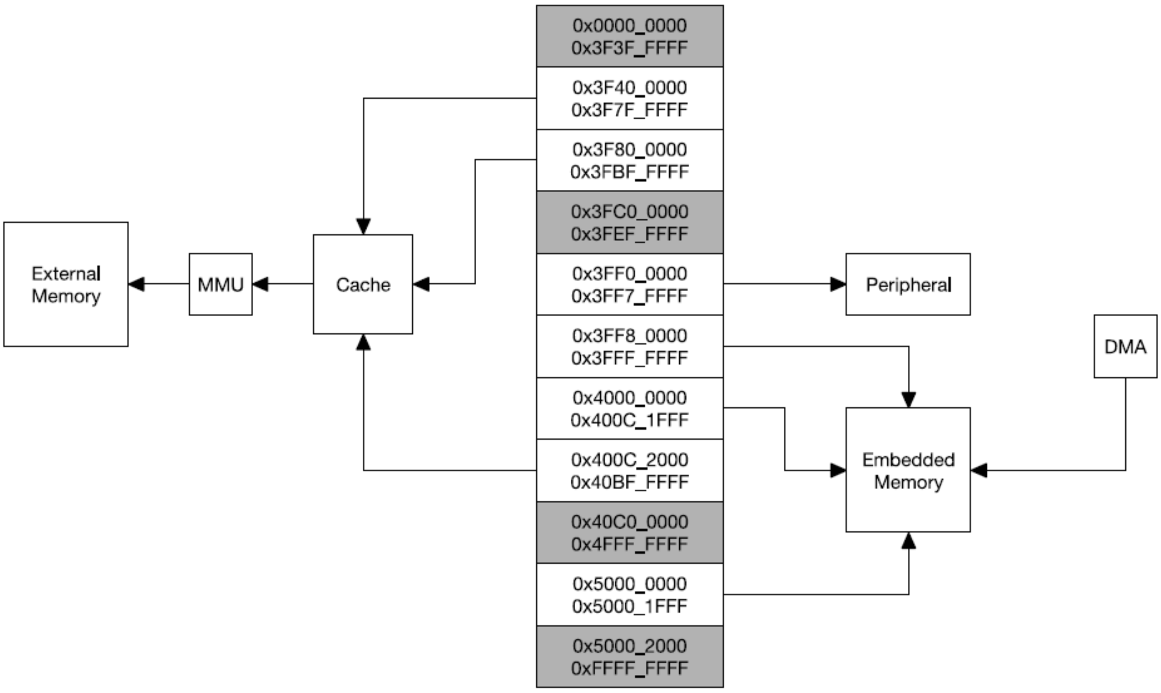


Figure 5: Address Mapping Structure Table

Category	Target	Start Address	End Address	Size
Embedded Memory	Internal ROM 0	0×4000_0000	0×4005_FFFF	384 KB
	Internal ROM 1	0×3FF9_0000	0×3FF9_FFFF	64 KB
	Internal SRAM 0	0×4007_0000	0×4009_FFFF	192 KB
	Internal SRAM 1	0×3FFE_0000	0×3FFF_FFFF	128 KB
		0×400A_0000	0×400B_FFFF	

Category	Target	Start Address	End Address	Size
Embedded Memory	Internal SRAM 2	0×3FFA_E000	0×3FFD_FFFF	200 KB
	RTC FAST Memory	0×3FF8_0000	0×3FF8_1FFF	8 KB
		0×400C_0000	0×400C_1FFF	
	RTC SLOW Memory	0×5000_0000	0×5000_1FFF	8 KB
External Memory	External Flash	0×3F40_0000	0×3F7F_FFFF	4 MB
		0×400C_2000	0×40BF_FFFF	11 MB+248 KB
	External RAM	0×3F80_0000	0×3FBF_FFFF	4 MB
Peripheral	DPort Register	0×3FF0_0000	0×3FF0_0FFF	4 KB
	AES Accelerator	0×3FF0_1000	0×3FF0_1FFF	4 KB
	RSA Accelerator	0×3FF0_2000	0×3FF0_2FFF	4 KB
	SHA Accelerator	0×3FF0_3000	0×3FF0_3FFF	4 KB
	Secure Boot	0×3FF0_4000	0×3FF0_4FFF	4 KB
	Cache MMU Table	0×3FF1_0000	0×3FF1_3FFF	16 KB
	PID Controller	0×3FF1_F000	0×3FF1_FFFF	4 KB
	UART0	0×3FF4_0000	0×3FF4_0FFF	4 KB
	SPI1	0×3FF4_2000	0×3FF4_2FFF	4 KB
	SPI0	0×3FF4_3000	0×3FF4_3FFF	4 KB
	GPIO	0×3FF4_4000	0×3FF4_4FFF	4 KB
	RTC	0×3FF4_8000	0×3FF4_8FFF	4 KB
	IO MUX	0×3FF4_9000	0×3FF4_9FFF	4 KB
	SDIO Slave	0×3FF4_B000	0×3FF4_BFFF	4 KB
	UDMA1	0×3FF4_C000	0×3FF4_CFFF	4 KB
	I2S0	0×3FF4_F000	0×3FF4_FFFF	4 KB
	UART1	0×3FF5_0000	0×3FF5_0FFF	4 KB
	I2C0	0×3FF5_3000	0×3FF5_3FFF	4 KB
	UDMA0	0×3FF5_4000	0×3FF5_4FFF	4 KB
	SDIO Slave	0×3FF5_5000	0×3FF5_5FFF	4 KB
	RMT	0×3FF5_6000	0×3FF5_6FFF	4 KB
	PCNT	0×3FF5_7000	0×3FF5_7FFF	4 KB
	SDIO Slave	0×3FF5_8000	0×3FF5_8FFF	4 KB
	LED PWM	0×3FF5_9000	0×3FF5_9FFF	4 KB
	Flash Encryption	0×3FF5_B000	0×3FF5_BFFF	4 KB
	PWM0	0×3FF5_E000	0×3FF5_EFFF	4 KB
	TIMG0	0×3FF5_F000	0×3FF5_FFFF	4 KB
	TIMG1	0×3FF6_0000	0×3FF6_0FFF	4 KB
	SPI2	0×3FF6_4000	0×3FF6_4FFF	4 KB
	SPI3	0×3FF6_5000	0×3FF6_5FFF	4 KB
	SYSCON	0×3FF6_6000	0×3FF6_6FFF	4 KB
	I2C1	0×3FF6_7000	0×3FF6_7FFF	4 KB
	SDMMC	0×3FF6_8000	0×3FF6_8FFF	4 KB
	EMAC	0×3FF6_9000	0×3FF6_AFFF	8 KB
	TWAI	0×3FF6_B000	0×3FF6_BFFF	4 KB
	PWM1	0×3FF6_C000	0×3FF6_CFFF	4 KB
	I2S1	0×3FF6_D000	0×3FF6_DFFF	4 KB
	UART2	0×3FF6_E000	0×3FF6_EFFF	4 KB

Category	Target	Start Address	End Address	Size
Peripheral	PWM2	0×3FF6_F000	0×3FF6_FFFF	4 KB
	PWM3	0×3FF7_0000	0×3FF7_0FFF	4 KB
	RNG	0×3FF7_5000	0×3FF7_5FFF	4 KB

18: Memory and Peripheral Mapping

5.1.5. Cache

ESP32 uses a two-way set-associative cache. Each of the two CPUs has 32 KB of cache featuring a block size of 32 bytes for accessing external storage.

For details, see *ESP32 Technical Reference Manual* > Chapter *System and Memory* > Section *Cache*.

5.2. System Clocks

5.2.1. CPU Clock

Upon reset, an external crystal clock source is selected as the default CPU clock. The external crystal clock source also connects to a PLL to generate a high-frequency clock (typically 160 MHz).

In addition, ESP32 has an internal 8 MHz oscillator. The application can select the clock source from the external crystal clock source, the PLL clock or the internal 8 MHz oscillator. The selected clock source drives the CPU clock directly, or after division, depending on the application.

5.2.2. RTC Clock

The RTC clock has five possible sources:

- external low-speed (32 kHz) crystal clock
- external crystal clock divided by 4
- internal RC oscillator (typically about 150 kHz, and adjustable)
- internal 8 MHz oscillator
- internal 31.25 kHz clock (derived from the internal 8 MHz oscillator divided by 256)

When the chip is in the normal power mode and needs faster CPU accessing, the application can choose the external high-speed crystal clock divided by 4 or the internal 8 MHz oscillator. When the chip operates in the low-power mode, the application chooses the external low-speed (32 kHz) crystal clock, the internal RC clock or the internal 31.25 kHz clock.

5.2.3. Audio PLL Clock

The audio clock is generated by the ultra-low-noise fractional-N PLL.
For details, see *ESP32 Technical Reference Manual* > Chapter *Reset and Clock*.

5.3. RTC and Low-power Management

5.3.1. Power Management Unit (PMU)

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

- Power modes

– **Active mode**: The chip radio is powered up. The chip can receive, transmit, or listen.

– **Modem-sleep mode**: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.

– **Light-sleep mode**: The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP coprocessor are running. Any wake-up events (MAC, SDIO host, RTC timer, or external interrupts) will wake up the chip.

– **Deep-sleep mode**: Only the RTC memory and RTC peripherals are powered up. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP coprocessor is functional.

– **Hibernation mode**: The internal 8 MHz oscillator and ULP coprocessor 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.

Power mode	Description			Power Consumption
Active (RF working)	Wi-Fi Tx packet			Please refer to Table 6 for details.
	Wi-Fi/BT Tx packet			
	Wi-Fi/BT Rx and listening			
Modem-sleep	The CPU is powered up.	240 MHz*	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz*	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	The ULP coprocessor is powered up.			150 μA
	ULP sensor-monitored pattern			100 μA @1% duty
	RTC timer + RTC memory			10 μA
Hibernation	RTC timer only			5 μA
Power off	CHIP_PU is set to low level, the chip is powered down.			1 μA

Table 19: Power Consumption by Power Modes

- * Among the ESP32 series of SoCs, ESP32-D0WD-V3, ESP32-D0WDR2-V3, ESP32-U4WDH, ESP32-D0WD (NRND), ESP32-D0WDQ6 (NRND), and ESP32-D0WDQ6-V3 (NRND) have a maximum CPU frequency of 240 MHz, ESP32-S0WD (NRND) has a maximum CPU frequency of 160 MHz.
- When Wi-Fi is enabled, the chip switches between Active and Modem-sleep modes. Therefore, power consumption changes accordingly.
- In Modem-sleep mode, the CPU frequency changes automatically. The frequency depends on the CPU load and the peripherals used.
- During Deep-sleep, when the ULP coprocessor is powered on, peripherals such as GPIO and RTC I2C are able to operate.
- When the system works in the ULP sensor-monitored pattern, the ULP coprocessor works with the ULP sensor periodically and the ADC works with a duty cycle of 1%, so the power consumption is 100 μA.

5.3.2. Ultra-Low-Power Coprocessor

The ULP coprocessor and RTC memory remain powered on during the Deep-sleep mode. Hence, the developer can store a program for the ULP coprocessor in the RTC slow memory to access the peripheral devices, internal timers and internal sensors during the Deep-sleep mode. This is useful for designing applications where the CPU needs to be woken up by an external event, or a timer, or a combination of the two, while maintaining minimal power consumption.

For details, see *ESP32 Technical Reference Manual* > Chapter *ULP Coprocessor*.

5.4. Timers and Watchdogs

5.4.1. General Purpose Timers

There are four general-purpose timers embedded in the chip. 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 feature:

- A 16-bit clock prescaler, from 2 to 65536
- A 64-bit timer
- Configurable up/down timer: incrementing or decrementing
- Halt and resume of time-base counter
- Auto-reload at alarming
- Software-controlled instant reload
- Level and edge interrupt generation

For details, see *ESP32 Technical Reference Manual* > Chapter *Timer Group*.

5.4.2. Watchdog Timers

The chip has three watchdog timers: one in each of the two timer modules (called the Main Watchdog Timer, or MWDT) and one in the RTC module (called the RTC Watchdog Timer, or RWDT). These watchdog timers are intended to recover from an unforeseen fault causing the application program to abandon its normal sequence. A watchdog timer has four stages. Each stage may trigger one of three or four possible actions upon the expiry of its programmed time period, unless the watchdog is fed or disabled. The actions are:

interrupt, CPU reset, core reset, and system reset. Only the RWDT can trigger the system reset, and is able to reset the entire chip, including the RTC itself. A timeout value can be set for each stage individually.

During flash boot the RWDT and the first MWDT start automatically in order to detect, and recover from, booting problems.

The watchdogs have the following features:

- Four stages, each of which can be configured or disabled separately

- A programmable time period for each stage
- One of three or four possible actions (interrupt, CPU reset, core reset, and system reset) upon the expiry of each stage
- 32-bit expiry counter
- Write protection that prevents the RWDT and MWDT configuration from being inadvertently altered
- SPI flash boot protection

If the boot process from an SPI flash does not complete within a predetermined time period, the watchdog will reboot the entire system.

For details, see *ESP32 Technical Reference Manual >* Chapter *Watchdog Timers*.

5.5. Cryptographic Hardware Accelerators

ESP32 is equipped with hardware accelerators of general algorithms, such as AES (FIPS PUB 197), SHA (FIPS PUB 180-4), RSA, and ECC. The chip also supports independent arithmetic, such as large-number modular multiplication and large-number multiplication. The maximum operation length for RSA, ECC, large-number modular multiplication, and large-number multiplication is 4096 bits.

The hardware accelerators greatly improve operation speed and reduce software complexity. They also support code encryption and dynamic decryption, which ensures that code in the flash will not be hacked.

5.6. Radio and Wi-Fi

The radio module consists of the following blocks:

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

5.6.1. 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 in the chip.

5.6.2. 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, enabling state-of-the-art performance in delivering up to +20.5 dBm of power for an 802.11b transmission and +18 dBm for an 802.11n transmission. Additional calibrations are integrated to cancel any radio imperfections, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities

- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the amount of time required for product testing, and render the testing equipment unnecessary.

5.6.3. 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.

5.6.4. Wi-Fi Radio and Baseband

ESP32 implements a TCP/IP and 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 with minimal host interaction to minimize the active-duty period.

The ESP32 Wi-Fi Radio and Baseband support the following features:

- 802.11b/g/n
- 802.11n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11n MCS32 (RX)
- 802.11n 0.4 μs guard-interval
- up to 150 Mbps of data rate
- Receiving STBC 2×1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity

  ESP32 supports antenna diversity with an external RF switch. One or more GPIOs control the RF switch and selects the best antenna to minimize the effects of channel fading.

5.6.5. Wi-Fi MAC

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

- Four virtual Wi-Fi interfaces
- Simultaneous Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- RTS protection, CTS protection, Immediate Block ACK
- Defragmentation
- TX/RX A-MPDU, RX A-MSDU
- TXOP
- WMM
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4) and CRC
- Automatic beacon monitoring (hardware TSF)

5.7. Bluetooth

The chip integrates a Bluetooth link controller and Bluetooth baseband, which carry out the baseband protocols and other low-level link routines, such as modulation/demodulation, packet processing, bit stream processing, frequency hopping, etc.

5.7.1. Bluetooth Radio and Baseband

The Bluetooth Radio and Baseband support the following features:

- Class-1, class-2 and class-3 transmit output powers, and a dynamic control range of up to 21 dB
- $\pi/4$ DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with a minimum sensitivity of -94 dBm
- Class-1 operation without external PA
- Internal SRAM allows full-speed data-transfer, mixed voice and data, and full piconet operation
- Logic for forward error correction, header error control, access code correlation, CRC, demodulation,
- encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO, and AFH
- A-law, μ-law, and CVSD digital audio CODEC in PCM interface
- SBC audio CODEC
- Power management for low-power applications
- SMP with 128-bit AES

5.7.2. Bluetooth Interface

- Provides UART HCI interface, up to 4 Mbps
- Provides SDIO/SPI HCI interface
- Provides PCM/I2S audio interface

5.7.3. Bluetooth Stack

The Bluetooth stack of the chip is compliant with the Bluetooth v4.2 BR/EDR and Bluetooth LE specifications.

5.7.4. Bluetooth Link Controller

The link controller operates in three major states: standby, connection and sniff. It enables multiple connections, and other operations, such as inquiry, page, and secure simple-pairing, and therefore enables Piconet and Scatternet. Below are the features:

- Classic Bluetooth
  - Device Discovery (inquiry, and inquiry scan)
  - Connection establishment (page, and page scan)

  - Multi-connections
  - Asynchronous data reception and transmission
  - Synchronous links (SCO/eSCO)
  - Master/Slave Switch
  - Adaptive Frequency Hopping and Channel assessment
  - Broadcast encryption
  - Authentication and encryption
  - Secure Simple-Pairing
  - Multi-point and scatternet management
  - Sniff mode
  - Connectionless Slave Broadcast (transmitter and receiver)
  - Enhanced power control
  - Ping
- Bluetooth Low Energy
  - Advertising
  - Scanning
  - Simultaneous advertising and scanning
  - Multiple connections
  - Asynchronous data reception and transmission
  - Adaptive Frequency Hopping and Channel assessment
  - Connection parameter update
  - Data Length Extension
  - Link Layer Encryption
  - LE Ping

5.8. Digital Peripherals

5.8.1. General Purpose Input / Output Interface (GPIO)

ESP32 has 34 GPIO pins which can be assigned various functions by programming the appropriate registers. There are several kinds of GPIOs: digital-only, analog-enabled, capacitive-touch-enabled, etc. Analog-enabled GPIOs and Capacitive-touch-enabled GPIOs can be configured as digital GPIOs.

Most of the digital GPIOs can be configured as internal pull-up or pull-down, or set to high impedance. When configured as an input, the input value can be read through the register. The input can also be set to edge-trigger or level-trigger to generate CPU interrupts. Most of the 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 SDIO, UART, SPI, etc. (More details can be found in the Appendix, Table IO_MUX.) For low-power operations, the GPIOs can be set to hold their states.

For details, see Section Peripheral *Pin Configurations* and *ESP32 Technical Reference Manual* > Chapter *IO_MUX and GPIO Matrix*.

5.8.2. Serial Peripheral Interface (SPI)

ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes.

Features of General Purpose SPI (GP-SPI)

- Programmable data transfer length, in multiples of 1 byte
- Four-line full-duplex/half-duplex communication and three-line half-duplex communication support
- Master mode and slave mode
- Programmable CPOL and CPHA
- Programmable clock

For details, see *ESP32 Technical Reference Manual* > Chapter *SPI Controller*.

Pin Assignment

For SPI, the pins are multiplexed with GPIO6 ~ GPIO11 via the IO MUX. For HSPI, the pins are multiplexed with GPIO2, GPIO4, GPIO12 - GPIO15 via the IO MUX. For VSPI, the pins are multiplexed with GPIO5, GPIO18 - GPIO19, GPIO21 - GPIO23 via the IO MUX.

For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.3. Universal Asynchronous Receiver Transmitter (UART)

The UART in the ESP32 chip facilitates the transmission and reception of asynchronous serial data between the chip and external UART devices. It consists of two UARTs in the main system, and one low-power LP UART.

Feature List

- Programmable baud rate
- RAM shared by TX FIFOs and RX FIFOs
- Supports input baud rate self-check
- Support for various lengths of data bits and stop bits
- Parity bit support
- Asynchronous communication (RS232 and RS485) and IrDA support
- Supports DMA to communicate data in high speed
- Supports UART wake-up
- Supports both software and hardware flow control

For details, see *ESP32 Technical Reference Manual* > Chapter *UART Controller*.

Pin Assignment

The pins for UART can be chosen from any GPIOs via the GPIO Matrix.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.4. I2C Interface

ESP32 has two I2C bus interfaces which can serve as I2C master or slave, depending on the user�s configuration.

Feature List

- Two I2C controllers: one in the main system and one in the low-power system
- Standard mode (100 Kbit/s)
- Fast mode (400 Kbit/s)
- Up to 5 MHz, yet constrained by SDA pull-up strength
- Support for 7-bit and 10-bit addressing, as well as dual address mode
- Supports continuous data transmission with disabled Serial Clock Line (SCL)
- Supports programmable digital noise filter

Users can program command registers to control I2C interfaces, so that they have more flexibility. For details, see *ESP32 Technical Reference Manual* > Chapter *I2C Controller*.

Pin Assignment

For regular I2C, the pins used can be chosen from any GPIOs via the GPIO Matrix.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.5. I2S Interface

The I2S Controller in the ESP32 chip provides a flexible communication interface for streaming digital data in multimedia applications, particularly digital audio applications.

Feature List

- Master mode and slave mode
- Full-duplex and half-duplex communications
- A variety of audio standards supported
- Configurable high-precision output clock
- Supports PDM signal input and output
- Configurable data transmit and receive modes

For details, see *ESP32 Technical Reference Manual* > Chapter *I2S Controller*.

Pin Assignment

The pins for the I2S Controller can be chosen from any GPIOs via the GPIO Matrix.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.6. Remote Control Peripheral

The Remote-Control Peripheral (RMT) controls the transmission and reception of infrared remote-control signals.

Feature List

- Eight channels for sending and receiving infrared remote control signals
- Independent transmission and reception capabilities for each channel
- Clock divider counter, state machine, and receiver for each RX channel
- Supports various infrared protocols

For details, see *ESP32 Technical Reference Manual* > Chapter *Remote Control Peripheral*.

Pin Assignment

The pins for the Remote-Control Peripheral can be chosen from any GPIOs via the GPIO Matrix.

For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.7. Pulse Counter

The pulse counter is designed to count input pulses by tracking rising and falling edges of the input pulse signal.

Feature List

- Four independent pulse counters with two channels each
- Counter modes: increment, decrement, or disable
- Glitch filtering for input pulse signals and control signals
- Selection between counting on rising or falling edges of the input pulse signal

For details, see *ESP32 Technical Reference Manual* > Chapter *Pulse Count Controller*.

Pin Assignment

The pins for the Pulse Count Controller can be chosen from any GPIOs via the GPIO Matrix.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.8. LED PWM Controller

The LED PWM Controller (LEDC) is designed to generate PWM signals for LED control.

Feature List

- Sixteen independent PWM generators
- Maximum PWM duty cycle resolution of 20 bits
- Eight independent timers with 20-bit counters, configurable fractional clock dividers and counter
- overflow values
- Adjustable phase of PWM signal output
- PWM duty cycle dithering
- Automatic duty cycle fading

For details, see *ESP32 Technical Reference Manual* > Chapter *LED PWM Controller*.

Pin Assignment

The pins for the LED PWM Controller can be chosen from any GPIOs via the GPIO Matrix.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.9. Motor Control PWM

The Pulse Width Modulation (PWM) controller can be used for driving digital motors and smart lights. The controller consists of PWM timers, the PWM operator and a dedicated capture sub-module. Each timer provides timing in synchronous or independent form, and each PWM operator generates a waveform for one PWM channel. The dedicated capture sub-module can accurately capture events with external timing.

Feature List

- Three PWM timers for precise timing and frequency control
  - Every PWM timer has a dedicated 8-bit clock prescaler
  - The 16-bit counter in the PWM timer can work in count-up mode, count-down mode, or count-up-down mode
  - A hardware sync can trigger a reload on the PWM timer with a phase register. It will also trigger the prescaler□restart, so that the timer□s clock can also be synced, with selectable hardware synchronization source
- Three PWM operators for generating waveform pairs
  - Six PWM outputs to operate in several topologies
  - Configurable dead time on rising and falling edges; each set up independently
  - Modulating of PWM output by high-frequency carrier signals, useful when gate drivers are insulated with a transformer
- Fault Detection module
  - Programmable fault handling in both cycle-by-cycle mode and one-shot mode
  - A fault condition can force the PWM output to either high or low logic levels
- Capture module for hardware-based signal processing
  - Speed measurement of rotating machinery
  - Measurement of elapsed time between position sensor pulses
  - Period and duty cycle measurement of pulse train signals
  - Decoding current or voltage amplitude derived from duty-cycle-encoded signals of current/voltage sensors
  - Three individual capture channels, each of which with a 32-bit time-stamp register
  - Selection of edge polarity and prescaling of input capture signals
  - The capture timer can sync with a PWM timer or external signals

For details, see *ESP32 Technical Reference Manual* > Chapter *Motor Control PWM*.

Pin Assignment

The pins for the Motor Control PWM can be chosen from any GPIOs via the GPIO Matrix.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.10. SD/SDIO/MMC Host Controller

An SD/SDIO/MMC host controller is available on ESP32.

Feature List

- Supports two external cards
- Supports SD Memory Card standard: version 3.0 and version 3.01)
- Supports SDIO Version 3.0
- Supports Consumer Electronics Advanced Transport Architecture (CE-ATA Version 1.1)
- Supports Multimedia Cards (MMC version 4.41, eMMC version 4.5 and version 4.51)

The controller allows up to 80 MHz clock output in three different data-bus modes: 1-bit, 4-bit, and 8-bit modes. It supports two SD/SDIO/MMC4.41 cards in a 4-bit data-bus mode. It also supports one SD card operating at 1.8 V.

For details, see *ESP32 Technical Reference Manual* > Chapter *SD/MMC Host Controller*.

Pin Assignment

The pins for SD/SDIO/MMC Host Controller are multiplexed with GPIO2, GPIO6 ~ GPIO15 via IO MUX. For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.11. SDIO/SPI Slave Controller

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

The SDIO/SPI slave controller supports the following features:

- SPI, 1-bit SDIO, and 4-bit SDIO transfer modes over the full clock range from 0 to 50 MHz
- Configurable sampling and driving clock edge
- Special registers for direct access by host
- Interrupts to host for initiating 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 details, see *ESP32 Technical Reference Manual* > Chapter *SDIO Slave Controller*.

Pin Assignment

The pins for SDIO/SPI Slave Controller are multiplexed with GPIO2, GPIO6 ~ GPIO15 via IO MUX.
For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

Two-wire Automotive Interface

The Two-wire Automotive Interface (TWAI®) is a multi-master, multi-cast communication protocol designed for automotive applications. The TWAI controller facilitates the communication based on this protocol.

Feature List

- 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 25 Kbit/s to 1 Mbit/s in chip revision v0.0/v1.0/v1.1
  - from 12.5 Kbit/s to 1 Mbit/s in chip revision v3.0/v3.1
- 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 details, see *ESP32 Technical Reference Manual* > Chapter *Two-wire Automotive Interface (TWAI)*.

Pin Assignment

The pins for the Two-wire Automotive Interface can be chosen from any GPIOs via the GPIO Matrix. For more information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.8.12. Ethernet MAC Interface

An IEEE-802.3-2008-compliant Media Access Controller (MAC) is provided for Ethernet LAN communications. ESP32 requires an external physical interface device (PHY) to connect to the physical LAN bus (twisted-pair, fiber, etc.). The PHY is connected to ESP32 through 17 signals of MII or nine signals of RMII.

Feature List

- 10 Mbps and 100 Mbps rates
- Dedicated DMA controller allowing high-speed transfer between the dedicated SRAM and Ethernet MAC
- Tagged MAC frame (VLAN support)
- Half-duplex (CSMA/CD) and full-duplex operation
- MAC control sublayer (control frames)
- 32-bit CRC generation and removal
- Several address-filtering modes for physical and multicast address (multicast and group addresses)

- 32-bit status code for each transmitted or received frame
- Internal FIFOs to buffer transmit and receive frames. The transmit FIFO and the receive FIFO are both 512 words (32-bit)
- Hardware PTP (Precision Time Protocol) in accordance with IEEE 1588 2008 (PTP V2)
- 25 MHz/50 MHz clock output

For details, see *ESP32 Technical Reference Manual* > Chapter *Ethernet Media Access Controller (MAC)*.

Pin Assignment

For information about the pin assignment of Ethernet MAC Interface, see Section *Peripheral Pin Configurations.*

5.9. Analog Peripherals

5.9.1. Analog-to-Digital Converter (ADC)

ESP32 integrates two 12-bit SAR ADCs and supports measurements on 18 channels (analog-enabled pins). The ULP coprocessor in ESP32 is also designed to measure voltage, while operating in the sleep mode, which enables low-power consumption. The CPU can be woken up by a threshold setting and/or via other triggers.

Parameter	Description	Min	Max	Unit
DNL (Differential nonlinearity)	RTC controller; ADC connected to an external 100 nF capacitor; DC signal input; ambient temperature at 25 °C; Wi-Fi&Bluetooth off	-7	7	LSB
INL (Integral nonlinearity)		-12	12	LSB
Sampling rate	RTC controller	—	200	ksps
	DIG controller	—	2	Msps

Table 20: ADC Characteristics

Notes:

- When atten = 3 and the measurement result is above 3000 (voltage at approx. 2450 mV), the ADC accuracy will be worse than described in the table above.
- To get better DNL results, users can take multiple sampling tests with a filter, or calculate the average value.
- The input voltage range of GPIO pins within VDD3P3_RTC domain should strictly follow the DC characteristics. Otherwise, measurement errors may be introduced, and chip performance may be affected.

By default, there are ±6% differences in measured results between chips. ESP-IDF provides couple of calibration methods for ADC1. Results after calibration using eFuse Vref value are shown in Table 21 For higher accuracy, users may apply other calibration methods provided in ESP-IDF, or implement their own.

Parameter	Description	Min	Max	Unit
Total error	Atten = 0, effective measurement range of 100 ～950 mV	-23	23	mV
	Atten = 1, effective measurement range of 100 ～1250 mV	-30	30	mV
	Atten = 2, effective measurement range of 150 ～1750 mV	-40	30	mV
	Atten = 3, effective measurement range of 150 ～2450 mV	-60	60	mV

Table 21: ADC Calibration Results

For details, see *ESP32 Technical Reference Manual* > Chapter *On-Chip Sensors and Analog Signal Processing*.

Pin Assignment

With appropriate settings, the ADCs can be configured to measure voltage on 18 pins maximum. For detailed information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.9.2. Digital-to-Analog Converter (DAC)

Two 8-bit DAC channels can be used to convert two digital signals into two analog voltage signal outputs. The design structure is composed of integrated resistor strings and a buffer. This dual DAC supports power supply as input voltage reference. The two DAC channels can also support independent conversions.

For details, see *ESP32 Technical Reference Manual* > Chapter *On-Chip Sensors and Analog Signal Processing*.

Pin Assignment

The DAC can be configured by GPIO 25 and GPIO 26. For detailed information about the pin assignment, see Section *Peripheral Pin Configurations.*

5.9.3. Touch Sensor

ESP32 has 10 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.

Pin Assignment

The 10 capacitive-sensing GPIOs are listed in Table 22.

Capacitive-Sensing Signal Name	Pin Name
T0	GPIO4
T1	GPIO0
T2	GPIO2
T3	MTDO
T4	MTCK
T5	MTDI
T6	MTMS

Capacitive-Sensing Signal Name	Pin Name
T7	GPIO27
T8	32K_XN
T9	32K_XP

Table 22: Capacitive-Sensing GPIOs Available on ESP32

For details, see *ESP32 Technical Reference Manual* > Chapter *On-Chip Sensors and Analog Signal Processing*.

5.10. Peripheral Pin Configurations

Interface	Signal	Pin	Function
ADC	ADC1_CH0	SENSOR_VP	Two 12-bit SAR ADCs
	ADC1_CH1	SENSOR_CAPP	
	ADC1_CH2	SENSOR_CAPN	
	ADC1_CH3	SENSOR_VN	
	ADC1_CH4	32K_XP	
	ADC1_CH5	32K_XN	
	ADC1_CH6	VDET_1	
	ADC1_CH7	VDET_2	
	ADC2_CH0	GPIO4	
	ADC2_CH1	GPIO0	
	ADC2_CH2	GPIO2	
	ADC2_CH3	MTDO	
	ADC2_CH4	MTCK	
	ADC2_CH5	MTDI	
	ADC2_CH6	MTMS	
	ADC2_CH7	GPIO27	
	ADC2_CH8	GPIO25	
	ADC2_CH9	GPIO26	
DAC	DAC_1	GPIO25	Two 8-bit DACs
	DAC_2	GPIO26	

Interface	Signal	Pin	Function
Touch Sensor	TOUCH0	GPIO4	Capacitive touch sensors
	TOUCH1	GPIO0	
	TOUCH2	GPIO2	
	TOUCH3	MTDO	
	TOUCH4	MTCK	
	TOUCH5	MTDI	
	TOUCH6	MTMS	
	TOUCH7	GPIO27	
	TOUCH8	32K_XN	
	TOUCH9	32K_XP	
JTAG	MTDI	MTDI	JTAG for software debugging
	MTCK	MTCK	
	MTMS	MTMS	
	MTDO	MTDO	
SD/SDIO/MMC Host Controller	HS2_CLK	MTMS	Supports SD memory card V3.01 standard
	HS2_CMD	MTDO	
	HS2_DATA0	GPIO2	
	HS2_DATA1	GPIO4	
	HS2_DATA2	MTDI	
	HS2_DATA3	MTCK	
UART	U0RXD_in	Any GPIO Pins	Three UART devices with hardware flow-control and DMA
	U0CTS_in		
	U0DSR_in		
	U0TXD_out		
	U0RTS_out		
	U0DTR_out		
	U1RXD_in		
	U1CTS_in		
	U1TXD_out		
	U1RTS_out		
	U2RXD_in		
	U2CTS_in		
	U2TXD_out		
	U2RTS_out		
I2C	I2CEXT0_SCL_in	Any GPIO Pins	Two I2C devices in slave or master mode
	I2CEXT0_SDA_in		
	I2CEXT1_SCL_in		
	I2CEXT1_SDA_in		
	I2CEXT0_SCL_out		

Interface	Signal	Pin	Function
I2C	I2CEXT0_SDA_out	Any GPIO Pins	Two I2C devices in slave or master mode
	I2CEXT1_SCL_out		
	I2CEXT1_SDA_out		
LED PWM	ledc_hs_sig_out0~7	Any GPIO Pins	16 independent channels @80 MHz clock/RTC CLK. Duty accuracy: 16 bits.
	ledc_ls_sig_out0~7		
I2S	I2S0I_DATA_in0~15	Any GPIO Pins	Stereo input and output from/to the audio codec; parallel LCD data output; parallel camera data input. Note: I2S0_CLK and I2S1_CLK can only be mapped to GPIO0, U0RXD (GPIO3), or U0TXD (GPIO1) via IO MUX by selecting GPIO functions CLK_OUT1, CLK_OUT2, and CLK_OUT3. For more information, see ESP32 Technical Reference Manual > Chapter *IO_MUX and GPIO Matrix* > Table *IO MUX Pad Summary*.
	I2S0O_BCK_in		
	I2S0O_WS_in		
	I2S0I_BCK_in		
	I2S0I_WS_in		
	I2S0I_H_SYNC		
	I2S0I_V_SYNC		
	I2S0I_H_ENABLE		
	I2S0O_BCK_out		
	I2S0O_WS_out		
	I2S0I_BCK_out		
	I2S0I_WS_out		
	I2S0O_DATA_out0~23		
	I2S1I_DATA_in0~15		
	I2S1O_BCK_in		
	I2S1O_WS_in		
	I2S1I_BCK_in		
	I2S1I_WS_in		
	I2S1I_H_SYNC		
	I2S1I_V_SYNC		
	I2S1I_H_ENABLE		
	I2S1O_BCK_out		
	I2S1O_WS_out		
	I2S1I_BCK_out		
	I2S1I_WS_out		
	I2S1O_DATA_out0~23		
	I2S0_CLK	GPIO0, U0RXD, or U0TXD	
	I2S1_CLK		
RMT	RMT_SIG_IN0~7	Any GPIO Pins	Eight channels for an IR transmitter and receiver of various waveforms
	RMT_SIG_OUT0~7		

Interface	Signal	Pin	Function
General Purpose SPI	HSPIQ_in/_out	Any GPIO Pins	Standard SPI consists of clock, chip-select, MOSI and MISO. These SPIs can be connected to LCD and other external devices. They support the following features: • Both master and slave modes; • Four sub-modes of the SPI transfer format; • Configurable SPI frequency; • Up to 64 bytes of FIFO and DMA.
	HSPID_in/_out		
	HSPICLK_in/_out		
	HSPI_CS0_in/_out		
	HSPI_CS1_out		
	HSPI_CS2_out		
	VSPIQ_in/_out		
	VSPID_in/_out		
	VSPICLK_in/_out		
	VSPI_CS0_in/_out		
	VSPI_CS1_out		
	VSPI_CS2_out		
Parallel QSPI	SPIHD	SD_DATA_2	Supports Standard SPI, Dual SPI, and Quad SPI that can be connected to the external flash and SRAM
	SPIWP	SD_DATA_3	
	SPICS0	SD_CMD	
	SPICLK	SD_CLK	
	SPIQ	SD_DATA_0	
	SPID	SD_DATA_1	
	HSPICLK	MTMS	
	HSPICS0	MTDO	
	HSPIQ	MTDI	
	HSPID	MTCK	
	HSPIHD	GPIO4	
	HSPIWP	GPIO2	
	VSPICLK	GPIO18	
	VSPICS0	GPIO5	
	VSPIQ	GPIO19	
	VSPID	GPIO23	
	VSPIHD	GPIO21	
	VSPIWP	GPIO22	
EMAC	EMAC_TX_CLK	GPIO0	Ethernet MAC with MII/RMII interface
	EMAC_RX_CLK	GPIO5	
	EMAC_TX_EN	GPIO21	
	EMAC_TXD0	GPIO19	
	EMAC_TXD1	GPIO22	
	EMAC_TXD2	MTMS	
	EMAC_TXD3	MTDI	
	EMAC_RX_ER	MTCK	
	EMAC_RX_DV	GPIO27	

Interface	Signal	Pin	Function
EMAC	EMAC_RXD0	GPIO25	Ethernet MAC with MII/RMII interface
	EMAC_RXD1	GPIO26	
	EMAC_RXD2	U0TXD	
	EMAC_RXD3	MTDO	
	EMAC_CLK_OUT	GPIO16	
	EMAC_CLK_OUT_180	GPIO17	
	EMAC_TX_ER	GPIO4	
	EMAC_MDC_out	Any GPIO Pins	
	EMAC_MDI_in	Any GPIO Pins	
	EMAC_MDO_out	Any GPIO Pins	
	EMAC_CRS_out	Any GPIO Pins	
	EMAC_COL_out	Any GPIO Pins	
Pulse Counter	pcnt_sig_ch0_in0	Any GPIO Pins	Operating in seven different modes, the pulse counter captures pulse and counts pulse edges.
	pcnt_sig_ch1_in0		
	pcnt_ctrl_ch0_in0		
	pcnt_ctrl_ch1_in0		
	pcnt_sig_ch0_in1		
	pcnt_sig_ch1_in1		
	pcnt_ctrl_ch0_in1		
	pcnt_ctrl_ch1_in1		
	pcnt_sig_ch0_in2		
	pcnt_sig_ch1_in2		
	pcnt_ctrl_ch0_in2		
	pcnt_ctrl_ch1_in2		
	pcnt_sig_ch0_in3		
	pcnt_sig_ch1_in3		
	pcnt_ctrl_ch0_in3		
	pcnt_ctrl_ch1_in3		
	pcnt_sig_ch0_in4		
	pcnt_sig_ch1_in4		
	pcnt_ctrl_ch0_in4		
	pcnt_ctrl_ch1_in4		
	pcnt_sig_ch0_in5		
	pcnt_sig_ch1_in5		
	pcnt_ctrl_ch0_in5		
	pcnt_ctrl_ch1_in5		
	pcnt_sig_ch0_in6		
	pcnt_sig_ch1_in6		
	pcnt_ctrl_ch0_in6		

Interface	Signal	Pin	Function
Pulse Counter	pcnt_ctrl_ch1_in6	Any GPIO Pins	Operating in seven different modes, the pulse counter captures pulse and counts pulse edges.
	pcnt_sig_ch0_in7		
	pcnt_sig_ch1_in7		
	pcnt_ctrl_ch0_in7		
	pcnt_ctrl_ch1_in7		
TWAI	twai_rx	Any GPIO Pins	Compatible with ISO 11898-1 protocol (CAN Specification 2.0)
	twai_tx		
	twai_bus_off_on		
	twai_clkout		

Table 23: Peripheral Pin Configurations

6. Boot Configurations

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

- **Chip boot mode**
– Strapping pin: GPIO0 and GPIO2
- **Internal LDO (VDD_SDIO) Voltage**
– Strapping pin: MTDI
– eFuse bit: EFUSE_SDIO_FORCE and EFUSE_SDIO_TIEH
- **U0TXD printing**
– Strapping pin: MTDO
- **Timing of SDIO Slave**
– Strapping pin: MTDO and GPIO5
- **JTAG signal source**
– Strapping pin: MTCK, MTMS, MTDI and MTDO
– eFuse bit: EFUSE_DISABLE_JTAG

The default values of all the above eFuse bits are 0, which means that they are not burnt. Given that eFuse isone-time programmable, once an eFuse bit is programmed to 1, it can never be reverted to 0. For how to program eFuse bits, please refer to *ESP32 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.

Strapping Pin	Default Configuration	Bit Value
GPIO0	Pull-up	1
GPIO2	Pull-down	0
MTDI	Pull-down	0
MTDO	Pull-up	1
GPIO5	Pull-up	1

Table 24: Default Configuration of Strapping Pins

To change the bit values, the strapping pins should be connected to external pull-down/pull-up resistances. If the ESP32 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 25 and Figure 24.

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.	1

Table 25: Description of Timing Parameters for the Strapping Pins

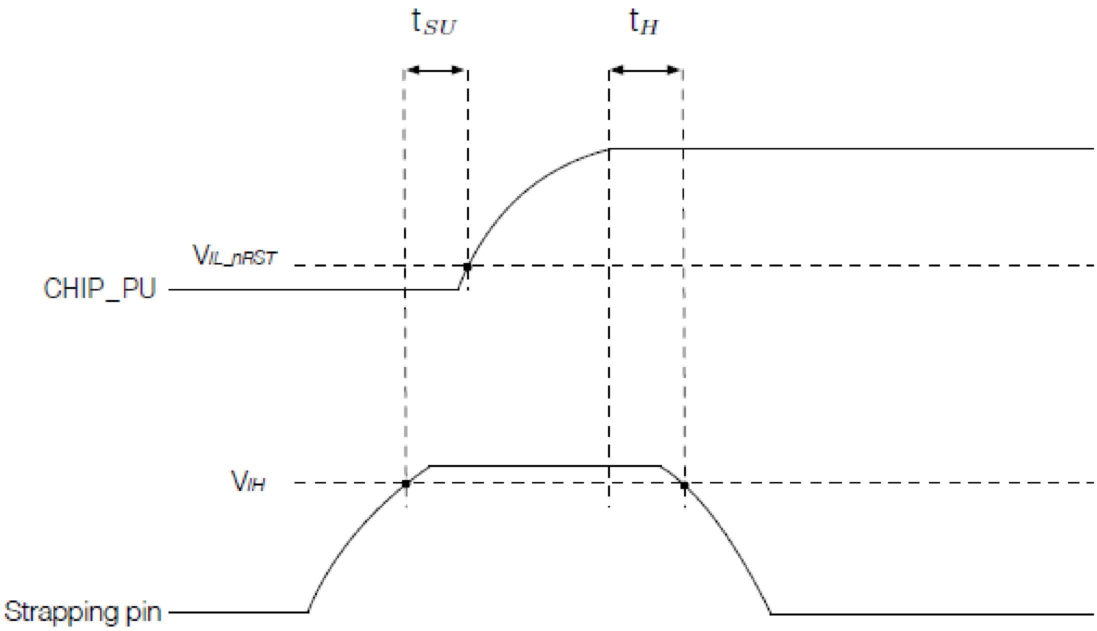


Figure 6: Visualization of Timing Parameters for the Strapping Pins

6.1. Chip Boot Mode Control

GPIO0 and GPIO2 control the boot mode after the reset is released. See Table 26 *Chip Boot Mode Control*.

Boot Mode	**GPIO0**	**GPIO2**
SPI Boot Mode	1	Any value
Joint Download Boot Mode 2	0	0

Table 26: Chip Boot Mode Control

[1] Bold marks the default value and configuration.

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

- SDIO Download Boot
- UART Download Boot

In Joint Download Boot mode, the detailed boot flow of the chip is put below 25.

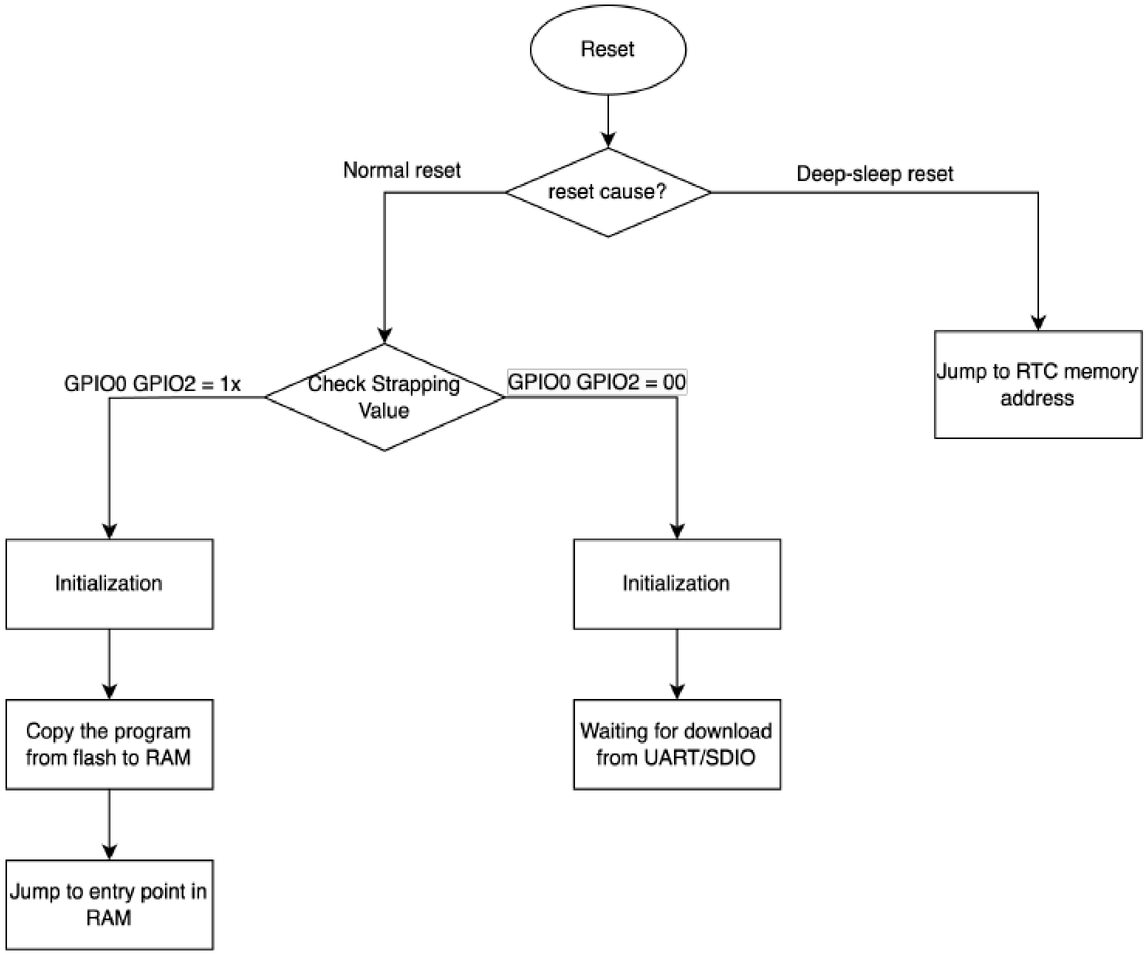


Figure 7: Chip Boot Flow

6.2. Internal LDO (VDD_SDIO) Voltage Control

The required VDD_SPI voltage for the chips of the ESP32 Series.

MTDI is used to select the VDD_SDIO power supply voltage at reset:

- MTDI = 0 (by default), VDD_SDIO pin is powered directly from VDD3P3_RTC. Typically, this voltage is 3.3 V.
- MTDI = 1, VDD_SDIO pin is powered from internal 1.8 V LDO.

This functionality can be overridden by setting EFUSE_SDIO_FORCE to 1, in which case the EFUSE_SDIO_TIEH determines the VDD_SDIO voltage:

- EFUSE_SDIO_TIEH = 0, VDD_SDIO connects to 1.8 V LDO.
- EFUSE_SPI_TIEH = 1, VDD_SDIO connects to VDD3P3_RTC.

6.3. U0TXD Printing Control

During booting, the strapping pin MTDO can be used to control the U0TXD Printing, as Table 27 shows.

U0TXD Printing Control	MTDO
Enabled [1]	1
Disabled	0

[1] Bold mark the default value and configuration

Table 27: U0TXD Printing Control

6.4. Timing Control of SDIO Slave

The strapping pin MTDO and GPIO5 can be used to control the timing of SDIO slave, see Table 28 *Timing Control of SDIO Slave*.

Edge behavior	MTDO	GPIO5
Falling edge sampling, falling edge output	0	0
Falling edge sampling, rising edge output	0	1
Rising edge sampling, falling edge output	1	0
Rising edge sampling, rising edge output	1	1

[1] Bold mark the default value and configuration.

Table 28: Timing Control of SDIO Slave

6.5. JTAG Signal Source Control

The strapping pin MTCK, MTMS, MTDI and MTDO can be used to control the source of JTAG signals during the early boot process.

If EFUSE_DISABLE_JTAG is set to 1, the source of JTAG signals can be disabled.

7. Schematics

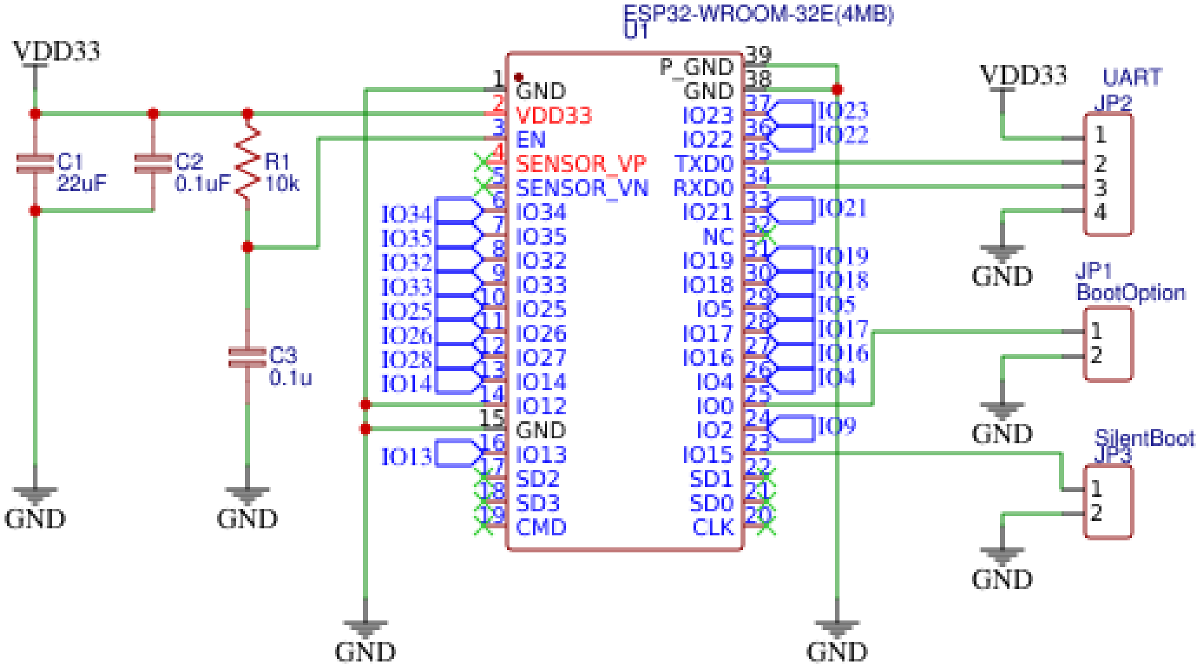


Figure 8: ESP32-WROOM-32E Schematic

8. Physical Dimensions

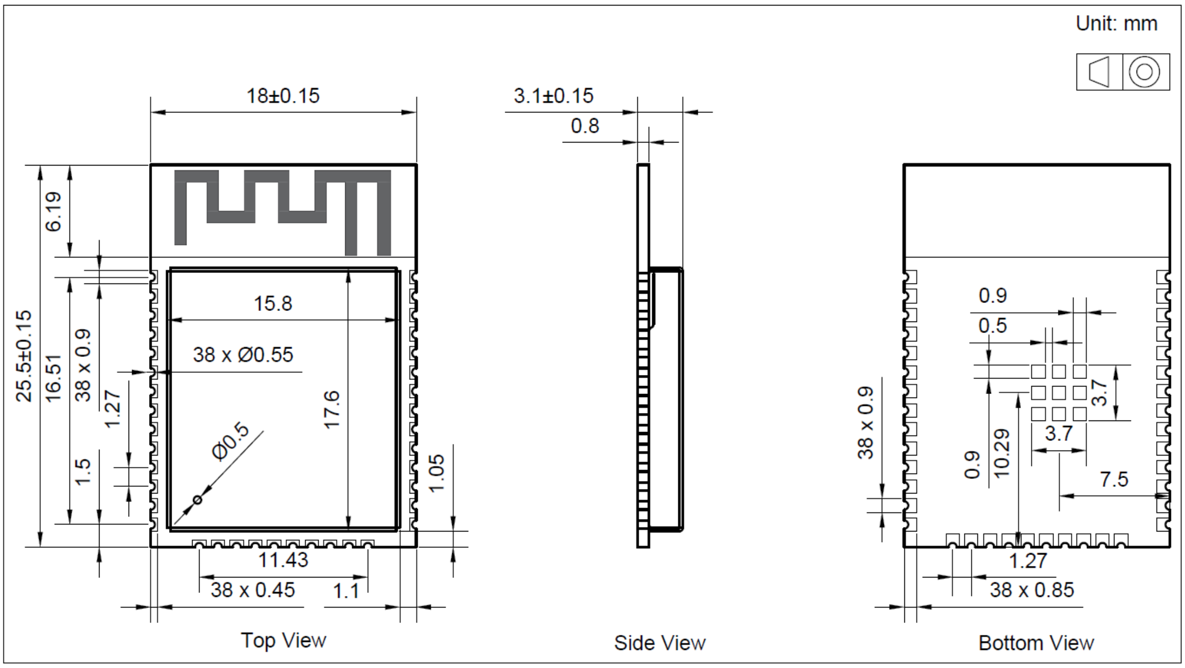
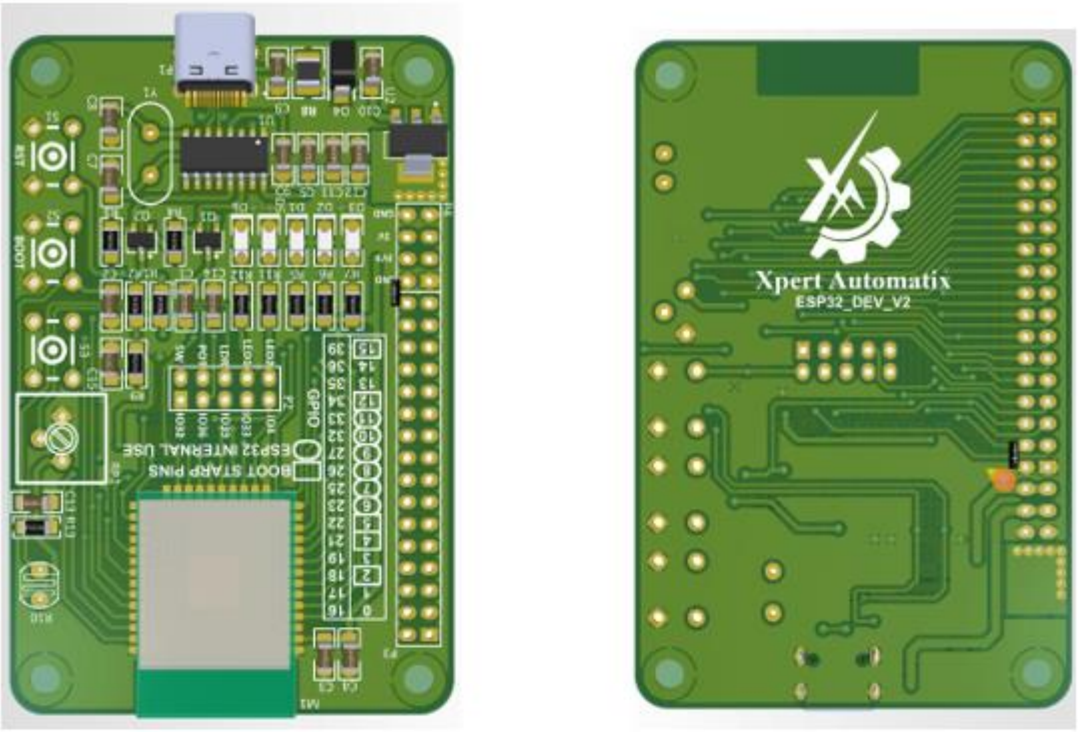


Figure 9: ESP32-WROOM-32E Physical Dimensions



FRONT SIDE VIEW

BACK SIDE VIEW

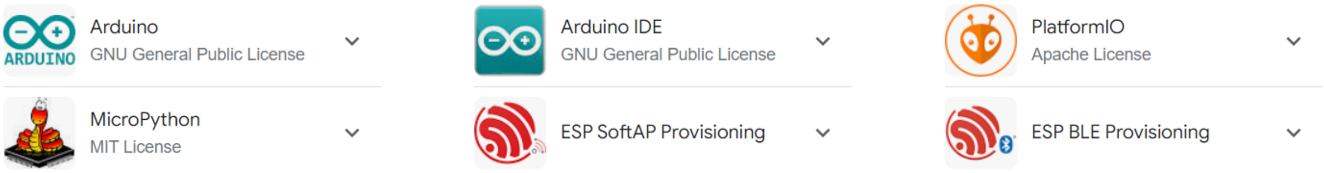
Figure 10: ESP32-WROOM-32E DEV MODULE VIEW

9. Basic program Knowledge Guidance

Requirements

- Software: Arduino IDE (We Choose this Software)
- Hardware: ESP32 Dev Module
- Data transfer cable: Type A to Type Micro B
- Programming Language: Embedded C, Micro Python, Lua

Software to program for ESP32



Arduino
GNU General Public License

Arduino IDE
GNU General Public License

PlatformIO
Apache License

MicroPython
MIT License

ESP SoftAP Provisioning

ESP BLE Provisioning

Figure 11: ESP32 PROGRAMMING SOFTWARES

How to program?

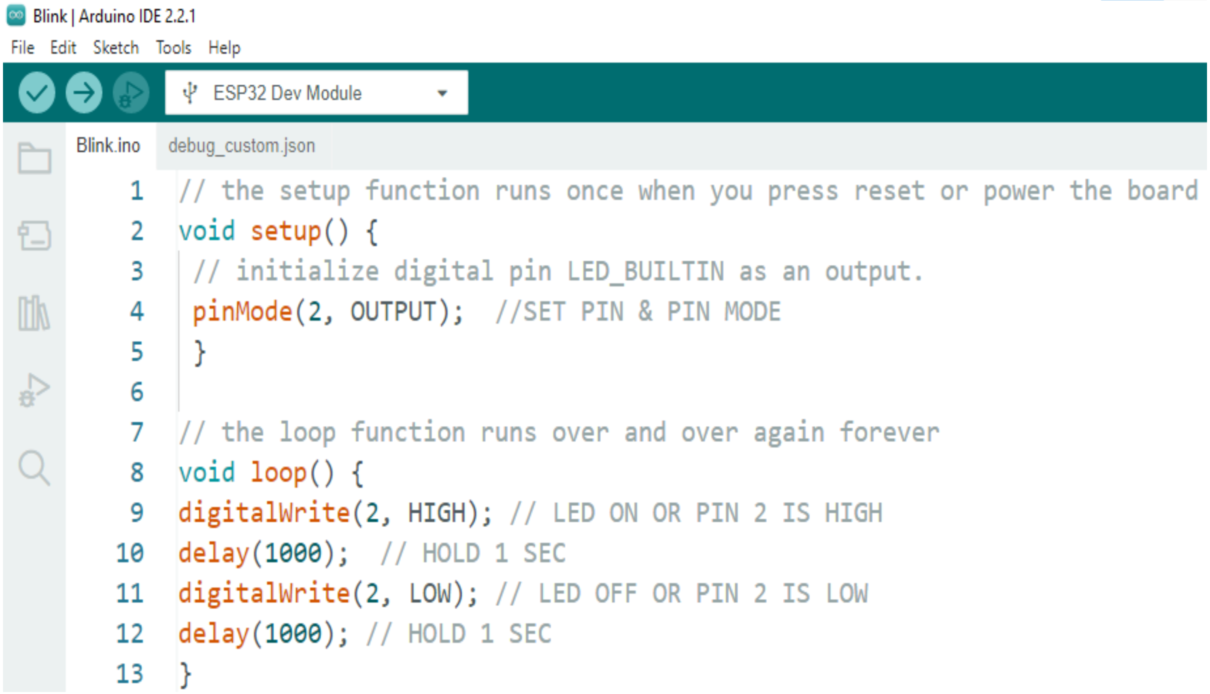
- Install Arduino IDE (Mostly used software)
- Open Board Manager Download ESP32 Boards
- Open New Sketch
- Open Tool then select the board ESP32 DEV MODULE
- Select port
- Write a code and compile and upload to the controller

Basic commands for Arduino

- pinMode();
- digitalWrite();
- digitalRead();
- analogWrite();
- analogRead();
- delay();

Example programs

Code for Led Blinking

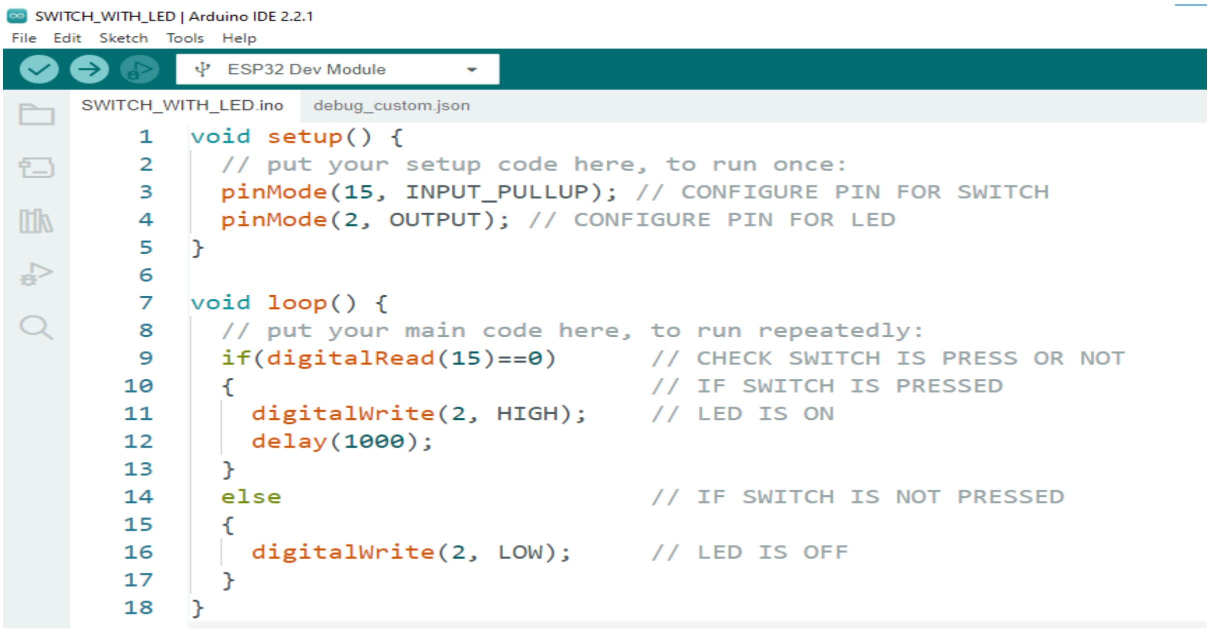


```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(2, OUTPUT);  //SET PIN & PIN MODE
  }

// the loop function runs over and over again forever
void loop() {
digitalWrite(2, HIGH); // LED ON OR PIN 2 IS HIGH
delay(1000);  // HOLD 1 SEC
digitalWrite(2, LOW); // LED OFF OR PIN 2 IS LOW
delay(1000); // HOLD 1 SEC
}
```

Figure 12: Led Blink

Code for Switch with Led

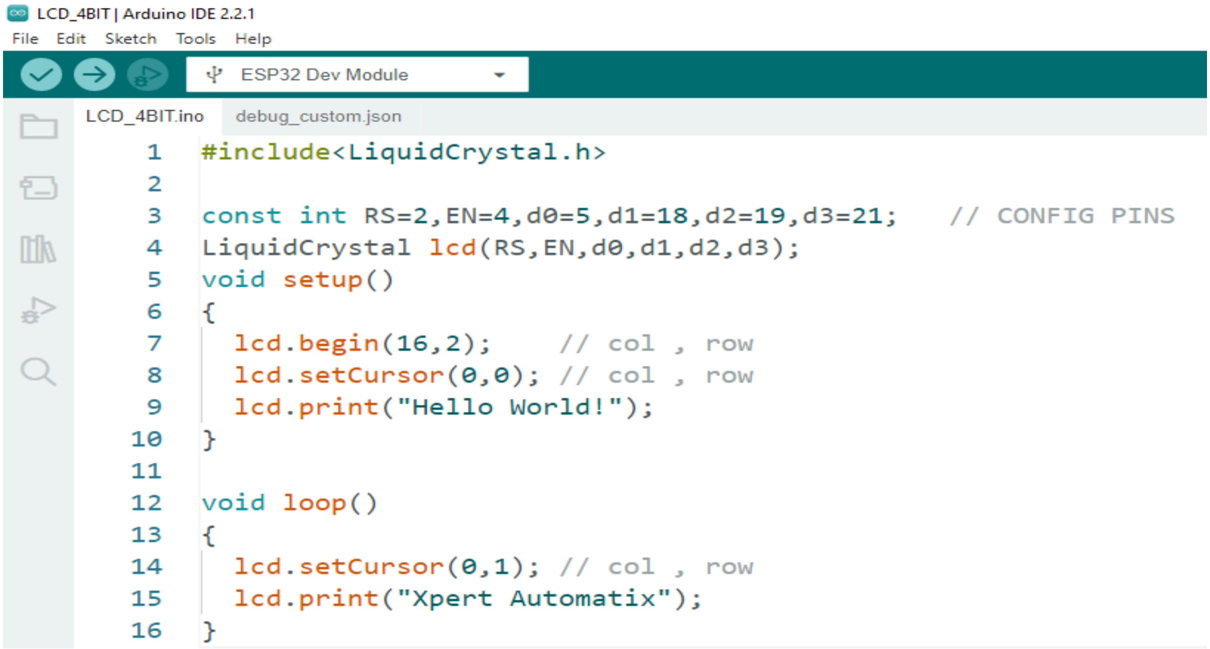

SWITCH_WITH_LED | Arduino IDE 2.2.1
File Edit Sketch Tools Help
ESP32 Dev Module
SWITCH_WITH_LED.ino
debug_custom.json

```
void setup() {
  // put your setup code here, to run once:
  pinMode(15, INPUT_PULLUP); // CONFIGURE PIN FOR SWITCH
  pinMode(2, OUTPUT); // CONFIGURE PIN FOR LED
}

void loop() {
  // put your main code here, to run repeatedly:
  if(digitalRead(15)==0)       // CHECK SWITCH IS PRESS OR NOT
  {                            // IF SWITCH IS PRESSED
    digitalWrite(2, HIGH);     // LED IS ON
    delay(1000);
  }
  else                         // IF SWITCH IS NOT PRESSED
  {
    digitalWrite(2, LOW);      // LED IS OFF
  }
}
```

Figure 13: Switch with Led

Code for LCD 16x2 Display 4bit Mode



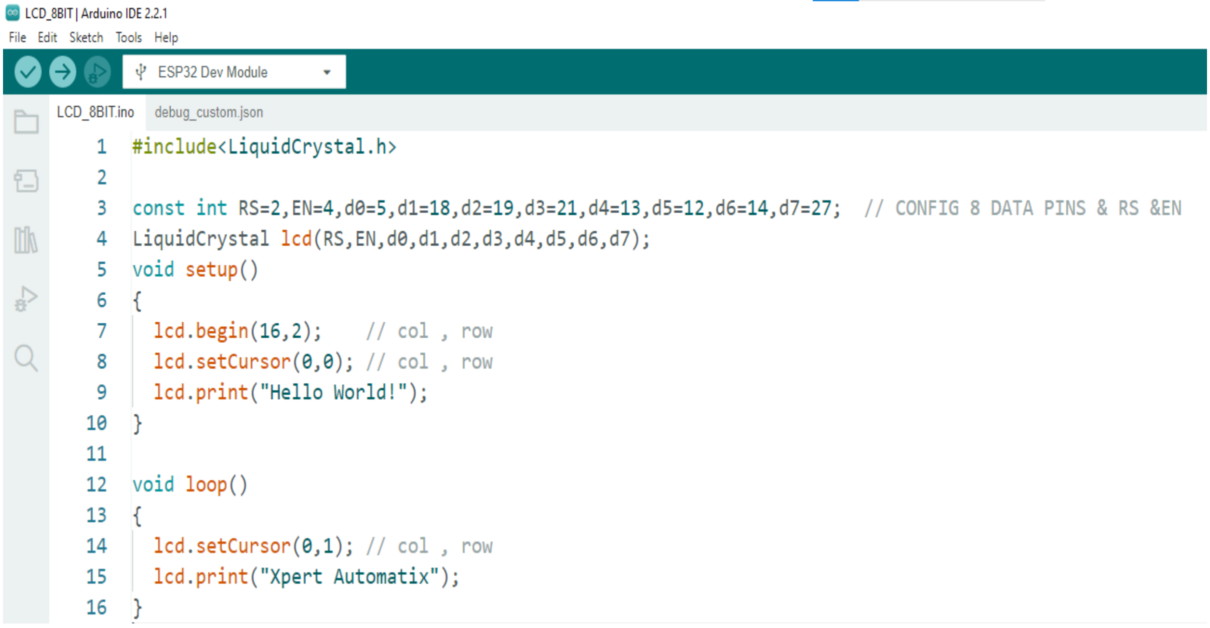
```
#include<LiquidCrystal.h>

const int RS=2,EN=4,d0=5,d1=18,d2=19,d3=21;   // CONFIG PINS
LiquidCrystal lcd(RS,EN,d0,d1,d2,d3);
void setup()
{
  lcd.begin(16,2);    // col , row
  lcd.setCursor(0,0); // col , row
  lcd.print("Hello World!");
}

void loop()
{
  lcd.setCursor(0,1); // col , row
  lcd.print("Xpert Automatix");
}
```

Figure 14: LCD 16x2 Display 4bit Mode

Code for LCD 16x2 Display 8bit Mode



```
#include<LiquidCrystal.h>

const int RS=2,EN=4,d0=5,d1=18,d2=19,d3=21,d4=13,d5=12,d6=14,d7=27;  // CONFIG 8 DATA PINS & RS &EN
LiquidCrystal lcd(RS,EN,d0,d1,d2,d3,d4,d5,d6,d7);
void setup()
{
  lcd.begin(16,2);    // col , row
  lcd.setCursor(0,0); // col , row
  lcd.print("Hello World!");
}

void loop()
{
  lcd.setCursor(0,1); // col , row
  lcd.print("Xpert Automatix");
}
```

Figure 15: LCD 16x2 Display 8bit Mode

Libraries

Peripherals

- Led Blink
- Switch with Led
- LCD 4Bit Mode
- LCD 8Bit Mode
- PWM – Led Fade
- ADC

Interrupt

- External Interrupt
- Timer Interrupt

Communication Protocols

- UART - HC:05 Bluetooth Module
- I2C – LCD
- I2C – RTC
- SPI – RFID RC522

Bluetooth

- Serial to SerialBT
- SerialBT to Control GPIO'S

Wi-Fi

- Wi-Fi Access point Mode
- Wi-Fi Station Mode

Cloud Platform

- ThingSpeak Read Data
- ThingSpeak Write Data

SD Card Module

- SD – Test

GitHub link

Follow the QR or Use the link below for the Example codes for the ESP32 Dev Module.

https://github.com/Xpertautomatix/Products/tree/main/XA_ESP32%20DEVELOPMENT%20BOARD

DEMO VIDEO

Follow the QR or Use the link below for the ESP32 demo video.

https://www.youtube.com

ESP32 DEV MODULE