ESPRESSIF ESP32-S3-MINI-1 Development Board User Manual

ESP32-S3-MINI-1 Development Board
User Manual

Contents

ESP32-S3-MINI-1 Development Board

Small-sized module supporting 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth® 5 (LE) Built around ESP32-S3 series of SoCs, Xtensa’® dual-core 32-bit LX7 microprocessor 8 MB flash 39 GPIOs, a rich set of peripherals On-board PCB antenna or external antenna connector.

ESP32-S3-MINI-1

ESP32-S3-MINI-1U

Module Overview

1.1 Features

CPU and On-Chip Memory

- ESP32-S3FN8 embedded, Xtensa® dual-core 32-bit LX7 microprocessor, up to 240 MHz
- 384 KB ROM
- 512 KB SRAM
- 16 KB SRAM in RTC
- 8 MB SPI flash

Wi-Fi

- 802.11 b/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 ~ 2462 MHz

Bluetooth

- Bluetooth LE: Bluetooth 5, Bluetooth mesh
- Speed: 125 Kbps, 500 Kbps, 1 Mbps, 2 Mbps
- Advertising extensions
- Multiple advertisement sets
- Channel selection algorithm #2

Peripherals

- GPIO, SPI, LCD interface, Camera interface, UART, I2C, I2S, remote control, pulse counter, LED PWM, USB 1.1 OTG, USB Serial/JTAG controller, MCPWM, SDIO host, GDMA, TWAI® controller (compatible with ISO

11898-1, i.e. CAN Specification 2.0), ADC, touch sensor, temperature sensor, timers, and watchdogs.

Integrated Components on Module

- 40 MHz crystal oscillator

Antenna Options

- On-board PCB antenna (ESP32-S3-MINI-1)
- External antenna via a connector (ESP32-S3-MINI-1U)

Operating Conditions

- Operating voltage/Power supply: 3.0 ~ 3.6 V
- Operating ambient temperature: –40 ~ 85 °C

1.2 Description

ESP32-S3-MINI-1 and ESP32-S3-MINI-1U are two powerful, generic Wi-Fi + Bluetooth LE MCU modules that feature a rich set of peripherals, yet an optimized size. They are an ideal choice for a wide variety of application scenarios related to the Internet of Things (IoT), such as embedded systems, smart homes, wearable electronics, etc.

ESP32-S3-MINI-1 comes with a PCB antenna. ESP32-S3-MINI-1U comes with an external antenna connector.

The ordering information of the module is shown in Table 1.

The information in this datasheet is applicable to both modules.

Table 1: Ordering Information

Module	ESP32-S3-MINI-1	ESP32-S3-MINI-1U
Variants	ESP32-S3-MINI-1-N8	ESP32-S3-MINI-1U-N8
Chip Embedded	ESP32-S3FN8	
Flash	8 MB (Quad SPI)	
PROGRAM	0	
Dimensions	15.4 x 20.5 x 2.4	15.4 x 15.4 x 2.4

At the core of the modules is an ESP32-S3FN8, an Xtensa® 32-bit LX7 CPU that operates at up to 240 MHz.

You can power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds.

ESP32-S3FN8 integrates a rich set of peripherals including SPI, LCD, Camera interface, UART, I2C, I2S, remote control, pulse counter, LED PWM, USB Serial/Jtag, MCPWM, SDIO host, GDMA, TWAI® controller (compatible with ISO 11898-1, i.e. CAN Specification 2.0), ADC, touch sensor, temperature sensor, timers, and watchdogs, as well as up to 45 GPIOs. It also includes a full-speed USB 1.1 On-The-Go (OTG) interface to enable USB communication.

Note:

* For more information on ESP32-S3FN8, please refer to ESP32-S3 Series Datasheet.

Pin Definitions

2.1 Pin Layout

The pin diagram below shows the approximate location of pins on the module. The pin diagram is applicable for ESP32-S3-MINI-1 and ESP32-S3-MINI-1U, but the latter has no keep-out zone.

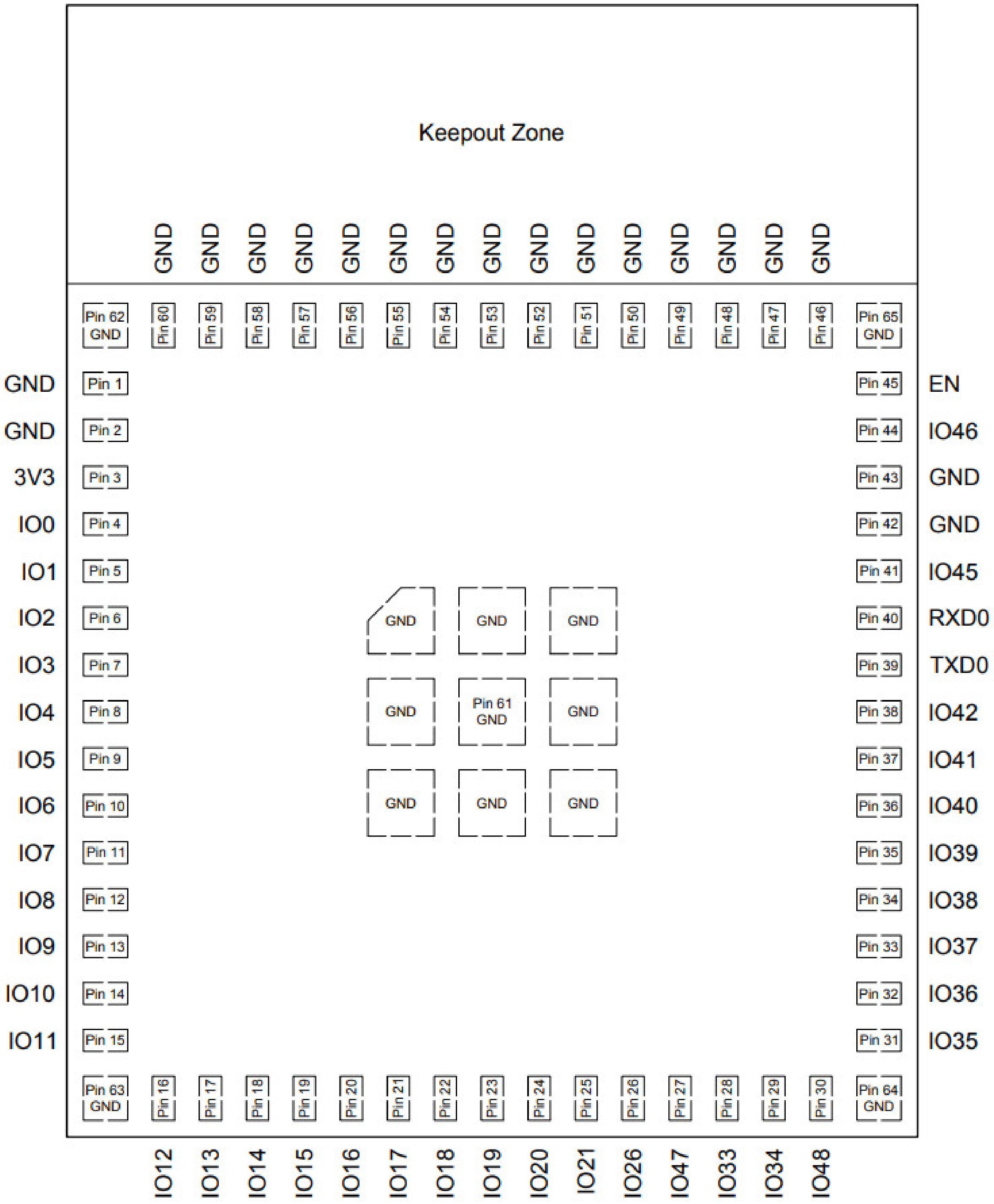


Figure 1: Pin Layout (Top View)

2.2 Pin Description

The module has 65 pins. See pin definitions in Table 2.

For explanations of pin names and function names, as well as configurations of peripheral pins, please refer to ESP32-S3 Series Datasheet.

Table 2: Pin Definitions

Name	No.	Type S	Function
GND	1, 2, 42, 43, 46-65	P	GND

3V3	3	P	Power supply
100	4	I/0/T	RTC_GPI00, GP100
101	5	1/0/1	RTC_GPI 01 , GPI01, TOUCH 1 , ADC1 _CHO
102	6	I/0/T	RTC_GPIO2, GPIO2, TOUCH2, ADC1_CH1
103	7	I/0/T	RTC_GPI03, GPIO3, TOUCH3, ADC1_CH2
104	8	I/0/T	RTC_GPI04, GPIO4, TOUCH4, ADC1_CH3
105	9	1/0/1	RTC_GPI05, GPIO5, TOUGHS, ADC1_CH4
106	10	I/0/T	RTC_GPI06, GPI06, TOUCH6, ADC1_CH5
107	11	I/0/T	RTC_GPI07, GPIO7, TOUCH7, ADC1_CH6
108	12	I/0/T	RTC_GPI08, GPI08, TOUGHS, ADC1_CH7, SUBSPICS1
109	13	I/0/T	RTC_GPI09, GPIO9, TOUCH9, ADC1_CH8, FSPIHD, SUBSPIHD
1010	14	I/0/T	RTC_GPI010, GPI010, TOUCH 10, ADC1_CH9, FSPICSO, FSP1104, SUBSPICSO
1011	15	I/0/T	RTC_GPI011, GPIO11, TOUCH11, ADC2_CHO, FSPID, FSPII05, SUBSPID
1012	16	I/0/T	RTC_GPI012, GPI012, TOUCH12, ADC2_CH1, FSPICLK, FSPII06, SUBSPICLK
1013	17	I/0/T	RTC_GPI013, GPIO13, TOUCH13, ADC2_CH2, FSPIQ, FSPIIO7, SUBSPIQ
1014	18	1/0/T	RTC_GPI014, GPI014, TOUCH14, ADC2_CH3, FSPIWP, FSPIDQS, SUBSPIWP
1015	19	I/0/T	RTC_GPI015, GPIO15, UORTS, ADC2_CH4, XTAL_32K_P
1016	20	I/0/T	RTC_GPI016, GPIO16. UOCTS, ADC2_CH5, XTAL_32K_N
1017	21	I/0/T	RTC_GP101 7, GPIO17, UITXD, ADC2_CH6
1018	22	I/0/T	RTC_GPI018, GPIO18, U1RXD, ADC2_CH7, CLK_OUT3
1019	23	I/0/T	RTC_GPI019, GPI019, U1 RTS, ADC2_CH8, CLK_OUT2, USB_D-
1020	24	I/0/T	RTC_GPIO20, GPIO20, U1 CTS, ADC2_CH9, CLK_OUT1, USB_D+
1021	25	I/0/T	RTC_GPIO21, GPIO21
1026	26	I/0/T	SPICS1 , GPIO26
1047	27	I/0/T	SPICLK_P_DIFF, GPI047, SUBSPICLK_P_DIFF
1033	28	I/0/T	SPII04, GPI033, FSPIHD, SUBSPIHD
1034	29	I/0/T	SPII05, GPI034, FSPICSO, SUBSPICSO
1048	30	I/0/T	SPICLK_N_DIFF, GPI048, SUBSPICLK_N_DIFF
1035	31	1/0/1	SPII06, GPI035, FSPID, SUBSPID

1036	32	1/0/1	SPII07, GPI036, FSPICLK, SUBSPICLK
1037	33	I/0/T	SPIDQS, GPI037, FSPIQ, SUBSPIQ
1038	34	I/0/T	GPI038, FSPIWP, SUBSPIWP
1039	35	I/0/T	MTCK, GPI039, CLK_OUT3, SUBSPICS1
1040	36	I/0/T	MTDO, GPI040, CLK_OUT2
1041	37	1/0/T	MTDI, GPI041, CLK_OUT1

Name	No.	Type [a]	Function
1042	38	I/0/T	MTMS, GPI042
TXDO	39	1/0/1-	UOTXD, GPI043, CLK_OUT1
RXDO	40	I/0/T	UORXD, GPI044, CLK_OUT2
1045	41	I/0/T	GP1045
1046	44	I/0/T	GPI046
EN	45	I	High: on, enables the chip. Low: off, the chip powers off. Note: Do not leave the EN pin floating.

[a] P: power supply; I: input; O: output; T: high impedance. Pin functions in bold font are the default pin functions. For pin 28 ~ 29, 31 ~ 33, the default function is decided by the eFuse bit.

Get Started

3.1 What You Need

To develop applications for the module you need:

- 1 x ESP32-S3-MINI-1 or ESP32-S3-MINI-1U
- 1 x Espressif RF testing board
- 1 x USB-to-Serial board
- 1 x Micro-USB cable
- 1 x PC running Linux
  In this user guide, we take Linux operating system as an example. For more information about the configuration on Windows and macOS, please refer to ESP-IDF Programming Guide.

3.2 Hardware Connection

1. Solder the ESP32-S3-MINI-1 or ESP32-S3-MINI-1U module to the RF testing board as shown in Figure 2.

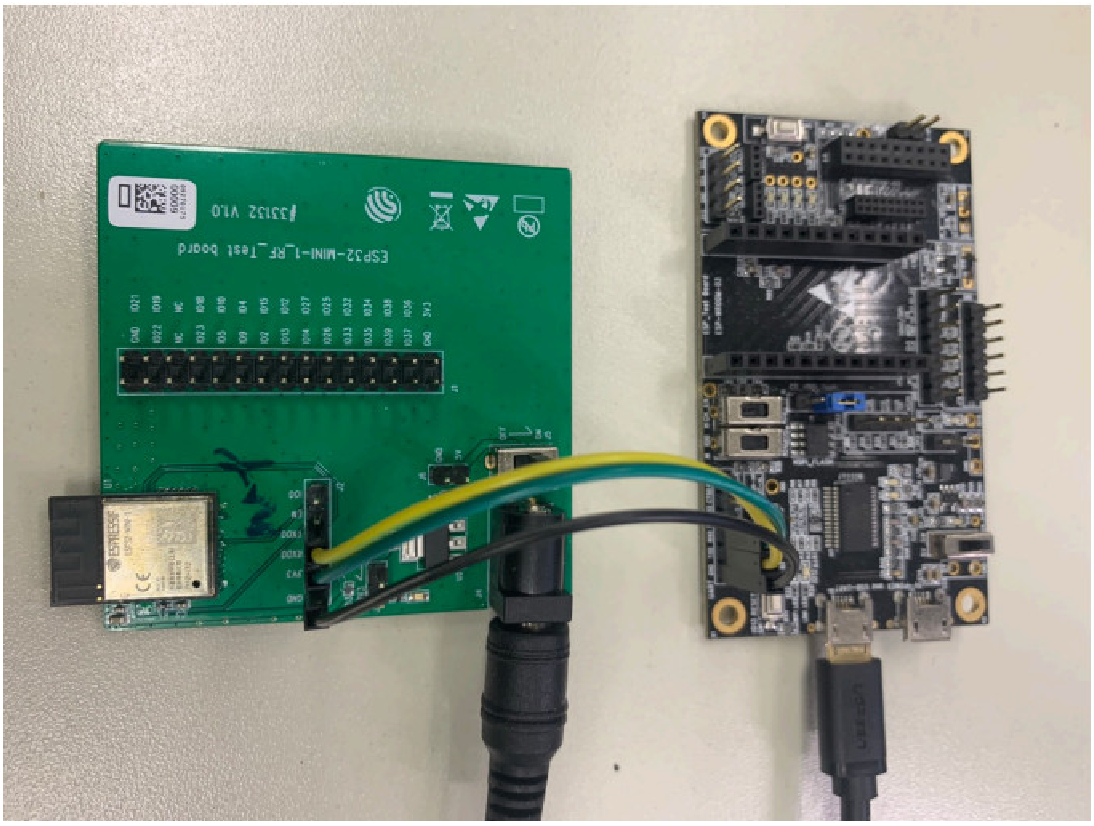


Figure 2: Hardware Connection

2. Connect the RF testing board to the USB-to-Serial board via TXD, RXD, and GND.
3. Connect the USB-to-Serial board to the PC.
4. Connect the RF testing board to the PC or a power adapter to enable 5 V power supply, via the Micro-USB cable.
5. During download, connect IO0 to GND via a jumper. Then, turn ”ON” the testing board.
6. Download firmware into flash. For details, see the sections below.
7. After downloading, remove the jumper on IO0 and GND.
8. Power up the RF testing board again. The module will switch to working mode. The chip will read programs from flash upon initialization.

Note:
IO0 is internally logic high. If IO0 is set to pull-up, the Boot mode is selected. If this pin is pull-down or left floating, the Download mode is selected. For more information on ESP32-S3-MINI-1 or ESP32-S3-MINI-1U, please refer to ESP32-S3 Series Datasheet.

3.3 Set up Development Environment

The Espressif IoT Development Framework (ESP-IDF for short) is a framework for developing applications based on the Espressif ESP32. Users can develop applications with ESP32-S3 in Windows/Linux/macOS based on ESP-IDF. Here we take Linux operating system as an example.

3.3.1 Install Prerequisites

To compile with ESP-IDF you need to get the following packages:

- CentOS 7 & 8:
  1. sudo yum -y update && sudo yum install git wget flex bison gperf python3 python3-pip
  2. python3-setuptools CMake ninja-build ccache dfu-util libusbx
- Ubuntu and Debian:

1. sudo apt-get install git wget flex bison gperf python3 python3-pip python3-setuptools
2. CMake ninja-build ccache life-dev libssl-dev dfu-util libusb-1.0-0

- Arch:
  1. sudo Pacman -S –needed GCC git make flex bison gperf python-pip CMake ninja ccache
  2. dfu-util libusb

Note:

- This guide uses the directory ~/esp on Linux as an installation folder for ESP-IDF.
- Keep in mind that ESP-IDF does not support spaces in paths.

3.3.2 Get ESP-IDF

To build applications for the ESP32-S3-MINI-1 or ESP32-S3-MINI-1U module, you need the software libraries provided by Espressif in the ESP-IDF repository.
To get ESP-IDF, create an installation directory (~/esp) to download ESP-IDF to and clone the repository with ‘git clone:
ESP-IDF will be downloaded into ~/esp/esp-idf. Consult ESP-IDF Versions for information about which ESP-IDF version to use in a given situation.

3.3.3 Set up Tools

Aside from the ESP-IDF, you also need to install the tools used by ESP-IDF, such as the compiler, debugger, Python packages, etc. ESP-IDF provides a script named ’install.sh’ to help set up the tools in one go.

1. cd ~/esp/esp-idf
2. ./install.sh

3.3.4 Set up Environment Variables

The installed tools are not yet added to the PATH environment variable. To make the tools usable from the command line, some environment variables must be set. ESP-IDF provides another script ’export.sh’ which does that. In the terminal where you are going to use ESP-IDF, run:

1. $HOME/esp/esp-idf/export.sh

   Now everything is ready, you can build your first project on the ESP32-S3-MINI-1 or ESP32-S3-MINI-1U module.

3.4 Create Your First Project

3.4.1 Start a Project

Now you are ready to prepare your application for the ESP32-S3-MINI-1 or ESP32-S3-MINI-1U module. You can start with the get-started/hello_world project from the examples directory in ESP-IDF.
Copy get-started/hello_world to ~/esp directory:

```
1 cd ~/esp
2 cp -r $IDF_PATH/examples/get-started/hello_world .
```

There is a range of example projects in the examples directory in ESP-IDF. You can copy any project in the same way as presented above and run it. It is also possible to build examples in place, without copying them first.

3.4.2 Connect Your Device

Now connect your module to the computer and check under what serial port the module is visible. Serial ports in Linux start with ‘/dev/tty’ in their names. Run the command below two times, first with the board unplugged, then with plugged in. The port which appears the second time is the one you need:

```
1 ls /dev/tty*
```

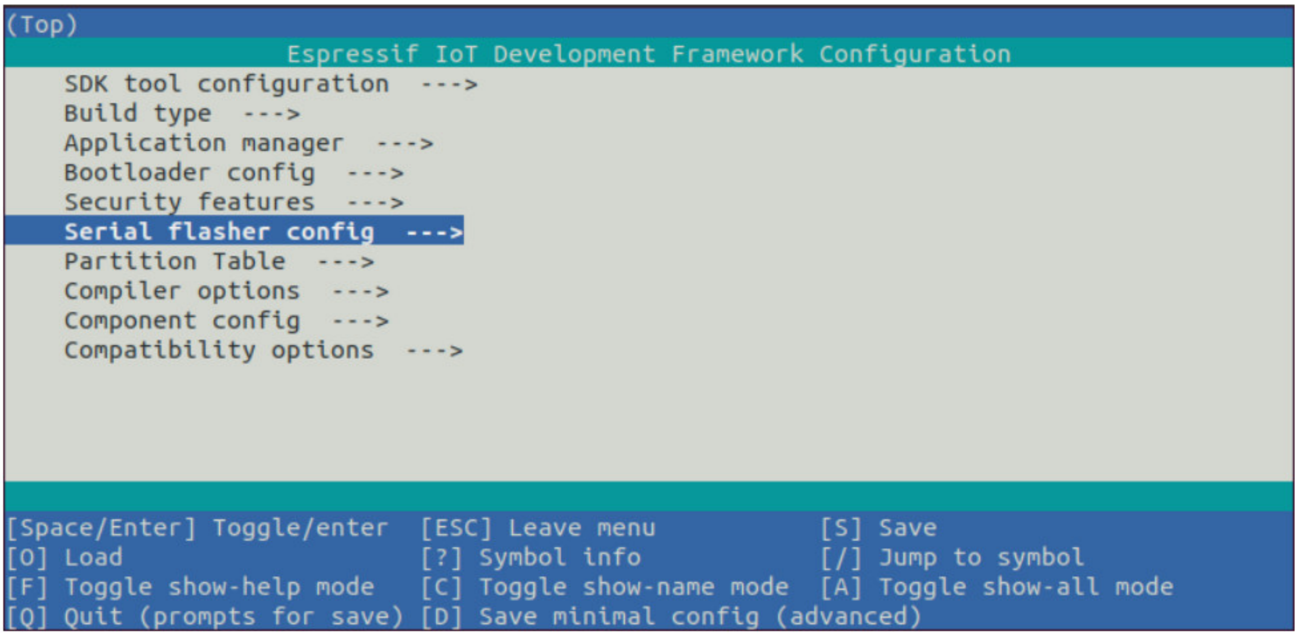
Note:
Keep the port name handy as you will need it in the next steps.

3.4.3 Configure

Navigate to your ‘hello_world’ directory from Step 3.4.1. Start a Project, set ESP32-S3 chip as the target, and run the project configuration utility ‘menuconfig’.

1. cd ~/esp/hello_world
2. idf.py set-target esp32s3
3. idf.py menuconfig

Setting the target with ‘idf.py set-target ESP32-S3’ should be done once, after opening a new project. If the project contains some existing builds and configurations, they will be cleared and initialized. The target may be saved in the environment variable to skip this step. See Selecting the Target for additional information.
If the previous steps have been done correctly, the following menu appears:



```
(Top)
                  Espressif IoT Development Framework Configuration
    SDK tool configuration  --->
    Build type  --->
    Application manager  --->
    Bootloader config  --->
    Security features  --->
    Serial flasher config  --->
    Partition Table  --->
    Compiler options  --->
    Component config  --->
    Compatibility options  --->



[Space/Enter] Toggle/enter  [ESC] Leave menu           [S] Save
[O] Load                    [?] Symbol info            [/] Jump to symbol
[F] Toggle show-help mode   [C] Toggle show-name mode  [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Figure 3: Project Configuration - Home Window

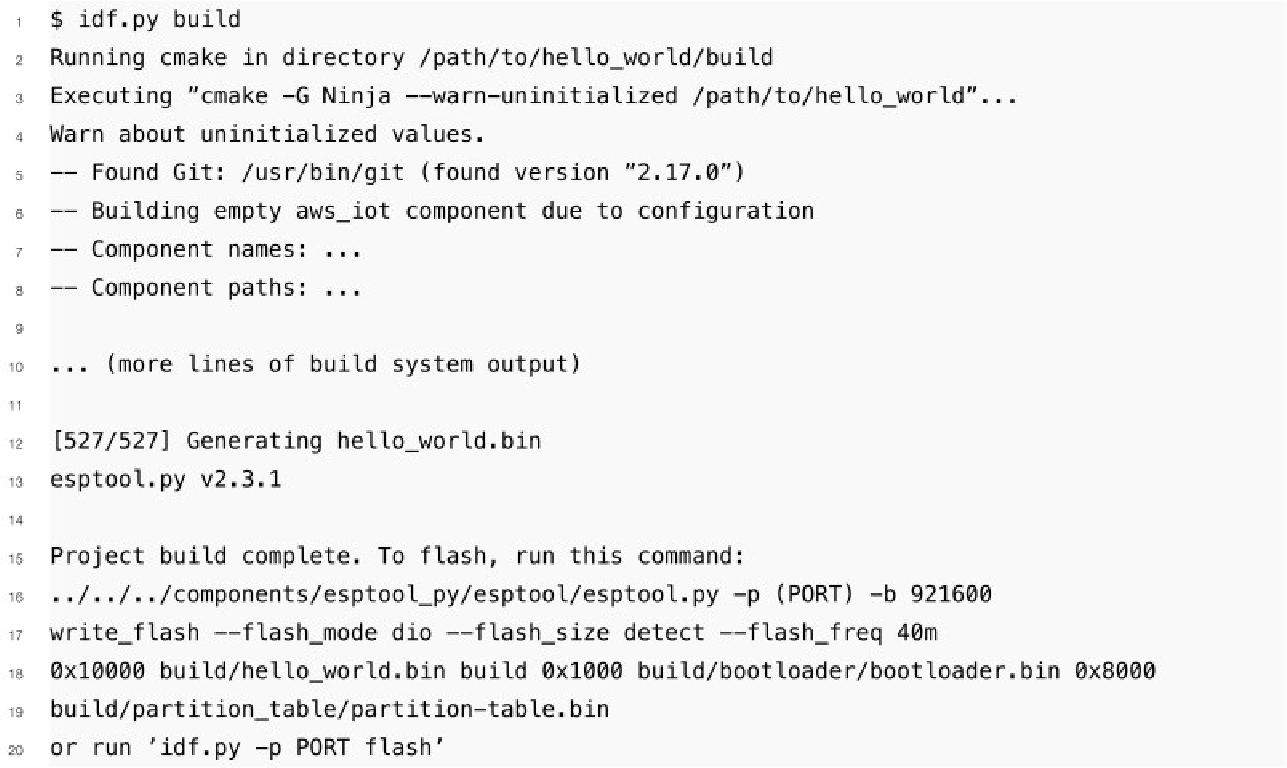
You are using this menu to set up project-specific variables, e.g. Wi-Fi network name and password, processor speed, etc. Setting up the project with menuconfig may be skipped for “hello_word”. This example will run with the default configuration
The colors of the menu could be different in your terminal. You can change the appearance with the option ‘- -style’. Please run ‘idf.py menuconfig – -help’ for further information.

3.4.4 Build the Project

Build the project by running:

1. idf.py build

This command will compile the application and all ESP-IDF components, then it will generate the bootloader, partition table, and application binaries.



```
$ idf.py build
Running cmake in directory /path/to/hello_world/build
Executing ”cmake -G Ninja --warn-uninitialized /path/to/hello_world”...
Warn about uninitialized values.
-- Found Git: /usr/bin/git (found version ”2.17.0”)
-- Building empty aws_iot component due to configuration
-- Component names: ...
-- Component paths: ...

... (more lines of build system output)

[527/527] Generating hello_world.bin
esptool.py v2.3.1

Project build complete. To flash, run this command:
../../../components/esptool_py/esptool/esptool.py -p (PORT) -b 921600
write_flash --flash_mode dio --flash_size detect --flash_freq 40m
0x10000 build/hello_world.bin build 0x1000 build/bootloader/bootloader.bin 0x8000
build/partition_table/partition-table.bin
or run ’idf.py -p PORT flash’
```

3.4.5 Flash onto the Device

Flash the binaries that you just built onto your module by running: idf.py -p PORT [-b BAUD] flash
Replace PORT with your ESP32-S3 board’s serial port name from Step: Connect Your Device.
You can also change the flasher baud rate by replacing BAUD with the baud rate you need. The default baud rate is 460800.
For more information on idf.py arguments, see **idf.py**.

Note:

The option ‘flash‘ automatically builds and flashes the project, so running ‘idf.py build‘ is not necessary.
When flashing, you will see the output log similar to the following:

1. …
2. esptool.py esp32s3 -p /dev/ttyUSB0 -b 460800 –before=default_reset –after=hard_reset
3. write_flash –flash_mode dio –flash_freq 80m –flash_size 2MB 0x0 bootloader/bootloader. bin
4. 0x10000 hello_world.bin 0x8000 partition_table/partition-table.bin
5. esptool.py v3.2-dev
6. Serial port /dev/ttyUSB0
7. Connecting…
8. Chip is ESP32-S3

   Features: WiFi, BLE

   Crystal is 40MHz

   MAC: 7c:df:a1:e0:00:64

   Uploading stub…

   Running stub…

   Stub running…

   Changing baud rate to 460800

   Changed.

   Configuring flash size…

   Flash will be erased from 0x00000000 to 0x00004fff…

```
Flash will be erased from 0x00010000 to 0x00039fff…
Flash will be erased from 0x00008000 to 0x00008fff…
Compressed 18896 bytes to 11758…
Writing at 0x00000000… (100 %)
Wrote 18896 bytes (11758 compressed) at 0x00000000 in 0.5 seconds (effective 279.9 kbit/s…)
Hash of data verified.
Compressed 168208 bytes to 88178…
Writing at 0x00010000… (16 %)
Writing at 0x0001a80f… (33 %)
Writing at 0x000201f1… (50 %)
Writing at 0x00025dcf… (66 %)
Writing at 0x0002d0be… (83 %)
Writing at 0x00036c07… (100 %)
Wrote 168208 bytes (88178 compressed) at 0x00010000 in 2.4 seconds (effective 569.2 kbit/s).
Hash of data verified.
Compressed 3072 bytes to 103…
Writing at 0x00008000… (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 478.9 kbit/s)…
Hash of data verified.
Leaving…
Hard resetting via RTS pin…
Done
```

If there are no issues by the end of the flash process, the board will reboot and start up the “hello_world” application.

3.4.6 Monitor

To check if “hello_world” is indeed running, type ‘idf.py -p PORT monitor‘ (Do not forget to replace PORT with your serial port name).
This command launches the IDF Monitor application:

```
$ idf.py -p /dev/ttyUSB0 monitor
Running idf_monitor in directory […]/esp/hello_world/build
Executing ”python […]/esp-idf/tools/idf_monitor.py -b 115200 […]/esp/hello_world/build/hello-world.elf”…
— idf_monitor on /dev/ttyUSB0 115200 —
— Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H —ets Jun 8 2016 00:22:57
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
ets Jun 8 2016 00:22:57
…
```

After startup and diagnostic logs scroll up, you should see “Hello world!” printed out by the application.

```
…
Hello, world!
Restarting in 10 seconds…
This is esp32s3 chip with 2 CPU core(s), This is esp32s3 chip with 2 CPU core(s), WiFi/BLE,
silicon revision 0, 2MB external flash
Minimum free heap size: 390684 bytes
Restarting in 9 seconds…
Restarting in 8 seconds…
Restarting in 7 seconds…
```

To exit the IDF monitor use the shortcut Ctrl+].
That's all that you need to get started with the ESP32-S3-MINI-1 or ESP32-S3-MINI-1U module! Now you are ready to try some other examples in ESP-IDF or go right to developing your own applications.

U.S. FCC Statement

The devices comply with KDB 996369 D03 OEM Manual v01. Below are integration instructions for host product manufacturers according to the KDB 996369 D03 OEM Manual v01.
List of Applicable FCC Rules
FCC Part 15 Subpart C 15.247 & 15.209
Specific Operational Use Conditions
The modules have WiFi, BR, EDR, and BLE functions.

- Operation Frequency:
  - WiFi: 2412 ~ 2462 MHz
  - Bluetooth: 2402 ~ 2480 MHz
- Number of Channel:
  - WiFi: 12
  - Bluetooth: 40
- Modulation:
  - WiFi: DSSS; OFDM
  - Bluetooth: GFSK; π/4 DQPSK; 8 DPSK
- Type: On-board PCB antenna or external antenna connector
- Gain: 4.54 dBi Max

The modules can be used for IoT applications with a maximum of 3.96 dBi antenna. The host manufacturer installing the modules into their product must ensure that the final composite product complies with the FCC requirements by a technical assessment or evaluation of the FCC rules, including the transmitter operation. The host manufacturer has to be aware not to provide information to the end user regarding how to install or remove the RF modules in the user's manual of the end product which integrates the modules. The end user manual shall include all required regulatory information/warnings as shown in this manual.

Limited Module Procedures
Not applicable. The modules are single modules and comply with the requirement of FCC Part 15.212.
Trace Antenna Designs
Not applicable. The modules have their own antenna and do not need a host's printed board microstrip trace antenna, etc.

RF Exposure Considerations
The modules must be installed in the host equipment such that at least 20cm is maintained between the antenna and the users' body; and if the RF exposure statement or module layout is changed, then the host product manufacturer is required to take responsibility for the modules through a change in FCC ID or new application. The FCC ID of the modules cannot be used on the final product. In these circumstances, the host manufacturer will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

Antennas
Antenna specification are as follows:

- Type: On-board PCB antenna

- Gain: 3.96 dBi
- Type: External antenna connector
- Gain: 4.54 dBi

This device is intended only for host manufacturers under the following conditions:

- The transmitter module may not be co-located with any other transmitter or antenna.
- The modules shall be only used with the external antenna(s) that has been originally tested and certified with the modules.
- The antenna must be either permanently attached or employ a ‘unique’ antenna coupler.

As long as the conditions above are met, further transmitter tests will not be required. However, the host manufacturer is still responsible for testing their end-product for any additional compliance requirements required with the modules installed (for example, digital device emissions, PC peripheral requirements, etc.).

Label and Compliance Information

Host product manufacturers need to provide a physical or e-label stating “Contains FCC ID: 2AC7Z-ESPS3MINI1” with their finished product.

Information on test modes and additional testing requirements

- Operation Frequency:
  – WiFi: 2412 ~ 2462 MHz
  – Bluetooth: 2402 ~ 2480 MHz
- Number of Channels:
  – WiFi: 12
  – Bluetooth: 40
- Modulation:
  – WiFi: DSSS; OFDM
  – Bluetooth: GFSK; π/4 DQPSK; 8 DPSK

Host manufacturers must perform tests of radiated and conducted emission and spurious emission, etc., according to the actual test modes for a stand-alone modular transmitter in a host, as well as for multiple simultaneously transmitting modules or other transmitters in a host product. Only when all the test results of test modes comply with FCC requirements, then the end product can be sold legally.

Additional testing, Part 15 Subpart B compliant

The modular transmitter is only FCC authorized for FCC Part 15 Subpart C 15.247 & 15.209 and the host product manufacturer is responsible for compliance with any other FCC rules that apply to the host not covered by the modular transmitter grant of certification. If the grantee markets their product as being Part 15 Subpart B compliant (when it also contains unintentional-radiator digital circuity), then the grantee shall provide a notice stating that the final host product still requires Part 15 Subpart B compliance testing with the modular transmitter installed.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications.

However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

The devices comply with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

- This device may not cause harmful interference.
- This device must accept any interference received, including interference that may cause undesired operation.

Caution:

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

The equipment complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. This device and its antenna must not be co-located or operating in conjunction with any other antenna or transmitter.

The antennas used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

OEM Integration Instructions

The devices are intended only for OEM integrators under the following conditions:

- The transmitter module may not be co-located with any other transmitter or antenna.
- The modules shall be only used with the external antenna(s) that has been originally tested and certified with the modules.

As long as the conditions above are met, further transmitter tests will not be required. However, the OEM integrator is still responsible for testing their end-product for any additional compliance requirements required with the modules installed (for example, digital device emissions, PC peripheral requirements, etc.).

Validity of Using the Module Certification

In the event that these conditions cannot be met (for example certain laptop configurations or co-location with another transmitter), then the FCC authorization for the modules in combination with the host equipment is no longer considered valid and the FCC ID of the modules cannot be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate FCC authorization.

End Product Labeling

The final end product must be labeled in a visible area with the following: "Contains Transmitter Module FCC ID: 2AC7Z-ESPS3MINI1".

Industry Canada Statement

This device complies with Industry Canada's license-exempt RSSs. Operation is subject to the following two conditions:

- This device may not cause interference; and
- This device must accept any interference, including interference that may cause undesired operation of the device.

Radiation Exposure Statement

This equipment complies with IC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with a minimum distance of 20 cm between the radiator and your body.

RSS-247 Section 6.4 (5)
The device could automatically discontinue transmission in case of the absence of information to transmit or operational failure. Note that this is not intended to prohibit transmission of control or signaling information or the use of repetitive codes where required by the technology.

This device is intended only for OEM integrators under the following conditions (For module device use):

- The antenna must be installed such that 20 cm is maintained between the antenna and users, and
- The transmitter module may not be co-located with any other transmitter or antenna.

As long as the 2 conditions above are met, further transmitter tests will not be required. However, the OEM integrator is still responsible for testing their end product for any additional compliance requirements required with this module installed.

IMPORTANT NOTE:
In the event that these conditions can not be met (for example certain laptop configurations or colocation with another transmitter), then the Canada authorization is no longer considered valid and the IC ID can not be used on the final product. In these circumstances, the OEM integrator will be responsible for re-evaluating the end product (including the transmitter) and obtaining a separate Canada authorization.

End Product Labeling
This transmitter module is authorized only for use in devices where the antenna may be installed such that 20 cm may be maintained between the antenna and users. The final end product must be labeled in a visible area with the following: “Contains IC: 21098-ESPS3MINI1”.

Manual Information to the End User
The OEM integrator has to be aware not to provide information to the end user regarding how to install or remove this RF module in the user’s manual of the end product which integrates this module. The end user manual shall include all required regulatory information/warnings as shown in this manual.

Related Documentation and Resources

Related Documentation

- ESP32-S3 Technical Reference Manual – Detailed information on how to use the ESP32-S3 memory and peripherals.
- ESP32-S3 Series Datasheet – Specifications of the ESP32-S3 hardware.
- ESP32-S3 Hardware Design Guidelines – Guidelines on how to integrate the ESP32-S3 into your hardware product.
- Certificates
  http://espressif.com/en/support/documents/certificates
- Documentation Updates and Update Notification Subscription
  http://espressif.com/en/support/download/documents

Developer Zone

- ESP-IDF Programming Guide for ESP32-S3 – Extensive documentation for the ESP-IDF development framework.

- ESP-IDF and other development frameworks on GitHub.
  http://github.com/espressif
- ESP32 BBS Forum – Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
  http://esp32.com/
- The ESP Journal – Best Practices, Articles, and Notes from Espressif folks.
  http://medium.com/the-esp-journal
- See the tabs SDKs and Demos, Apps, Tools, AT Firmware.
  http://espressif.com/en/support/download/sdks-demos

Products

- ESP32-S3 Series SoCs – Browse through all ESP32-S3 SoCs.
  http://espressif.com/en/products/socs?id=ESP32-S3
- ESP32-S3 Series Modules – Browse through all ESP32-S3-based modules.
  http://espressif.com/en/products/modules?id=ESP32-S3
- ESP32-S3 Series DevKits – Browse through all ESP32-S3-based devkits.
  http://espressif.com/en/products/devkits?id=ESP32-S3
- ESP Product Selector – Find an Espressif hardware product suitable for your needs by comparing or applying filters.
  http://products.espressif.com/#/product-selector?language=en

Contact Us

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

Revision History

Date	Version	Release notes
2022-02-24	v0.6	Overall update for chip revision 1
2021-03-30	v0.1	Preliminary release, for chip revision 0

Disclaimer and Copyright Notice

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

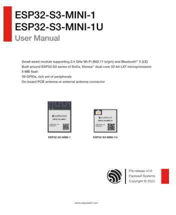
ALL THIRD-PARTY'S INFORMATION IN THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES TO ITS AUTHENTICITY AND ACCURACY.

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 the 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.

www.espressif.com

Documents / Resources

 ESP32-S3-MINI-1 ESP32-S3-MINI-1U User Manual Small-sized module supporting 2.4 GHz Wi-Fi (802.11 b/g/n) and Bluetooth® 5 (LE) Built around ESP32-S3 series of SoCs, Xtensa® dual-core 32-bit LX7 microprocessor 8 MB flash 39 GPIOs, rich set of peripherals On-board PCB antenna or external antenna connector ESP32-S3-MINI-1 ESP32-S3-MINI-1U Pre-release v0.6 Espressif Systems Copyright © 2022 www.espressif.com	ESPRESSIF ESP32-S3-MINI-1 Development Board [pdf] User Manual ESPS3MINI1, 2AC7Z-ESPS3MINI1, 2AC7ZESPS3MINI1, ESP32-S3-MINI-1 Development Board, ESP32-S3-MINI-1, ESP32-S3-MINI-1U, Development Board

References

- ESP32 Forum - Index page
- Sales Questions | Espressif Systems
- Development Boards | Espressif Systems
- Modules | Espressif Systems
- Chipsets | Espressif Systems
- Certificates | Espressif Systems
- Technical Documents | Espressif Systems
- SDKs & Demos | Espressif Systems
- Espressif Systems · GitHub
- The ESP Journal
- ESP Product Selector
- Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems
- Build System - ESP32 - — ESP-IDF Programming Guide latest documentation
- ESP-IDF Versions - ESP32 - — ESP-IDF Programming Guide latest documentation
- Build System - ESP32-S3 - — ESP-IDF Programming Guide latest documentation
- Get Started - ESP32-S3 - — ESP-IDF Programming Guide latest documentation
- Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems
- Documentation Feedback | Espressif Systems