ESPRESSIF EK058 2.4 GHz WiFi Bluetooth LE Module User Manual

Contents

ESPRESSIF

EK058 2.4 GHz WiFi Bluetooth LE Module

Product Information: EK058 Module

The EK058 module is a development board designed for the Espressif ESP32 microcontroller. It comes with a RF testing boar and a USB-to-Serial board for hardware connection. The development framework used for programming applications on the ESP32 is the Espressif IoT Development Framework (ESP-IDF).

Contents of User Manual:

Get Started on EK058:

Hardware Connection:

1. Solder the EK058 module to the RF testing board.
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 via the Micro-USB cable for the power supply.
5. During download, connect IO0 to GND via a jumper. Then, turn ON the testing board.
6. Download firmware into Flash. For details, refer to the user manual.
7. After downloading, remove the jumper on IO0 and GND.
8. Power up the RF testing board again. EK058 will switch to working mode. The chip will read programs from Flash upon initialization.

Set up Development Environment:

1. Install prerequisites depending on your operating system. Refer to the user manual for details.
2. Get ESP-IDF.
3. Set up tools and environment variables.
4. Create your first project by starting a project, connecting your device, configuring, building the project, flashing it onto the device, and monitoring it. Refer to user manual for detailed instructions.

Document
This user manual shows how to get started with the EK058 module.

Document Updates
Please always refer to the latest version on **https://www.espressif.com/en/support/download/documents**.

Revision History
For revision history of this document, please refer to the last page.

Documentation Change Notification
Espressif provides email notifications to keep you updated on changes to technical documentation. Please subscribe at **www.espressif.com/en/subscribe**.

Certification
Download certificates for Espressif products from **www.espressif.com/en/certificates**.

Overview

Module Overview

EK058 is a customized Wi-Fi + Bluetooth + Bluetooth LE MCU module and comes with a PCB antenna and an 8 MB SPI flash.

At the core of EK058 is the ESP32-D0WDR2-V3. 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. This ESP32 chip integrates a rich set of peripherals, ranging from SD card interface, capacitive touch sensors, ADC, DAC, and Two-Wire Automotive Interface, to Ethernet, high-speed SPI, UART, I2S, I2C, etc.
The power supply voltage range of this module is 3.0 V ~ 3.6 V, and the center frequency range of operating channel is 2412 MHz ~ 2462 MHz. For more information about the hardware and pins, please refer to EK058 Datasheet.

Get Started on EK058

What You Need
To develop applications for module you need:

- 1 x EK058 module
- 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.

Hardware Connection

1. Solder the EK058 module to the RF testing board as shown in **Figure 1.**

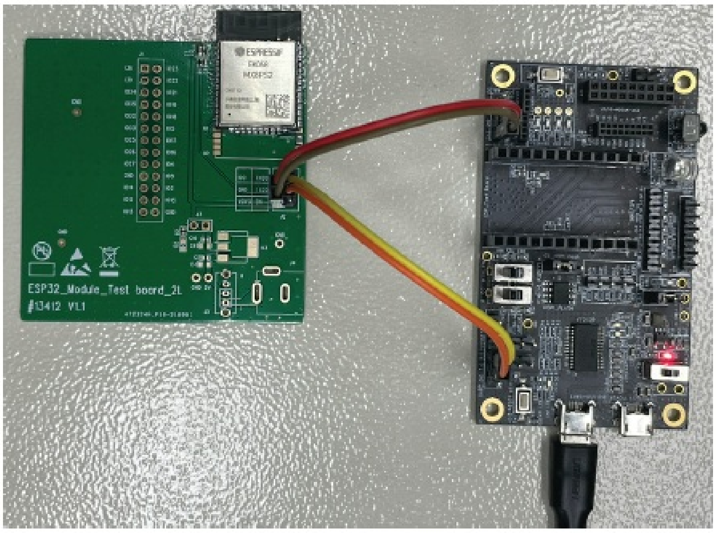


Figure 1: Hardware Connection

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 download, remove the jumper on IO0 and GND.
8. Power up the RF testing board again. EK058 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 EK058, please refer to EK058 Datasheet.

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 in Windows/Linux/macOS based on ESP-IDF. Here we take Linux operating system as an example.

Install Prerequisites

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

- **CentOS 7:**

  ```
  sudo yum install git wget flex bison gperf python cmake ninja-build ccache dfu-util
  ```

- **Ubuntu and Debian (one command breaks into two lines):**

  ```
  sudo apt-get install git wget flex bison gperf python python-pip python-setuptools cmake ninja-build ccache
  libffi-dev libssl-dev dfu-util
  ```

- **Arch:**

```
sudo pacman -S –needed gcc git make flex bison gperf python-pip cmake ninja ccache dfu-util
```

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.

Get ESPIDF
To build applications for EK058 module, you need the software libraries provided by Espressif in ESP-IDF repository.
To get ESP-IDF, create an installation directory (~/esp) to download ESP-IDF to and clone the repository with 'git clone':

```
mkdir -p ~/esp
cd ~/esp
git clone –recursive https://github.com/espressif/esp-idf.git
```

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.

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.

```
cd ~/esp/esp-idf
./install.sh
```

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:

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

Now everything is ready, you can build your first project on EK058 module.

Create Your First Project

Start a Project
Now you are ready to prepare your application for EK058 module. You can start with get-started/hello_world project from examples directory in ESP-IDF.
Copy get-started/hello_world to ~/esp directory:

```
cd ~/esp
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.

Connect Your Device

Now connect your EK058 module to the computer and check under what serial port the module is visible. Se-rial ports in Linux start with '/dev/tty' in their names. Run the command below two times, first with the board

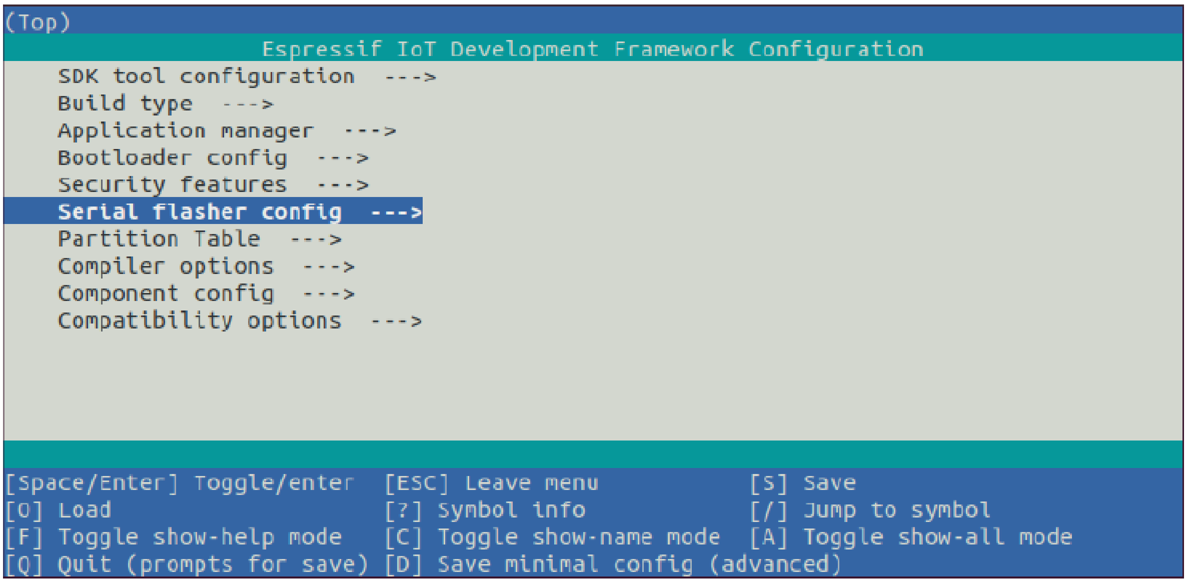
unplugged, then with plugged in. The port which appears the second time is the one you need:
ls /dev/tty*

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

Configure
Navigate to your ‘hello_world’ directory from Step 2.4.1. Start a Project, set ESP32 chip as the target and run the project configuration utility ‘menuconfig’.

```
cd ~/esp/hello_world
idf.py set-target esp32
idf.py menuconfig
```

Setting the target with ‘idf.py set-target esp32’ should be done once, after opening a new project. If the project contains some existing builds and configuration, they will be cleared and initialized. The target may be saved in environment variable to skip this step at all. 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 2: Project Configuration - Home Window

The colors of the menu could be different in your terminal. You can change the appearance with the option ‘style’. Please run ‘idf.py menu configuration help for further information.

Build the Project
Build the project by running:

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'
```

If there are no errors, the build will finish by generating the firmware binary .bin file.

Flash onto the Device
Flash the binaries that you just built onto your EK058 module by running:

```
idf.py -p PORT [-b BAUD] flash
```

Replace PORT with your module'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. If everything goes well, the "hello_world" application starts running after you remove the jumper on IO0 and GND, and re-power up the testing board.

Note: The option 'flash' automatically builds and flashes the project, so running 'idf.py build' is not necessary.

```
Running esptool.py in directory […]/esp/hello_world
Executing "python […]/esp-idf/components/esptool_py/esptool/esptool.py -b 460800 write_flash
@flash_project_args"…
esptool.py -b 460800 write_flash –flash_mode dio –flash_size detect –flash_freq 40m 0x1000
bootloader/bootloader.bin 0x8000 partition_table/partition-table.bin 0x10000 hello-world.bin esptool.py v2.3.1
Connecting….
Detecting chip type… ESP32
Chip is ESP32
Features: WiFi, BT, Dual Core
Uploading stub…
Running stub…
Stub running…
Changing baud rate to 460800
Changed.
Configuring flash size…
Auto-detected Flash size: 8MB
Flash params set to 0x0220
Compressed 22992 bytes to 13019…
Wrote 22992 bytes (13019 compressed) at 0x00001000 in 0.3 seconds (effective 558.9 kbit/s)… Hash of data
verified.
Compressed 3072 bytes to 82…
Wrote 3072 bytes (82 compressed) at 0x00008000 in 0.0 seconds (effective 5789.3 kbit/s)… Hash of data
verified.
Compressed 136672 bytes to 67544…
Wrote 136672 bytes (67544 compressed) at 0x00010000 in 1.9 seconds (effective 567.5 kbit/s)… Hash of data
verified.
Leaving…
Hard resetting via RTS pin…
```

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 esp32 chip with 2 CPU cores, WiFi/BT/BLE, silicon revision 3, 8MB flash
Restarting in 9 seconds…
Restarting in 8 seconds…
Restarting in 7 seconds…
```

To exit IDF monitor use the shortcut Ctrl+].

That’s all what you need to get started with EK058 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 device complies 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

Specific Operational Use Conditions
The module has WiFi, BR, EDR, and BLE functions.

- **Operation Frequency:**
  - WiFi: 2412 ~ 2462 MHz
  - Bluetooth: 2402 ~ 2480 MHz
- **Number of Channel:**
  - WiFi: 11
  - Bluetooth: 40
- **Modulation:**
  - WiFi: DSSS; OFDM
  - Bluetooth: GFSK; π/4 DQPSK; 8 DPSK
- **Type:** On-board PCB Antenna
- **Gain:** 3.26 dBi Max

The module can be used for IoT applications with a maximum 3.26 dBi antenna. The host manufacturer installing this module into their product must ensure that the final composit product complies with the FCC requirements by a technical assessment or evaluation to 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 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/warning as show in this manual.

Limited Module Procedures
Not applicable. The module is a single module and complies with the requirement of FCC Part 15.212.

Trace Antenna Designs
Not applicable. The module has its own antenna, and does not need a host's printed board microstrip trace antenna, etc.

RF Exposure Considerations
The module must be installed in the host equipment such that at least 20cm is maintained between the antenna and users' body; and if RF exposure statement or module layout is changed, then the host product manufacturer required to take responsibility of the module through a change in FCC ID or new application. The FCC ID of the module 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 specifications are as follows:

- **Type:** PCB Antenna
- **Gain:** 3.26 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 module shall be only used with the external antenna(s) that has been originally tested and certified with this module.
- The antenna must be either permanently attached or employ a 'unique' antenna coupler.

As long as the conditions above are met, further transmitter test will not be required. However, the host manufacturer is still responsible for testing their end-product for any additional compliance requirements required with this module 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-EK058" with their finished product.

Information on test modes and additional testing requirements

- **Operation Frequency:**
  - **WiFi:** 2412 ~ 2462 MHz
  - **Bluetooth:** 2402 ~ 2480 MHz
- **Number of Channel:**
  - **WiFi:** 11
  - **Bluetooth:** 40
- **Modulation:**
  - **WiFi:** DSSS; OFDM
  - **Bluetooth:** GFSK; π/4 DQPSK; 8 DPSK

Host manufacturer must perform test 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 and the host product manufacturer is responsible for compliance to 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 Part 15 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.

This device complies 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.

This 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

This device is intended only for OEM integrators under the following conditions:

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

As long as the conditions above are met, further transmitter test 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 (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 this module in combination with the host equipment is no longer considered valid and the FCC ID of the module 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-EK058”.

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 minimum distance 20 cm between the radiator and your body.

RSS247 Section 6.4 (5)
The device could automatically discontinue transmission in case of absence of information to transmit, or operational failure. Note that this is not intended to prohibit the 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 2 conditions above are met, further transmitter test 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 device 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-EK058”.

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/warning as show in this manual.

Related Documentation and Resources

Related Documentation

- **ESP32 Technical Reference Manual –** Detailed information on how to use the ESP32 memory and peripherals.
- **ESP32 Series Datasheet – S**pecifications of the ESP32 hardware.
- **ESP32 Hardware Design Guidelines –** Guidelines on how to integrate the ESP32 into your hardware product.
- **ESP32 ECO and Workarounds for Bugs –** Correction of ESP32 design errors.
- **Certificates**
  http://espressif.com/en/support/documents/certificates
- **ESP32 Product/Process Change Notifications (PCN)**
  http://espressif.com/en/support/documents/pcns
- **ESP32 Advisories – Information on security, bugs, compatibility, component reliability.**
  http://espressif.com/en/support/documents/advisories
- **Documentation Updates and Update Notification Subscription**
  http://espressif.com/en/support/download/documents

Developer Zone

- ESP-IDF Programming Guide for ESP32 – 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 Series SoCs – Browse through all ESP32 SoCs.
  http://espressif.com/en/products/socs?id=ESP32
- ESP32 Series Modules – Browse through all ESP32-based modules.
  http://espressif.com/en/products/modules?id=ESP32
- ESP32 Series DevKits – Browse through all ESP32-based devkits.
  http://espressif.com/en/products/devkits?id=ESP32
- 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

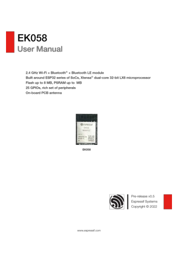
Contact Us

- 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

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 ARISE OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.
All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.
The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.
All trade names, trademarks, and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Documents / Resources

 EK058 User Manual 2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE module Built around ESP32 series of SoCs, Xtensa® dual-core 32-bit LX6 microprocessor Flash up to 8 MB, PSRAM up to MB 25 GPIOs, rich set of peripherals On-board PCB antenna EK058 Pre-release v0.5 Espressif Systems Copyright © 2022 www.espressif.com	**ESPRESSIF EK058 2.4 GHz WiFi Bluetooth LE Module** [pdf] User Manual EK058, 2AC7Z-EK058, 2AC7ZEK058, EK058 2.4 GHz Wi Fi Bluetooth LE Module, 2.4 GHz Wi Fi Bluetooth LE Module, Wi Fi Bluetooth LE Module, Bluetooth LE Module, LE Module

References

- Get Started - ESP32 - — ESP-IDF Programming Guide latest documentation
- ESP32 Forum - Index page
- Certificates | Espressif Systems
- Sales Questions | Espressif Systems
- Development Boards | Espressif Systems
- Modules | Espressif Systems
- Chipsets | Espressif Systems
- Subscribe | Espressif Systems
- Advisories | Espressif Systems
- Certificates | Espressif Systems
- PCNs | 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
- Build System - ESP32 - — ESP-IDF Programming Guide latest documentation
- ESP-IDF Versions - ESP32 - — ESP-IDF Programming Guide latest documentation
- Wi-Fi & Bluetooth MCUs and AIoT Solutions I Espressif Systems
- Technical Documents | Espressif Systems