

```
Administrator: ESP-IDF 4.3 CN | Search hello_world

Setting PYTHONNOUSERSITE, was not set
Using Python in C:\Espressif\python_env\idf4.3_py3.11_env\Scripts\
Using Git in C:\Espressif\tools\idf-git\2.43.0\cmd\
git version 2.43.0.windows.1
Setting IDF_PATH: C:\Espressif\frameworks\esp-idf-v4.3

Adding ESP-IDF tools to PATH...
C:\Espressif\tools\xtensa-esp32-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32-elf\bin
C:\Espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s2-elf\bin
C:\Espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s3-elf\bin
C:\Espressif\tools\risccv32-esp-elf\esp-2021r2-patch3-8.4.0\risccv32-esp-elf\bin
C:\Espressif\tools\esp32ulp-elf\esp-2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Espressif\tools\esp32s2ulp-elf\esp-2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Espressif\tools\cmake\3.16.4\bin
C:\Espressif\tools\openocd-esp32\0.11.0-esp32-20220706\openocd-esp32\bin
C:\Espressif\tools\ninja\1.10.2\
C:\Espressif\tools\idf-exe\1.0.1\
C:\Espressif\tools\ccache\3.7\
C:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
C:\Espressif\frameworks\esp-idf-v4.3\tools

Checking if Python packages are up to date...
Python requirements from C:\Espressif\frameworks\esp-idf-v4.3\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

    idf.py build
```

```
Administrator: ESP-IDF 4.3 CN | Search hello_world

[80/89] Linking C static library esp-idf\esp_hw_support\libesp_hw_support.a
[81/89] Linking C static library esp-idf\efuse\libefuse.a
[82/89] Linking C static library esp-idf\bootloader_support\libbootloader_support.a
[83/89] Linking C static library esp-idf\spi_flash\libspi_flash.a
[84/89] Linking C static library esp-idf\micro-ecc\libmicro-ecc.a
[85/89] Linking C static library esp-idf\soc\libsoc.a
[86/89] Linking C static library esp-idf\hal\libhal.a
[87/89] Linking C static library esp-idf\main\libmain.a
[88/89] Linking C executable bootloader.elf
[89/89] Generating binary image from built executable
espotaol.py v3.3.4-dev
Creating esp32 image...
Merged 1 ELF section
Successfully created esp32 image.
Generated C:/esp/hello_world/build/bootloader/bootloader.bin
[997/997] Generating binary image from built executable
espotaol.py v3.3.4-dev
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
Generated C:/esp/hello_world/build/hello-world.bin

Project build complete. To flash, run this command:
C:\Espressif\python_env\idf4.3_py3.11_env\Scripts\python.exe ..\..\Espressif\frameworks\esp-idf-v4.3\components\espotaol\py\espotaol\espotaol.py -p (PORT) -b 460800 --before default_reset --after hard_reset --chip esp32 write_flash --flash_mode dio --flash_size detect --flash_freq 40m 0x1000 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition_table.bin or run 'idf.py -p (PORT) flash'

C:\esp\hello_world>
```

```
Administrator: ESP-IDF 4.3 CN | C:\esp\hello_world>idf.py -p COM5 -b 460800 flash
Executing action: flash
Running ninja in directory c:\esp\hello_world\build
Executing "ninja flash"...
[1/4] Performing build step for 'bootloader'
ninja: no work to do.
[1/2] cmd.exe /C "cd /D C:\Espressif\frameworks\esp-idf-v4.../esp-idf-v4.3/components/esptool_py/run_serial_tool.cmake"
esptool.py esp32 -p COM5 -b 460800 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_freq 4
Picutu 0M --flash_size 2MB 0x8000 partition_table/partition-table.bin 0x1000 bootloader/bootloader.bin 0x10000 hello-world.bin
Music esptool.py v3.4-dev
Video Serial port COMS
Local Connecting.....
Local
Final A fatal error occurred: Failed to connect to ESP32: Wrong boot mode detected (0x13)! The chip needs to be in download mode.
DIP Li de.
Hello. For troubleshooting steps visit: https://docs.espressif.com/projects/esptool/en/latest/troubleshooting.html
CMake Error at run_serial_tool.cmake:50 (message):
  C:/Espressif/python_env/idf4_3_py3.11_env/Scripts/python.exe
  C:/Espressif/frameworks/esp-idf-v4.3/components/esptool_py/esptool.py --chip esp32" -D SERIAL_TOOL
  --chip esp32 failed
Log Log
Netw FAILED: CMakeFiles/flash
cmd.exe /C "cd /D C:\Espressif\frameworks\esp-idf-v4.3\components\esptool_py && C:\Espressif\tools\cmake\3.16.4\bin\cmake.exe -D IDF_PATH="C:/Espressif/frameworks/esp-idf-v4.3" -D SERIAL_TOOL="C:/Espressif/python_env/idf4_3_py3.11_env/Scripts/python.exe C:/Espressif/frameworks/esp-idf-v4.3/components/esptool_py/esptool.py --chip esp32" -D SERIAL_TOOL
_ARGS="--before=default_reset --after=hard_reset write_flash @flash_args" -D WORKING_DIRECTORY="C:/esp/hello_world/build"
" -P C:/Espressif/frameworks/esp-idf-v4.3/components/esptool_py/run_serial_tool.cmake"
ninja: build stopped: subcommand failed.

7 items |
```

```
Administrator: ESP-IDF 4.3 CN | C:\esp\hello_world>idf.py -p COM5 -b 460800 flash
Executing "ninja flash"...
[1/4] Performing build step for 'bootloader'
ninja: no work to do.
[1/2] cmd.exe /C "cd /D C:\Espressif\frameworks\esp-idf-v4.../esp-idf-v4.3/components/esptool_py/run_serial_tool.cmake"
esptool.py esp32 -p COM5 -b 460800 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_freq 40m --flash_size 2MB 0x8000 partition_table/partition-table.bin 0x1000 bootloader/bootloader.bin 0x10000 hello-world.bin
esptool.py v3.3.4-dev
Serial port COMS
Connecting....
Chip is ESP32-DWD-V3 (revision v3.1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: e8:0b:ea:cf:4d:c4
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x00000ffff...
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00010000 to 0x00039fff...
Compressed 3072 bytes to 103...
Writing at 0x00000800... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00000800 in 0.1 seconds (effective 254.7 kbit/s)...
Hash of data verified.
Compressed 26192 bytes to 16229...
Writing at 0x00001000... (100 %)
Wrote 26192 bytes (16229 compressed) at 0x00001000 in 0.9 seconds (effective 229.4 kbit/s)...
Hash of data verified.
Compressed 170624 bytes to 89228...
Writing at 0x00010000... (16 %)
Writing at 0x000106ab... (33 %)
Writing at 0x00020e46... (50 %)
Writing at 0x0002654a... (66 %)
Writing at 0x0002edc4... (83 %)
Writing at 0x00036ec8... (100 %)
Wrote 170624 bytes (89228 compressed) at 0x00010000 in 2.7 seconds (effective 509.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
Done
```

```
C:\esp\hello_world>idf.py -p COM5 monitor
Executing action: monitor
Running idf_monitor in directory c:\esp\hello_world
Executing "C:\Espressif\python_env\idf4.3_py3.11_env\Scripts\python.exe C:\Espressif\frameworks\esp-idf-v4.3\tools\idf_monitor.py -p COM5 -b 115200 --toolchain-prefix xtensa-esp32-elf- c:\esp\hello_world\build\hello-world.elf -m 'C:\Espressif\python_env\idf4.3_py3.11_env\Scripts\python.exe' 'C:\Espressif\frameworks\esp-idf-v4.3\tools\idf.py' '-p' 'COM5'..."
--- WARNING: GDB cannot open serial ports accessed as COMx
--- Using `.\COM5` instead...
C:\Espressif\frameworks\esp-idf-v4.3\tools\idf_monitor.py:518: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
  if StrictVersion(serial.VERSION) < StrictVersion('3.3.0'):
--- idf_monitor on `.\COM5` 115200 ---
--- Quit: Ctrl+Q | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:7256
load:0x40078000,len:15132
ho 0 tail 12 room 4
load:0x40080040,len:3712
0x40080400: _init at ???:?

entry 0x40080680
I (29) boot: ESP-IDF -128-NOTFOUND 2nd stage bootloader
I (29) boot: compile time 12:15:13
I (29) boot: chip revision: v3.1
I (33) boot.esp32: SPI Speed : 40MHz
I (37) boot.esp32: SPI Mode : DIO
I (42) boot.esp32: SPI Flash Size : 2MB
I (47) boot: Enabling RNG early entropy source...
I (52) boot: Partition Table:
I (56) boot: ## Label           Usage         Type ST Offset  Length
I (63) boot:  0 nvs            WiFi data    01 02 00009000 00006000
I (78) boot:  1 phy_init       RF data     01 01 0000F000 00001000
I (78) boot:  2 factory        factory app  00 00 00010000 00100000
I (85) boot: End of partition table
I (89) esp_image: segment 0: paddr=00010020 vaddr=3f400020 size=08258h ( 33368) map
I (110) esp_image: segment 1: paddr=00018280 vaddr=3ffb0000 size=02134h ( 8500) load
I (114) esp_image: segment 2: paddr=0001a3bc vaddr=40080000 size=05c5ch ( 23644) load

12:26 PM
```

```
I (114) esp_image: segment 2: paddr=0001a3bc vaddr=40080000 size=05c5ch ( 23644) load
I (127) esp_image: segment 3: paddr=00020020 vaddr=400d0020 size=142a8h ( 82600) map
I (157) esp_image: segment 4: paddr=000342d0 vaddr=40085c5c size=05784h ( 22404) load
I (172) boot: Loaded app from partition at offset 0x10000
I (172) boot: Disabling RNG early entropy source...
I (184) cpu_start: Pro cpu up.
I (184) cpu_start: Starting app cpu, entry point is 0x4008102c
0x4008102c: call_start_cpu1 at C:/Espressif/frameworks/esp-idf-v4.3/components/esp_system/port/cpu_start.c:153

I (0) cpu_start: App cpu up.
I (198) cpu_start: Pro cpu start user code
I (198) cpu_start: cpu freq: 16000000
I (198) cpu_start: Application information:
I (202) cpu_start: Project name: hello-world
I (208) cpu_start: App version: 1
I (212) cpu_start: Compile time: Mar 28 2024 12:14:57
I (218) cpu_start: ELF file SHA256: 5addfe0e5743e06a...
I (224) cpu_start: ESP-IDF: -128-NOTFOUND
I (230) cpu_start: Min chip rev: v0.8
I (234) cpu_start: Max chip rev: v3.99
I (239) cpu_start: Chip rev: v3.1
I (244) heap_init: Initializing. RAM available for dynamic allocation:
I (251) heap_init: At 3FFAE6E0 len 00001920 (6 Kib): DRAM
I (257) heap_init: At 3FFB2A28 len 0002D5D8 (181 Kib): DRAM
I (263) heap_init: At 3FFE0440 len 00003AE0 (14 Kib): D/IRAM
I (270) heap_init: At 3FFE4358 len 0001BCB0 (111 Kib): D/IRAM
I (276) heap_init: At 400883E0 len 00014C20 (83 Kib): IRAM
I (284) spi_flash: detected chip: generic
I (287) spi_flash: flash io: dio
W (291) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
I (305) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
This is esp32 chip with 2 CPU core(s), WiFi/BT/BLE, silicon revision v3.1, 2MB external flash
Minimum free heap size: 296752 bytes
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
Restarting in 6 seconds...
Restarting in 5 seconds...
Restarting in 4 seconds...
Restarting in 3 seconds...
12:27 PM
```

Configuration (ESP-IDF)

The ESP-IDF provides great control over project configuration. This configuration can be updated by running the following command from your terminal.

```
idf.py menuconfig
```



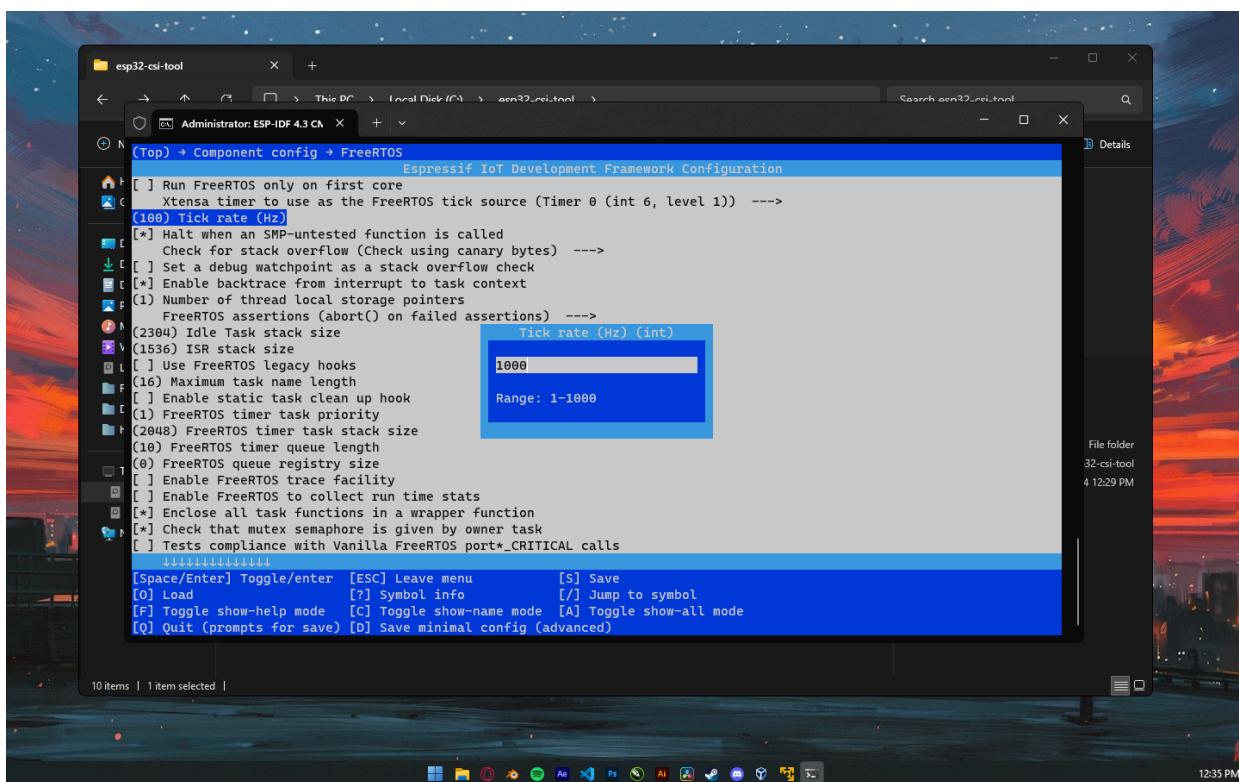
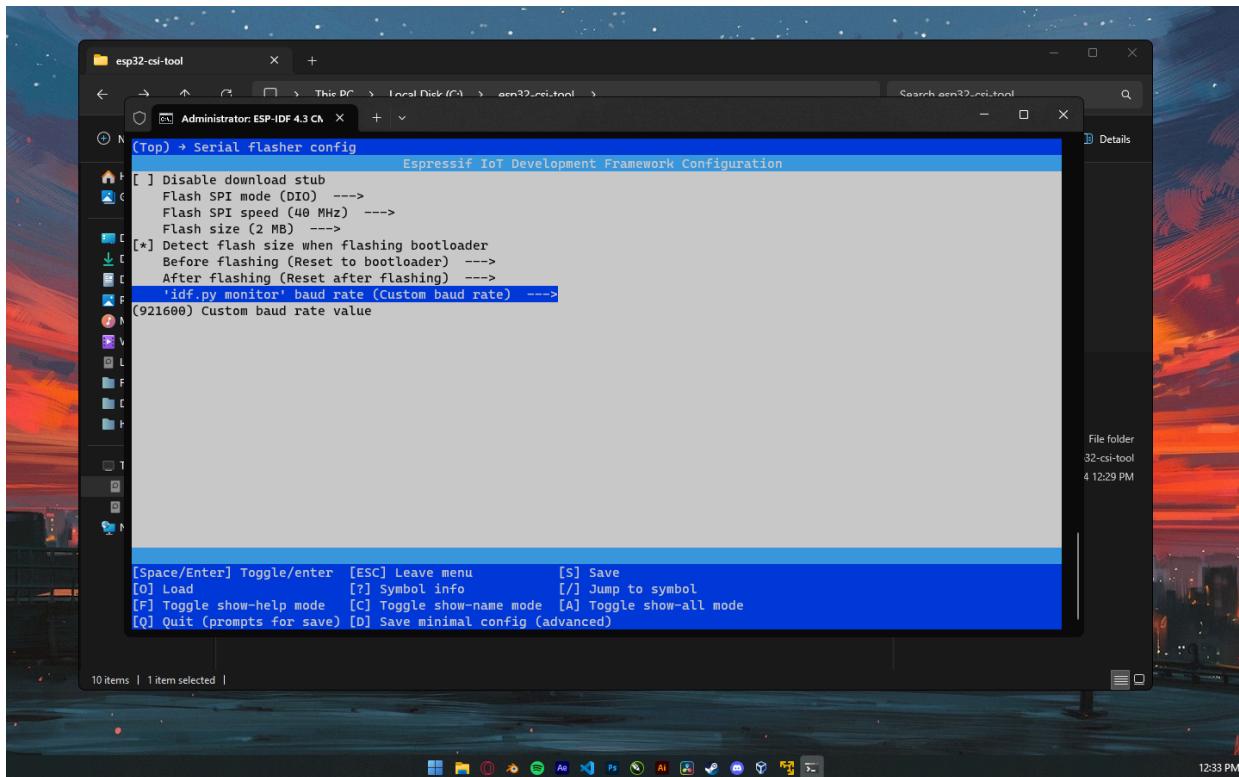
It is important to notice that these configurations are project specific and will not automatically be copied between sub-projects. So for example, make sure when you change the Wi-Fi password in *Active CSI collection (AP)*, you also update this configuration in the *Active CSI collection (Station)* project as well.

The following configurations are important for this project:

1. `Serial flasher config > 'idf.py monitor' baud rate > Custom Baud Rate`
2. `Serial flasher config > Custom baud rate value > 921600` This allows more data to be transmitted on the Serial port
3. `Component config > Common ESP32-related > Channel for console output > Custom UART`
4. `Component config > Common ESP32-related > UART console baud rate > 921600`
5. `Component config > Wi-Fi > WiFi CSI(Channel State Information) (Press space to select)`
6. `Component config > FreeRTOS > Tick rate (Hz) > 1000`
7. `ESP32 CSI Tool Config > ****` all options in this menu can be specified per your experiment requirements.

NOTE: For some systems, other baud rates may be required. Good options to try are `921600`, `1000000`, `1152000`, `1500000`, and `1552000`.

The higher baud rate the better! Baud rate is extremely important to achieve high sampling rates without lag!
If you have a problem, please leave any relevant information such as operating system, esp-idf version, list of all baud rates work and baud rates that do not work etc in [issue #5](#).



```
active_ap
Administrator: ESP-IDF 4.3 CN
Writing at 0x0000c5cea... (77 %)
Writing at 0x0000cc677... (80 %)
Writing at 0x0000d4395... (82 %)
Writing at 0x0000dd54d... (85 %)
Writing at 0x0000e48fb... (88 %)
Writing at 0x000ea108... (91 %)
Writing at 0x000ef851... (94 %)
Writing at 0x000f5352... (97 %)
Writing at 0x000fa90d... (100 %)
Wrote 963568 bytes (558744 compressed) at 0x00010000 in 14.8 seconds (effective 520.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
Done
C:\esp32-csif\active_ap>idf.py monitor | findstr "CSI_DATA" > test_data.csv
No serial ports found. Connect a device, or use '-p PORT' option to set a specific port.

--- WARNING: GDB cannot open serial ports accessed as COMx
--- Using \.\COM5 instead...
C:\Espressif\frameworks\esp-idf-v4.3\tools\idf_monitor.py:518: DeprecationWarning: distutils Version classes are deprecated. Use packaging.version instead.
if StrictVersion(serial.VERSION) < StrictVersion('3.3.0'):
--- idf_monitor on \\.\COM5 921600 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+H followed by Ctrl+H ---
```