

Partitions

Partition table

A file called **partitions.csv** is located in the root of the esp-idf project folder. With content similar to:

#	Name,	Type,	SubType,	Offset,	Size,	Flags
nvs,		data,	nvs,	0x9000,	0x4000,	
otadata,		data,	ota,	0xd000,	0x2000,	
phy_init,		data,	phy,	0xf000,	0x1000,	
ota_0,		app,	ota_0,	0x10000,	0x120000,	
ota_1,		app,	ota_1,	0x130000,	0x120000,	
config,		data,	phy,	0x250000,	0x4000,	
logging,		data,	phy,	0x254000,	0x145000,	

Updating a partition table

Switch to custom partitions

You must tell ESP-IDF to stop using the default layout and look at your new CSV file instead. In **idf.py menuconfig** navigate to Partition Table. Change Partition Table Type from “Single factory app” to Custom partition CSV file. Ensure the Custom partition CSV filename setting is set to partitions.csv. Press S to save and Q to quit.

Build partition table

Edit the contents of the partitions.csv file and run the following commands:

```
idf.py menuconfig
idf.py partition-table
idf.py build
```

Write Partition Table to ESP32

Flash the partition table by:

```
idf.py -p /dev/ttyUSB0 partition-table-flash
```

Or you can do a full flash which will also write the table.

```
idf.py -p /dev/ttyUSB0 flash
```

Read Partition Table from ESP32

Read the Partition Table Flashed on a Connected ESP32 If you want to pull the partition table off a physical ESP32 board to see what is currently flashed on it, you can use the target partition-table-dump.

```
idf.py -p [PORT] partition-table-dump
```

It invokes esptool.py in the background to read the binary partition data directly from the chip's flash memory (always located at offset 0x8000 or 0x9000 depending on your SDK configuration) and prints the structure to your terminal.

Erase a partition area

```
esptool_py/esptool/esptool.py --port /dev/ttyUSB0 erase_region 0x254000  
0x145000
```

Full Flash

```
idf.py -p /dev/ttyACM0 flash  
python -m esptool --chip esp32c3 -p /dev/ttyACM0 -b 460800 --before  
default_reset --after hard_reset write_flash --flash_mode dio --flash_size  
4MB --flash_freq 80m 0x0 build/bootloader/bootloader.bin 0x8000  
build/partition_table/partition-table.bin 0xd000 build/ota_data_initial.bin  
0x10000 build/LCR-Meter.bin 0x250000 default.config
```

Write Application to partition

```
python -m esptool --chip esp32c3 -p /dev/ttyACM0 -b 460800 --before  
default_reset --after hard_reset write_flash --flash_mode dio --flash_size  
4MB --flash_freq 80m 0x10000 build/ESR-Meter.bin
```

Read/Write config partition

Write Config to partition

```
esptool.py --port /dev/ttyACM0 write_flash --flash_mode qio 0x250000  
default.config
```

Read CONFIG from partition

```
esptool.py --port /dev/ttyACM0 -b 460800 --before default_reset --chip  
esp32c3 read_flash 0x250000 4096 downloaded.config
```

From:

<https://wiki.oscardegroot.nl/> - **HomeWiki**

Permanent link:

<https://wiki.oscardegroot.nl/doku.php?id=esp:esp32:partitions&rev=1788011503>

Last update: **2026/08/29 13:51**

