CEC Continuing Education Center

DesignNews

Designing Embedded Systems using the ESP32

DAY 3 : Programming and Writing the First Application

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

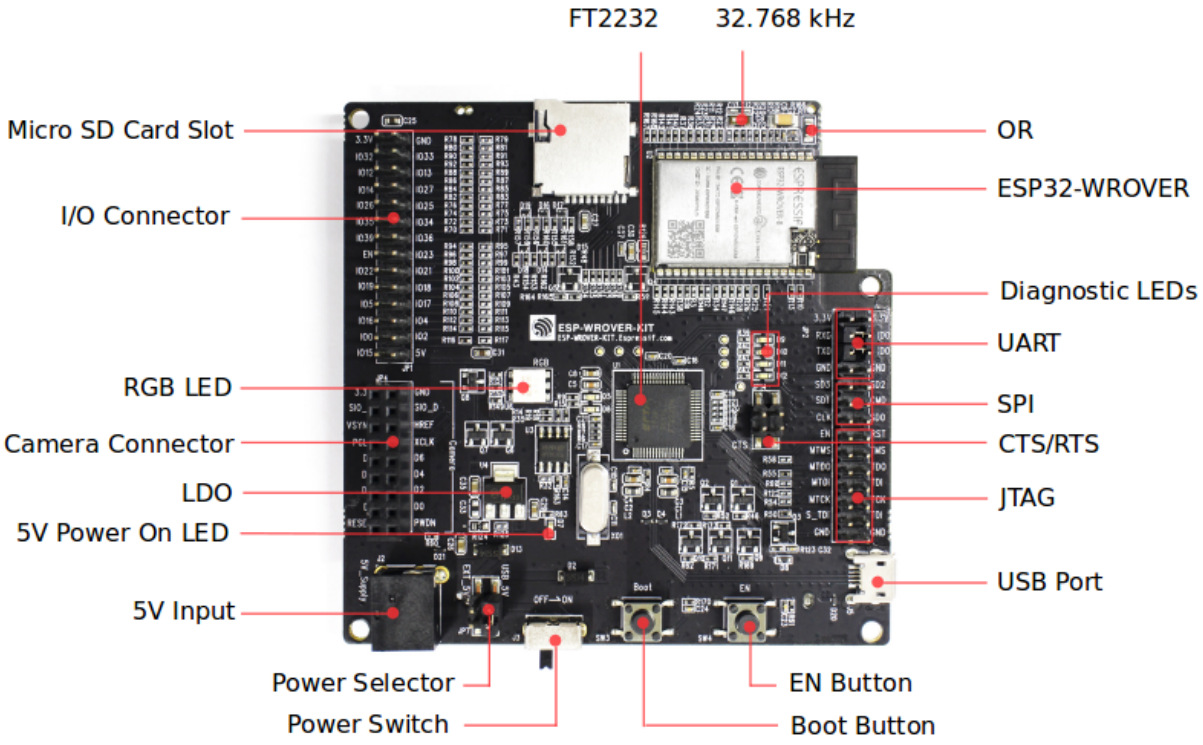
Course Sessions

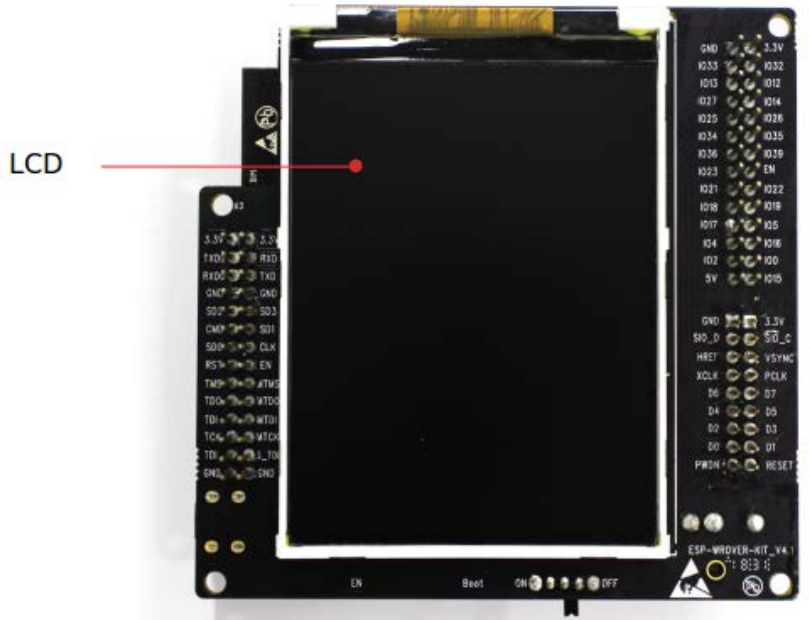
- Introduction to the ESP32 Wi-Fi Module
- Setting up and Exploring the SDK
- **Programming and Writing the First Application**
- It's all about Wi-Fi
- Jump-Starting Cloud Connectivity Applications with Amazon FreeRTOS

Will you be walking through the first application live?

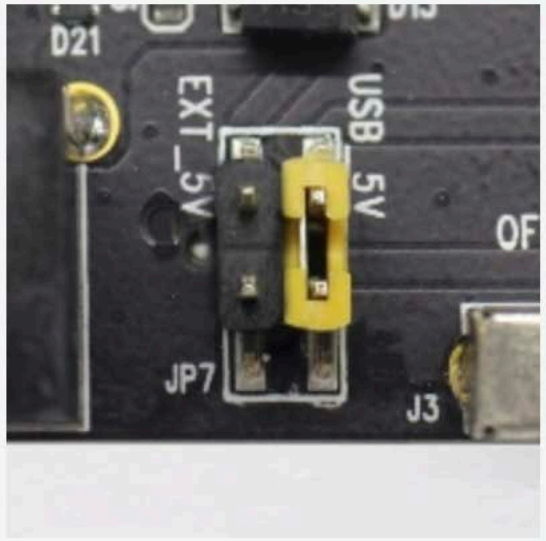
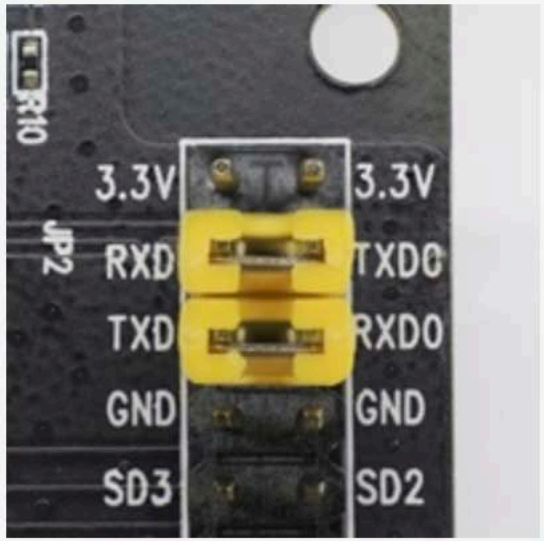
- Yes
- No

ESP32-WROVER-KIT





Jumper Configuration

Power up from USB port	Enable UART communication
	

Powering on with Boot (SW3) Pressed

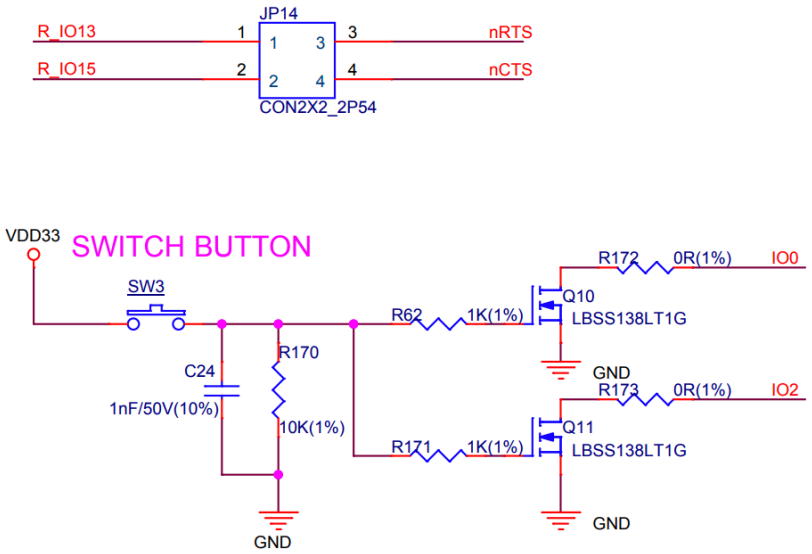
```
Setting IDF_PATH environment variable: /Users/beningo/esp/esp-idf
Executing action: monitor
Running idf_monitor in directory /Users/beningo/esp/hello_world
Executing "/Users/beningo/.espressif/python_env/idf4.2_py3.9_env/bin/python /Users/beningo/esp/esp-idf/tools/idf_monitor.py -p /dev/cu.usbserial-14301 -b 115200 --toolchain-pre
fix xtensa-esp32-elf- /Users/beningo/esp/hello_world/build/hello-world.elf -m '/Users/beningo/.espressif/python_env/idf4.2_py3.9_env/bin/python' '/Users/beningo/esp/esp-idf/too
ls/idf.py' '-p' '/dev/cu.usbserial-14301'"...
--- idf_monitor on /dev/cu.usbserial-14301 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:0x26 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_FEO_V2))
waiting for download
```

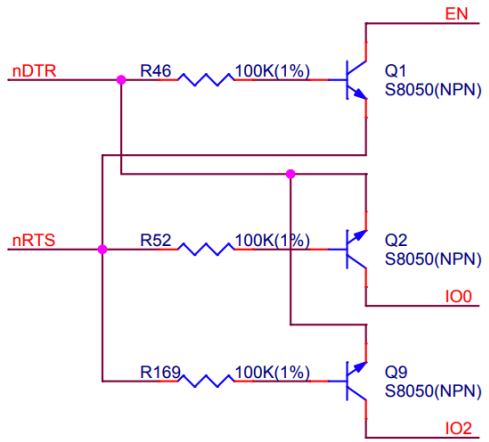
Pressing EN (SW4)

- Resets the processor

Programming Circuit

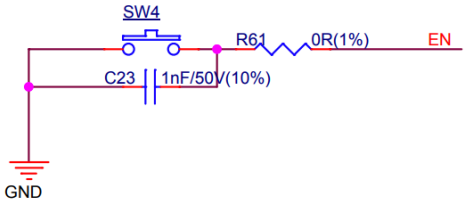
Switch:





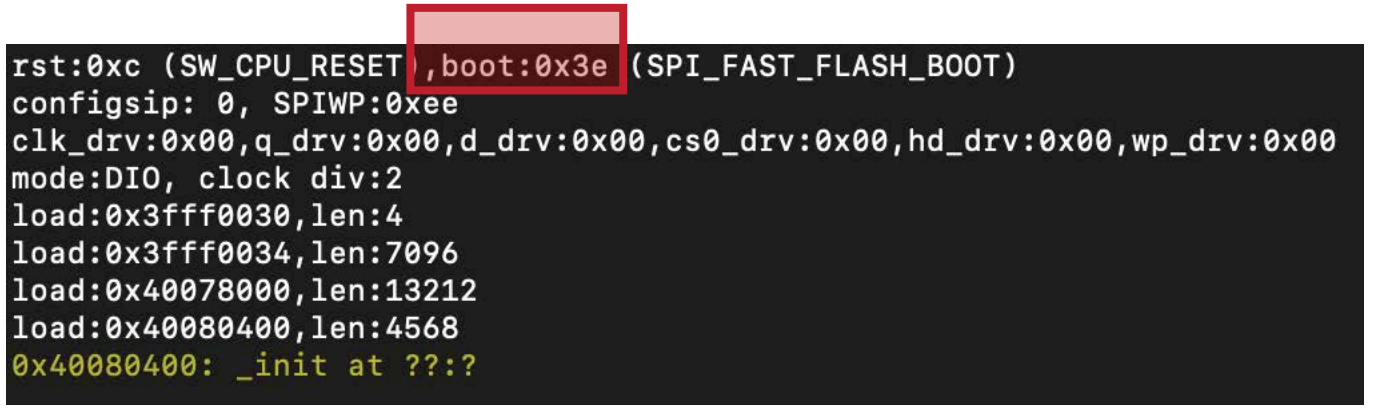
Auto program

DTR	RTS-->EN	IO0
1	1	1
0	0	1
1	0	0
0	1	1
1	1	0



Boot Capabilities

- 0x01 - GPIO5
- 0x02 - MTDO (GPIO15)
- 0x04 - GPIO4
- 0x08 - GPIO2
- 0x10 - GPIO0
- 0x20 - MTDI (GPIO12)



Which boot mode do you typically prefer?

- Internal flash
- External flash
- SDIO
- Other

Run the Hello World Example

1) Connect your device
2) Slide J3 into the ON position
3) Identify the communication port the USB enumerates on

```
beningo@Jacobs-MacBook-Pro hello_world % ls /dev/cu.*
/dev/cu.Bluetooth-Incoming-Port /dev/cu.JBLCharge4-SPPDev       /dev/cu.usbserial-14300       /dev/cu.usbserial-14301
```

4) Copy the Hello World example by executing:

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

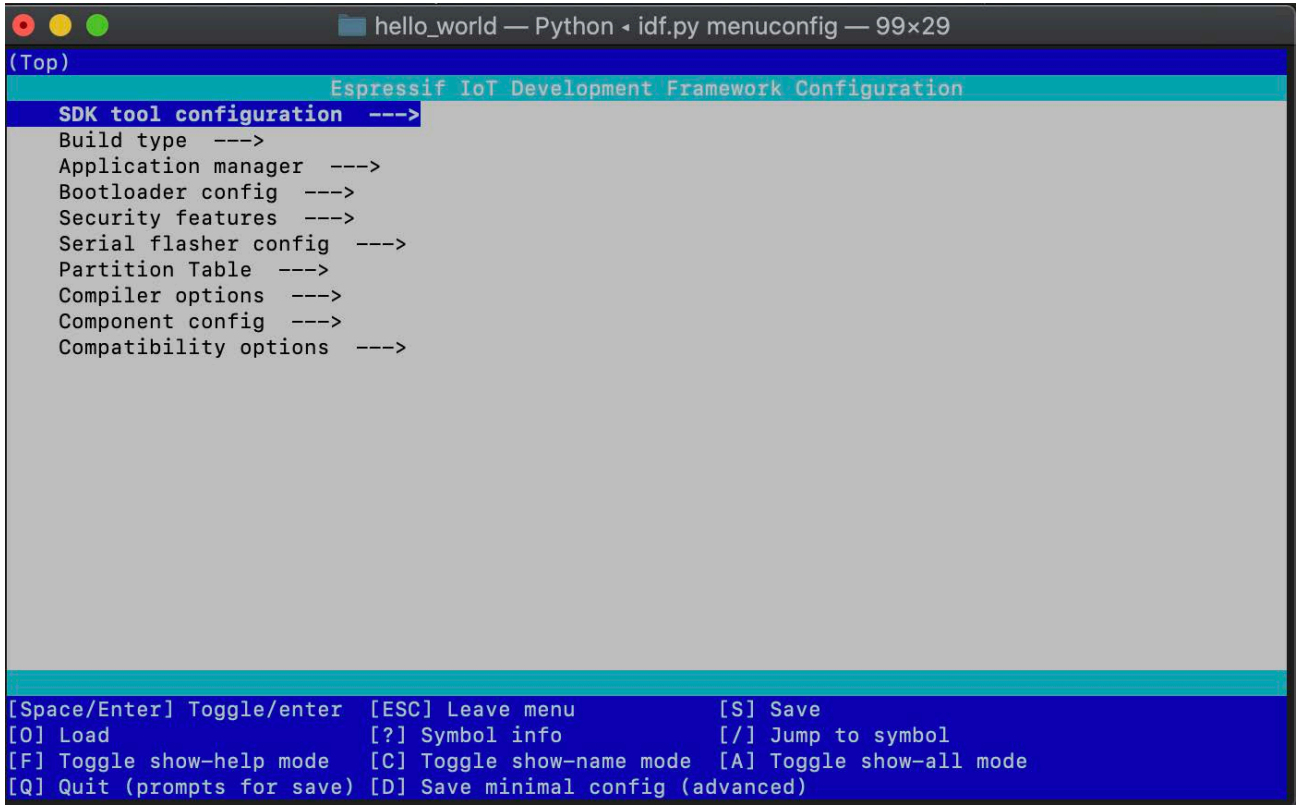
Run the Hello World Example

5) Run the following:

cd ~/esp/hello_world

idf.py set-target esp32

idf.py menuconfig



Run the Hello World Example

6) Run the following to build:

idf.py build

```
$ 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
or run 'idf.py -p PORT flash'
```

Run the Hello World Example

7) idf.py –p /dev/cu.usbserial14301 flash

8) idf.py –p /dev/cu.usbserial14301 monitor

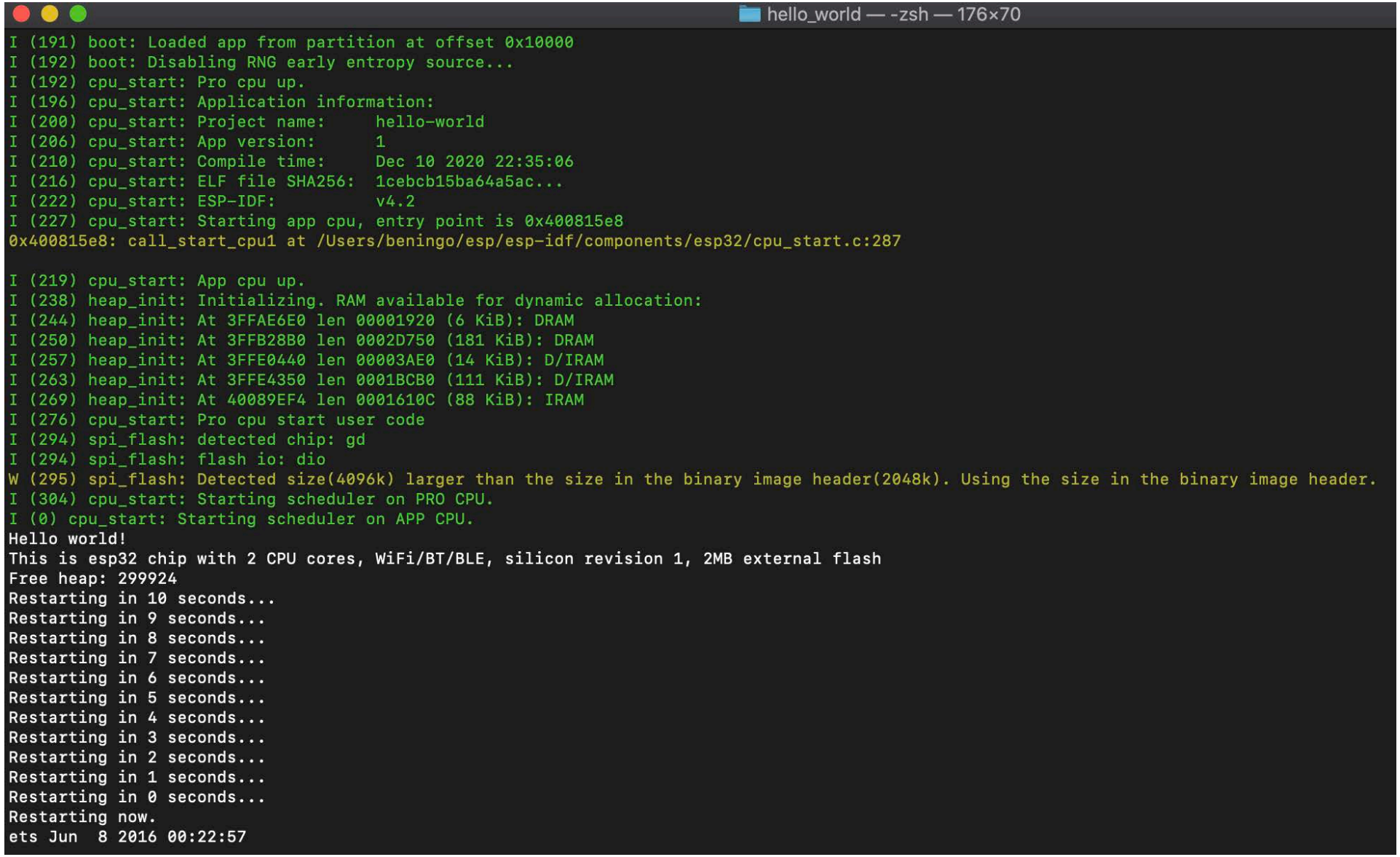
hello_world — -zsh — 176×70

```
rst:0xc (SW_CPU_RESET),boot:0x3e (SPI_FAST_FLASH_BOOT)
configsip: 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:4
load:0x3fff0034,len:7096
load:0x40078000,len:13212
load:0x40080400,len:4568
0x40080400: _init at ??:?

entry 0x400806f4
I (29) boot: ESP-IDF v4.2 2nd stage bootloader
I (29) boot: compile time 22:35:16
I (29) boot: chip revision: 1
I (32) boot_comm: chip revision: 1, min. bootloader chip revision: 0
I (41) boot.esp32: SPI Speed      : 40MHz
I (43) boot.esp32: SPI Mode       : DIO
I (48) boot.esp32: SPI Flash Size : 2MB
I (52) boot: Enabling RNG early entropy source...
I (58) boot: Partition Table:
I (61) boot: ## Label            Usage          Type ST Offset   Length
I (69) boot:  0 nvs              WiFi data        01 02 00009000 00006000
I (76) boot:  1 phy_init         RF data          01 01 0000f000 00001000
I (84) boot:  2 factory          factory app      00 00 00010000 00100000
I (91) boot: End of partition table
I (95) boot_comm: chip revision: 1, min. application chip revision: 0
I (102) esp_image: segment 0: paddr=0x00010020 vaddr=0x3f400020 size=0x05b64 ( 23396) map
I (120) esp_image: segment 1: paddr=0x00015b8c vaddr=0x3ffb0000 size=0x02074 (  8308) load
I (124) esp_image: segment 2: paddr=0x00017c08 vaddr=0x40080000 size=0x00404 (  1028) load
0x40080000: _WindowOverflow4 at /Users/beningo/esp/esp-idf/components/freertos/xtensa/xtensa_vectors.S:1730

I (130) esp_image: segment 3: paddr=0x00018014 vaddr=0x40080404 size=0x08004 ( 32772) load
I (153) esp_image: segment 4: paddr=0x00020020 vaddr=0x400d0020 size=0x12fe4 ( 77796) map
0x400d0020: _stext at ??:?

I (183) esp_image: segment 5: paddr=0x0003300c vaddr=0x40088408 size=0x01aec (  6892) load
0x40088408: rtc_init at /Users/beningo/esp/esp-idf/components/soc/src/esp32/rtc_init.c:32
```

What is the partition table?

- Defines what is stored in flash memory such as calibration data, filesystems, parameters, applications, etc
- Length is 0xC00 (95 entries maximum)
- Default location is 0x8000
- Can store multiple applications for OTA updates
- Each entry contain a name, subtype and offset

```
# ESP-IDF Partition Table
# Name,   Type, SubType, Offset,  Size, Flags
nvs,      data, nvs,     0x9000,  0x6000,
phy_init, data, phy,     0xf000,  0x1000,
factory,  app,  factory, 0x10000, 1M,
```


hello_world — -zsh — 176×70
I (191) boot: Loaded app from partition at offset 0x10000
I (192) boot: Disabling RNG early entropy source...
I (192) cpu_start: Pro cpu up.
I (196) cpu_start: Application information:
I (200) cpu_start: Project name: hello-world
I (206) cpu_start: App version: 1
I (210) cpu_start: Compile time: Dec 10 2020 22:35:06
I (216) cpu_start: ELF file SHA256: 1cebcb15ba64a5ac...
I (222) cpu_start: ESP-IDF: v4.2
I (227) cpu_start: Starting app cpu, entry point is 0x400815e8
0x400815e8: call_start_cpu1 at /Users/beningo/esp/esp-idf/components/esp32/cpu_start.c:287
I (219) cpu_start: App cpu up.
I (238) heap_init: Initializing. RAM available for dynamic allocation:
I (244) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (250) heap_init: At 3FFB28B0 len 0002D750 (181 KiB): DRAM
I (257) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (263) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (269) heap_init: At 40089EF4 len 0001610C (88 KiB): IRAM
I (276) cpu_start: Pro cpu start user code
I (294) spi_flash: detected chip: gd
I (294) spi_flash: flash io: dio
W (295) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the binary image header.
I (304) 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 cores, WiFi/BT/BLE, silicon revision 1, 2MB external flash
Free heap: 299924
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...
Restarting in 2 seconds...
Restarting in 1 seconds...
Restarting in 0 seconds...
Restarting now.
ets Jun 8 2016 00:22:57

What next steps are you going to take?

- Review the Hello World application code
- Dig into Wi-Fi
- Dig into Bluetooth
- Other

Thank you for attending

Please consider the resources below:

- www.beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm

From www.beningo.com under

- Blog > CEC – Designing Embedded Systems using the ESP32

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets