CEC Continuing Education Center

DesignNews

Designing Embedded Systems using the ESP32

DAY 5 : Jump-Starting Cloud Connectivity Applications with Amazon FreeRTOS

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.

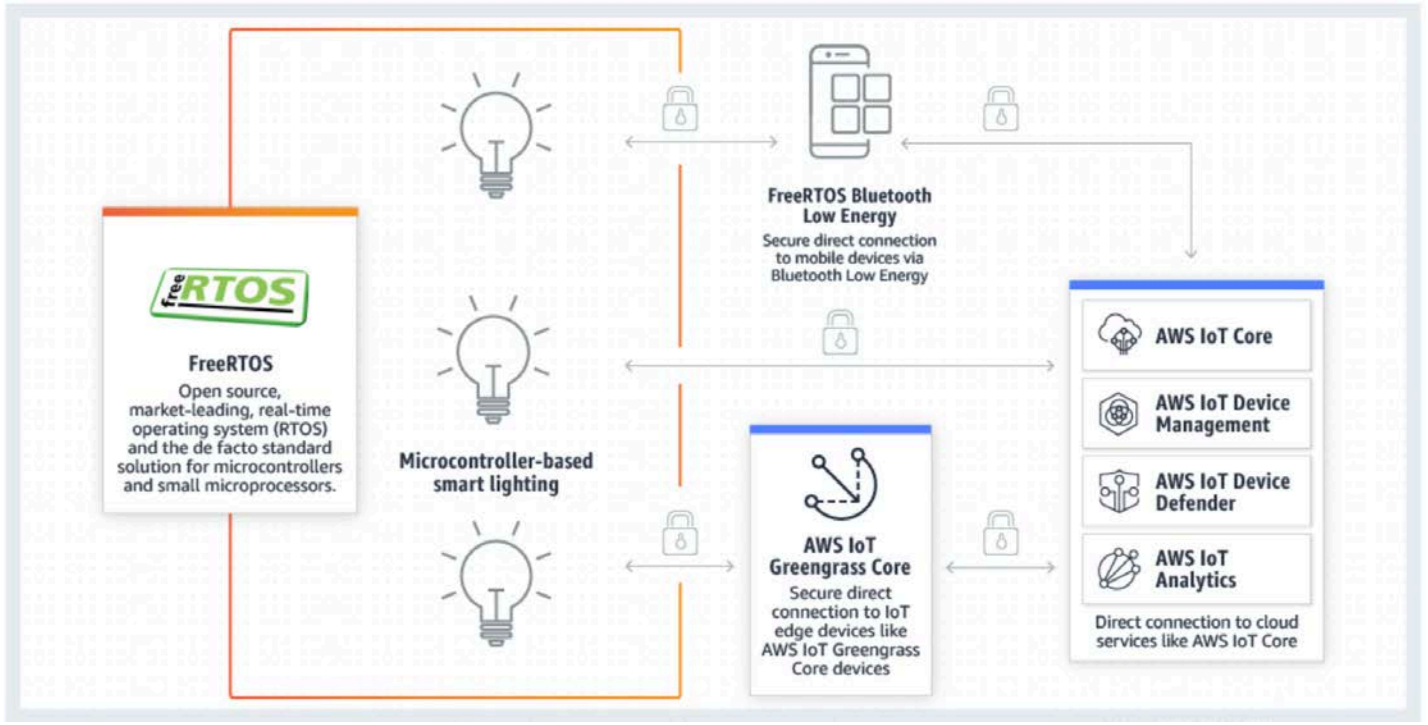
Course Sessions

- 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**

“Amazon” FreeRTOS

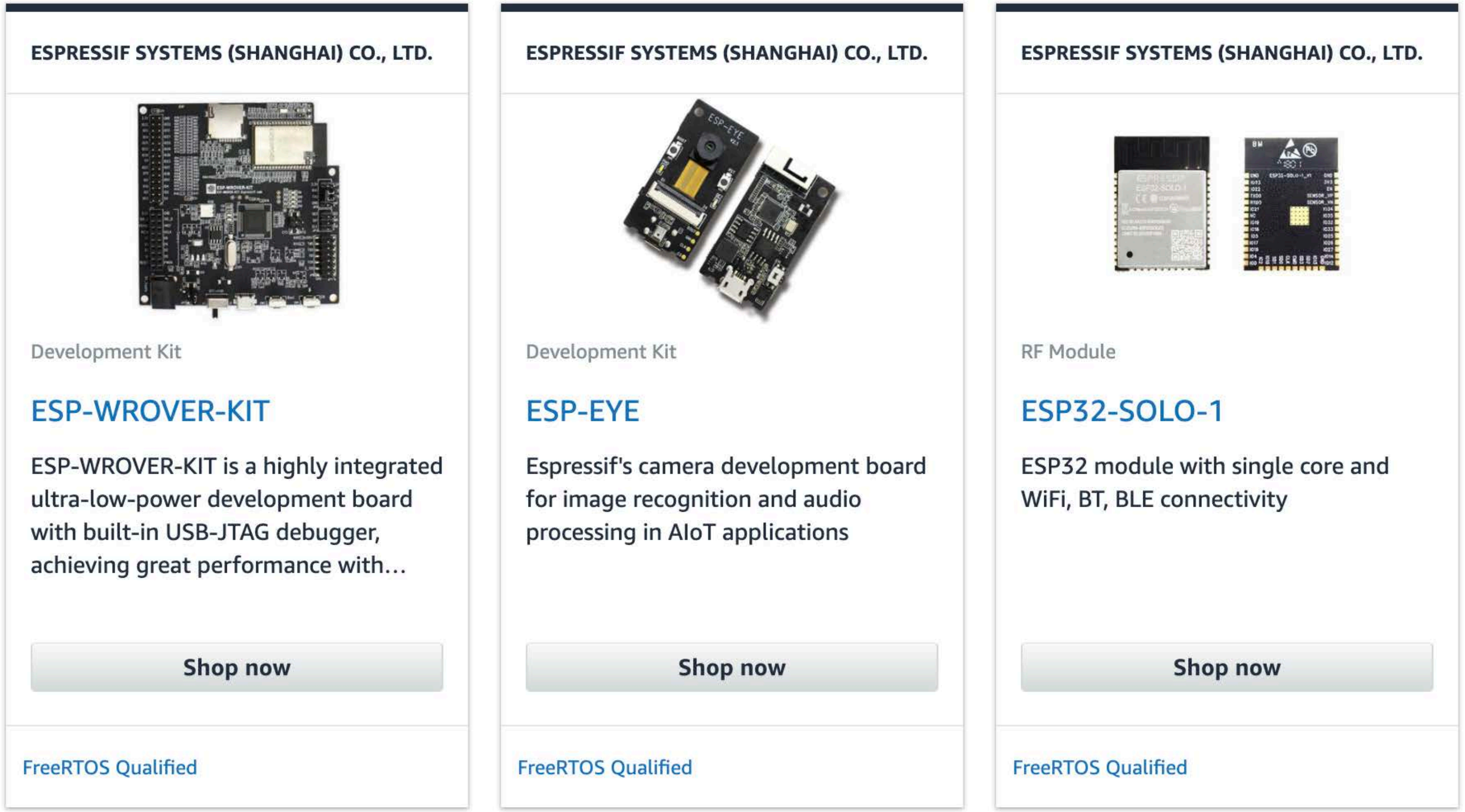
Main Features

- FreeRTOS
- Easy AWS connectivity
- MQTT demo example
- Manage connected devices
- OTA demo example
- Integrated libraries
- Open source


freeRTOS
FreeRTOS
Open source, market-leading, real-time operating system (RTOS) and the de facto standard solution for microcontrollers and small microprocessors.
Microcontroller-based smart lighting
FreeRTOS Bluetooth Low Energy
Secure direct connection to mobile devices via Bluetooth Low Energy
AWS IoT Greengrass Core
Secure direct connection to IoT edge devices like AWS IoT Greengrass Core devices
AWS IoT Core
AWS IoT Device Management
AWS IoT Device Defender
AWS IoT Analytics
Direct connection to cloud services like AWS IoT Core

Source: Amazon

Amazon FreeRTOS ESP32 Development Boards



ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD.

Development Kit

ESP-WROVER-KIT

ESP-WROVER-KIT is a highly integrated ultra-low-power development board with built-in USB-JTAG debugger, achieving great performance with…

Shop now

FreeRTOS Qualified

ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD.

Development Kit

ESP-EYE

Espressif's camera development board for image recognition and audio processing in AIoT applications

Shop now

FreeRTOS Qualified

ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD.

RF Module

ESP32-SOLO-1

ESP32 module with single core and WiFi, BT, BLE connectivity

Shop now

FreeRTOS Qualified

Image Source: Amazon

What is the greatest advantage to using Amazon FreeRTOS?

- Integrated development environment
- More development libraries
- Cloud connectivity
- Security solutions

Getting Started

https://docs.aws.amazon.com/freertos/latest/userguide/getting_started_espressif.html

1) Setup AWS user and permissions
2) Clone the repository:

   https://github.com/aws/amazon-freertos.git

3) Walk through “Configure the FreeRTOS demo applications” in getting started guide.

Configure ESP32

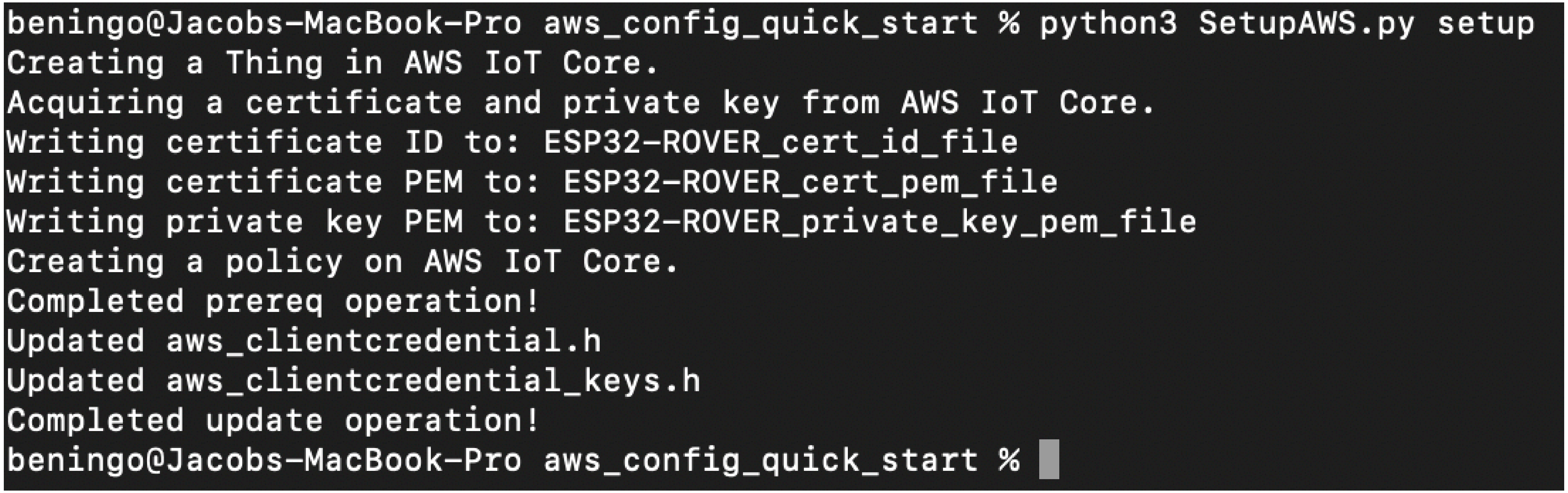
freertos/tools/aws_config_quick_start/configure.json

```
{
    "afr_source_dir":"../..",
    "thing_name":"$thing_name",
    "wifi_ssid":"$wifi_ssid",
    "wifi_password":"$wifi_password",
    "wifi_security":"$wifi_security"
}
```

Full path to afr

eWiFiSecurityWPA2

Run SetupAWS.py



```
beningo@Jacobs-MacBook-Pro aws_config_quick_start % python3 SetupAWS.py setup
Creating a Thing in AWS IoT Core.
Acquiring a certificate and private key from AWS IoT Core.
Writing certificate ID to: ESP32-ROVER_cert_id_file
Writing certificate PEM to: ESP32-ROVER_cert_pem_file
Writing private key PEM to: ESP32-ROVER_private_key_pem_file
Creating a policy on AWS IoT Core.
Completed prereq operation!
Updated aws_clientcredential.h
Updated aws_clientcredential_keys.h
Completed update operation!
beningo@Jacobs-MacBook-Pro aws_config_quick_start %
```

How familiar are you with AWS and the CLI?

- Beginner
- Intermediate
- Advanced
- Never heard of it

Build the application

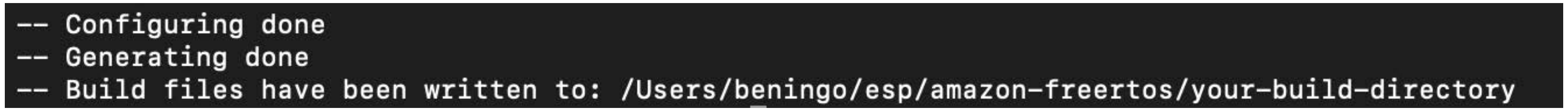
From the freeRTOS root directory:

```
cmake -DVENDOR=espressif -DBOARD=esp32_wrover_kit –
DCOMPILER=xtensa-esp32 -S . -B your-build-directory
```

cmake will clone various submodules and take several minutes to generate the build scripts.

You may need to clear your IDF_PATH using: export IDF_PATH=

Build the application



```
-- Configuring done
-- Generating done
-- Build files have been written to: /Users/beningo/esp/amazon-freertos/your-build-directory
```

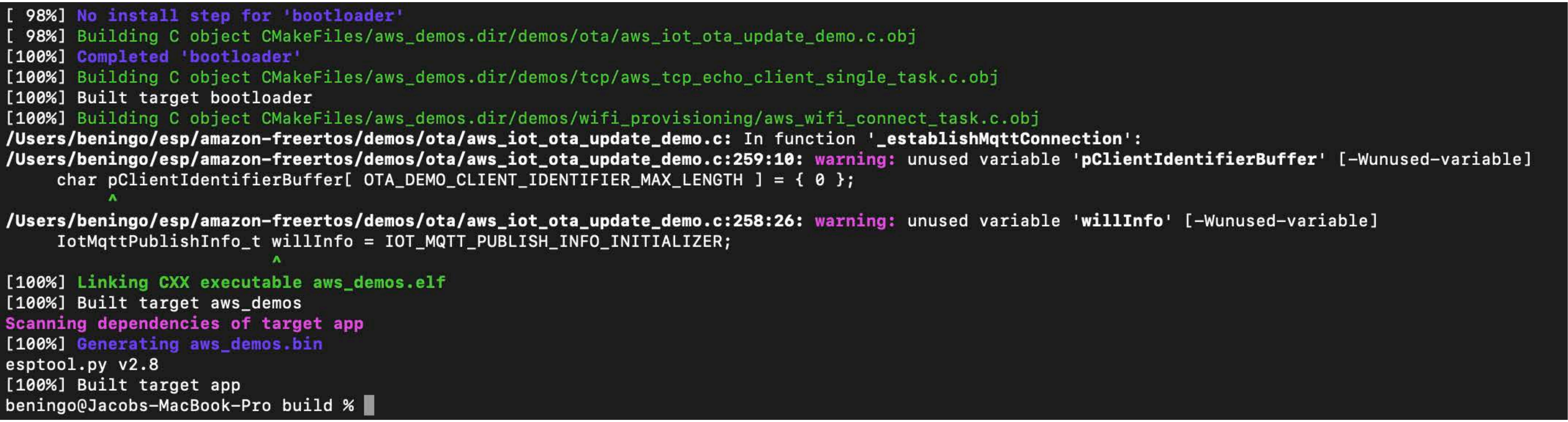
Additional flags for generating the build files:
-DCMAKE_BUILD_TYPE=Debug
-DAFR_ENABLE_TESTS=1
–DAFR_ESP_FREERTOS_TCP
    - switch between the FREERTOS or LWIP libraries

Build the final image using:
    make all -j4

Continuing Education Center

Sponsored By

Build the application



```
[ 98%] No install step for 'bootloader'
[ 98%] Building C object CMakeFiles/aws_demos.dir/demos/ota/aws_iot_ota_update_demo.c.obj
[100%] Completed 'bootloader'
[100%] Building C object CMakeFiles/aws_demos.dir/demos/tcp/aws_tcp_echo_client_single_task.c.obj
[100%] Built target bootloader
[100%] Building C object CMakeFiles/aws_demos.dir/demos/wifi_provisioning/aws_wifi_connect_task.c.obj
/Users/beningo/esp/amazon-freertos/demos/ota/aws_iot_ota_update_demo.c: In function '_establishMqttConnection':
/Users/beningo/esp/amazon-freertos/demos/ota/aws_iot_ota_update_demo.c:259:10: warning: unused variable 'pClientIdentifierBuffer' [-Wunused-variable]
     char pClientIdentifierBuffer[ OTA_DEMO_CLIENT_IDENTIFIER_MAX_LENGTH ] = { 0 };
          ^
/Users/beningo/esp/amazon-freertos/demos/ota/aws_iot_ota_update_demo.c:258:26: warning: unused variable 'willInfo' [-Wunused-variable]
     IotMqttPublishInfo_t willInfo = IOT_MQTT_PUBLISH_INFO_INITIALIZER;
                          ^
[100%] Linking CXX executable aws_demos.elf
[100%] Built target aws_demos
Scanning dependencies of target app
[100%] Generating aws_demos.bin
esptool.py v2.8
[100%] Built target app
beningo@Jacobs-MacBook-Pro build % 
```

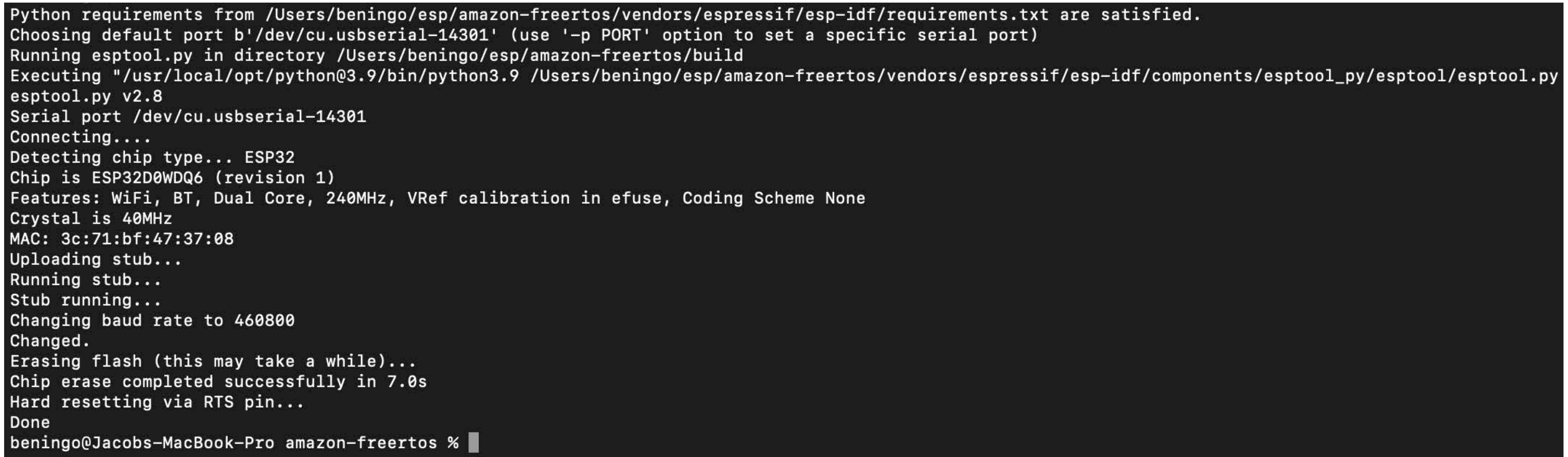
CEC Continuing Education Center

Sponsored By

Programming the Board

1) From <freertos>, run:

./vendors/espressif/esp-idf/tools/idf.py erase_flash -B build-directory



```
Python requirements from /Users/beningo/esp/amazon-freertos/vendors/espressif/esp-idf/requirements.txt are satisfied.
Choosing default port b'/dev/cu.usbserial-14301' (use '-p PORT' option to set a specific serial port)
Running esptool.py in directory /Users/beningo/esp/amazon-freertos/build
Executing "/usr/local/opt/python@3.9/bin/python3.9 /Users/beningo/esp/amazon-freertos/vendors/espressif/esp-idf/components/esptool_py/esptool/esptool.py
esptool.py v2.8
Serial port /dev/cu.usbserial-14301
Connecting....
Detecting chip type... ESP32
Chip is ESP32D0WDQ6 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 3c:71:bf:47:37:08
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Erasing flash (this may take a while)...
Chip erase completed successfully in 7.0s
Hard resetting via RTS pin...
Done
beningo@Jacobs-MacBook-Pro amazon-freertos %
```

Programming the Board

To monitor what is going on use:

```
./vendors/espressif/esp-idf/tools/idf.py monitor -B build-directory
```

Examine your cloud MQTT Client for topic messages.

Ideas for where to go from here:

1) Try additional Amazon FreeRTOS demos
2) Try to Bluetooth demos
3) Modify the default application for your own purposes
4) Experiment with OTA examples

What capabilities are you looking to try next?

- Try additional Amazon FreeRTOS demos
- Try to Bluetooth demos
- Modify the default application for your own purposes
- Experiment with OTA examples

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