



life.augmented

A night-time aerial view of a city skyline with numerous skyscrapers. Overlaid on the image is a network diagram with white nodes and lines connecting them. Several of these nodes are accompanied by blue Wi-Fi signal icons, representing IoT connectivity across the urban environment.

STM32U5 AWS IoT Hands-on

Slim JALLOULI

October 2022

Agenda

- 1 STM32U5 Introduction
- 2 FreeRTOS STM32U5 Reference Integration
- 3 STM32U5 AWS QuickConnect
- 4 Boards distribution
- 5 Hands-on

STM32U5 Microcontrollers

The new reference for secure and smart IoT applications



Higher Security

Certified PSA L3 and SESIP L3

Lower Power

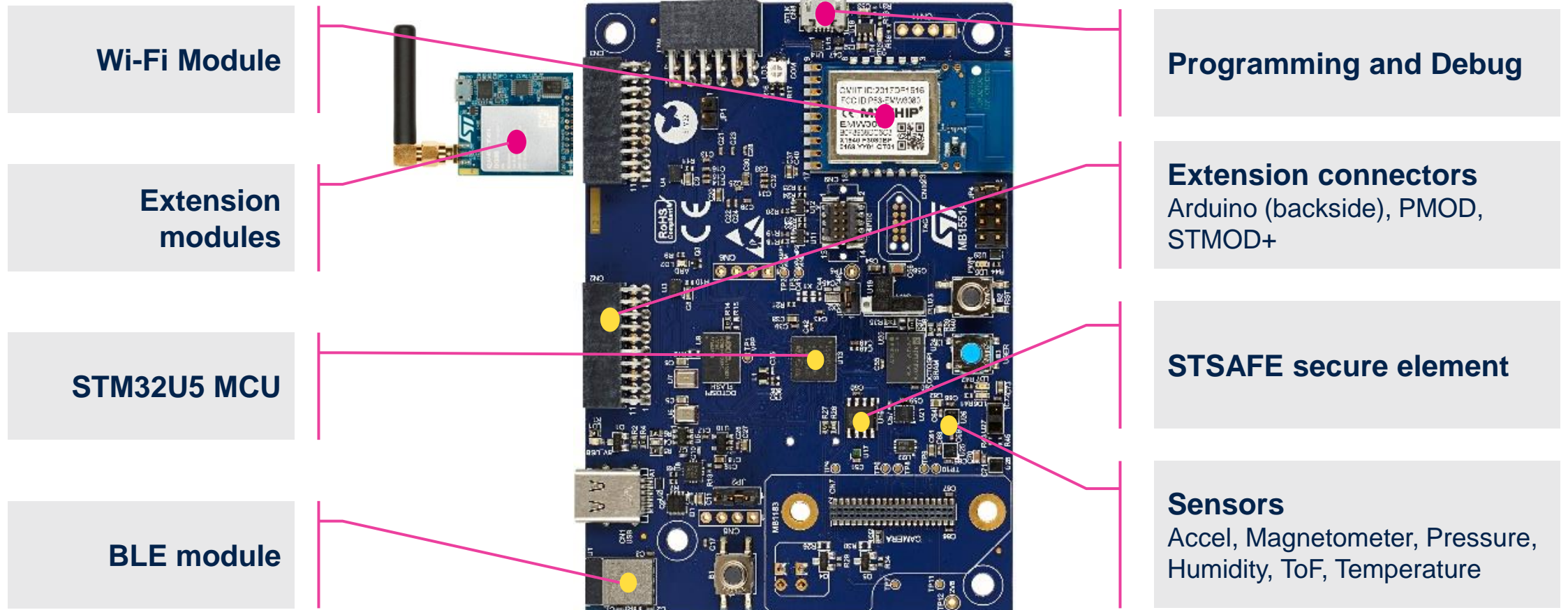
58 ULPMark-CM

Richer applications

Cortex-M33 @ 160MHz, extended features set

STM32U5 IoT Kit

Your reference board for IoT Proof-of-Concept



[B-U585I-IOT02A](#)

AWS IoT on STM32U5

X-CUBE-AWS reference integration simplifying your development



STM32
CubeExpansion
X-CUBE-AWS

AWS Certified

Leveraging ARM Trusted Firmware-M (TF-M)

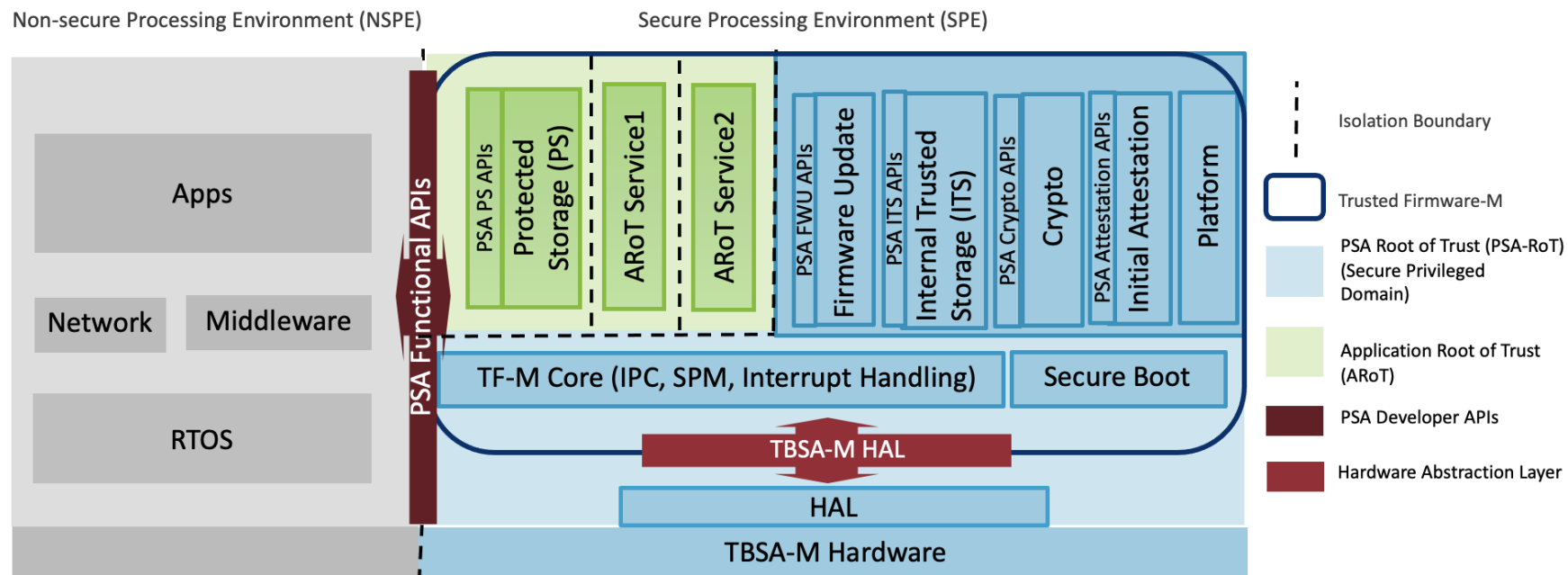
Based on FreeRTOS LTS Library

Over The Air Update

AWS IoT Defender

Trusted Firmware-M (TF-M)


- Trusted Firmware for Cortex M (TF-M) implements the Secure Processing Environment (SPE) for Armv8-M, Armv8.1-M
- **Trusted Firmware-M consists of:**
 - Secure Boot, Control the isolation, communication and execution within SPE and with NSPE
 - Crypto, Internal Trusted Storage (ITS), Protected Storage (PS) and Attestation secure services



STM32U5 FreeRTOS reference

Featured FreeRTOS IoT Integration

- <https://www.freertos.org/STM32U5/>



[KERNEL](#) [LIBRARIES](#) [SUPPORT](#) [PARTNERS](#) [COMMUNITY](#) [Download FreeRTOS](#)

LIBRARIES

- [Home](#)
- [LTS Libraries](#)
- [All libraries](#)

WHAT'S NEW

FreeRTOS-Plus-TCP v3.0.0 released:
We've added comprehensive unit tests and penetration and protocol testing. See the [blog post](#).

Featured FreeRTOS IoT Integrations:
Introducing three featured integrations for more secure IoT applications. See the [blog post](#).

Extended Maintenance Plan (EMP):
Announcing the EMP for FreeRTOS, provided by AWS. See the [blog post](#).

Featured FreeRTOS IoT Integration

Targeting an STM32U5 Arm Cortex-M33 MCU

- On this page:
 - [Introduction](#)
 - [Demonstrated security features and functions](#)
 - [Reducing the potential for attack by isolating critical security firmware and data](#)
 - [Cryptographic operations](#)
 - [Keeping device identity and secrets secure](#)
 - [Secure TLS communication with mutual authentication](#)
 - [Secure over the air updates \(OTA\)](#)
 - [Anti-Rollback protection](#)
 - [Memory safety proofs](#)
 - [Getting started with the demo](#)

Introduction

This demo shows how to integrate modular FreeRTOS software with hardware enforced security to help create more secure cloud connected applications. The projects are preconfigured to run on the [B-U585-IOT02A](#) IoT discovery kit which includes an [STM32U5](#) microcontroller (MCU).

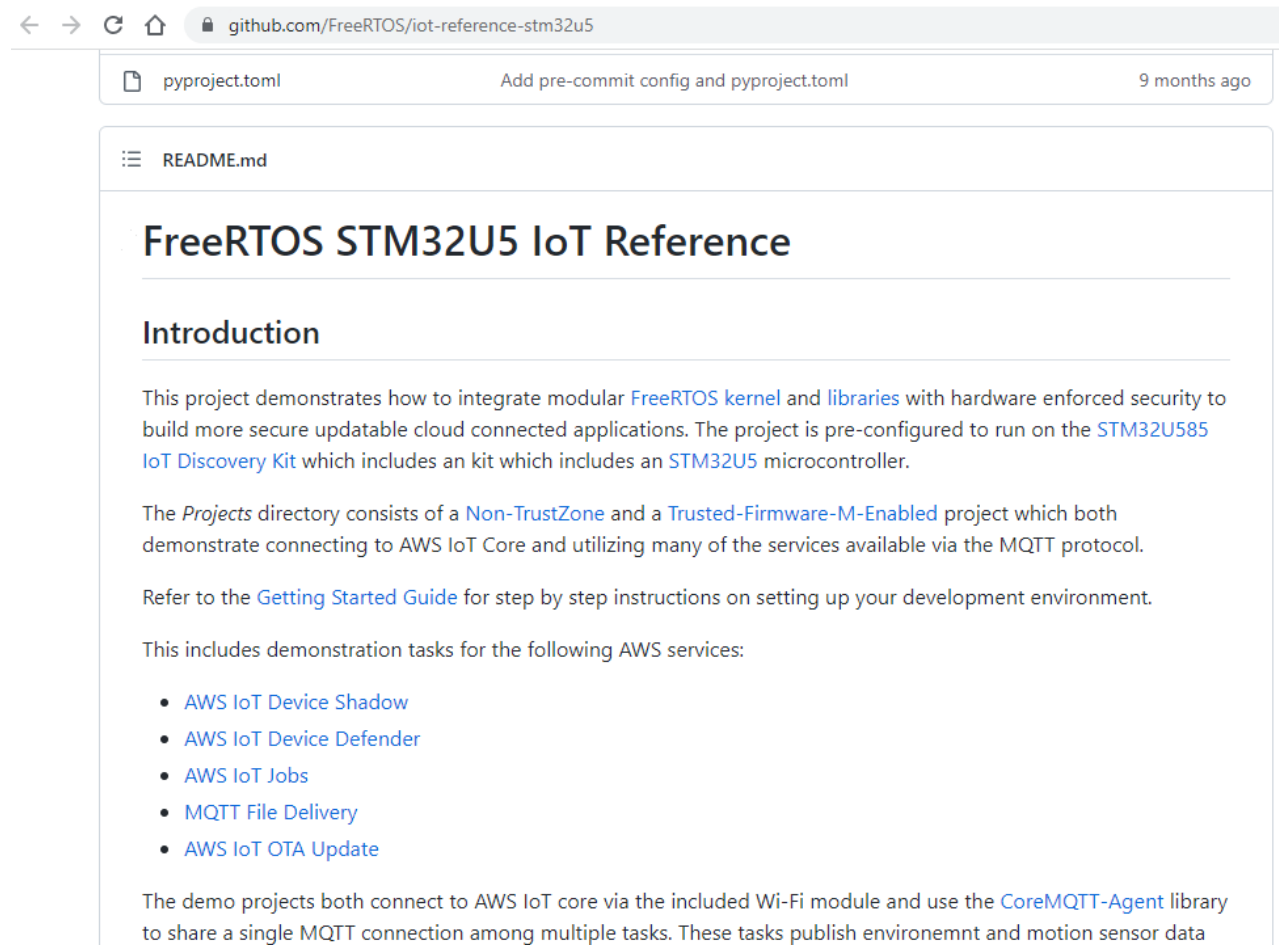
The STM32U5 is an Arm® Cortex®-M33 MCU and includes [Arm TrustZone technology](#) to help protect critical security code and data with hardware-enforced isolation built into the CPU. There are two projects, one without and one with TrustZone enabled. The MCU also provides built-in security functions, some of which are used in this demo such as secure boot, secure storage, and a [True Random Number Generator](#) (TRNG). The STM32U5 has been independently certified to [PSA Level 3](#) and [SESIP Level 3](#).



FreeRTOS STM32U5 GitHub repository

- <https://github.com/FreeRTOS/iot-reference-stm32u5>

```
git clone https://github.com/FreeRTOS/iot-reference-stm32u5.git --recurse-submodules
```



The screenshot shows the GitHub repository page for `FreeRTOS/iot-reference-stm32u5`. The browser address bar displays the repository URL. Below the header, a commit message "Add pre-commit config and pyproject.toml" is visible, dated "9 months ago". The file explorer shows `pyproject.toml` and `README.md`. The `README.md` content includes the title "FreeRTOS STM32U5 IoT Reference", an "Introduction" section, and a description of the project's purpose: integrating FreeRTOS with hardware-enforced security for secure IoT applications. It mentions the use of the STM32U585 IoT Discovery Kit and the STM32U5 microcontroller. The README also describes the project structure, including a `Projects` directory with `Non-TrustZone` and `Trusted-Firmware-M-Enabled` projects, and provides links to a `Getting Started Guide` and a list of AWS services demonstrated (AWS IoT Device Shadow, AWS IoT Device Defender, AWS IoT Jobs, MQTT File Delivery, and AWS IoT OTA Update). The final paragraph states that the demo projects connect to AWS IoT core via Wi-Fi and use the `CoreMQTT-Agent` library to share a single MQTT connection.

github.com/FreeRTOS/iot-reference-stm32u5

pyproject.toml Add pre-commit config and pyproject.toml 9 months ago

README.md

FreeRTOS STM32U5 IoT Reference

Introduction

This project demonstrates how to integrate modular [FreeRTOS kernel](#) and [libraries](#) with hardware enforced security to build more secure updatable cloud connected applications. The project is pre-configured to run on the [STM32U585 IoT Discovery Kit](#) which includes an kit which includes an [STM32U5](#) microcontroller.

The *Projects* directory consists of a [Non-TrustZone](#) and a [Trusted-Firmware-M-Enabled](#) project which both demonstrate connecting to AWS IoT Core and utilizing many of the services available via the MQTT protocol.

Refer to the [Getting Started Guide](#) for step by step instructions on setting up your development environment.

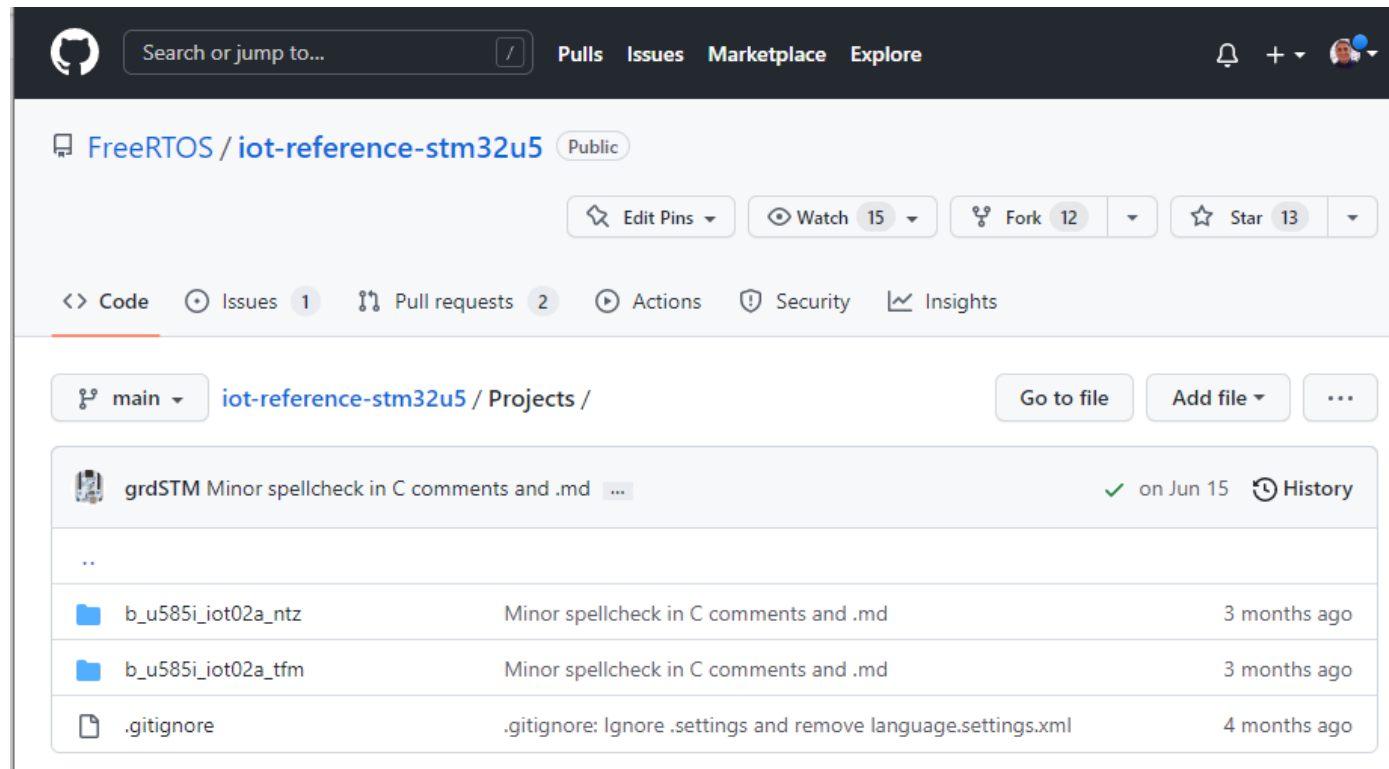
This includes demonstration tasks for the following AWS services:

- [AWS IoT Device Shadow](#)
- [AWS IoT Device Defender](#)
- [AWS IoT Jobs](#)
- [MQTT File Delivery](#)
- [AWS IoT OTA Update](#)

The demo projects both connect to AWS IoT core via the included Wi-Fi module and use the [CoreMQTT-Agent](#) library to share a single MQTT connection among multiple tasks. These tasks publish environemnt and motion sensor data

Projects

- Two projects
 - b_u585i_iot02a_ntz (For POC only)
 - b_u585i_iot02a_tfm (For production)

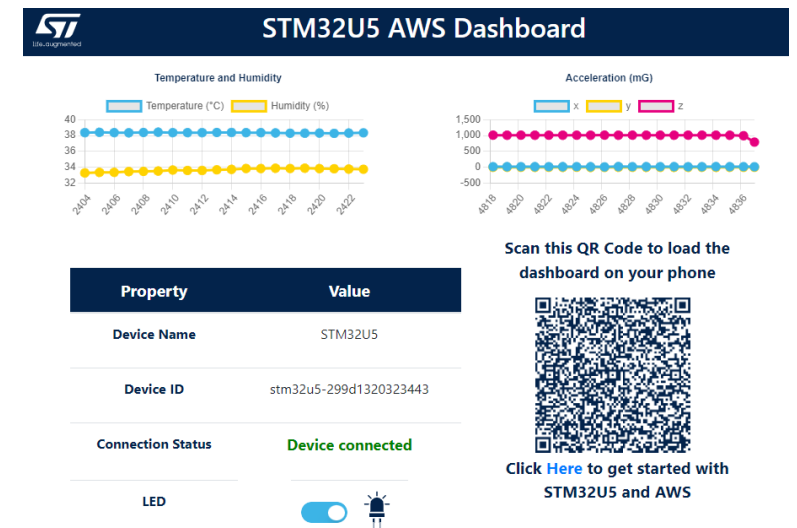


STM32U5 AWS Quick Connect

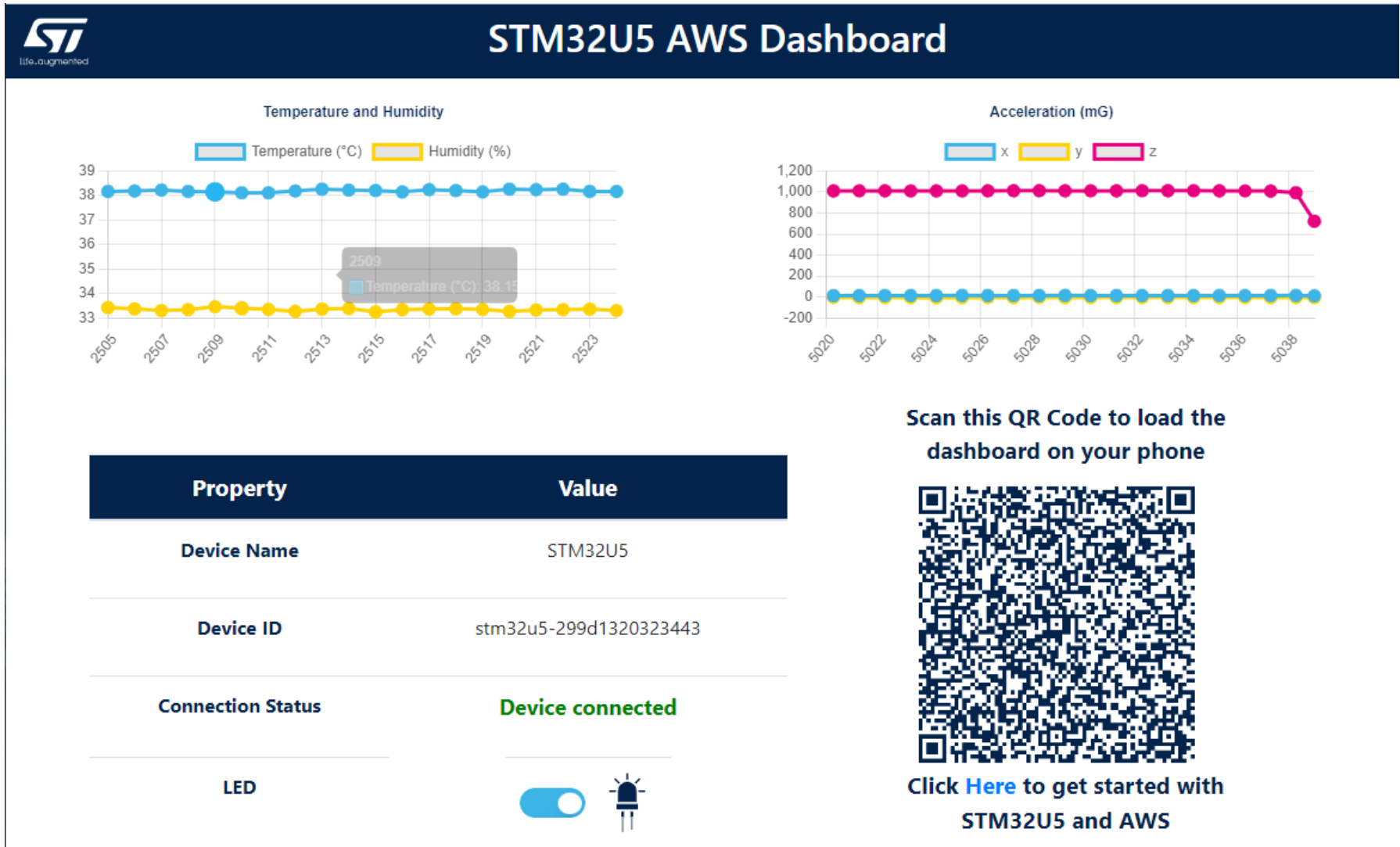
AWS Quick Connect

Abstracts Firmware customization and registration process

- Allows Cloud IoT/Data architects to focus on developing power of the Cloud IoT platform proof of concepts.
- Connect to AWS IoT and perform telemetry in minutes
- Solution Components:
 - *B-U585I-IOT02A Discovery Kit*
 - *Reference Binary*
 - *Quick connect scripts*
 - *Cloud visualization*



STM32U AWS Dashboard



Scan this QR Code to load the dashboard on your phone



Click [Here](#) to get started with STM32U5 and AWS

Boards distribution

Boards distribution

- Believe it or Not, even at ST we have hard time to get boards. Unfortunately, we need to collect the board at the end of the workshop to use them in the next one.
- We'll disinfect the boards at the end of the workshop and before re-distribution.
- A voucher is handed at the end of the workshop



Lab 1

Lab 1: System preparation

- In this lab we'll make sure that all the tools are properly installed and that your PC is ready to run the STM32U5_AWS_QuickConnect script

System Check


- Clone the workshop repo:

```
git clone https://github.com/SlimJallouli/ST_DevCon_2022_AWS_Workshop.git --recurse-submodules
```

- Open a command window (example PowerShell or bash)

- Run:

- `aws --version`
- `python --version`

 Windows PowerShell

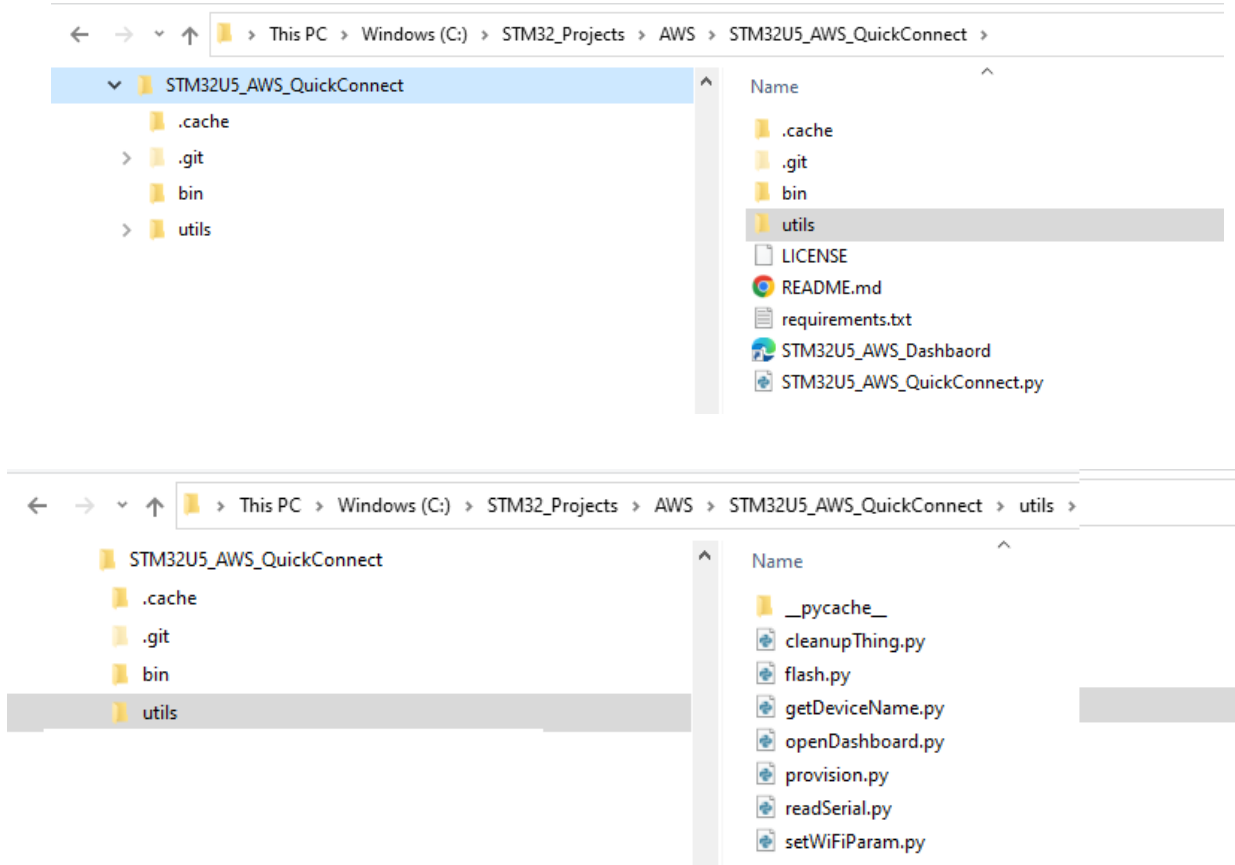
```
PS C:\STM32_Projects\AWS\STM32U5_AWS_QuickConnect> aws --version
aws-cli/2.0.53 Python/3.7.7 windows/10 exe/AMD64
PS C:\STM32_Projects\AWS\STM32U5_AWS_QuickConnect> python --version
Python 3.10.4
PS C:\STM32_Projects\AWS\STM32U5_AWS_QuickConnect> _
```

- Navigate to the `STM32U5_AWS_QuickConnect` directory
- Run: `pip install -r requirements.txt`

AWS CLI profiles

- Navigate to the `STM32_AWS_QuickConnect` directory
- Open a command window (example PowerShell or bash)
 - For Windows users double click on `AWS_CLI_ProfileConfig.bat`
 - For Linux and MAC users run `AWS_CLI_ProfileConfig.sh`
- The scripts will add two AWS CLI profiles called `provision` and `dashboard`.
- The first profile is used to provision your board with AWS IoT core
- The second profile is used to open the STM32U5 AWS Dashboard

STM32U5 AWS QuickConnect



pip install -r requirements.txt

Link to your device dashboard

STM32U5 QuickConnect script

Utils:

- Flash the binary
- Generate a device name
- Change Wi-Fi ssid and password
- Provision the board
- Open dashboard and create shortcut
- Read and print the serial port

Lab 2

Lab 2: Connect the board to AWS IoT Core

- In this lab we'll use the STM32_AWS_QuickConnect to connect your board AWS IoT Core and open a dashboard to visualize the sensor data and control the LED.

Connect your board

- Connect your board to the PC

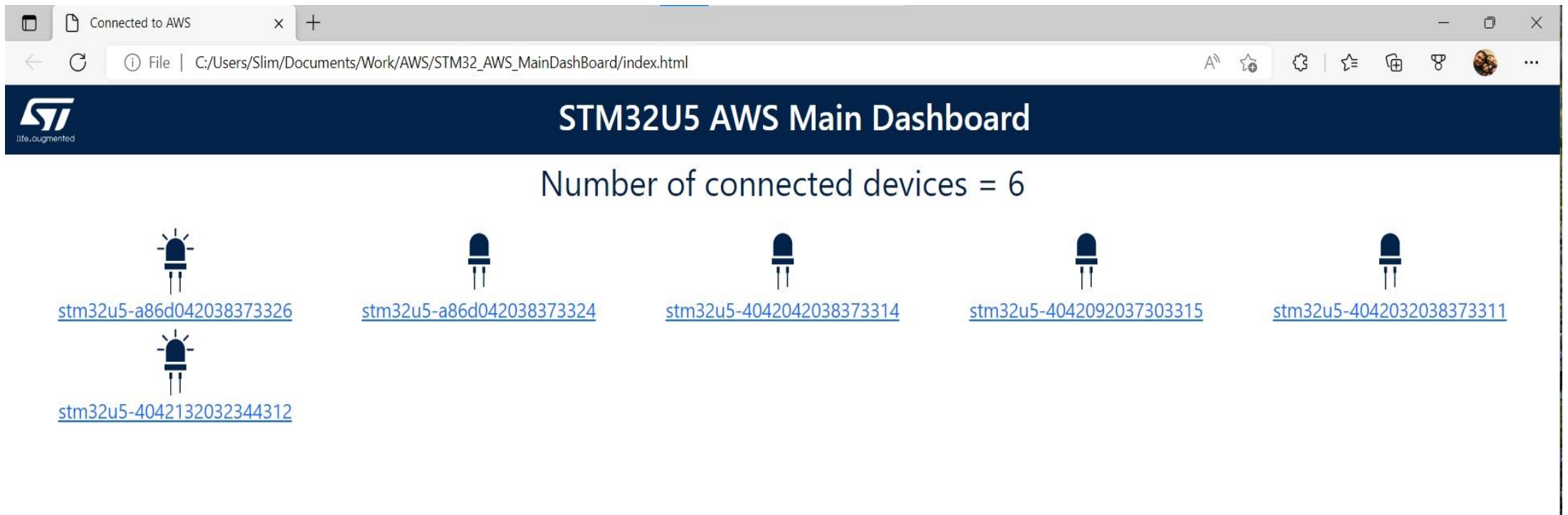


Run the quick connect script

- Navigate to `STM32U5_AWS_QuickConnect` directory
- Open a PowerShell console
- Type `python .\STM32U5_AWS_QuickConnect.py -i`
- Accept all the default settings
- The script will:
 - Flash your board with the binary
 - Provision your board with AWS IoT Core
 - Set the Wi-Fi SSID and password
 - Create a shortcut link to the dashboard specifically for your board
 - Open the dashboard for your board

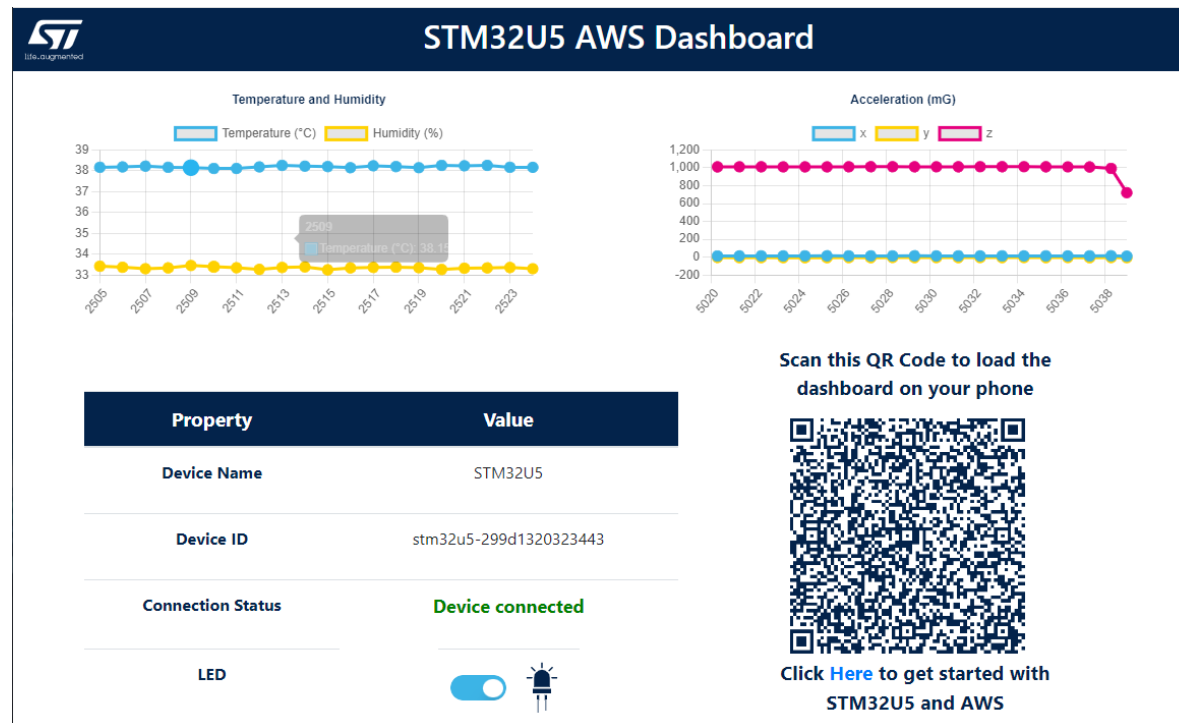
Main Dashboard

- I'll have a special dashboard showing the number of connected devices, the device ID and the corresponding LED status in real time as your boards get connected.



STM32U5 AWS Dashboard

- Scan the QR code with your phone camera
- Move the board to see the sensor data changing
- Use the toggle button to toggle the LED On/Off



Scan this QR Code to load the dashboard on your phone



Click [Here](#) to get started with STM32U5 and AWS

Lab 3

Lab 3: FreeRTOS-Plus-CLI

- In this lab we'll use the **FreeRTOS-Plus-CLI** to check and change the board configuration and check the application status

FreeRTOS-Plus-CLI

https://www.freertos.org/FreeRTOS-Plus/FreeRTOS_Plus_CLI/FreeRTOS_Plus_Command_Line_Interface.html

The screenshot shows the FreeRTOS-Plus-CLI webpage. The navigation menu includes KERNEL, LIBRARIES (selected), SUPPORT, PARTNERS, and COMMUNITY. A green button labeled 'Download FreeRTOS' is visible. The left sidebar contains links for Home, Getting started, All libraries, and a dropdown menu for FreeRTOS-Plus-CLI with sub-links for Introduction, Documentation, and Demos. The main content area features the title 'FreeRTOS-Plus-CLI' and the subtitle 'An Extensible Command Line Interface Framework'. Below this is the 'Introduction' section, which describes the framework's purpose and includes a four-step process diagram for adding a command.

LIBRARIES

- [Home](#)
- [Getting started](#)
- [All libraries](#)
- FreeRTOS-Plus-CLI
 - [Introduction](#)
 - [Documentation](#)
 - [Demos](#)

WHAT'S NEW

FreeRTOS-Plus-TCP v3.0.0 released:
We've added comprehensive unit tests and penetration and protocol testing. See the [blog post](#).

Featured FreeRTOS IoT Integrations:
Introducing three featured integrations for more secure IoT applications.

FreeRTOS-Plus-CLI

An Extensible Command Line Interface Framework

Introduction

FreeRTOS-Plus-CLI (Command Line Interface) provides a simple, small, extensible and RAM efficient method of enabling your FreeRTOS application to process command line input. The steps required to add a command are shown in the clickable diagram below - **click each stage in the process individually** to be taken to a worked example.

```
graph TD; A[Provide a function that implements the command behaviour] --> B[Map the command to the function that implements its behaviour]; B --> C[Register the command with FreeRTOS+CLI]; C --> D[Run the command interpreter];
```

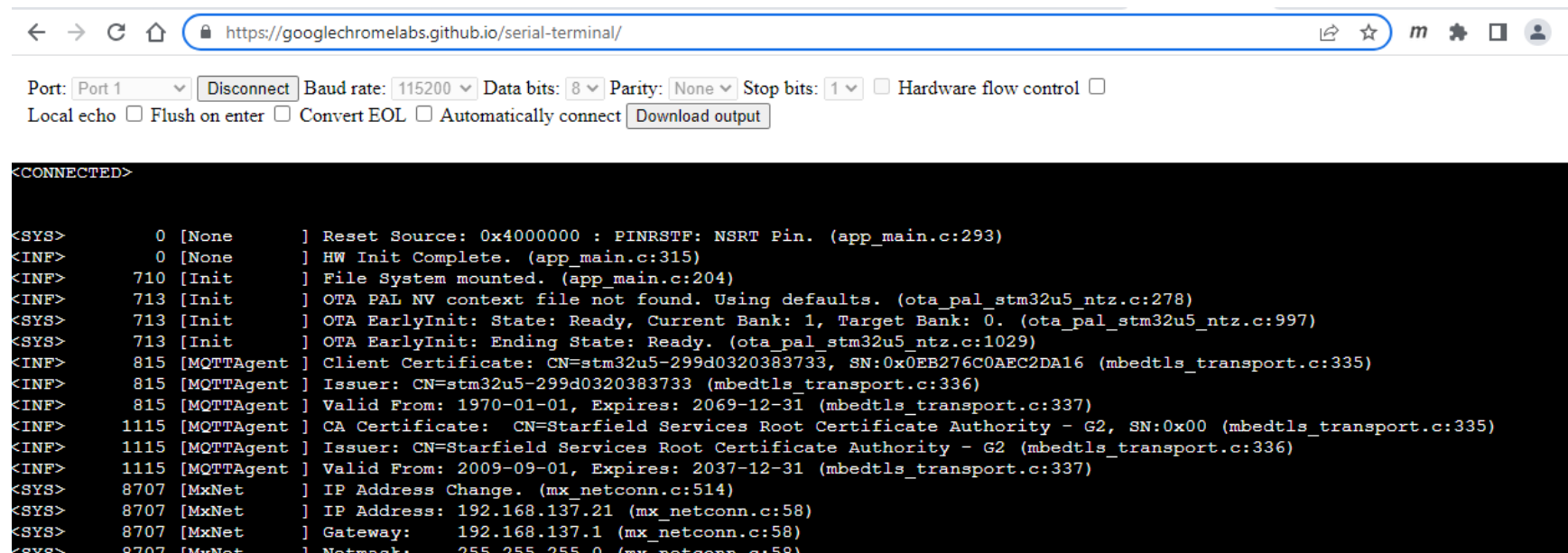
Adding a command to FreeRTOS-Plus-CLI. This diagram is clickable.

Connect to the board over serial port

- Close the quick connect script window
- You can use a serial terminal like TeraTerm or this web based serial terminal

<https://googlechromelabs.github.io/serial-terminal/>

- Connect to the board (8-bits, 1-stop, 115200)



The screenshot shows a web browser window with the URL <https://googlechromelabs.github.io/serial-terminal/>. The interface includes a control bar with the following settings: Port: Port 1, Baud rate: 115200, Data bits: 8, Parity: None, Stop bits: 1, Hardware flow control: unchecked. There are also checkboxes for Local echo, Flush on enter, Convert EOL, and Automatically connect, along with a 'Disconnect' button and a 'Download output' button.

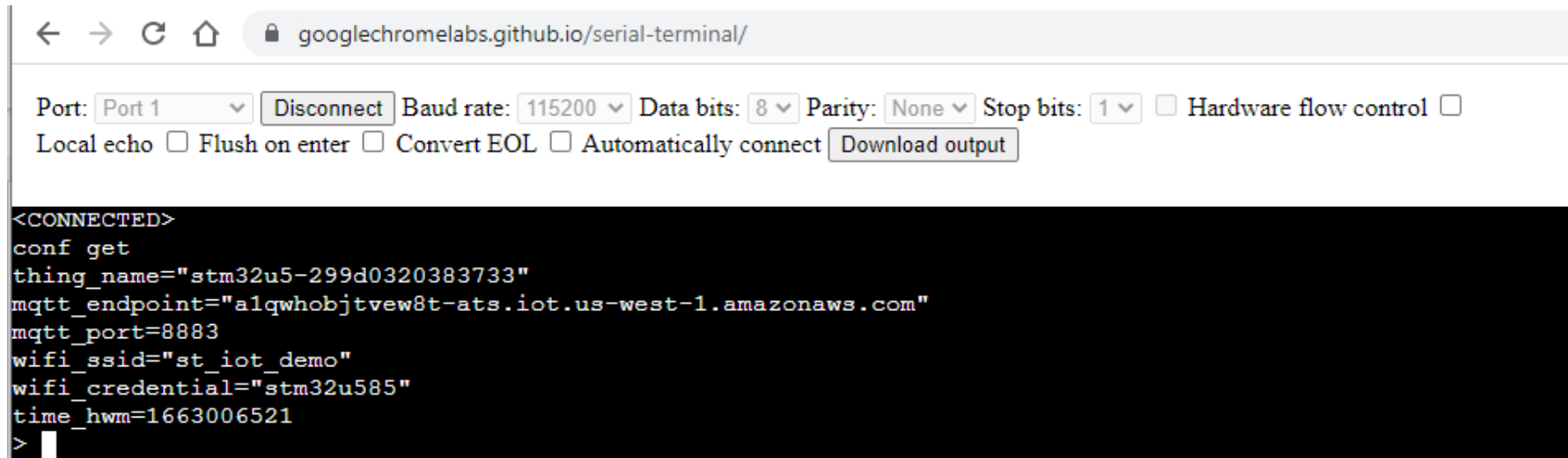
The terminal window displays the following log output:

```
<CONNECTED>

<SYS>      0 [None      ] Reset Source: 0x4000000 : PINRSTF: NSRT Pin. (app_main.c:293)
<INF>      0 [None      ] HW Init Complete. (app_main.c:315)
<INF>     710 [Init       ] File System mounted. (app_main.c:204)
<INF>     713 [Init       ] OTA PAL NV context file not found. Using defaults. (ota_pal_stm32u5_ntz.c:278)
<SYS>     713 [Init       ] OTA EarlyInit: State: Ready, Current Bank: 1, Target Bank: 0. (ota_pal_stm32u5_ntz.c:997)
<SYS>     713 [Init       ] OTA EarlyInit: Ending State: Ready. (ota_pal_stm32u5_ntz.c:1029)
<INF>     815 [MQTTAgent ] Client Certificate: CN=stm32u5-299d0320383733, SN:0x0EB276C0AEC2DA16 (mbedtls_transport.c:335)
<INF>     815 [MQTTAgent ] Issuer: CN=stm32u5-299d0320383733 (mbedtls_transport.c:336)
<INF>     815 [MQTTAgent ] Valid From: 1970-01-01, Expires: 2069-12-31 (mbedtls_transport.c:337)
<INF>    1115 [MQTTAgent ] CA Certificate: CN=Starfield Services Root Certificate Authority - G2, SN:0x00 (mbedtls_transport.c:335)
<INF>    1115 [MQTTAgent ] Issuer: CN=Starfield Services Root Certificate Authority - G2 (mbedtls_transport.c:336)
<INF>    1115 [MQTTAgent ] Valid From: 2009-09-01, Expires: 2037-12-31 (mbedtls_transport.c:337)
<SYS>     8707 [MxNet       ] IP Address Change. (mx_netconn.c:514)
<SYS>     8707 [MxNet       ] IP Address: 192.168.137.21 (mx_netconn.c:58)
<SYS>     8707 [MxNet       ] Gateway: 192.168.137.1 (mx_netconn.c:58)
<SYS>     8707 [MxNet       ] Netmask: 255.255.255.0 (mx_netconn.c:58)
```

FreeRTOS CLI: Check your board configuration

- On the terminal type `conf get`



The screenshot shows a web browser window with the address bar displaying `googlechromelabs.github.io/serial-terminal/`. Below the address bar, there are configuration controls for a serial terminal, including a port selector (set to 'Port 1'), a 'Disconnect' button, a baud rate selector (set to '115200'), data bits (set to '8'), parity (set to 'None'), stop bits (set to '1'), and a checkbox for 'Hardware flow control'. There are also checkboxes for 'Local echo', 'Flush on enter', 'Convert EOL', and 'Automatically connect', along with a 'Download output' button. The terminal window itself is black with white text, showing the output of the `conf get` command. The output is as follows:

```
<CONNECTED>
conf get
thing_name="stm32u5-299d0320383733"
mqtt_endpoint="alqwhobjtvew8t-ats.iot.us-west-1.amazonaws.com"
mqtt_port=8883
wifi_ssid="st_iot_demo"
wifi_credential="stm32u585"
time_hwm=1663006521
>
```

- Type `help` for help menu

FreeRTOS CLI: Change your board Wi-Fi settings

- Two ways to change the board Wi-Fi settings

1- You can use the terminal and type the following commands

```
> conf set wifi_ssid myssid
wifi_ssid="myssid"
> conf set wifi_credential mypasswd
wifi_credential="mypasswd"
> conf commit
Configuration saved to NVM.
> reset
```

2- Use the STM32_AWS_QuickConnect\utils\setWiFiParams.py

```
python setWiFiParams.py --ssid=mysid --password=mypassword
```

Our technology starts with You



Find out more at www.st.com

© STMicroelectronics - All rights reserved.

ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.

For additional information about ST trademarks, please refer to www.st.com/trademarks.

All other product or service names are the property of their respective owners.



life.augmented