



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, and several blue Wi-Fi signal icons, representing an IoT network.

# STM32U5 AWS IoT Hands-on

Slim JALLOULI

October 2022

# Agenda

1 Workshop prerequisites

2 STM32U5 Introduction

3 FreeRTOS STM32U5 Reference Integration

4 STM32U5 AWS QuickConnect

5 Board distribution

6 Hands-on

7 Help and Support

# 1- Workshop prerequisites

# Clone the workshop repo

- You can access the repo using this url: <https://tinyurl.com/stm32u5aws>

- Clone the workshop repo:

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

- Git can be downloaded from <https://git-scm.com/downloads>

# Workshop Directory

This PC > Documents > ST_DevCon_2022_AWS_Workshop >			
Name	Status	D	
.git		9,	
firmware		9,	
Presentation		9,	
STM32U5_AWS_QuickConnect		9,	
.gitmodules		9,	
CODE_OF_CONDUCT.md		9,	
CONTRIBUTING.md		9,	
LICENSE		9,	
README.md		9,	
SECURITY.md		9,	

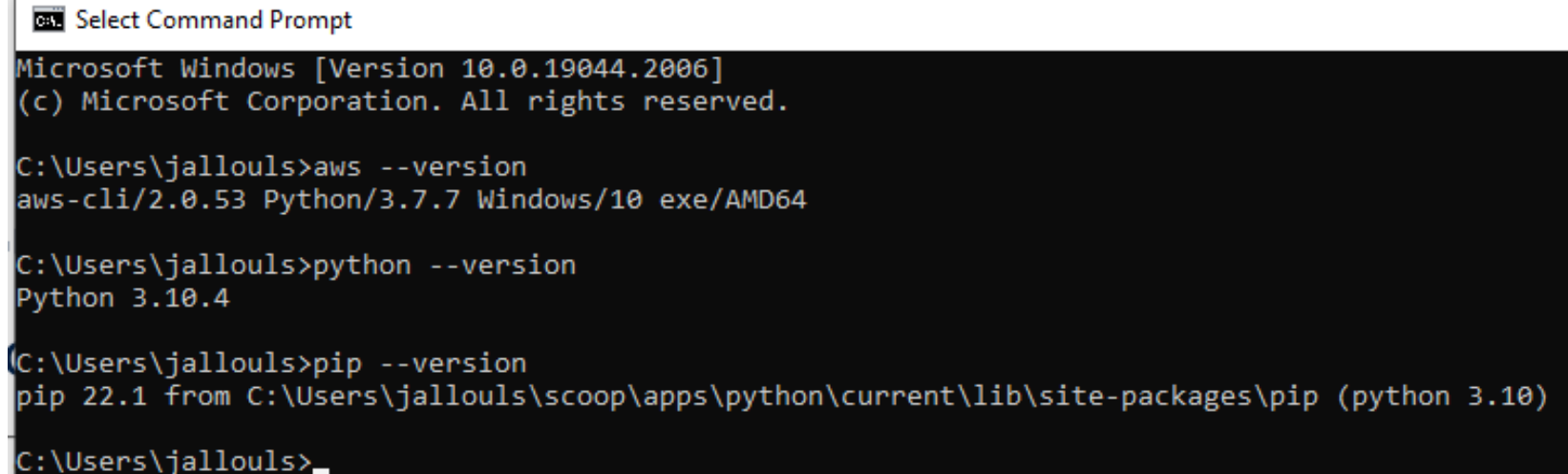
Binary image

The current presentation

STM32U5 Quick Connect script

# Install software

- Please make sure you have [AWS CLI](#) and latest Python installed
- Make sure that [Python](#) is installed with pip and added to path
- To verify, run the following commands in a command prompt
  - `aws --version`
  - `Python --version`
  - `pip --version`



```
CA: Select Command Prompt
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jallouls>aws --version
aws-cli/2.0.53 Python/3.7.7 Windows/10 exe/AMD64

C:\Users\jallouls>python --version
Python 3.10.4

C:\Users\jallouls>pip --version
pip 22.1 from C:\Users\jallouls\scoop\apps\python\current\lib\site-packages\pip (python 3.10)

C:\Users\jallouls>_
```

## 2- STM32U5 Introduction

# STM32U5 Microcontrollers

The new reference for secure and Cloud IoT applications



## Higher Security

Certified PSA L3 and SESIP L3

## Lower Power

58 ULPMark-CM

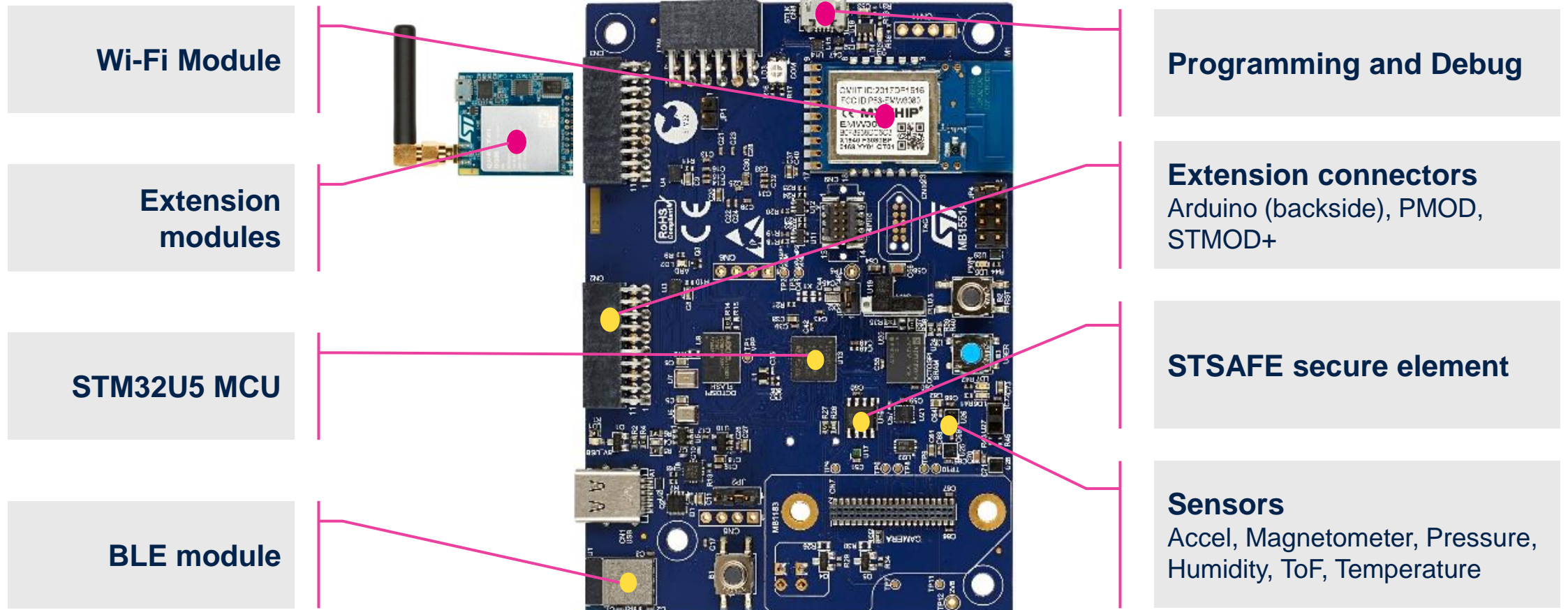
## Richer applications

Cortex-M33 @ 160MHz, extended features set



# STM32U5 IoT Kit

Your reference board for Cloud IoT Proof-of-Concepts



[B-U585I-IOT02A](#)

# AWS IoT on STM32U5

**X-CUBE-AWS reference integration simplifies your development**



**STM32**  
**CubeExpansion**  
**X-CUBE-AWS**

**AWS Certified**

**Leverages ARM Trusted Firmware-M (TF-M)**

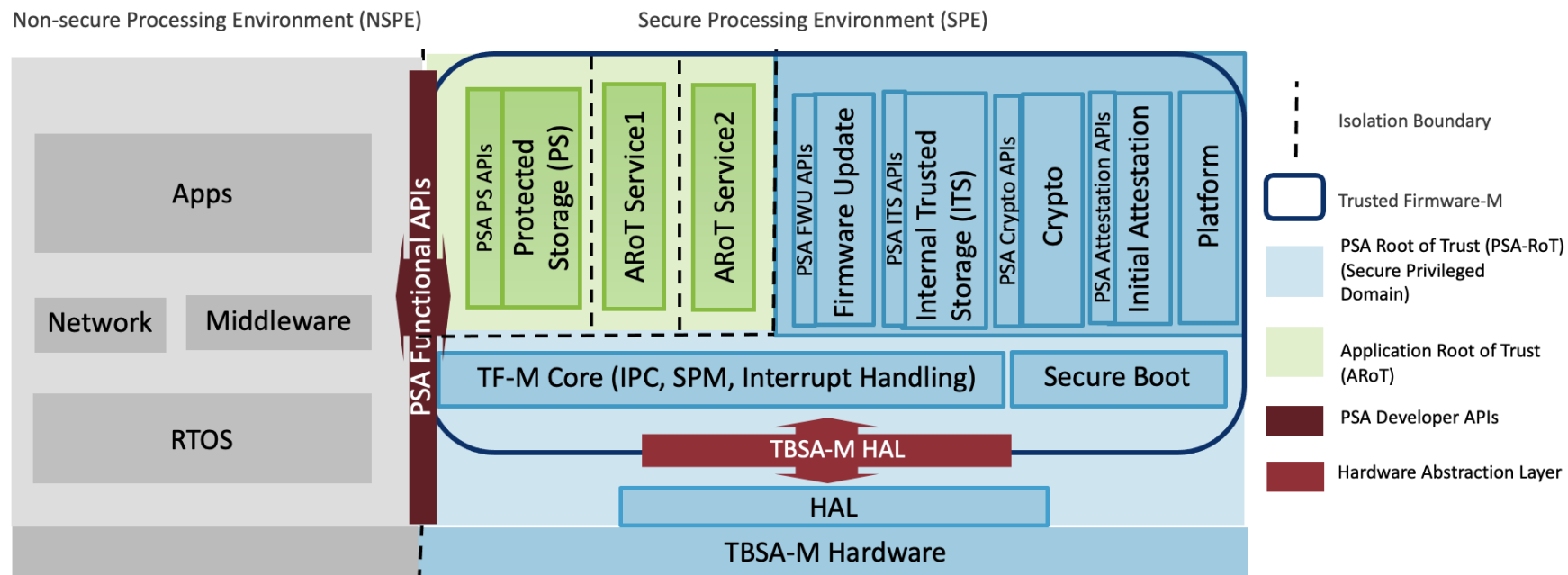
**Based on FreeRTOS LTS Library**

**Over The Air Update**

**AWS IoT Defender**

# Trusted Firmware-M (TF-M)

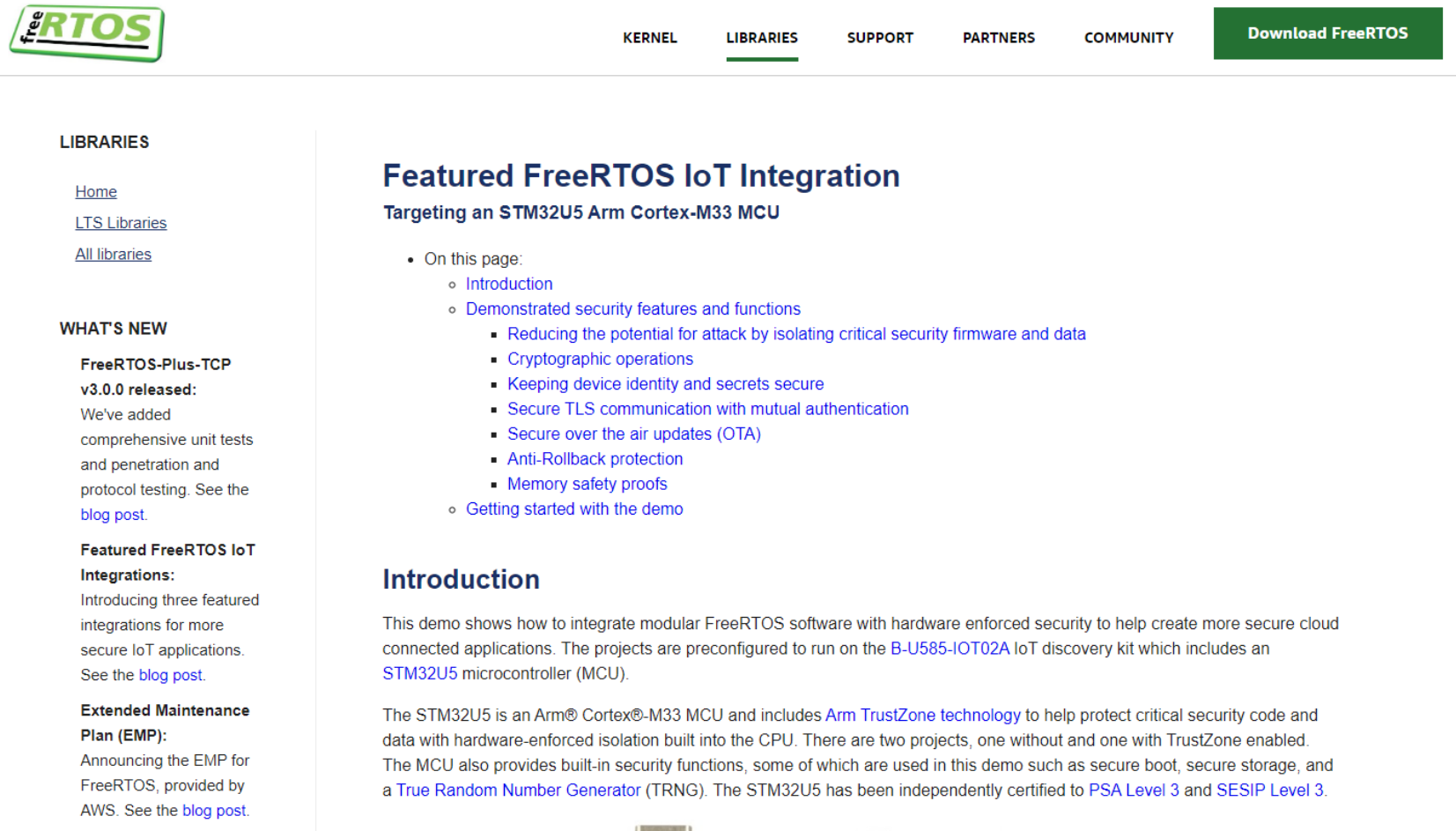
- Trusted Firmware for Cortex M (TF-M) implements the Secure Processing Environment (SPE) for Armv8-M, Armv8.1-M
- **Trusted Firmware-M consists of:**
  - Secure Boot, Isolation Control, communication and execution within SPE and with NSPE
  - Secure Services -Crypto, Internal Trusted Storage (ITS),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



## 3- STM32U5 FreeRTOS reference

# Featured FreeRTOS IoT Integration

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



The screenshot shows the FreeRTOS website with the 'LIBRARIES' tab selected. The main heading is 'Featured FreeRTOS IoT Integration' with the subtitle 'Targeting an STM32U5 Arm Cortex-M33 MCU'. A list of links is provided: 'On this page:' followed by 'Introduction', 'Demonstrated security features and functions' (which includes a bulleted list of security features), and 'Getting started with the demo'. The 'Introduction' section describes the demo's purpose and the hardware used (STM32U5 microcontroller).

**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**

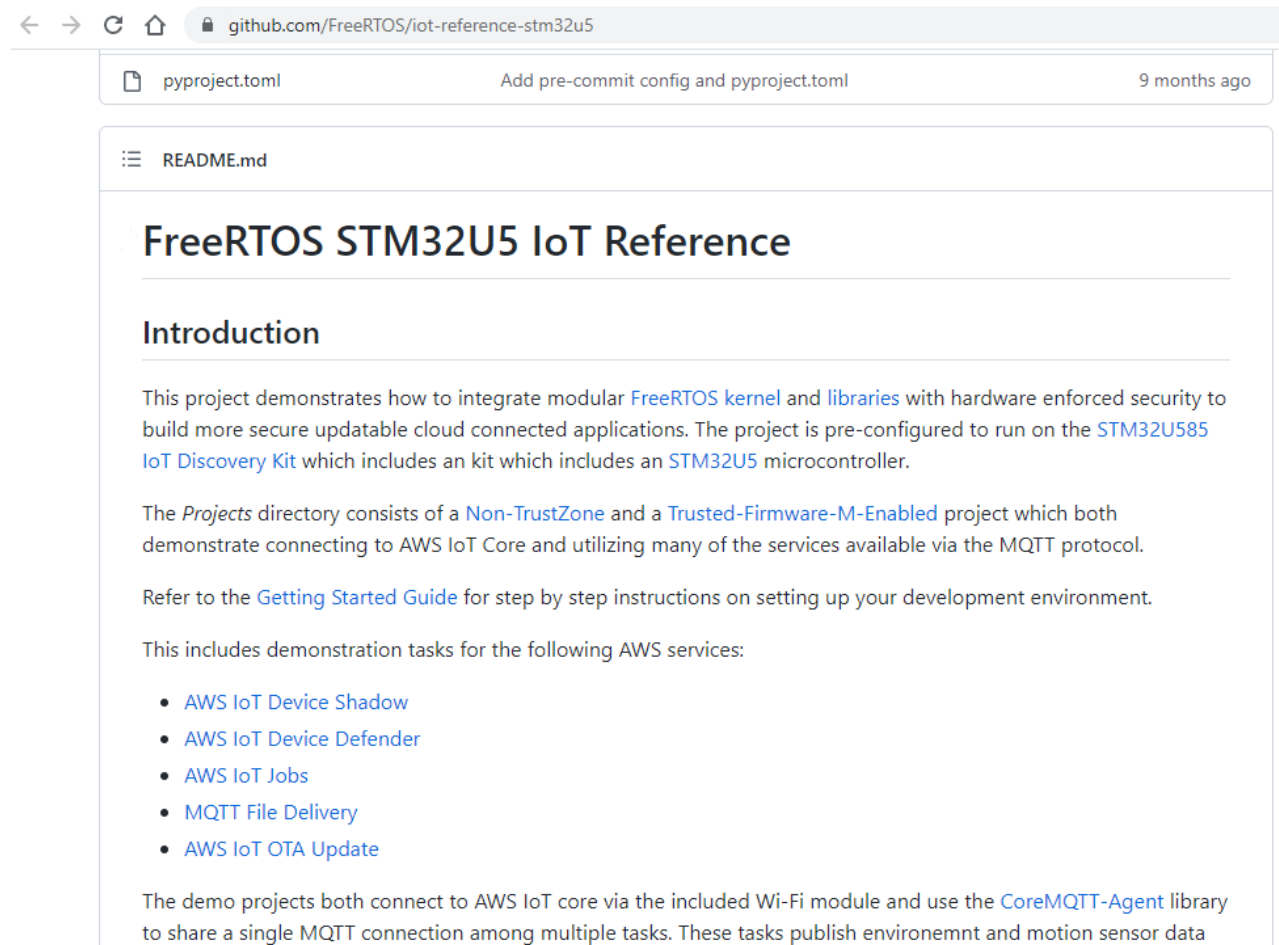
This demo shows how to integrate modular FreeRTOS software with hardware enforced security to help create more secure cloud connected applications. The projects are preconfigured to run on the [B-U585-IOT02A](#) IoT discovery kit which includes an [STM32U5](#) microcontroller (MCU).

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
```



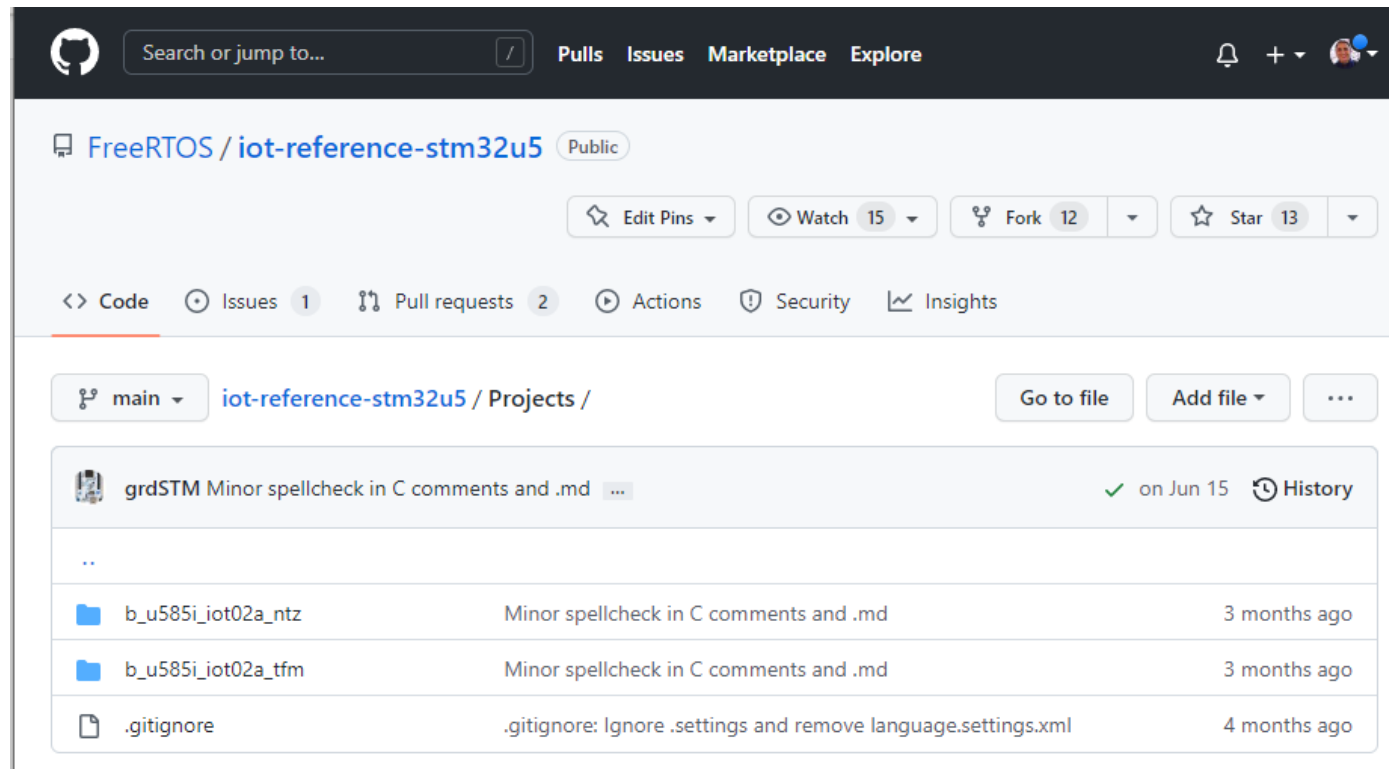
The screenshot shows the GitHub repository page for `FreeRTOS/iot-reference-stm32u5`. The browser address bar displays `github.com/FreeRTOS/iot-reference-stm32u5`. Below the address bar, a commit message `pyproject.toml Add pre-commit config and pyproject.toml` is shown, dated `9 months ago`. The repository's `README.md` file is displayed, featuring the title `FreeRTOS STM32U5 IoT Reference` and an `Introduction` section. The introduction text states: "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." It further explains that the `Projects` directory contains a `Non-TrustZone` and a `Trusted-Firmware-M-Enabled` project, both demonstrating connections to AWS IoT Core via MQTT. A reference to the `Getting Started Guide` is provided for setup instructions. A list of AWS services used in the demonstration is shown: 

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

The text concludes by stating that the demo projects connect to AWS IoT Core via a Wi-Fi module and use the `CoreMQTT-Agent` library to share a single MQTT connection among multiple tasks, publishing environment and motion sensor data.

# Projects

- Two projects
  - b\_u585i\_iot02a\_ntz (For POC only)
  - b\_u585i\_iot02a\_tfm (For production)



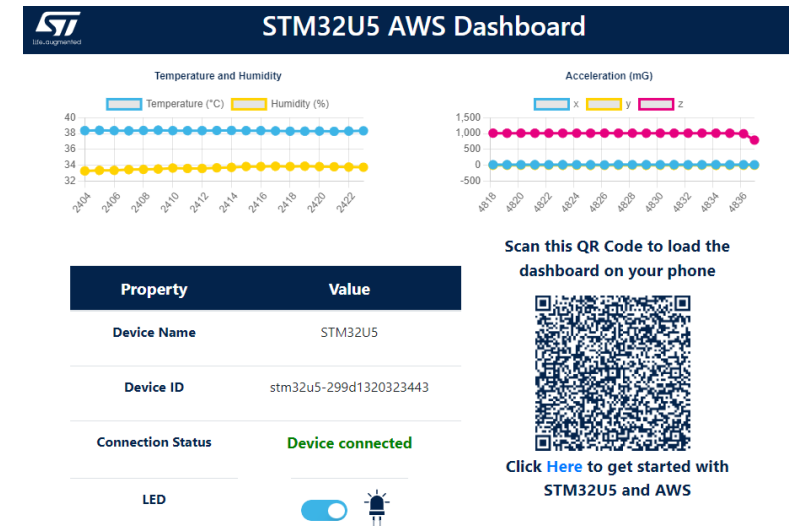
## 4- STM32U5 AWS Quick Connect



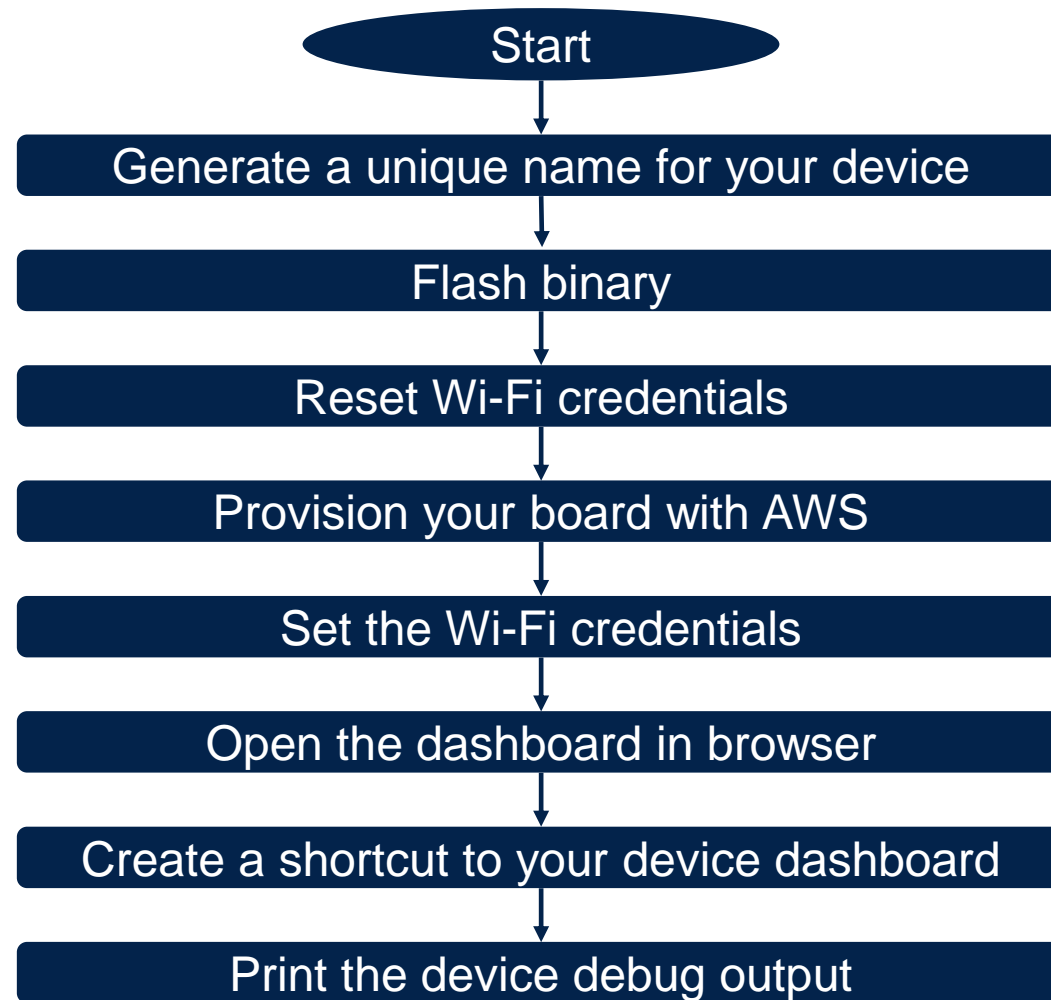
# AWS Quick Connect

## 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*



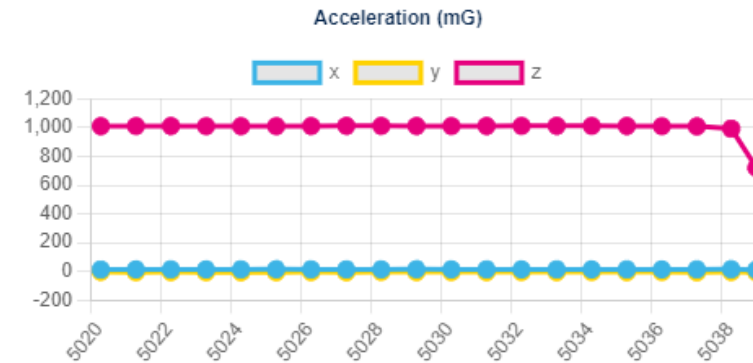
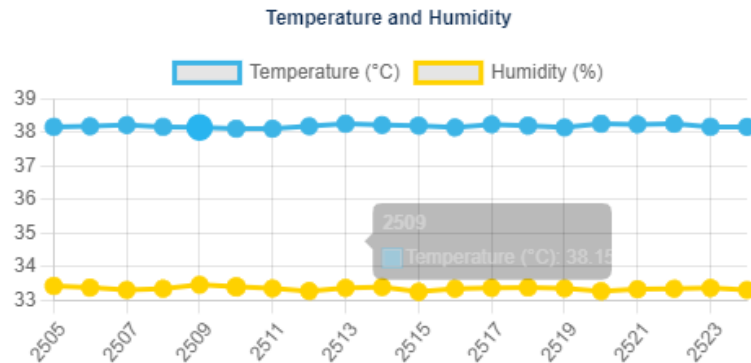
# AWS Quick Connect Flow



# STM32U AWS Dashboard



## STM32U5 AWS Dashboard



Property	Value
Device Name	STM32U5
Device ID	stm32u5-299d1320323443
Connection Status	Device connected
LED	<input checked="" type="checkbox"/>

Scan this QR Code to load the dashboard on your phone



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

## 5- Board Distribution

# Loaner Board Handout & Return

- **Please Take Note:** Due to supply limitations, ST will need to reuse the boards in this workshop for our near-team future workshops.
- **Board Logistics**
  - ST will hand out boards & USB cables
  - Please remove the boards from the packaging
    - The cardboard box can be put in the recycle bin
    - Please keep the antistatic bag for when the board is returned
  - Enjoy the workshop
  - Put the board and USB cable back into the anti-static bag
  - At the end of workshop, ST will provide a voucher to purchase the same board at a discounted price from a dedicated stock reserved specifically for workshop participants when we collect the loaner boards
- We appreciate your understanding and thank you for your cooperation!



## 6- Labs

# Lab 1: System preparation

# Lab 1: System preparation

- In this lab we'll make sure all the tools are properly installed and your PC is ready to run the STM32U5\_AWS\_QuickConnect script



# Clone the workshop repo

- You can access the repo using this url: <https://tinyurl.com/stm32u5aws>
- Clone the workshop repo:

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

# Workshop Directory

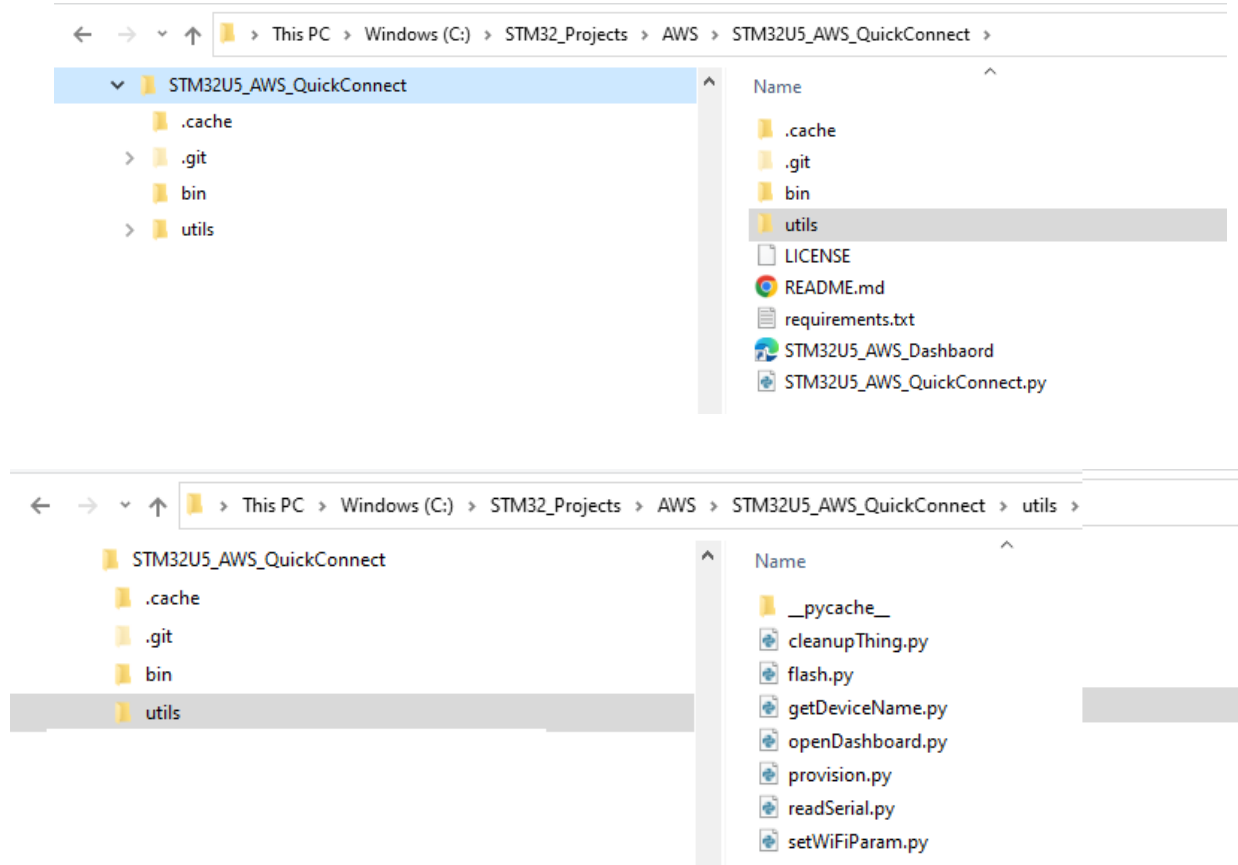
This PC > Documents > ST_DevCon_2022_AWS_Workshop >			
Name	Status	D	
.git		9,	
firmware		9,	
Presentation		9,	
STM32U5_AWS_QuickConnect		9,	
.gitmodules		9,	
CODE_OF_CONDUCT.md		9,	
CONTRIBUTING.md		9,	
LICENSE		9,	
README.md		9,	
SECURITY.md		9,	

Binary image

The current presentation

STM32U5 Quick Connect script

# STM32U5 AWS QuickConnect



`pip install -r requirements.txt`

STM32U5 QuickConnect script

Link to your device dashboard

## 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

# Install required python libraries

- Navigate to `ST_DevCon_2022_AWS_Workshop`
- 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 save your current AWS CLI `config` and `credentials` files and create new ones with 2 profiles (`default` and `dashboard`).
- The `default` profile is used to provision your board with AWS IoT core
- The `dashboard` profile is used to open the STM32U5 AWS Dashboard

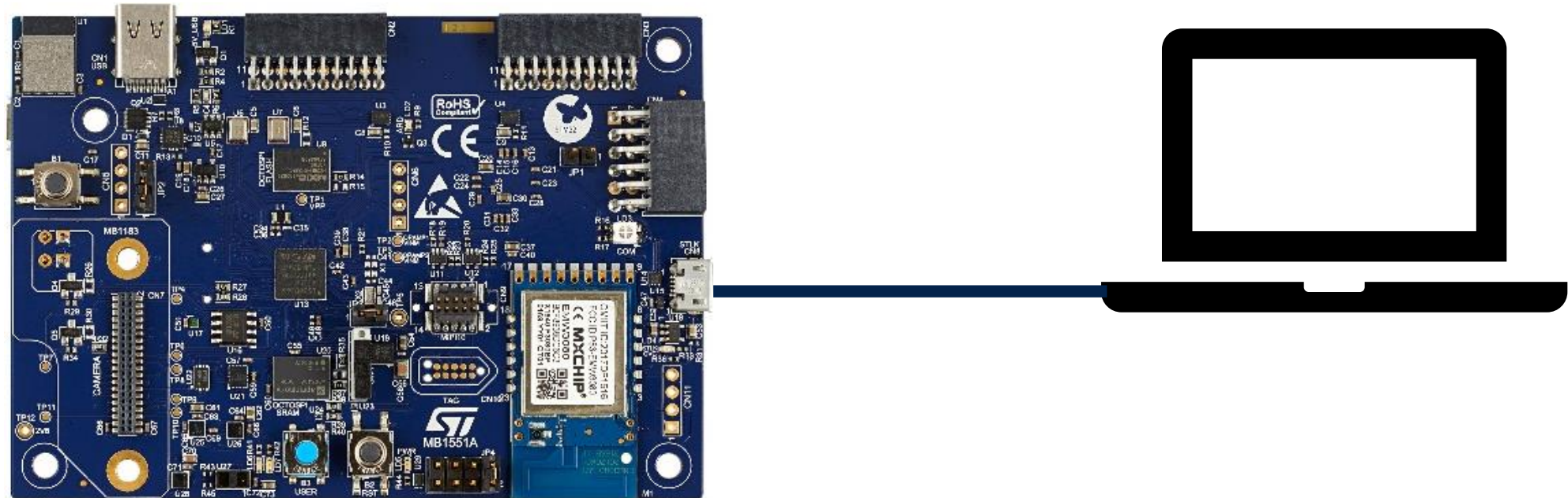
# Lab 2: Connect to AWS IoT Core

# Lab 2: Connect to AWS IoT Core

- In this lab we'll use the STM32\_AWS\_QuickConnect to connect your board to 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



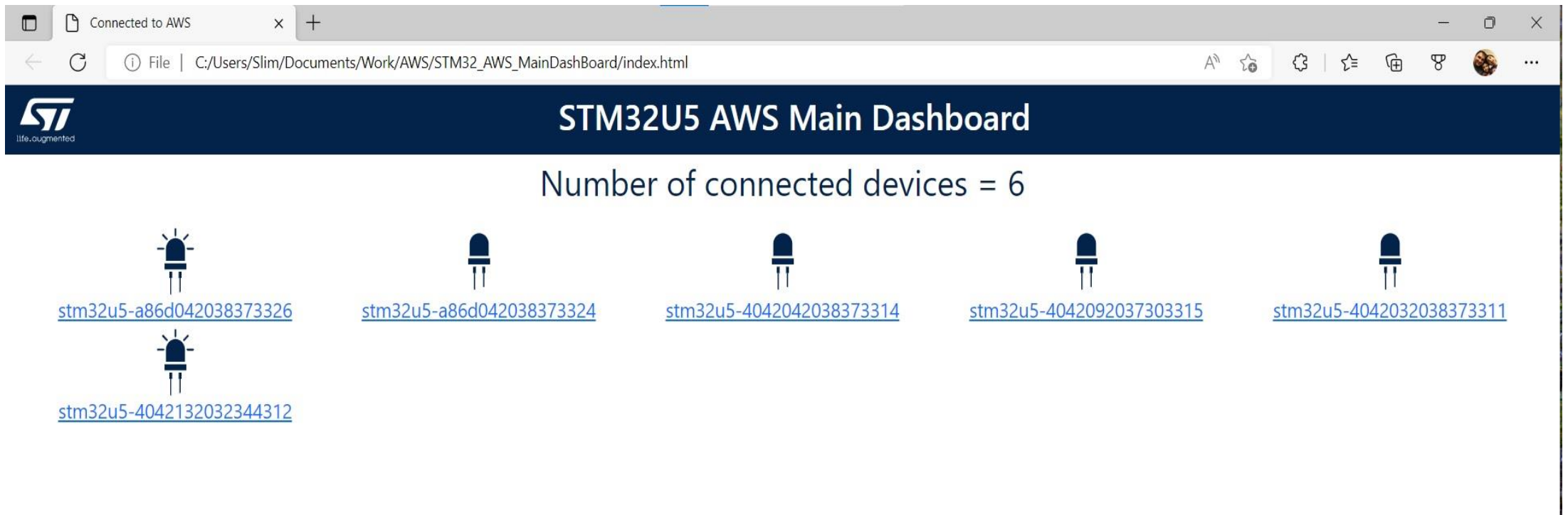


# Run the quick connect script

- 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

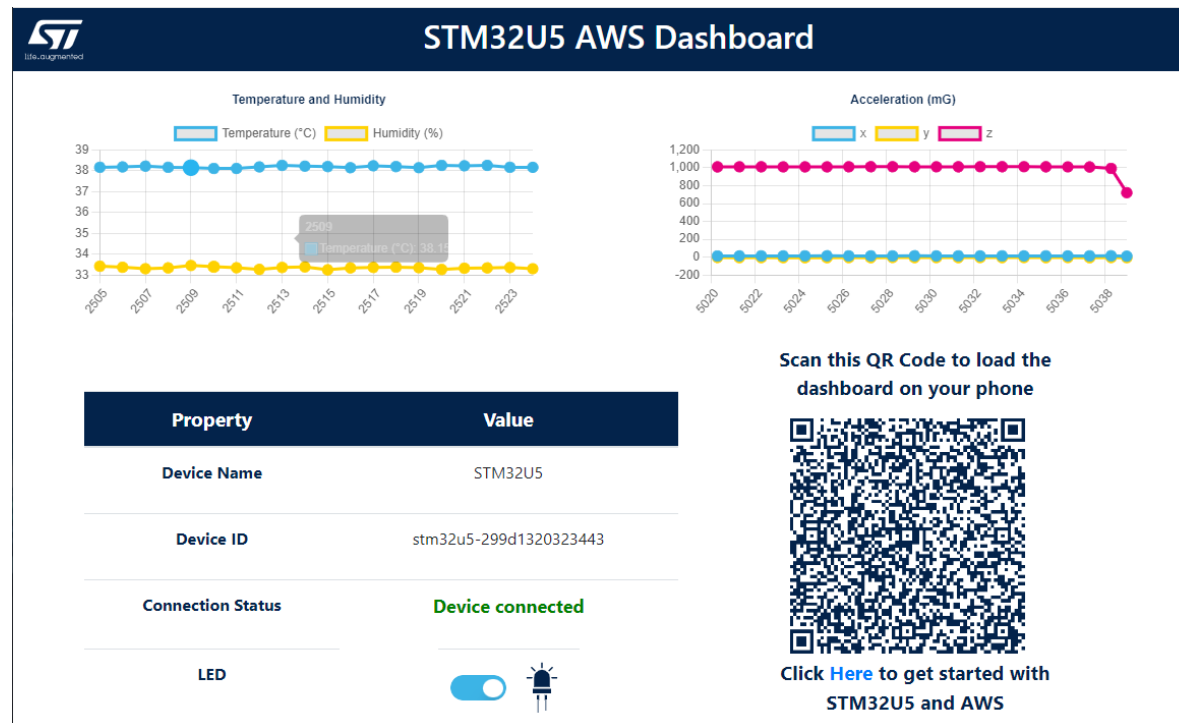
# Main Dashboard

- 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 turn the LED on your board On/Off



# Lab 3: FreeRTOS-Plus-CLI

# 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](https://www.freertos.org/FreeRTOS-Plus/FreeRTOS_Plus_CLI/FreeRTOS_Plus_Command_Line_Interface.html)

The screenshot shows the FreeRTOS-Plus-CLI website. 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 section for FreeRTOS-Plus-CLI with links to Introduction, Documentation, and Demos. The main content area is titled 'FreeRTOS-Plus-CLI' and 'An Extensible Command Line Interface Framework'. It includes an 'Introduction' section with a paragraph explaining the framework and 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];
```

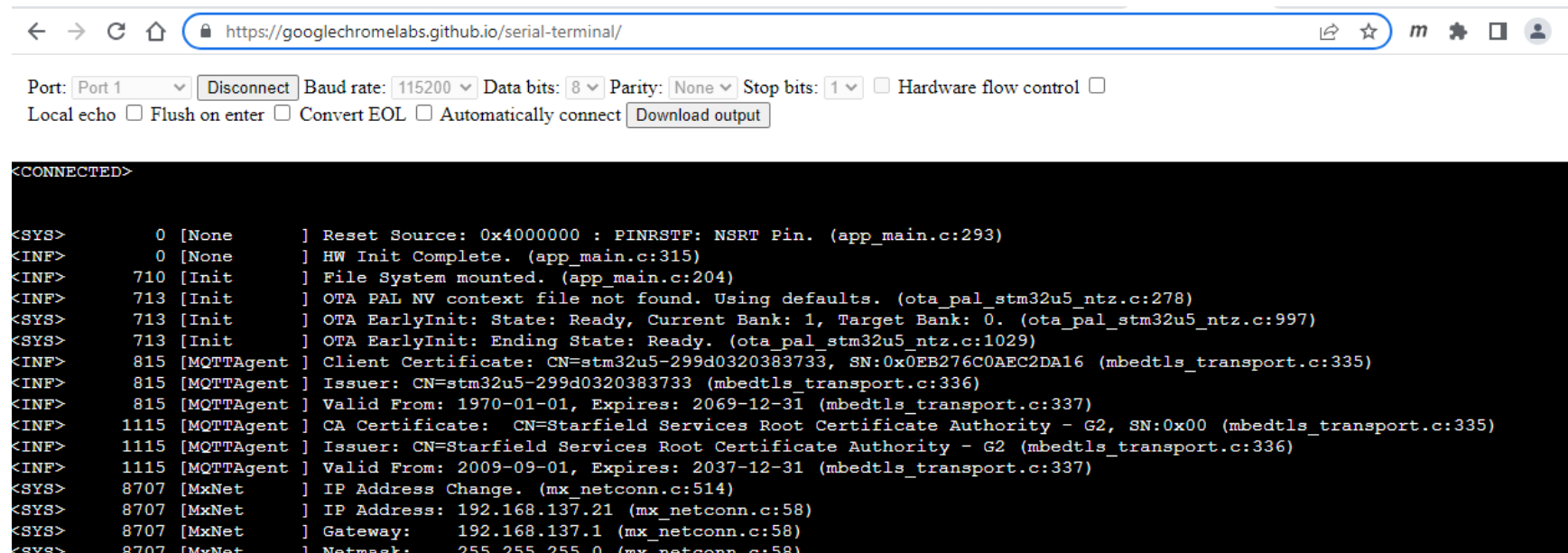
Adding a command to FreeRTOS-Plus-CLI. This diagram is clickable.

# Connect to the board over the 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)

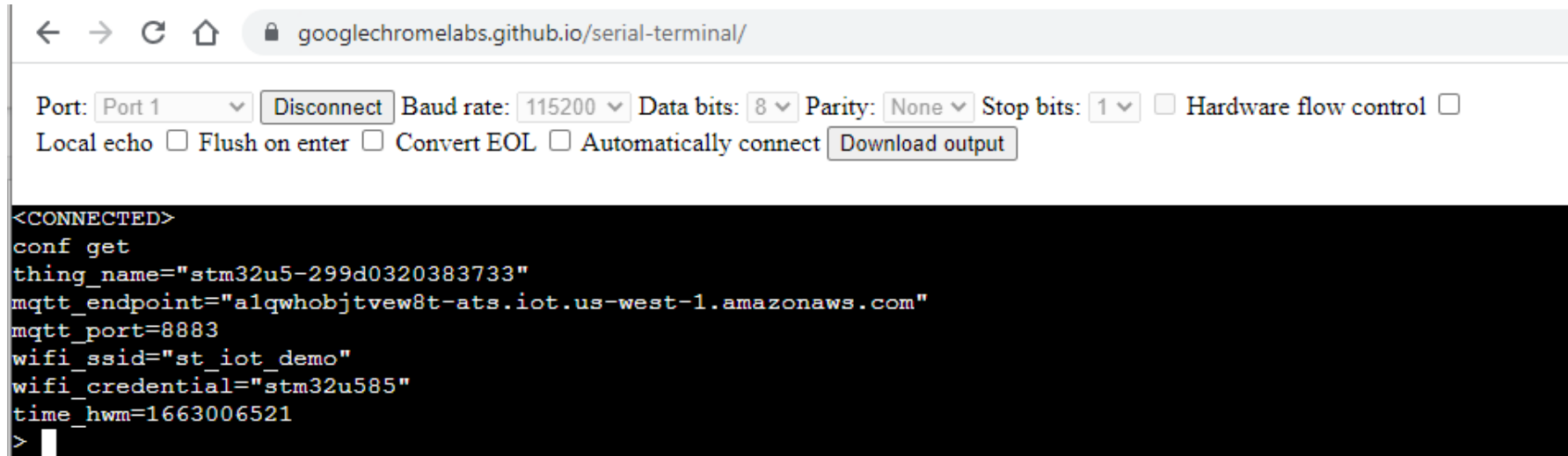


The screenshot shows a web browser window with the URL <https://googlechromelabs.github.io/serial-terminal/>. The interface includes a control bar with a 'Port' dropdown set to 'Port 1', a 'Disconnect' button, a 'Baud rate' dropdown set to '115200', 'Data bits' set to '8', 'Parity' set to 'None', 'Stop bits' set to '1', and checkboxes for 'Hardware flow control', 'Local echo', 'Flush on enter', 'Convert EOL', and 'Automatically connect'. A 'Download output' button is also present. The main terminal area 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: a dropdown menu for 'Port' (set to 'Port 1'), a 'Disconnect' button, a 'Baud rate' dropdown (set to '115200'), a 'Data bits' dropdown (set to '8'), a 'Parity' dropdown (set to 'None'), a 'Stop bits' dropdown (set to '1'), and a checkbox for 'Hardware flow control' (unchecked). Below these are checkboxes for 'Local echo' (unchecked), 'Flush on enter' (unchecked), 'Convert EOL' (unchecked), and 'Automatically connect' (unchecked), followed by a 'Download output' button. The main terminal area is a black rectangle with white text. It starts with `<CONNECTED>`, followed by the command `conf get` and its output: `thing_name="stm32u5-299d0320383733"`, `mqtt_endpoint="algwhobjtview8t-ats.iot.us-west-1.amazonaws.com"`, `mqtt_port=8883`, `wifi_ssid="st_iot_demo"`, `wifi_credential="stm32u585"`, and `time_hwm=1663006521`. The prompt `>` is visible at the bottom of the terminal area.

```
<CONNECTED>
conf get
thing_name="stm32u5-299d0320383733"
mqtt_endpoint="algwhobjtview8t-ats.iot.us-west-1.amazonaws.com"
mqtt_port=8883
wifi_ssid="st_iot_demo"
wifi_credential="stm32u585"
time_hwm=1663006521
>
```



# FreeRTOS CLI: Change your board Wi-Fi settings

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
```

# Use Help menu

- Type `help` for help menu to get a list of all the possible command

```
> help
help:
  List available commands and their arguments.
  Usage:

  help
    Print help for all recognized commands

  help <command>
    Print help test for a specific command

conf:
  Get/ Set/ Commit runtime configuration values
  Usage:
  conf get
    Outputs the value of all runtime config options supported by the system.

  conf get <key>
    Outputs the current value of a given runtime config item.

  conf set <key> <value>
    Set the value of a given runtime config item. This change is staged
    in volatile memory until a commit operation occurs.

  conf commit
    Commit staged config changes to nonvolatile memory.

pki:
  Perform public/private key operations.
  Usage:
  pki <verb> <object> <args>
    Valid verbs are { generate, import, export, list }
    Valid object types are { key, csr, cert }
    Arguments should be specified in --<arg_name> <value>

  pki generate key <label_public> <label_private> <algorithm> <algorithm_param>
    Generates a new private key to be stored in the specified labels
```

# Heap statistics

- Type `heapstat` to display the heap usage

```
> heapstat
```

Metric	Dec (Bytes)	Hex (Bytes)	% Total
Heap Total	307200	0x4B000	100 %
Heap Free	106960	0x1A1D0	34 %
Min. Heap Free	93200	0x16C10	30 %
Heap Alloc.	200240	0x30E30	65 %
Max. Heap Alloc.	214000	0x343F0	69 %

# List the running tasks and statistics

- Type `ps` to display information about the the running tasks

```
> ps
Total Runtime: 81725
```

Task ID	State	Task Name	__Priority__		%CPU	Stack		Stack HWM	Stack Usage
			Base	Cur.		Alloc			
4	RUNNING	cli	10	10	0%	2048	1906	6%	
2	READY	IDLE	0	0	96%	1025	1001	2%	
6	BLOCKED	uartTx	24	24	0%	1024	954	6%	
15	BLOCKED	MotionS	5	5	0%	2048	1784	12%	
5	BLOCKED	uartRx	30	30	0%	1024	990	3%	
9	BLOCKED	lwIP	25	25	0%	4096	3964	3%	
10	BLOCKED	MxData	25	25	0%	4096	4026	1%	
14	BLOCKED	EnvSense	6	6	0%	1024	612	40%	
7	BLOCKED	Heartbeat	0	0	0%	128	104	18%	
8	BLOCKED	MxNet	23	23	0%	1024	838	18%	
13	BLOCKED	OTAUpdate	1	1	0%	4096	3972	3%	
17	BLOCKED	AWSDefender	5	5	0%	2048	1608	21%	
12	BLOCKED	MQTTAgent	10	10	2%	2048	1394	31%	
1	SUSPENDED	Init	8	8	0%	1024	738	27%	
11	SUSPENDED	MxCtrl	24	24	0%	4096	3972	3%	
19	BLOCKED	OTAAgent	3	3	0%	4096	3846	6%	
16	BLOCKED	ShadowDevice	5	5	0%	1024	880	14%	
3	BLOCKED	Tmr Svc	24	24	0%	2049	2019	1%	

```
>
```

# Reset (reboot) the system

- Type `reset` to reset the device

```
> reset
Resetting device.

<SYS>      0 [None      ] Reset Source: 0x14000000 : SFTRSTF: Software. (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>     809 [MQTTAgent ] Client Certificate: CN=stm32u5-6eaa0320383733M3, SN:0x00F23EE9F59262DF40 (mbedtls_transport.c:335)
<INF>     809 [MQTTAgent ] Issuer: CN=stm32u5-6eaa0320383733M3 (mbedtls_transport.c:336)
<INF>     809 [MQTTAgent ] Valid From: 1970-01-01, Expires: 2069-12-31 (mbedtls_transport.c:337)
<INF>    1109 [MQTTAgent ] CA Certificate: CN=Starfield Services Root Certificate Authority - G2, SN:0x00 (mbedtls_transport.c:335)
<INF>    1109 [MQTTAgent ] Issuer: CN=Starfield Services Root Certificate Authority - G2 (mbedtls_transport.c:336)
<INF>    1109 [MQTTAgent ] Valid From: 2009-09-01, Expires: 2037-12-31 (mbedtls_transport.c:337)
>
```

# Collateral and help

# Help and Support

- Useful links
  - <https://www.freertos.org/STM32U5/>
  - <https://github.com/FreeRTOS/iot-reference-stm32u5>
  - [www.st.com/x-cube-aws](http://www.st.com/x-cube-aws)
  - <https://community.st.com/s/article/getting-started-with-stm32u5-iot-discovery-kit-and-aws>
  - <https://community.st.com/s/article/how-to-do-an-ota-update-with-stm32u5-and-aws>
- Support
  - <https://community.st.com/s/online-support>

# Our technology starts with You



Find out more at [www.st.com](http://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](http://www.st.com/trademarks).

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



life.augmented



# Loaner Board Handout & Return

- **Please Take Note:** Due to supply limitations, ST will need to reuse the boards in this workshop for our near-team future workshops.
- **Board Logistics**
  - ST will hand out boards & USB cables
  - Please remove the boards from the packaging
    - The cardboard box can be put in the recycle bin
    - Please keep the antistatic bag for when the board is returned
  - Enjoy the workshop
  - Put the board and USB cable back into the anti-static bag
  - At the end of workshop, ST will provide a voucher to purchase the same board at a discounted price from a dedicated stock reserved specifically for workshop participants when we collect the loaner boards
- We appreciate your understanding and thank you for your cooperation!

