



CQ Bootloaders

Version V1.00

Table of Contents

1	Scope	2
2	Reference Material	2
3	Target Hardware	2
4	IEF Architecture.....	3
4.1	IEF Security.....	3
5	Bootloader Description	4
5.1	Espressif ESP32	4
5.1.1	Secure Boot V2.....	4
5.1.2	eFuse.....	5
5.1.3	Virtual eFuse.....	5
5.1.4	Bootloader Provisioning	6
5.1.5	Bootloader Modes.....	9
5.1.6	Bootloader Key	10
5.1.7	Key Points About Flash Encryption	11
5.2	M5Stack Edu Kit.....	12
5.2.1	Bootloader Modes.....	12
6	Revision History	14

1 Scope

This document is targeted at embedded software engineers who require additional details regarding the design of the secure bootloaders that are utilised, in the client hardware, as part of the CQ QuarkLink security platform. The QuarkLink makes use of secure bootloaders during the provisioning process of the target hardware. The secure bootloaders are proprietary to Crypto Quantique and are customised depending on the security functions available on the target hardware. A secure bootloader image is provisioned into target IoT devices during the **Provisioning** function of a QuarkLink (see QuarkLink User Guide). The secure bootloader is closely linked to the Initial Enrolment Firmware (IEF) that is initially provisioned into a target IoT device. The IEF is also specific to the target hardware being provisioned.

2 Reference Material

The following additional reference material is recommended for review:

- QuarkLink User Guide (available in GitHub)
- CQ Initial Enrolment Firmware Application note (available in GitHub)
- QuarkLink Demonstration Videos
(<https://www.youtube.com/@CryptoQuantique>)
- Crypto Quantique GitHub (<https://github.com/cryptoquantique>)

3 Target Hardware

The QuarkLink supports multiple hardware devices, typically secure microcontrollers from silicon manufacturers such as Espressif, Renesas, ST Microelectronics, NXP etc. The secure microcontrollers are used to build IoT devices which can be onboarded to secure cloud services using a QuarkLink.

Crypto Quantique provides QuarkLink users with embedded firmware to ensure that they (the users) can build secure IoT products that can be managed, throughout their life-cycle, by including capabilities such as firmware Over-the-Air (OTA) updates, certificate renewal, certificate revocation and onboarding to cloud service providers, MQTT brokers and databases.

Each target hardware is supported by a secure bootloader and an IEF instance. The secure bootloaders are available in source from the Espressif GitHub whereas the IEF is binary only (both developed by Crypto Quantique to ensure security).

The QuarkLink is preconfigured with all bootloaders and IEF instances required to provision all supported target hardware.

Each target hardware supported by QuarkLink has unique customisations which are dependant of the design of the security hardware in the device. This document includes appendices that cover the differences for each device.

4 IEF Architecture

The IEF is an application that is executed on the target hardware during the provisioning process and carries out the following functions:

- Configures a UART to allow communication with the QuarkLink provisioning tool
- Executes the Crypto Quantique QuarkLink client library
- Connects the target hardware to the local WiFi network (to communicate with the QuarkLink instance)
- Generates the target hardware identity cryptographic keys and stores them securely
- Configures the QuarkLink batch with the target hardware device ID.
- Checks the QuarkLink instance for firmware updates
- Executes firmware update process

4.1 IEF Security

The IEF is a proprietary firmware image that is designed by Crypto Quantique. The IEF image is signed by the QuarkLink instance that is carrying out the provisioning task. During provisioning, a secure bootloader is flashed into the target hardware first, followed by the IEF. On a reset the bootloader verifies the signature of the IEF image prior to executing it.

5 Bootloader Description

As mentioned, both the IEF and bootloader are customised for each target hardware. The bootloader design and implementation for each supported target hardware is covered separately in this section. Further information regarding the IEF is covered in the documents listed in section 2.

5.1 Espressif ESP32

ESP32 is a Wi-Fi and Bluetooth 5 (LE) System-On-Chip (SoC). It is classed as a secure MCU due to its integration of secure peripherals and functions such as Secure Boot. The ESP32 has experienced issues with older devices whereby the security features have been bypassed by the use of fault injection techniques. These issues have been resolved in newer versions and are now supported by QuarkLink. QuarkLink utilises the Secure Boot V2 (ECO 3 onwards) scheme.

5.1.1 Secure Boot V2

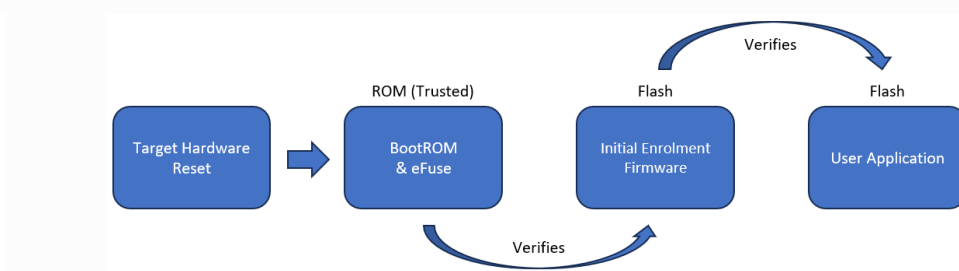
Secure Boot protects a device from running any unauthorized (i.e., unsigned) code by checking that each piece of software that is being booted is signed. On an ESP32, these pieces of software include the second stage bootloader and each application binary. Note that the first stage bootloader does not require signing as it is ROM code thus cannot be changed.

A RSA based Secure Boot verification scheme (Secure Boot V2) is implemented on ESP32 (ECO 3 onwards).

The Secure Boot process on the ESP32 **AFTER** the QuarkLink provisioning process involves the following steps:

1. When the first stage bootloader loads the second stage bootloader (IEF), the IEF's RSA-PSS (**RSA-Probabilistic Signature Scheme**) signature is verified. If the verification is successful, the IEF is executed.
2. When the IEF loads a particular user application image, the application's RSA-PSS signature is verified. If the verification is successful, the user application image is executed.

The diagram below shows the process, often referred to as the chain of trust.



The advantages of this scheme are as follows:

- The RSA-PSS public key is stored on the device. The corresponding RSA-PSS private key is kept within the Hardware Security Module (HSM) of the QuarkLink and is never accessed by the device.
- Only one public key can be generated and stored in the chip during QuarkLink provisioning.

CQ Bootloaders

Version V1.00

- Same image format and signature verification method is applied for user applications and IEF.
- No secrets are stored on the device. Therefore, it is immune to passive side-channel attacks (timing or power analysis, etc.)

For more detailed information on the secure boot process visit the Espressif website:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/security/secure-boot-v2.html>

5.1.2 eFuse

The eFuse plays an important role in the functioning of the ESP32 security features. The ESP32 has a 1024-bit eFuse, which is a one-time programmable memory. This eFuse is divided into 4 blocks of 256-bits each.



ESP32 eFuse Overview

Of primary interest to QuarkLink users are blocks 1 and 2. These blocks store keys for flash encryption and secure boot respectively. Also, once the keys are stored in the eFuse, the IEF configures it in such a way that any software running on ESP32 cannot read (or update) these keys (**disable software readout**). Once **disable software readout** has been enabled by the IEF, only the ESP32 hardware can read and use these keys for ensuring secure boot and flash encryption.

Important Note: Once the Secure Boot eFuse has been blown, the target hardware cannot be updated with firmware unless it has been correctly signed by the associated QuarkLink instance. Also, the target hardware cannot be erased to restore it to the factory default condition, it is non-recoverable. Please ensure that when provisioning via the QuarkLink the production/release version of the bootloader is not provisioned unless the user is fully conversant with the implications of enabling maximum security for their target hardware.

5.1.3 Virtual eFuse

To help users during prototyping, Crypto Quantique have implemented a **Virtual eFuse** or **veFuse**. This is a software modification that allows virtual keys to be programmed into SRAM memory and, therefore, modified or erased at any time. During startup, the eFuses are copied to SRAM. All eFuse operations (read and write) are performed with SRAM instead of the real eFuse registers. This allows users to fully test their connected device with temporary cryptographic keys. Once all unit tests and prototyping are complete, the user firmware can be rebuilt using the correct configuration. Please review the CQ Initial Enrolment Firmware Application note for more detailed information.

5.1.4 Bootloader Provisioning

This section describes the process of configuring the target device during the QuarkLink provisioning process.

5.1.4.1 First Stage Bootloader

The provisioning process requires that the target hardware proceeds through two boot sequences. The first boot sequence (see Figure 1) is responsible for generating the Flash encryption key and storing it in the NVS partition. This is carried out if the Flash Encryption flag is set.

Firmware and partition table information are programmed into the target devices' off-chip Flash memory in plain text (unencrypted). The ESP32 factory bootloader is configured to enable Flash encryption. Once the factory bootloader is run it will encrypt both the plain text firmware and partition table and reprogram the off-chip memory with the encrypted images.

For a factory default device, the encryption key is not available and so part of the provisioning process is to also generate a key using the TRNG (True Random Number Generator). Once the key has been generated and the firmware encrypted, all the requisite information and configuration can be completed, locking the device forever in a secure mode.

Note: Self generation of the encryption key is highly secure as the key never leaves the target device and is not accessible by the debugger or firmware once secure boot is enabled. This key generation methodology is implemented during QuarkLink Provisioning of an ESP32 device.

Note : If ESP32 is reset / powered down during the 1st boot, it will begin the process again on next boot

Figure 1 shows the detail of the flow for provisioning the first stage of the process as described above.

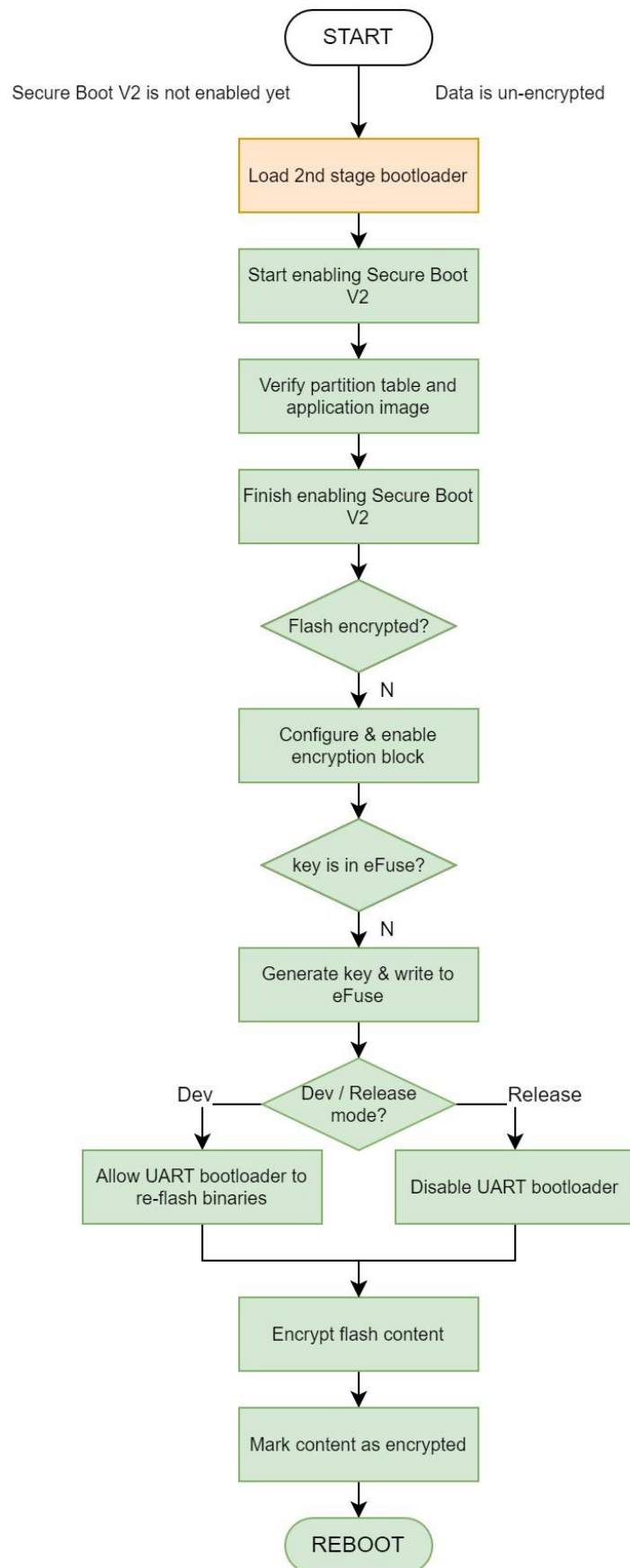


Figure 1: First stage provisioning process

CQ Bootloaders

Version V1.00

5.1.4.2 Second Stage Bootloader

Once the first stage provisioning process is complete, the target device is reset and the normal secure boot process is initiated. This second boot process is now the default process of the ESP32 and is locked i.e. firmware updates to the device are not possible unless the firmware has been signed. Also the device cannot be reset to its factory default. Figure 2 shows the flow of the second (default) boot process.

Notes:

During the second (default) boot process, if any of the verification steps fail, the process is aborted

Secure Boot v2 is not enabled until **after** a valid partition table and app image have been flashed

The cryptographic key used for off-chip memory encryption is an AES-256 key, generated using the TRNG module. It is written, by the module, to eFuse and R/W protected, meaning it can't be accessed from software or external hardware.

The encryption process can take up to 1 min for large partitions.

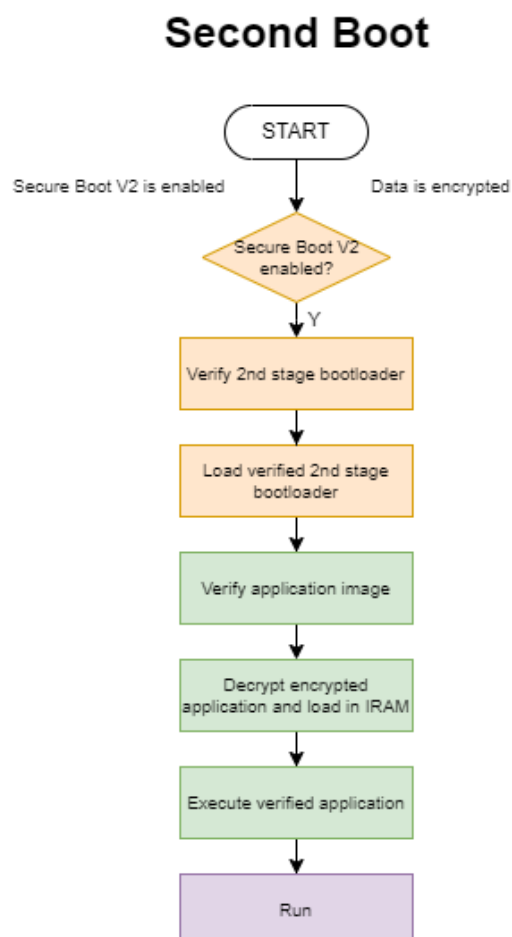
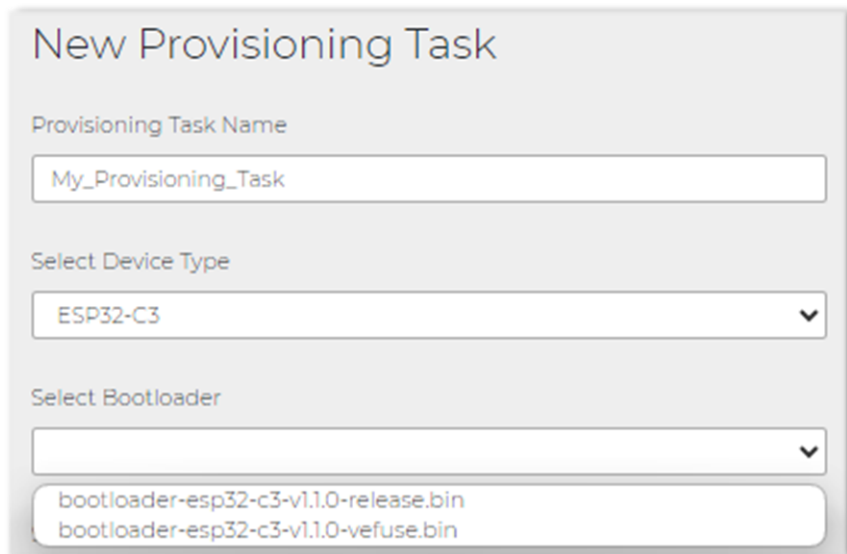


Figure 2: Second (default) boot process

5.1.5 Bootloader Modes

The ESP32 has two bootloader modes. The QuarkLink is preconfigured with two bootloader images, allowing the user to select their preferred mode (see Figure 3).



The screenshot shows a web form titled "New Provisioning Task". It contains three main sections: "Provisioning Task Name" with a text input field containing "My_Provisioning_Task"; "Select Device Type" with a dropdown menu showing "ESP32-C3"; and "Select Bootloader" with a dropdown menu. The "Select Bootloader" dropdown is open, showing two options: "bootloader-esp32-c3-v1.1.0-release.bin" and "bootloader-esp32-c3-v1.1.0-vefuse.bin".

Figure 3: Bootloader mode image selection

The following flash encryption modes are available:

veFuse – This bootloader type (Virtual eFuse) is used for development and prototyping as the cryptographic security keys are stored in the target hardware volatile memory (SRAM). This ensures that the target device is recoverable once the bootloader has been flashed. It is not recommended that this bootloader type is used for production IoT devices.

Release – Release bootloaders have their cryptographic security keys programmed into OTP (One Time Programmable) memory. The device cannot be reprogrammed with new keys or with a different bootloader once the release version has been provisioned. This bootloader type should be used in production only.

Please refer to the ESP32 Programming Guide for further details of the bootloader modes and how to use them.

CQ Bootloaders

Version V1.00

5.1.6 Bootloader Key

As mentioned in section 5.1.1, a firmware signing key is programmed into the eFuse memory area that is used to verify the signature of the second stage bootloader. The key is an RSA public key and is programmed into the target device during the QuarkLink provisioning task. Figure 3 shows details of the key configuration.

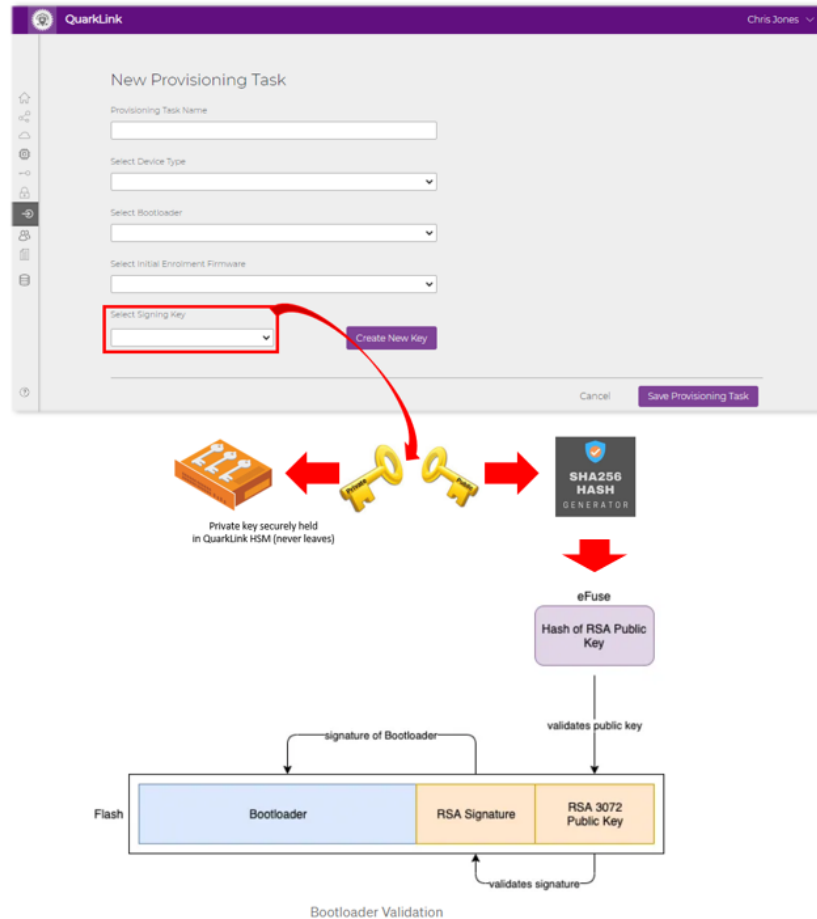


Figure 3: Secure boot signing key configuration

- **Bootloader Image:** This is the bootloader executable that is selected by the QuarkLink user during **Provisioning**.
- **RSA Signature:** This is the RSA3072 based signature of the bootloader image. The key is generated by QuarkLink and is created/selected during the creation of a **Provisioning Task**.
- **RSA 3072 Public Key:** The public key that can be used to validate the signature.

For more information regarding the secure bootloader implemented in the QuarkLink ESP32 provisioning process, please review document **ESP-IDF Programming Guide – Flash Encryption**.

5.1.7 Key Points About Flash Encryption

- Flash memory contents are encrypted using AES-256. The flash encryption key is stored in the `flash_encryption` eFuse internal to the chip and, by default, is protected from software access.
- The flash encryption algorithm is AES-256, where the key is “tweaked” with the offset address of each 32 byte block of flash. This means that every 32-byte block (two consecutive 16 byte AES blocks) is encrypted with a unique key derived from the flash encryption key.
- Flash access is transparent via the flash cache mapping feature of ESP32 - any flash regions which are mapped to the address space will be transparently decrypted when read.

Some data partitions might need to remain unencrypted for ease of access or might require the use of flash-friendly update algorithms which are ineffective if the data is encrypted.

- If flash encryption might be used in future, the user must keep it in mind and take certain precautions when writing code that uses encrypted flash.

Enabling flash encryption will increase the size of bootloader, which might require updating the partition table offset.

5.2 M5Stack Edu Kit

The M5Stack Edu Kit is the reference hardware kit primarily developed for AWS as an IoT Kit. It was developed to enable users to learn how to build IoT applications using AWS services. It comes equipped with a Microchip ATECC608 Trust&GO pre-provisioned secure element, in addition to the existing features of the standard M5Stack Core2. QuarkLink utilises the cryptographic key generation and storage capabilities of the ATECC608 for enhanced security.

Crypto Quantique have taken advantage of the features of the kit to promote the QuarkLink security platform. The QuarkLink provides more advanced features such as onboarding the kit to MQTT brokers, databases and other cloud service providers (not only AWS).

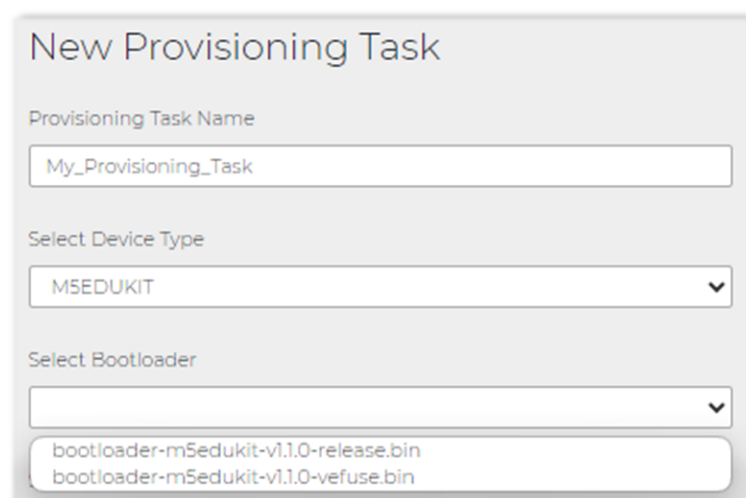
The M5Stack Edu Kit comes with an ESP32-D0WDQ6-V3 micro controller, which features dual Xtensa 32-bit LX6 cores and a main frequency up to 240Mhz, with 2.4GHz Wi-Fi. It includes 8MB PSRAM and 16MB flash on board.

The main unit is equipped with a 2.0-inch capacitive touch screen that provides a smooth and responsive human machine interface. The built-in vibration motor can be used to provide haptic feedback or alerts. Onboard RTC module provides accurate time of day. Power is supplied through an AXP192 power management chip, to monitor and control power attributes of the device. The included TF card slot supports microSD cards up to 16GB. The on-board speaker is paired with an I2S digital audio interface power amplifier chip to reduce signal distortion and provide clearer audio output. There are independent physical power and reset (RST) buttons on the sides of M5Stack Edu Kit, with 3 programmable touch buttons on the front of the screen.

The ESP32-D0WDQ6-V3 is classed as a secure MCU due to its integration of secure peripherals and functions such as Secure Boot. The ESP32 has experienced issues with older devices whereby the security features have been bypassed by the use of fault injection techniques. These issues have been resolved in newer versions and are now supported by QuarkLink. QuarkLink utilises the Secure Boot V2 (ECO 3 onwards) scheme.

5.2.1 Bootloader Modes

The M5Stack Edu Kit has two bootloader modes. The QuarkLink is preconfigured with two bootloader images, allowing the user to select their preferred mode (see Figure 4).



The screenshot shows a web form titled "New Provisioning Task". It contains three main sections: "Provisioning Task Name" with a text input field containing "My_Provisioning_Task"; "Select Device Type" with a dropdown menu showing "M5EDUKIT"; and "Select Bootloader" with a dropdown menu showing two options: "bootloader-m5edukit-v1.1.0-release.bin" and "bootloader-m5edukit-v1.1.0-vefuse.bin".

Figure 4: Bootloader mode image selection

The following flash encryption modes are available:

veFuse – This bootloader type (Virtual eFuse) is used for development and prototyping as the cryptographic security keys are stored in the target hardware volatile memory (SRAM). This ensures that the target device is recoverable once the bootloader has been flashed. It is not recommended that this bootloader type is used for production IoT devices.

Release – Release bootloaders have their cryptographic security keys programmed into OTP (One Time Programmable) memory. The device cannot be reprogrammed with new keys or with a different bootloader once the release version has been provisioned. This bootloader type should be used in production only.

Please refer to the ESP32 Programming Guide for further details of the bootloader modes and how to use them.

6 Revision History

CQ Bootloaders

Rev.	Date	Owner	Description
1.00	22.9.2023	CDJ	Original document

CQ Bootloaders

Version V1.00

Legal Notice Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. CRYPTO QUANTIQUE MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Crypto Quantique disclaims all liability arising from this information and its use. Use of Crypto Quantique devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Crypto Quantique from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Crypto Quantique intellectual property rights unless otherwise stated.



United Kingdom

Unit 304-5,
164-180 Union Street,
London
SE1 0LH

General contact email:
info@cryptoquantique.com

QUANTUM DRIVEN CYBERSECURITY

The most advanced security product for the Internet of Things in the world