

Software Design Document

For

Firmware Over-The-Air Update in FreeRTOS

Version 1.0

Prepared By

Jinu Raju

Kiran Thomas George Tharakan

Navaneeth Sunil

Sharan K Lakshman

TKM College Of Engineering

5 April 2023

Table of Contents

1 . Introduction

1.1 Scope

1.2 Purpose

1.3 Intended Audience

1.4 References

2. System Architecture Design

3. Design

3.1 System Design

3.2 Module 1 Design: OTA Update module

3.3 Module 2 Design: Device Configuration module

3.4 Module 3 Design: Communication module

3.5 Module 4 Design: Security module

3.6 Module 5 Design: Logging module

3.7 Module 6 Design: Error handling module

3.8 Module 7 Design: User interface module

4. Basic Graphical User Interface

5. Technology Stack

6. Testing

1. OVERVIEW

1.1 Scope

This SDD document defines the design of a reusable library for FreeRTOS, with an update portal for over-the-air updates based on the BalenaOS reference design. The library provides flexible and scalable APIs for managing the update process, and a web-based portal for uploading and distributing updates. It can be integrated into existing FreeRTOS projects and is compatible with various hardware platforms. The SDD document describes the library's detailed design, including software architecture, interfaces, and customization guidance. It does not cover implementation or specific application development.

1.2 Purpose

The purpose of this document is to define the design requirements for a reusable library for FreeRTOS with an update portal, based on the reference design from BalenaOS runtime and over-the-air updates.

1.3 Intended audience

The intended audience for the SDD document of the project "Firmware Over-The-Update in FreeRTOS" includes project managers, developers, quality assurance teams, testers, and customers/stakeholders.

1.4 Definitions, Acronyms and Abbreviations

Acronym	Meaning
OTA	Over The Air Update
RTOS	Real Time Operating System
FOTA	Firmware Over The Update
SDD	Software Design Description

1.5 References

<https://www.freertos.org/ota/index.html>

<https://www.freertos.org/2022/01/delta-over-the-air-updates.html>

<https://www.balena.io>

2. System Architecture Design

Embedded Devices: These are the devices running FreeRTOS operating system that need to be updated with the latest firmware. They are connected to the internet through wireless communication protocols such as Wi-Fi

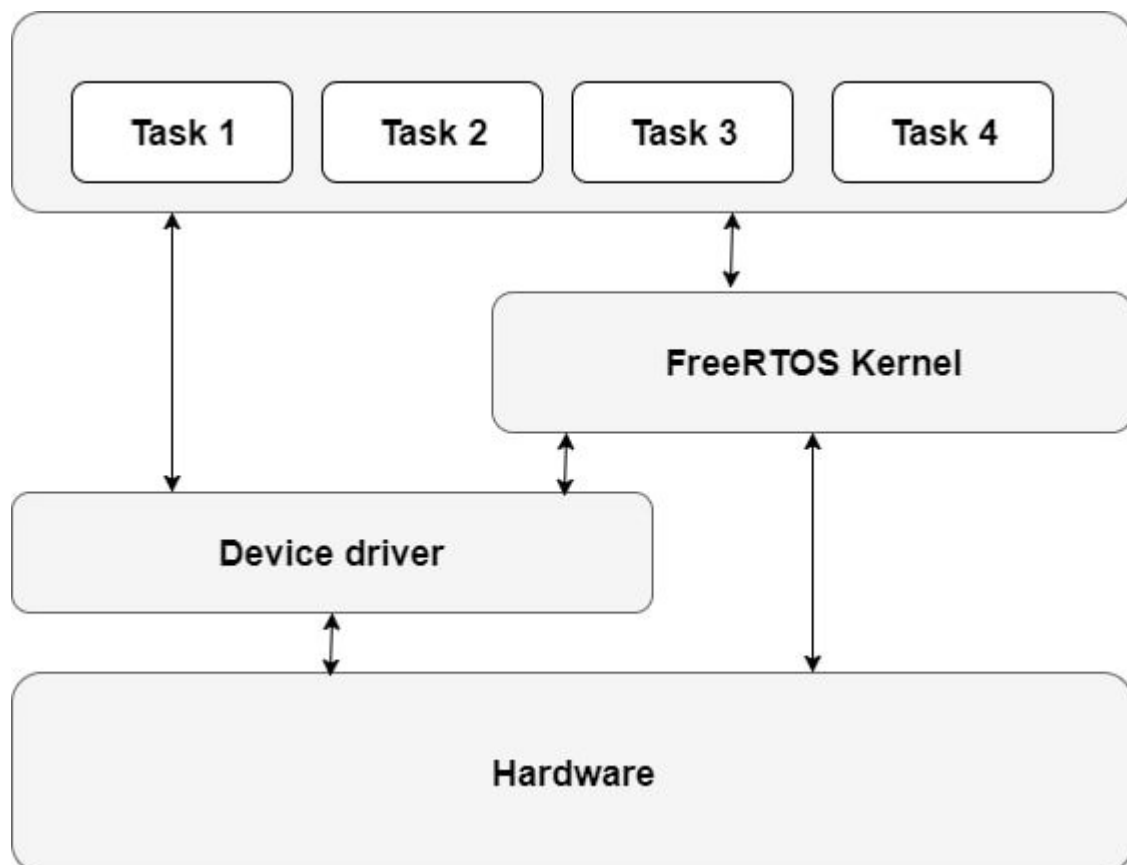
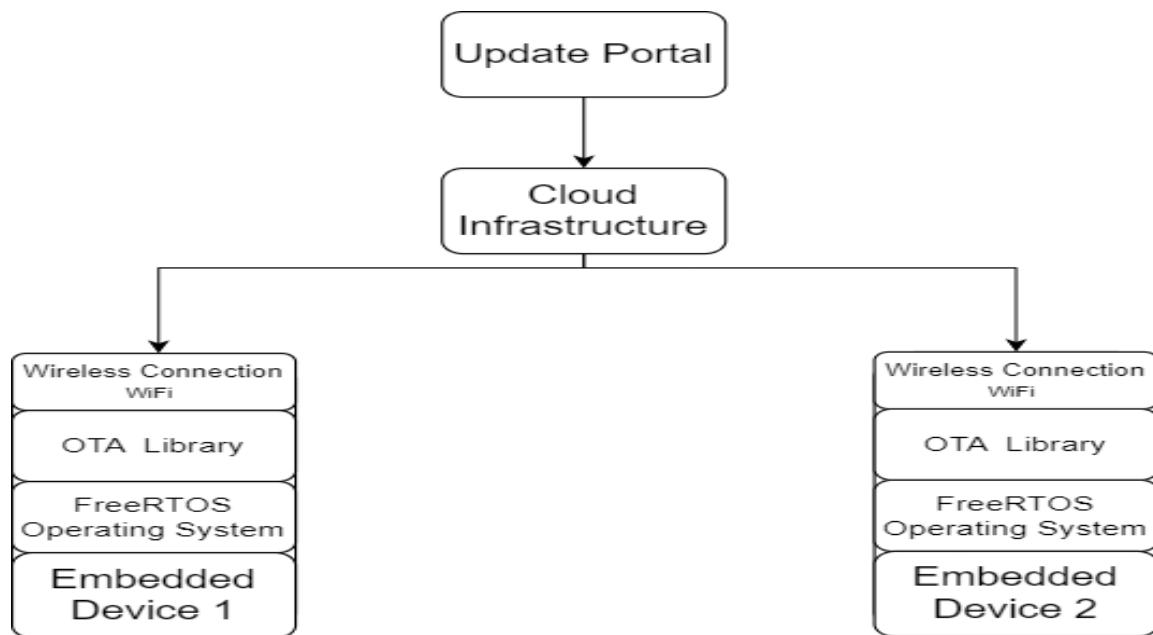
FOTA Library: This is a reusable software library for FreeRTOS that provides the functionality to download, verify and apply firmware updates over the air to the embedded devices. It will be developed to operate within the resource constraints of the hardware module.

Bootloader: This is a small software program that is responsible for updating the firmware on the embedded devices. It will interface with the FOTA library to download, verify and apply firmware updates to the devices.

Update Portal: This is a web-based portal that allows users to initiate and monitor over-the-air firmware updates. It provides a simple and intuitive interface that can be accessed from any device with a web browser. The portal will be responsible for managing user authentication and authorization.

Cloud Infrastructure: This component provides the necessary cloud resources to host the update portal and manage the firmware updates. It will include a cloud database to store device information, firmware updates, and other related data.

Security Components: The update process must be secure and must include encryption of data, certificate validation, and secure storage of sensitive information. This component will be responsible for implementing the necessary security features to ensure the integrity and confidentiality of the update process.

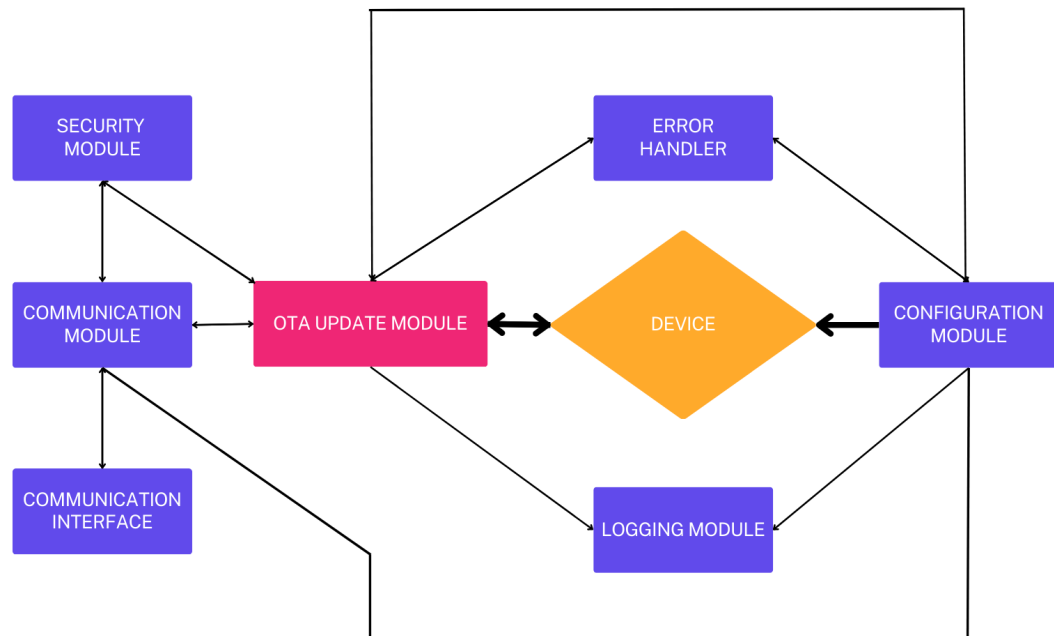


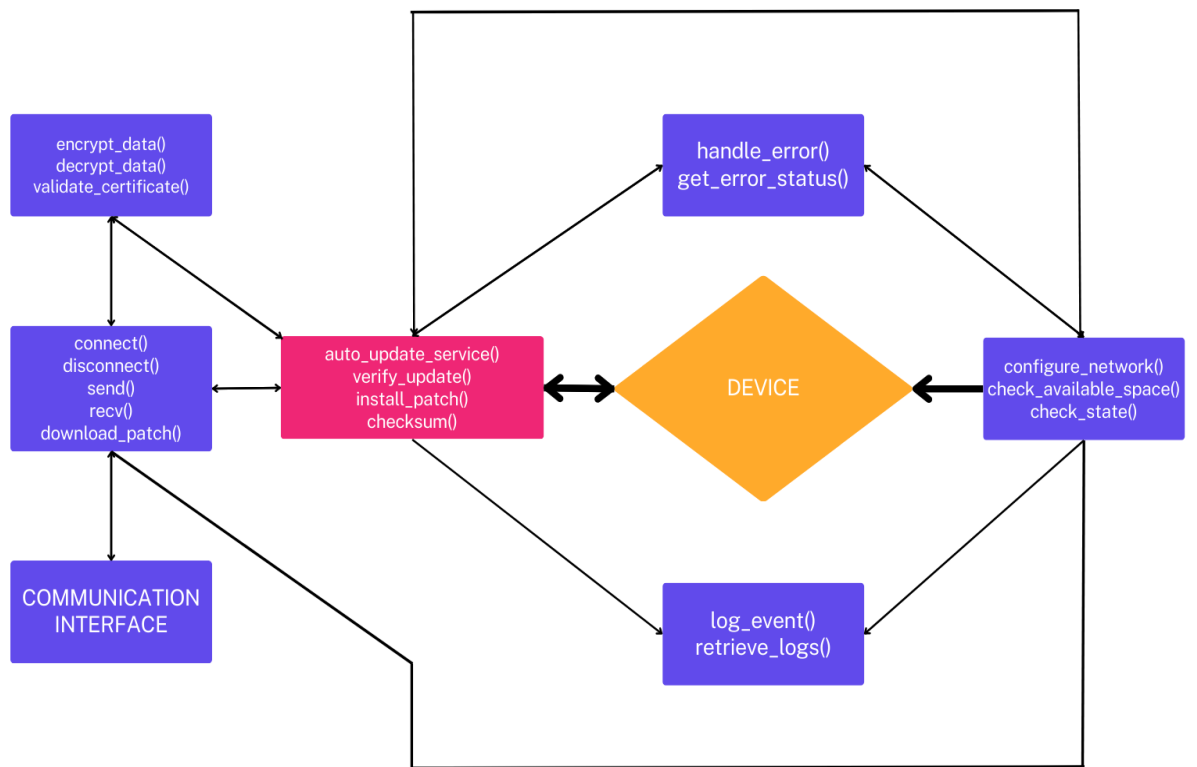
3.Design

3.1 System Design

The OTA Update system consists of several modules that work together to handle over-the-air updates. These modules include the OTA Update module, the Device Configuration module, the Communication module, the Security module, the Logging module, the Error Handling module, and the User Interface module (optional).

Each module in the OTA Update system has a specific function and interacts with other modules in a predefined way. The module design phase involves defining the interfaces for each module and the interactions between them. This ensures that each module can be developed independently, tested thoroughly, and integrated into the larger system.





3.2 Module 1 Design: OTA Update module

The OTA Update module is responsible for handling the over-the-air updates, verifying the authenticity of the update, and installing the update on the hardware module. The module will have the following inputs and outputs:

Inputs:

- Update file: This is the file containing the update to be installed on the hardware module.
- Verification data: This is data used to verify the authenticity of the update, such as a digital signature or checksum.

Outputs:

- Success/failure status: This indicates whether the update was installed successfully or if an error occurred.
- Error message: If the update installation failed, an error message will be generated indicating the cause of the failure.

Interactions with other modules:

- The OTA Update module will interact with the Device Configuration module to ensure that the device is in a safe state for the update and to verify that there is sufficient space to install the update.
- The module will also interact with the Security module to verify the authenticity of the update file and ensure that it has not been tampered with.
- Finally, the OTA Update module will interact with the Logging module to log events related to the update process.

3.3 Module 2 design : Device configuration module

The Device Configuration module is responsible for configuring the device for the update process. This includes setting up network connections, verifying available space for the update, and ensuring that the device is in a safe state for the update. The module will have the following inputs and outputs:

Inputs:

- Network configuration: This includes information about the device's network connection, such as its IP address, port number, and security settings.
- Available space: This indicates the amount of free space available on the device's storage.
- Device state: This indicates whether the device is in a safe state for the update.

Outputs:

- Success/failure status: This indicates whether the device has been successfully configured for the update process or if an error occurred.
- Error message: If the configuration process failed, an error message will be generated indicating the cause of the failure.

Interactions with other modules:

- The Device Configuration module will interact with the OTA Update module to ensure that the device is in a safe state for the update and to verify that there is sufficient space to install the update.
- The module will also interact with the Communication module to establish a secure connection to the update portal.

3.4 Module 3 Design: Communication module

The Communication module is responsible for establishing and maintaining secure communication between the hardware module and the update portal. The module will have the following inputs and outputs:

Inputs:

- Network configuration: This includes information about the device's network connection, such as its IP address, port number, and security settings.
- Update file: This is the file containing the update to be installed on the hardware module.

Outputs:

- Success/failure status: This indicates whether the communication has been established successfully or if an error occurred.
- Error message: If the communication failed, an error message will be generated indicating the cause of the failure.

Interactions with other modules:

- The Communication module will interact with the Device Configuration module to obtain the necessary network configuration information.
- The module will also interact with the Security module to establish a secure connection to the update portal and ensure that data is encrypted.

3.5 Module 4 Design: Security module

The Security module is responsible for ensuring the security of the update process. This includes encrypting data, validating certificates, and securing storage of sensitive information. The module will have the following inputs and outputs:

Inputs:

- Network configuration: This includes information about the device's network connection, such as its IP address, port number, and security settings.
- Verification data: This is data used to verify the authenticity of the update, such as a digital signature or checksum.
- Update file: This is the file containing the update to be installed on the hardware module.

Outputs:

- Success/failure status: This indicates whether the security checks were passed successfully or if an error occurred.
- Error message: If the security checks failed, an error message will be generated indicating the cause of the failure.

Interactions with other modules:

- The Security module will interact with the OTA Update module to verify the authenticity of the update file and ensure that it has not been tampered with.
- The module will also interact with the Communication module to establish a secure connection to the update portal and encrypt data.

3.6 Module 5 Design: Logging module

The Logging module will be responsible for recording all events that occur during the update process. It will capture information such as the start and end time of the update, any errors or warnings encountered, and the status of the update (successful or failed).

Input:

- The Logging module will receive data from other modules, such as error messages from the Error handling module and status updates from the OTA Update module.

Output:

- Logs: The Logging module will write the information it receives to a log file or database for later analysis and troubleshooting purposes.

Interactions with other modules:

- The Logging module will interact with all the other modules in the system, as it will need to capture events that occur during the update process. It will receive input from the OTA Update module, Device Configuration module, Communication module, Security module, and Error handling module. The Logging module will also provide output to any other module that requires access to the logs, such as a monitoring tool or a troubleshooting interface.

3.7 Module 6 Design: Error handling module

The Error handling module will be responsible for detecting and handling errors that occur during the update process. It will provide feedback to the user and take appropriate action to address the error.

Input:

- The Error handling module will receive error messages from other modules, such as the OTA Update module or the Device Configuration module.

Output:

- The Error handling module will provide feedback to the user in the form of an error message, explaining the nature of the error and any recommended actions.

Interactions with other modules:

- The Error handling module will interact with all the other modules in the system, as any module could encounter an error during the update process. It will receive input from the OTA Update module, Device Configuration module, Communication module, Security module, and Logging module. The Error handling module will also provide output to the User interface module, as it will need to display error messages to the user.

3.8 Module 7 Design: User interface module (optional)

The User interface module is an optional module that will provide a graphical interface for the update process. It will display progress bars, status messages, and error messages to the user, making the update process more user-friendly.

Input:

- OTA Update module: The User Interface module will receive input from the OTA Update module regarding the progress of the update process and any issues that arise.
- Error Handling module: The User Interface module will receive input from the Error Handling module regarding any errors that occur during the update process.

Output:

- Status messages: The User Interface module will output status messages that can be displayed to the user to provide feedback on the progress of the update process.
- Error messages: The User Interface module will output error messages that can be displayed to the user to provide feedback on any issues that arise during the update process..

Interactions with other modules:

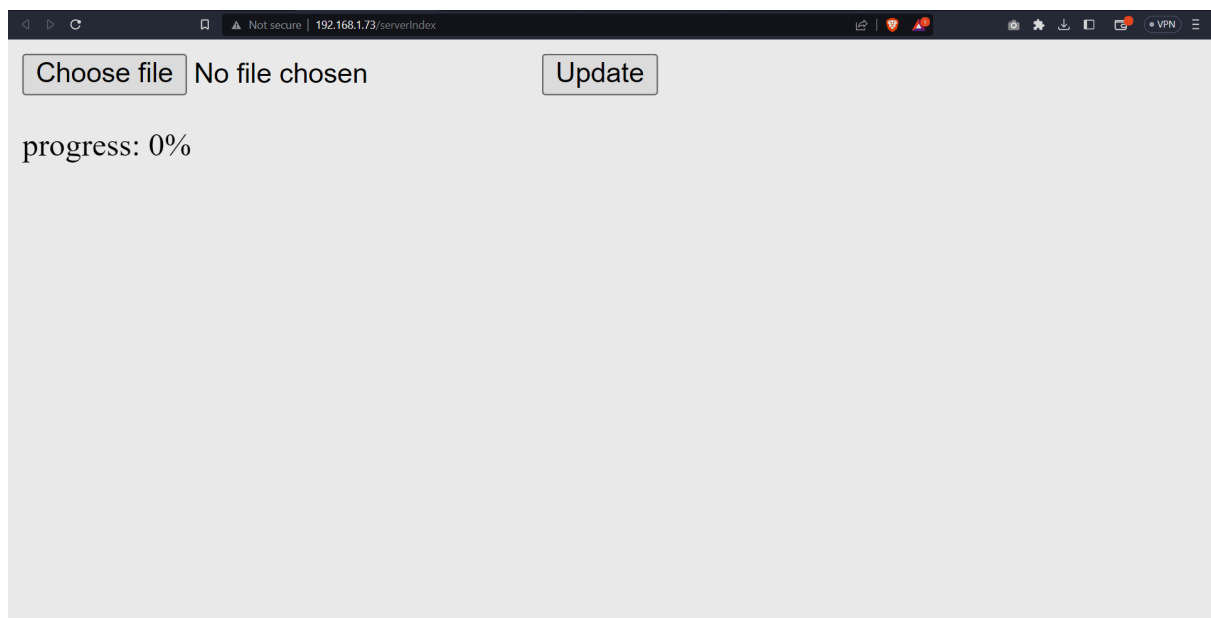
- OTA Update module: The User Interface module will interact with the OTA Update module to receive input on the progress of the update process and any issues that arise.
- Error Handling module: The User Interface module will interact with the Error Handling module to receive input on any errors that occur during the update process

4. Basic Graphical User Interface:



A screenshot of a web-based login interface for an ESP32. The page has a grey background. At the top, the title "ESP32 Login Page" is centered in a bold, black, serif font. Below the title, there are two input fields. The first is labeled "Username:" and the second is labeled "Password:". Both labels are in a black, serif font. The input fields are white with a thin black border. Below the password field, there is a "Login" button with a black border and a light grey background.

Log in screen at the server end



A screenshot of a web-based file upload interface. The browser's address bar shows "192.168.1.73/serverIndex". The page has a light grey background. At the top, there is a "Choose file" button, followed by the text "No file chosen", and then an "Update" button. Below this, the text "progress: 0%" is displayed. The rest of the page is a large, empty light grey area.

Update upload screen

Choose file ota1.ino.esp32.bin

Update

progress: 40%

Upload in progress

5. Technology Stack:

5.1 FreeRTOS - for the operating system.

FreeRTOS is a market-leading real-time operating system ([RTOS](#)) for microcontrollers and small microprocessors. Distributed freely under the MIT open source license, FreeRTOS includes a kernel and a growing set of IoT libraries suitable for use across all industry sectors. FreeRTOS is built with an emphasis on reliability and ease of use.

5.2 C programming language - for the development of the firmware update module

5.3 BalenaOS runtime - as a reference design for the over-the-air updates.

5.4 HTTP - for the communication between the update portal and the device

5.5 Version control system - Git

5.6 Integrated Development Environment (IDE) - Arduino IDE

6. Testing

Unit Testing: Individual units or components of the software are tested in isolation to ensure that each unit/component performs as expected. For the FreeRTOS update library, we write unit tests to test the functionality of each function, module, or component.

Integration Testing: Different units or components of the software are tested together to ensure that they work as expected when integrated. For the FreeRTOS update library, you can write integration tests to test the interaction between different components of the library.

System Testing: Entire system or software is tested as a whole to ensure that it meets the requirements and specifications. For the FreeRTOS update library, system testing is performed to verify that the library meets the functional and non-functional requirements and specifications.

Regression Testing: Previously tested functionality is retested after making changes or fixes to the software to ensure that the changes have not introduced any new bugs or issues.

Performance Testing: Performance and scalability of the software are tested under various load conditions to ensure that the software can handle the expected load and scale as needed.

Acceptance Testing: Acceptance testing is a type of testing where the software is tested to ensure that it meets the acceptance criteria and requirements of the stakeholders.

Security Testing: Security testing is a type of testing where the software is tested for vulnerabilities and weaknesses that can be exploited by attackers. Perform security testing to identify any security vulnerabilities and weaknesses in the library and take measures to mitigate them.