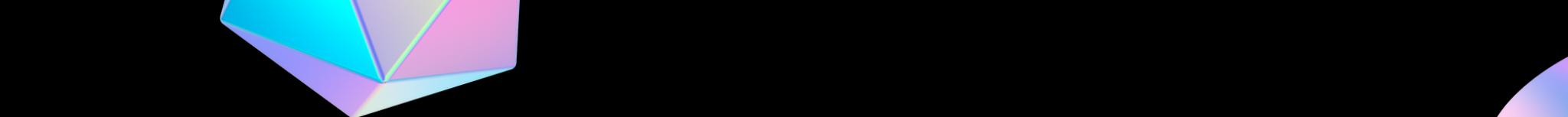




O-T-A FIRMWARE UPDATE MODULE

SHARAN K LAKSHMAN - 58
KIRAN THOMAS GEORGE THARAKAN - 71
JINU RAJU - 72
NAVANEETH S - 73



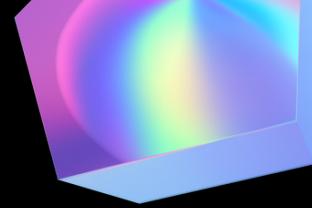
PROBLEM STATEMENT:

Firmware over the air update module in FreeRTOS

- Firmware updates are essential for embedded systems, but updating firmware can be challenging, especially for systems that are deployed remotely or in inaccessible locations.
- Firmware Over-the-Air (FOTA) updates offer a solution by enabling firmware updates to be delivered over the air using wireless networks.
- Implementing FOTA updates in FreeRTOS-based systems requires specialized knowledge and expertise.
- The problem statement is to develop a FOTA update module for FreeRTOS that is reliable, secure, and easy to use.
- The proposed solution will enable embedded systems designers to update firmware in their devices with confidence, ensuring that bugs are fixed, features are added, and performance is improved.

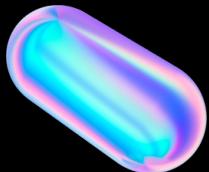
OBJECTIVES:

- Understand the basics of FreeRTOS and how it manages tasks and resources.
- Understand the importance of Firmware Over-the-Air (FOTA) updates in modern embedded systems and the advantages it offers.
- Learn how to implement a FOTA update module in FreeRTOS using OTA endpoints provided by Amazon Web Services (AWS) and the tools required for OTA updates.
- Understand the security aspects of FOTA updates and how to ensure the security of the update process.



MOTIVATION

The motivation for undertaking this project lies in the increasing demand for remote firmware updates in embedded systems. This can be especially useful in situations where it is not feasible or practical to physically access the device to update its firmware. By implementing FOTA, it is possible to improve the functionality, security, and stability of embedded systems without the need for physical intervention.



ABSTRACT

This project is focused on developing a system that enables Firmware Over-The-Air (FOTA) updates for embedded systems running FreeRTOS. The proposed solution aims to improve the security and reliability of the firmware update process by utilizing the robustness of FreeRTOS and modern OTA update techniques.

ABSTRACT

The system consists of an OTA server and client that communicate over the internet, the target embedded device running FreeRTOS that requires the firmware update. The OTA server stores the latest firmware version, which is downloaded and transferred to the target device. The firmware update process is executed in a secure and reliable manner by using FreeRTOS features such as task management, inter-task communication, and memory protection.

ABSTRACT

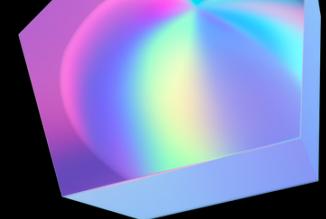
The system will include several key features, such as the ability to revert back to the previous firmware state in case of a failure during the update process. Additionally, the system will store the old firmware in a separate location to ensure that it is available in case of any issues. The project will make use of FreeRTOS's real-time operating system to ensure that the update process does not interfere with the normal operation of the device.

ABSTRACT

The system's feasibility is demonstrated through experimental testing on a FreeRTOS-based embedded device. The results indicate that the proposed solution can provide a reliable and secure FOTA update process for FreeRTOS-based embedded systems. The ultimate goal of this project is to provide a robust and reliable FOTA update mechanism that can be used in a wide range of embedded systems.

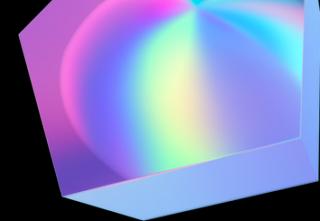
LITERATURE REVIEW

- BalenaOS is a container-based operating system
- One of the key features of BalenaOS is its support for Over The Air (OTA) updates, which allows users to remotely update their devices with new software releases without needing physical access to the device.
- However, many low-resource hardware modules do not support POSIX based systems, making it challenging to implement OTA updates on these devices.
- To address this challenge, a reusable library for FreeRTOS with an update portal can be created.



LITERATURE REVIEW

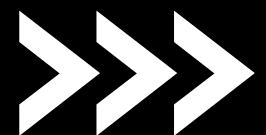
- FreeRTOS is a popular real-time operating system that is well suited for low-resource hardware.
- By incorporating an update portal into FreeRTOS, OTA updates can be implemented on devices that do not support POSIX-based systems
- The library can include features such as a bootloader, a mechanism for downloading and verifying updates, and a rollback mechanism in case of failed updates.

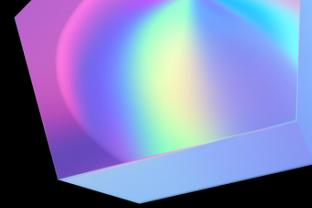


LITERATURE REVIEW

- Benefits of using a reusable library for OTA updates on low-resource hardware modules include easy and efficient management of software updates over the entire lifecycle of the device
- In conclusion, creating a reusable library for FreeRTOS with an update portal based on the BalenaOS reference design can provide a reliable and secure solution for implementing OTA updates on low-resource hardware modules that do not support POSIX-based systems.

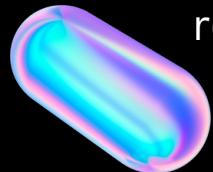
Project Plan

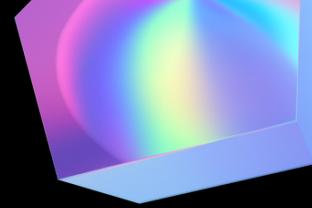




PHASES

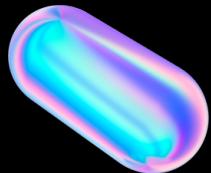
- **Requirements Gathering:** Gather requirements for the library by analyzing the BalenaOS reference design and Over The Air updates system. Identify the key features and functionalities required for the library.
- **Library Design:** Design the library architecture and module structure. Identify the different modules required for the library and the interactions between them.
- **Library Development:** Develop the library modules using the design from step 2. Ensure that the library is designed to work seamlessly with the update portal.
- **Update Portal Development:** Develop an update portal that will allow for the delivery of updates to the hardware module. The portal should support secure communication and over-the-air updates, as well as be designed with low-resource hardware modules in mind.

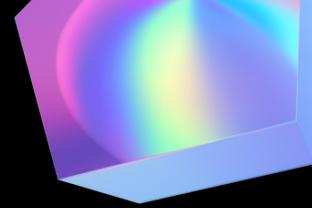




PHASES

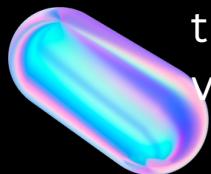
- **Testing:** Perform testing on the library and update portal to ensure that they meet the requirements. Conduct unit tests, integration tests, and system tests.
- **Documentation:** Document the library and update portal, including instructions for use and any known limitations.
- **Release:** Release the library and update portal as a re-usable library for FreeRTOS with an update portal.

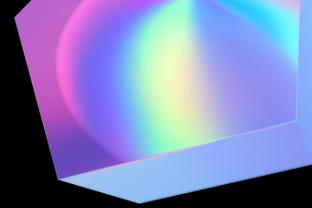




MODULES

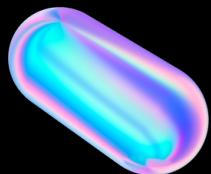
- **OTA Update module:** This module will be responsible for handling the over-the-air updates, verifying the authenticity of the update, and installing the update on the hardware module.
- **Device Configuration module:** This module will be responsible for configuring the device for the update process. This could include things like setting up network connections, verifying available space for the update, and ensuring that the device is in a safe state for the update.
- **Communication module:** This module will be responsible for establishing and maintaining secure communication between the hardware module and the update portal.
- **Security module:** This module will be responsible for ensuring the security of the update process. This could include things like encryption of data, certificate validation, and secure storage of sensitive information.

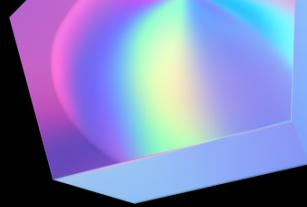




MODULES

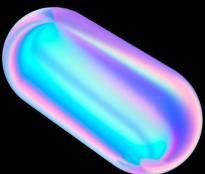
- **Logging module:** This module will be responsible for logging events during the update process for debugging and troubleshooting purposes.
- **Error handling module:** This module will be responsible for handling errors that occur during the update process, and providing appropriate feedback to the user.
- **User interface module (optional) :** This module will be responsible for providing a user interface for the update process. This could include things like progress bars, status messages, and error messages





TIMELINE

- Requirements Gathering: 1 week
- Library Design: 2 weeks
- Update Portal Development: 1 - 2 weeks
- Library Development: 3 - 4 weeks
- Documentation and Testing: 1 week





THANK YOU

