Ecole Nationale Supérieure d'Ingénieurs de Tunis

Université de Tunis
Ecole Nationale Supérieure
d'Ingénieurs de Tunis

الجمهورية التونسية
وزارة التعليم العالي والبحث العلمي
جامعة تونس
المدرسة الوطنية العليا للمهندسين بتونس

# Summer Internship Rapport

**Class : 2 GI B**

Presented by
AYARI Sarra

**Company** : SAGEMCOM

**Supervisor** : Ms BEN ROMDHAN asma

**Period :**

**Start** : 07/07/2022 **End** : 08/08/2022

**Academic year : 2022-2023**

5, Avenue Taha Hussein – Tunis
B. P. 56, Bab Menara 1008

Tel. : 71. 496. 066: الهاتف
Fax : 71. 391. 166: فاكس

5، شارع طه حسين ـ تونس
ص. ب.: 56 باب منارة 1008

# Dedication

First of all, I would like to direct my thanks to the director of Sagemcom and the HR of SagemCom, Mme Baffoun Hejer who graced me with the opportunity to do this internship within the organization and for the facilities provided to accomplish this internship.

Secondly, I'd like to express my gratitude toward my supervisor, Mme.Ben Romdhane Asma, for her sacrifice and patience all along this project.

Lastly, i wanted to mention the people that helped create a healthy workplace environment for me

I am extremely grateful to my department staff members and friends who helped me in the successful completion of this internship.

# Table of Content

# General introduction

With more and more connected devices around us, the chance that you've been hit by an update notification is high.

A software update (also known as patch) is a set of changes to a software to update, fix or improve it. Changes to the software will usually either fix bugs, fix security vulnerabilities, provide new features or improve performances and usability. Infrequently, patches may also be used to limit functionality, remove or disable features.

From a security standpoint, software updates have important implications. When an update includes a fix for security vulnerabilities, any device running an out-of-date version of the software is particularly vulnerable. This allows malicious actors to know what vulnerabilities exist on a given system and, consequently, puts devices running this software (version) more at risk.

# chapter I : actual mods of firmware upgrades in gateways

## I.1 Introduction :

The technology landscape is constantly changing, especially for cutting-edge Internet of Things products. Updating these devices is essential in order to add new functionality and keep them protected from malicious actors.

Years ago, software updates needed to be installed using CDs or floppy disks that you would buy at the store or receive in the mail. With faster Internet speeds, however, “over the air” (OTA) has become the preferred method of delivering updates. Nevertheless, manual updates are still an option, and may even be the preferred choice in some cases.

## I.2 Manual Firmware Updates :

Manual firmware updates were the only option before the introduction of OTA technology, and they’re still used in certain situations.

Typically, manual updates involve downloading the new firmware to storage such as a hard disk or flash drive and then connecting it with your device. The device usually needs to be taken apart, reprogrammed, reassembled, and restored so it can function again.

### I.2.1 Pros of manual firmware updates

The most obvious benefit of manual firmware updates is that nothing can be rolled out before the end user is ready. This can be crucial in situations where the device must operate in a stable and predictable manner at all times.

Manual firmware updates also have advantages in certain situations. For example, when a device refuses to turn on, or you have a weak or nonexistent Internet connection, a manual update is your only option.

### I.2.2 Cons of manual firmware updates

The most obvious disadvantage of manual firmware updates is having to retrieve and update the device yourself. This will require finding a maintenance window when the impact of downtime won’t be as great.

In addition, manual firmware updates aren’t scalable, meaning that the devices may be difficult or impossible to update once they’re out of your hands. If the update is too complicated for end users to do themselves, they may have to bring their devices to you in order to get serviced.

## I.3 OTA Firmware Updates:

Over-the-air (OTA) firmware updates are a vital component of any IoT system. Over-the-air firmware updates refers to the practice of remotely updating the code on an embedded device.

*Figure 1:Particle's all-in-one IoT platform offers industry leading OTA update capabilities*

The value of incorporating OTA update capabilities into a connected product include:

- The ability to add new software features to a product after a device has been deployed in the field to improve functionality over time
- The opportunity to rapidly respond to bugs and security vulnerabilities without the need for physical recalls of devices or truck rolls
- Ensuring embedded developers can quickly prototype and seamlessly roll out new versions of device firmware, speeding up innovation cycles

### I.3.1 OTA Updates with Particle :

Particle's platform is uniquely positioned to provide industry-leading OTA update capabilities for embedded devices. Using Particle for OTA updates give you the following benefits:

### *I.3.1.1 A Complete Solution:*

A successful OTA update requires complex coordination between IoT hardware, device firmware, network connectivity, and an IoT device cloud. Trust us, this is a very hard problem to solve correctly.

Lucky for you and your team, this is where Particle's fully integrated IoT platform shines. These four parts of the "IoT stack" are core pillars of Particle's platform, and enable you to seamlessly deliver OTA updates to devices at any scale.


- Hardware: All Particle dev kits and systems-on-a-module (SoMs) support OTA updates right out of the box. Device and cloud-side safeguards are combined to ensure that devices only receive compatible firmware that can run on its unique hardware platform.
- Device OS: Our embedded operating system which runs on all Particle devices, Device OS, is architected to reliably and resiliently accept firmware updates from the Device Cloud.
- Connectivity: OTA updates work seamlessly across Particle's suite of connectivity offerings. Devices connecting over Wi-Fi and Cellular can receive firmware updates using our OTA feature set.
- Device Cloud: The Device Cloud tightly integrates with Device OS to safely and effectively deliver OTA updates. It also provides a variety of flexible management tools to choose how OTA updates should be applied based on your needs.

In fact, one must have visibility and control over all four of these components of an IoT system to successfully deliver an OTA update. Without any one of these parts, or the integration of these parts, an OTA update is not possible.

Other IoT platforms may market an OTA feature, but in reality only provide a small sliver of the functionality required perform a complete, reliable, and secure update — leaving your

team to piece together a bespoke solution that distracts them from spending valuable time on the features that make your IoT product unique.

#### *I.3.1.2 Reliable and Resilient*

Sending an OTA update is arguably one of the riskiest actions you can take on a connected device. Mishandling OTA updates could at a minimum cause temporary disruption, or at worst force the device into an unrecoverable state.

Particle's OTA updates have been built from the ground up to be reliable and resilient to allow your team to deploy quickly while keeping your fleet functioning healthily:

- ***Atomic updates:*** A Particle device will only run a new version of firmware it has received after empirically verifying that it has successfully received the entire file from the cloud. Additionally, your firmware application can be updated independently of the Device OS.
- ***Automatic rollbacks:*** If for some reason an OTA is interrupted (like a disruption in connectivity or a device losing power), the device will fail gracefully by automatically reverting to the previous version of working firmware.
- ***Minimal disruption:*** A Particle device continues to run its current version of firmware while it receives an OTA update. After a brief reset, the device seamlessly begins running the new firmware. If at this point the version of Device OS running on the device is older than the minimum version of Device OS to run the user application, the device will enter safe mode and download the required Device OS. Once this is complete, the device will reset and begin running the new firmware on the updated Device OS.
- ***Application and Device OS version management:*** Particle's OTA capabilities make it easy to manage both the firmware applications your team writes, and the low-level Device OS that Particle manages. This lets you send updates to your device logic, but also enables you to stay up-to-date with the latest features and improvements in Device OS. You are in complete control of your adoption of Device OS versions.

- ***Support for updates for sleeping devices:*** Some applications use sleep modes to reduce power consumption on battery-powered devices. OTA updates can be released and applied when the device wakes up from sleep mode automatically.

#### *I.3.1.3 Secure*

Given the risk associated with an OTA update, it is especially critical that these updates are performed securely and safely.

***Encrypted communications:*** All messages between Particle devices and the Device Cloud are always encrypted, including firmware files. This eliminates potential man-in-the-middle attacks that seek to send fraudulent firmware to the device.

***Sender verification:*** Every OTA update attempt is first verified to ensure the identity of the sender is an approved device manager.

#### *I.3.1.4 Scalable*

Particle provides your team with tools as part of the Device Cloud that give you full control and flexibility in how OTA updates are delivered to your fleet. Particle offers different OTA tools that are appropriate at different phases of fleet growth.

- ***Prototyping***

When prototyping, you want to enable your development team with the ability to iterate quickly to provide viability and value in as little time as possible.

Particle's single device OTA functions help enable your embedded development team to rapidly prototype and innovate. OTA updates can be sent with a click of a button in our IDEs (available both in Workbench and our Web IDE), or via our developer-approved REST API.

- ***Moving to Production***

As you begin to deploy large numbers of devices, it is imperative to have the ability to safely batch OTA updates to many devices at one time. This is what allows you to roll out new software features, fix bugs, or patch security holes across your fleet.

For this purpose, Particle offers fleet-wide OTA updates. A variety of tools are available in the Particle Console to apply fleet-wide updates without sacrificing fine-grained control.

**Firmware releases:** Cut a version of firmware that will be automatically sent to your fleet, with sensible safeguards to roll out an update responsibly and monitor fleet health for changes.

**Release by device groups:** Target a subset of your fleet to receive a new version of firmware. This is useful when your product has variants that require different device behaviors, or when wanting to phase out a single release over time to reduce risk.

**Intelligent firmware releases:** Instead of waiting for devices to re-connect to receive an update, push a fleet-wide update as quickly as possible while still allowing the device control over the appropriate time to update.

#### *I.3.1.5 The firmware stack*

- ***Device OS***

The Particle Device OS provides the foundation for your application. It provides many features including:

Core networking and communication APIs
Standard C and C++ libraries
Hardware abstraction layer, allowing your code to be run on a variety of devices
Real-time operating system (RTOS)

The Device OS can be updated over-the-air as needed, however you are always in control over updates. You can choose whether you want to upgrade or not for any release.

- ***Application firmware***

Application firmware provides the features for your specific use case and support for any additional peripheral hardware like sensors, actuators, etc..

Particle application firmware is programmed in C/C++ using industry-standard gcc C++11 compilers.

Furthermore, to facilitate moving from prototyping to production, an Arduino compatibility layer allows the use of hardware libraries for sensors, displays, etc. that were designed for Arduino to be easily used on Particle devices.

Application firmware is designed to make it easy to compile your firmware with little or no modification when switching between devices. You can start prototyping with the Boron in prototyping breadboards and move to mass-production on the B series SoM (system on a module).

Splitting the firmware between application part (up to 128 Kbytes) and system parts (384K to 656K bytes, depending on the device) allows for faster updates of application firmware, with reduced data usage when only updating application firmware.

# chapter II : Presentation of the company

## II.1 Introduction :

In this chapter, we will be presenting the host company for the summer internship.

## II.2 Internship context :

### II.2.1 presentation of the host organization

A high-tech company with global operations and a consolidated turnover of several billion euros is the SagemCom group.

It was formed in many nations, was founded in 2008, and is currently ranked third in Europe for electronics defense and security. It is also the second French telecoms organization.

It is a large global industrial corporation with factories and R&D centers that work in the areas of new technologies (decoders, printing terminals, multimedia, digital TV, etc.). The whole planet is home to its subsidiaries.

Sagemcom, the largest stakeholder in the financial firm American Carlyle, has its headquarters in Rueil-Malmaison, France.

### II.2.2 Sagemcom

The group Sagem, established in 1925, is a multinational group of high technologies in the communication field. Since the merger Sagem-Snecma and the creation of the SAFRAN Group, the Sagem company, did not exist anymore under this name.

Sagem is from now on a mark detained by SAFRAN which grants its use either under composite shape for names of companies (Sagem Defense-Security, Sagem Industries, Sagem Wireless, Sagem Communications), as a trademark. From 2008, the branch communications of the ex Sagem is a deprived independent group , named Sagemcom, independent from the SAFRAN group.

Thus, Sagemcom is a high-technology international group, of French origin, specialist of the broadband (broadband solutions and Digital Set-top Boxes), Energy Telecom and Internet of Things.


*Figure 2:logo Sagemcom*

### II.2.3 Sagemcom Software & Technologies

SagemCom Software Technologies is a subsidiary of Sagemcom situated in Megrine, Tunisia.

It is a Research and Development (RD) facility that focuses on emerging technologies, notably telecommunications, information processing and digital transmission, and development.

As a result, it creates and develops a wide range of goods for the broadband communication sector, such as decoders, printers and multimedia terminals, and digital television.

The OHS unit is split into five directorates: operations management, human resources, information systems, quality, general services, and four poles: ETL, AVS, and BBS.

• ***Pôle ETL (Energy and Telecom Activities):*** This pôle's objective is to provide energy measurement systems (electrical converters) to energy distributors.

• ***Pôle AVS (Audio Video Solutions):*** This pôle's objective is to design, develop, and validate the software embedded in digital set-top boxes and IPTV (Internet Protocol).

• ***Pôle BBS (BroadBand Solutions):*** This pôle is in charge of developing and validating residential software embedded products.

# chapter III : Project presentation

## III.1 Literature review

This review will be dedicated to presenting the recent development stages and state of technologies, techniques and devices related to this project. Featured keywords will be included such as residential gateways, firmware

### III.1.1 Residential gateways

***Firmware***

Firmware exists because there are hundreds of different hardware modules (such as bluetooth, wifi, cameras etc.) produced by different manufacturers that all behave slightly differently. It would be very challenging for an operating system to be able to communicate directly with such a disparate array of hardware, so instead they communicate with the firmware which exposes a known method or action that the operating system understands.

In other words, the firmware acts as the translator between hardware and software (most of the time, an operating system). It receives the software instructions and further passes them on to the hardware components.

## III.2 Project overview

### III.2.1 project context

This work was realized in the context of a summer internship project for the first year as a computer science Engineer student in the National Superieur School of Engineers of Tunis (ENSIT).

This project started on the 7th of July 2022 and ended on the 31 th of august 2022 and was hosted by Sagemcom Software & Technologies (SS&T) which is a subsidiary of the multinational group Sagemcom. The project is entitled OTA Firmware Upgrade.

### III.2.2 project description

***Clients description***

This document was brought by a client of Sagemcom and it features the implementation of a solution for remote firmware upgrade.In other words, an OTA firmware upgrade of their gateway.

the gateway will be able to consult a specific URL for an xml file containing the version of the upgrade and the path to the image of the software upgrade.

Once the listed version is more advanced than the current installed version, it will be installed at the scheduled time in case no complications arise.

***OTA firmware upgrade advantages***

The value of incorporating On The Air update capabilities into a connected product include:

- The ability to add new software features to a product after a device has been deployed in the field to improve functionality over time
- The opportunity to rapidly respond to bugs and security vulnerabilities without the need for physical recalls of devices or truck rolls
- Ensuring embedded developers can quickly prototype and seamlessly roll out new versions of device firmware, speeding up innovation cycles
- The improved product behavior, even after the device is in the end user’s hands.
- You can even try out experimental features with A/B testing, sending different versions to different groups of users.

- OTA firmware updates also generally save on cost.
- You can manage the firmware across your fleet of devices from a seamless, unified interface.

#### III.2.3 Project statement

This service should meet the customer requirements:

MGT-032 the RG will download and install the firmware at the time specified in the XML file.

Enable/Disable this feature, URL of XML file and frequency of checking must be configurable via the Datastore.

by Default must be : enabled, http://fwupdate.frontier.com, 24 hours.

this feature is specifically only for the client's needs  standards, the end solution should leverage available communication and security standards authenticate a downloaded image is from a trusted source only authenticated parties may alter data (i.e.parameters) stored in the embedded system.

### III.3 Description of the solution

To tend to the client's needs , Sagemcom software & technologies suggests adding to the gateway's already existing services a new service dedicated to Firmware Upgrades.

Assuming that every step goes flawlessly ,Enabling this service will give access to upgrading the firmware when a newer update is presented in the form of an xml file at a specific URL address.

the URL will be consulted once a day by default, when a file meets the conditions ,the download of the version takes place then the over-writing of the flash chip will be scheduled just as the time in the file states.

When completed, the gateway will reboot automatically.

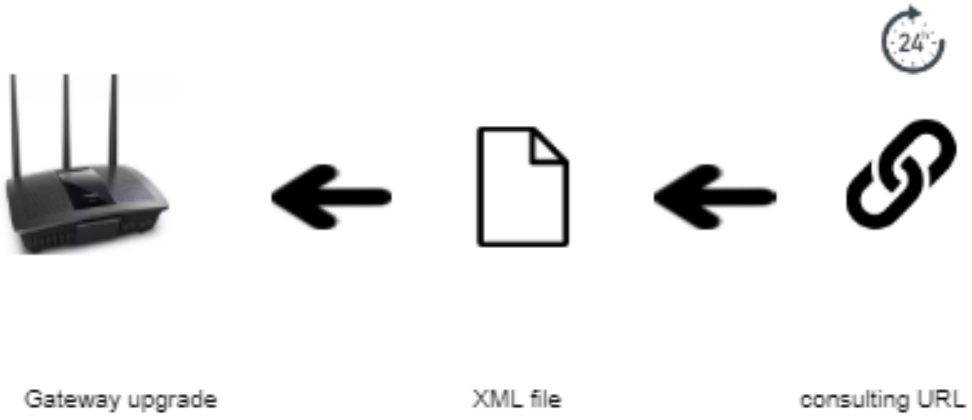


*Figure 3: schema solution*

## III.4 OTA firmware update design

### III.4.1 Steps of OTA fu

Our solution for the OTA firmware process has the following steps:

***Step 1:*** The files are uploaded to a server and a URL is obtained to point to the binary file.

***Step 2:*** A notification is sent to the IoT device (the gateway) that new firmware is available for download, and the URL for the firmware is provided.

***Step 3:*** IoT device will download the firmware file using HTTPS.

***Step 4:*** IoT device will check the authenticity of the files and will start to load in the newly downloaded firmware.

### III.4.2 Modules intervening with OTA firmware update

**firewall for Security Measures**

Having a consistent and scalable process for OTA firmware updates is especially important when the security of devices in the field is considered.gateway devices must also be considered and taken into account. Firmware upgrades must be delivered in a timely and systematic manner, otherwise devices in the fleet will be open to software vulnerabilities which increase the threat vectors for these devices and be exposed to malicious attackers to be penetrated.

A secure firmware update mechanism is developed to help devices authenticate the source of received firmware and verify the integrity of the received firmware.

We will discuss the key elements of a secure OTA process:

- The IoT device should securely download the firmware image from the cloud after connecting using HTTPS (TLS).
- IoT devices should authenticate the new firmware files to ensure that it was made by a trusted source. Using code signing techniques, once code is written and uploaded, the file will be signed.
- Then the IoT device will download and confirm the signed file if all keys match and then load in the new firmware. The device's bootloader should also perform additional steps to ensure that the image that is being loaded into is signed appropriately just in case an injected attack happens.

A firewall is a network security device, either hardware or software-based, which monitors all incoming and outgoing traffic and based on a defined set of security rules it accepts, rejects or drops that specific traffic.

Accept : allow the traffic

Reject : block the traffic but reply with an "unreachable error"

Drop : block the traffic with no reply

A firewall establishes a barrier between secured internal networks and outside untrusted network, such as the Internet.

Firewall matches the network traffic against the rule set defined in its table. Once the rule is matched, associate action is applied to the network traffic. From the perspective of a server, network traffic can be either outgoing or incoming.in this project we ll be working with Incoming traffic .

Most traffic which reaches on the firewall is one of these three major Transport Layer protocols- TCP, UDP or ICMP. All these types have a source address and destination address. Also, TCP and UDP have port numbers. ICMP uses type code instead of port number which identifies the purpose of that packet.

**NTP for synchronizing activities :**

Network Time Protocol (NTP) is a protocol that helps the computer's clock times to be synchronized in a network. This protocol is an application protocol that is responsible for the synchronization of hosts on a TCP/IP network. NTP was developed by David Mills in 1981 at the University of Delaware. This is required in a communication mechanism so that a seamless connection is present between the computers.

Features of NTP :

Some features of NTP are –

- NTP servers have access to highly precise atomic clocks and GPU clocks
- It uses Coordinated Universal Time (UTC) to synchronize CPU clock time.
- Avoids even having a fraction of vulnerabilities in information exchange communication.
- Provides consistent timekeeping for file servers
- Applications of NTP :
- Used in a production system where the live sound is recorded.
- Used in the development of Broadcasting infrastructures.

Advantages of NTP :

- It provides internet synchronization between the devices.
- It provides enhanced security within the premises.
- It is used in authentication systems like Kerberos.
- It provides network acceleration which helps in troubleshooting problems.
- Used in file systems that are difficult in network synchronization.

Disadvantages of NTP :

- When the servers are down the sync time is affected across a running communication.

Servers are prone to error due to various time zones and conflict may occur.

Minimal reduction of time accuracy.

When NTP packets are increased synchronization is conflicted.

Manipulation can be done in synchronization.

**Wan**

Call backs

onWanUp { check if (feature enable) and (URL not empty) call start_fu() }

onWanDown { if (feature enable) stop timer }

It is unquestionable that a WAN connection establishment is necessary for OTA firmware upgrade since the file is supposed to be downloaded from the internet. The update could only take place if the gateway has access to the wifi

## III.5 Algorithm conditions

### III.5.1 firmware upgrade start conditions:

As we got through the algorithm study, it is undeniable that a power outage could occur at any stage of the operation or the WAN could go down failing the download or the search or anything that relies on the internet.

The algorithm should also consider the fact that the file could be unreadable or invalid

taking into consideration wan.

deciding which variables could help make the program more optimal and efficient.

The firmware upgrade could only take place, firstly, if the feature is enabled in the gateway configuration

# Conclusion:

While they're highly convenient for both IoT manufacturers and users, OTA firmware updates do have some drawbacks. For one, they require the device to be working and connected to the Internet. In addition, the device needs to be shut down and restarted in order to complete the update, which can be a problem for scheduling the upgrade.

OTA updates also fail at a higher rate than manual updates. In general, getting software to update itself is a difficult problem. In some cases the update files will fail to download to the target device. Another example is if during the update the device crashes or is powered off.

Most IoT products will benefit from an OTA update delivery system, which has become the norm for the vast majority of devices. However, there are certain situations in which manual firmware updates would be a better fit for your IoT device.

Consider factors such as:

Fault tolerance: Technologies such as medical devices and spaceships require high fault tolerance, due to the catastrophic impact if they fail. If this describes your product, then manual updates are probably better because they give you more control over the update process.

Reverting to previous versions: It's easy to add features and make changes with an OTA update system. If this new code contains a bug, however, then it's not always easy to revert to a previous version of the firmware. You may have to release yet another update in order to fix the issue, and of course that update may also fail.

Control: Who should control whether to release an update for general availability to all devices? Who should control whether to accept the update? When and how should devices be updated?

Scale: Generally speaking, the more devices you need to update, the more benefit you get from OTA. Tens of devices could be updated manually, while hundreds or more can be very challenging.

Compatibility: Do you expect your IoT product to grow such that its firmware must be updated in order to work with the rest of your software? Such a forced upgrade could be impractical with manual updates.