University of Liège
Faculty of Applied Sciences

UX and Security improvements on connected glasses

by

FINTA Ionut Andrei

Master's Thesis submitted in fulfillment
of the requirements for a degree of
Master of Science in Computer Science
with professional focus on Computer systems security

Research and Development project owners:
MALHERBE A., JENCHENNE P., DESSAMBRE N.

Academic supervisor:
Pr. BOIGELOT Bernard

Academic year
2022 – 2023

University of Liège - Faculty of Applied Science
Title: UX and Security improvements on connected glasses

Author: FINTA Ionut Andrei
Master in Science in Computer Science
with professional focus in Computer systems security
Project owners: MALHERBE A., JENCHENNE P., DESSAMBRE N.
Academic supervisor: Pr. BOIGELOT Bernard
Academic year: 2022 – 2023

Abstract

Get Your Way, a Belgian start-up founded in 2020, develops a new kind of connected glasses in the domain of assisted Reality (aR). This Master's thesis aims to advance this project in two domains: User eXperience (UX) and information security.

First, the UX is improved by enabling screen orientation change when the monocular glass is reversed on the other eye. Then, the autonomy of the battery is extended by implementing a dynamic CPU frequency strategy.

Second, the security of the product is improved beginning with some suggestions for improving the security of the organization itself. Then, we address the problem of securing the Bluetooth functionality of the device.

Finally, some consistent bugs were found in the hardware design and current implementation of the Proof-Of-Concept (POC) firmware. A first issue was the inversion of two traces on the main board, and it was solved by the implementation of a custom I^2C driver. The second issue was the calibration of the LCD display and it was solved empirically by trial-and-error.

Keywords: embedded, glasses, Bluetooth Low-Energy, GPIO, I^2C, autonomy

Preface

This report is the result of hundreds of work hours mainly spread across the first semester of 2023 with Get Your Way. This start-up has for plan to become a big player in the assisted reality domain. The context and the product they develop will be explained in the first two chapters. Then, three chapters will cover the three main domains of work in which this thesis contributed. Finally, a conclusion will be drawn in the last chapter.

The three domains covered by this work are: User eXperience (UX), a bug solving in the embedded systems communication protocol I^2C and information security. It is difficult to work on so vast domains together in a single thesis, that is why the result may seem too abstract for experts. However, my personal desire has always been to have broad knowledge rather than very specific expertise, which is why I chose to do so.

Acknowledgements

This report would not be without all the persons that have enlightened my way and who deserves my warmest thanks. Especially, I would like to thank Pr Bernard Boigelot who performed a great follow up of this project and serious reviews of my work. I am also grateful to the founders of Get Your Way, Antoine Malherbe, Nicolas Dessambre and Pierre Jenchenne who not only supported me in my work and answered all my questions, but also allowed me to have deep insights into the world of start-up companies, entrepreneurship and product development. Taking part into technological events such as TEDx Brussels or Technocité was more than inspiring and allowed me to meet interesting people.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

I^2C	Inter-Integrated Circuit
API	Application Programming Interface
ARM	Advanced RISC Machines
BLE	Bluetooth Low Energy
DRP	Dynamically Re-configurable Processor
FreeRTOS	Free Real-Time Operating System
FSM	Finite State Machine
GDB	GNU DebuGger
GPIO	General-Purpose Input-Output
GUI	Graphical User Interface
GYW	Get Your Way
HMD	Head-Mounted Display
HMI	Human-Machine Interaction
IDE	Integrated Development Environment
IOT	Internet Of Things
ISR	Interrupt Service Routine
JW	Just Works
LCD	Liquid Cristal Display
LP	Low Power
MCU	Micro Controller Unit
MITM	Man-In-The-Middle
MVP	Minimum Viable Product
OCR	Optical Character Recognition
OOB	Out Of Band
OSTM	OS Timer Manager
OTA	Over The Air
PCB	Printed Circuit Board
PDF	Portable Document Format
PE	Passkey Entry
RGA	Renesas Graphics Architecture
RISC	Reduced Instruction Set Computer
SMEs	Small and medium-sized enterprises
SPI	Serial Peripheral Interface
TAM	Technology Acceptance Model
UML	Unified Modelling Language
UX	User eXperience
VDC	Video Display Controller
VGA	Video Graphics Array
WVGA	Wide VGA

Chapter 1

Introduction

1.1 Get Your Way

1.1.1 The company

Get Your Way is a Belgian start-up founded by 3 entrepreneurs during their studies at the University of Liège. Their project project is to build the simplest-to-use assisted reality device with the highest proposed value possible at a reasonable price. Successful achievement of this undertaking entails significant research and development efforts, as well as proficient management of various business tasks, including for example finances, sales and planning.

The idea is not totally new and a growing market already exists for such devices [8] at the time of writing. However, pioneer products in the domain are not widely adopted because of multiple factors, the most important of which being their perceived usefulness [6]. This goes in pair with the Technology Acceptance Model (TAM) proposed by F. D. Davis [3] in 1985 which states that the perceived usefulness and ease of use of a technology are its main success determining factors. Figure 1.1 shows the limit of smart glasses in the TAM.

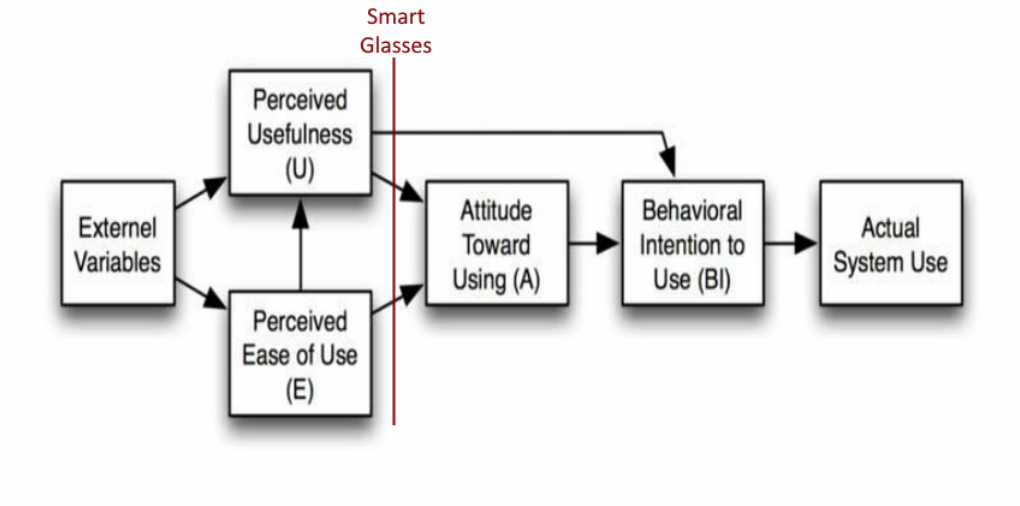


Figure 1.1: Smart glasses in the TAM. (source: [3])

1.1.2 The product

The product created by GYW is called the aRdent smart glasses. Thanks to the work performed during this thesis, it has now achieved the state of Minimum Viable Product (MVP), but was still a prototype at the beginning of this work. It is presented as a multi-purpose Head-Mounted Display (HMD) [8] that can be used to provide information in the eye-sight of the user with little to no discomfort. Figure 1.2 shows the glasses in their final form.

Figure 1.2: aRdent smart glasses overview. (source: getyourway.be)

The main requirements of the final product are listed below.

- The product must be comfortable to allow a professional to wear it as long as required while performing computer-assisted tasks. For example, a delivery person should be able to wear the glasses nearly all the working day.
- Being a wearable connected device requires the final product to have a small footprint and to be as efficient as possible. A small battery should power the device up during at least 4 consecutive hours without interruption. One main advantage of aRdent is that it allows users to quickly perform a battery swap when the device battery is discharged.
- The device must offer a user-friendly experience with a simple Human-Machine Interaction (HMI) allowing it to be widely adopted by professionals.

Finally, a careful reader would have by now noticed that numerous terms can be used to qualify this kind of product: Smart glasses, connected glasses, data glasses, and HMD, the last one being the most accurate as aRdent glasses are not "smart". Indeed, the functionality of the device is reduced to displaying texts and images in order to maximize the battery life.

Moreover, the user will not interact directly with the glasses but with any device that is paired to them via Bluetooth. This improves the ease of use and should result in a better adoption of the technology. For example, the aRdent display can be managed by a smartphone, tablet or computer, but also by a simple keypad also developed in-house as shown in Figure 1.3.

Figure 1.3: aRdent glasses with their keypad. (source: getyourway.be)

1.1.3 Target use cases

GYW's target audience is businesses that want to invest in technology to improve the efficiency and the comfort of their operators. Although the objective of the company is to create a general-purpose device, several use cases have been concretely identified and defined for the proof of concept. The common denominator of those use cases is a work that consists in the completion of tasks that are generally well-structured into several steps and require little interaction from the operator: checklists, number input, forward and backward navigation.

A first example where aRdent can be used is in the context of quality control processes where operators have to follow well-defined checklists and input quantitative or qualitative information at each step. Thanks to aRdent, the hands of the operator are free and there is no more need to type the data manually in a spreadsheet when the information is written by the operator on paper[1].

As stated above, other use cases are slightly different but comply with the common denominator: Inventory operators, laboratory workers, manufacturing procedures, etc. A simple backward and forward navigation application can be useful in the context of museums and expositions: aRdent can be used to provide visitors with textual tours instead of audio if they have hearing disabilities.

1.1.4 aRdent project status

At the beginning of this thesis, GYW had already made a demonstration prototype of the connected glasses with an early firmware in order to show the potential of the product to customers and investors.

This early firmware provides basic functionalities such as Bluetooth communication with a mobile app, displaying static images hard-coded in the firmware binary, and showing a list of instructions from JSON data received over Bluetooth. Moreover, a

[1]This could also be done via Optical Character Recognition (OCR), but this also requires manual intervention or some sort of investment in OCR automation.

contractor company, Quimesis[2], has enhanced the firmware during the first months of this thesis project, allowing Over The Air (OTA) updates via Bluetooth, the use of a FAT filesystem to store PNG/JPEG-encoded images instead of hard-coding them in the source code of the firmware, and methods to interact with this system via Bluetooth.

aRdent is little-by-little coming to market and some improvements can still be made in various topics. More particularly, from a computer science perspective, the glasses can benefit from several User eXperience (UX) refinements as well as a comprehensive assessment of the device security, eventually leading to further developments in order to make the firmware and/or the applications safer.

1.2 Project statement

The initial project statement proposed by GYW was mainly focused on integrating the 3D accelerometer and the gyroscope module in the firmware.

First, GYW suggested a workflow for this work which is listed below:

- Get the signal of the accelerometer via an I^2C interface.
- Read the accelerometer data (events and raw data).
- Generate interruptions for events.
- Transmit this information via Bluetooth to an external device.

Then, the statement also contained a list of events that could be interesting to implement such as performing a screen rotation when the device rotates, detect when the user taps on the device, shut the device down when long periods of inactivity, or detect accidents.

In addition to these initial objectives, security contributions to the project will also be included as part of this Master's Thesis. This is in line with the professional focus on computer systems security stated in the title. This work will therefore also consist in assessing the star-tup's information security, eventually suggesting improvements. Finally, the device security itself should be correctly assessed, risks relative to its usage in a corporate environment should be identified and response measures should be taken.

[2]For more information, consult `https://www.quimesis.be`.

1.3 Objectives and motivations

First and foremost, the main objective of this work is to propose a solution to the aforementioned problems using a methodical approach that will be defined in the next section.

Second, an important aspect of completing a Master's thesis is to learn new things and gain expertise in a given field. Therefore, a motivation for this work is to acquire and convey knowledge in the field of embedded systems and the related ones such as security and UX.

Finally, a pursued objective is to add value to the aRdent product, impacting positively the work of the company and its customers. This can be done for example by improving the developers documentation of the firmware which overall improves GYW's knowledge.

1.4 Methodology

This section describes the project management methodology that was used for these UX and security improvements. The initial choice was to proceed with Agile Scrum, but it was later changed to the waterfall method due to some challenges and limitations such as the difficulty to estimate the time required to solve subproblems. Moreover, Scrum is particularly suited for team working whereas this work was mainly carried by myself alone.

The waterfall method used for this project included the steps listed and described in the list below.

- **Problem description and requirements**: The problem is correctly identified and the requirements of the solution are listed.
- **Research**: The possible solutions are enumerated and the solution that fits the best this project is chosen.
- **Analysis**: If applicable, the chosen solution is modelled and/or specified.
- **Implementation**: The chosen solution is implemented.
- **Check**: The implementation is tested and we ensure that it meets the requirements listed in the first step. The tests are carried out manually as there is no test framework in place.
- **Commit**: The solution is reviewed by the project owners, and once accepted it is merged into the main project.

1.5 Organization of this document

After this introduction, the document will be divided into 4 main chapters.

First, a chapter will give an extended description of the aRdent product, the available development environment and documentation that allows to understand the following chapters.

Second, two User eXperience improvements will be brought in a chapter. First, we will solve the screen rotation problem. Then, we will explore the extension of the device's battery life.

Then, a problem related to a communication bus was unveiled while working on the screen rotation. The problem was located in the hardware design of the product and it will be addressed in a following chapter.

After that, a chapter will discuss the security aspects of Get Your Way and aRdent.

Finally, the results of the development will be presented in a conclusion which also includes a retrospective work and future perspectives.

Chapter 2

Product description

In this chapter, the aRdent product will be extensively described in all its forms. First, the hardware form will be set out in the first section. Then, we will explain the firmware followed by the development environment offered by Renesas, the manufacturer of the main chip of this product, and the various libraries offered by the other chips manufacturers.

2.1 aRdent hardware

As stated earlier in the introduction, one point of differentiation between aRdent and its competitors is its simplicity. This is already relatable in the hardware design of the product. For example, we can see in Figure 2.1 that the HMD is not bloated with components such as a camera or a microphone which most of the competition have.

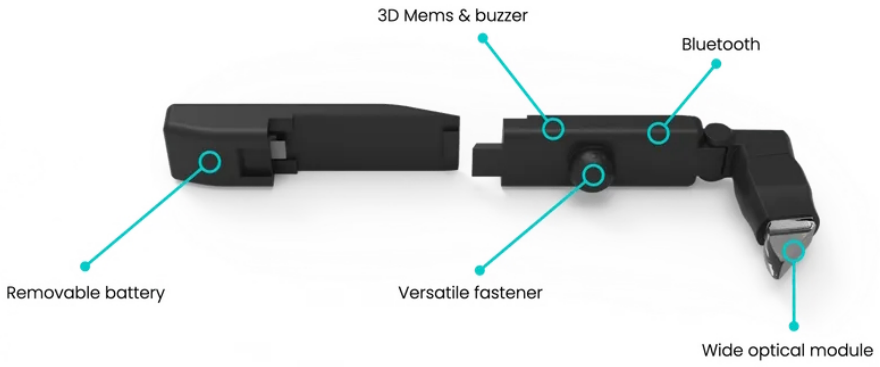


Figure 2.1: Main exploded view of aRdent components. (source: getyourway.be)

The first element on the left is the removable battery module. It is a classic Lithium-ion battery with a 450mAh capacity, and there is no sophisticated system to allow precise State of Charge (SoC) such as Coulomb-counting [18]. Indeed, when the battery is discharged, the device will simply turn off and this is not a concern since there is no business-oriented state that is persisted on the HMD. It is normally desired to do a battery swap during a break, but in the event that a battery runs out of current earlier, an application can automatically reconnect to the device and re-send the screen composition when it is back up and running.

On the right side of Figure 2.1, the part containing the main board and the optical module are visible.

In the following subsections, we will describe all the important components, that are schematized in Figure 2.2 below.

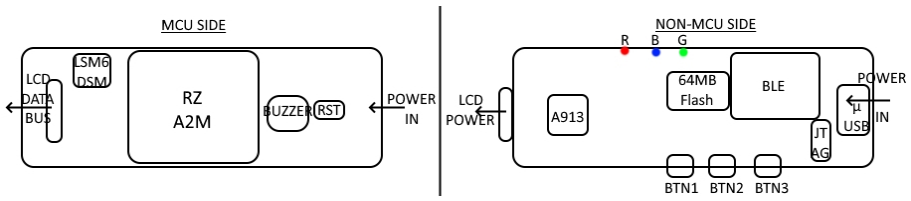


Figure 2.2: Schematic view of the main components on the main board.

2.1.1 Computing

aRdent is powered by a Renesas RZ/A2M[1] Micro Controller Unit (MCU). This low-power high performance microprocessor has one ARM[2] Cortex-A9 core that can be clocked up to 528MHz. Additionally, it has a Dynamically Re-configurable Processor (DRP) that enables high efficiency image processing or AI acceleration. The package of the MCU also contains 4MB of Static Random Access Memory (SRAM)[3] that is not extensible and there is an external Macronix 64MB flash memory for the firmware and application data. [12]

2.1.2 Connectivity

aRdent is equipped with a Bluetooth Low Energy (BLE) module which is the only interaction medium for users in production. This module is the ST Microelectronics BlueNRG-M0A, a model that supports the version 4.2 of the BLE protocol and has no dedicated computing core, RAM or flash memory. [14] The absence of computing capabilities makes it rely on the MCU for security keys management. [15] The communication between the MCU and the BlueNRG module is carried through a Serial Peripheral Interface (SPI).

The other possibility to interact with the device is through its JTAG interface. This is only for internal development and debugging as it required the main board to be accessible and wired unlike a solution based on Bluetooth.

2.1.3 Accelerometer and gyroscope module

The 3D accelerometer and gyroscope module that is present on the board is the ST Microelectronics' LSM6DSM. The communication with the MCU is made through a dedicated I^2C bus on which there is no other device. The MCU is the only master and the LSM6DSM the only slave connected to this bus. Additionally, the module has 2 configurable interrupt pins that are both connected to the appropriate pin of the MCU. [9]

[1]The precise part number is R7S921052VCBG, a variant that has the Dynamically Reconfigurable Processor, 272 pins and no hardware accelerated security features.

[2]Advanced RISC Machines

[3]A type of RAM that does not require periodic refreshes.

The configurable interrupts that are listed in the datasheet of this peripheral [9] are the following:

- Free-fall
- Wake-up
- 6D Orientation
- Click and double-click sensing
- (In-)Activity Recognition

2.1.4 Display

The HMD uses a Kopin Golden Pearl™display module controlled by a A913 driver which has a Full Wide Video Graphics Array (FWVGA) resolution of 854x480 pixels. [7]

The link between the MCU and the display driver is a 16-bit progressive video parallel bus meaning that the LCD array is sent line by line over 16 channels in parallel carrying pixel data (i.e., 16-bits are used as follows: 5-bit red, 6-bit green and 5-bit blue components). Additionally, the MCU uses a different I^2C channel to configure the driver's registers on which the MCU is the only master and the A913 is the only slave. These registers permit to tweak the driver extensively. For example, it allows to change the resolution, the brightness, the gamma value, etc.

2.1.5 Other components

There are several other components that were initially planned in the design of the PCB and are finally not exposed by the casing in order to remain waterproof. These components are: Three buttons, a reset button, 3 leds (green, red and blue), and the buzzer. Even if they are not exposed, some components can be used. For example, the blue led is used as a Bluetooth indicator and can be seen through the casing, and the buzzer is audible even though there are no specific holes for it in the case.

2.2 aRdent firmware

The firmware is based on Free Real-Time Operating System (FreeRTOS). It is composed of a bootloader that checks for available updates over Bluetooth and performs the update if possible, and then boots the main OS.

There are only two tasks that are running continuously in the system. The first task initializes the peripherals then makes the green led blink if everything went successfully or the red led if something went wrong. The second task manages Bluetooth communication. It waits to be notified by the Interrupt Service Routine (ISR) that is triggered when a Bluetooth packet is ready to be treated and handles this packet.

The 64MB flash memory is partitioned as follows: 32 MB are used for the firmware and the eventual OTA updated firmware. Then, a FAT12 filesystem of 4MB has been added in the unused 32MB space, but this partition can be extended later.

The 4MB SRAM is partitioned as described in Firgure 2.3. The memory map shows that 16KB are used for the page table, 65KB for the stack, about 3MB for the chached RAM, about 1 MB for the uncached RAM, and 65KB for the "Segger RTT" which is a wired "high performance" communication protocol used for debugging (Source: segger.com).

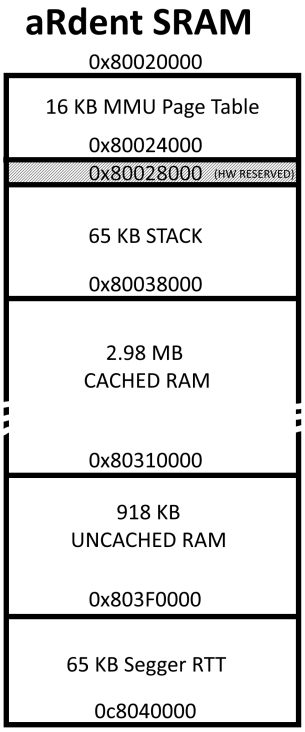


Figure 2.3: aRdent SRAM map.

2.3 Development environment and available documentation

In this section, we will cover the main tools provided by the MCU manufacturer, Renesas and what is provided by the accelerometer manufacturer. We will see that those tools have many advantages, but they also suffer from some disadvantages that the team needed to address.

2.3.1 Renesas

Renesas provides a complete Eclipse-based Integrated Development Environment (IDE), called "E2 Studio" containing, among other things, a plugin called "Smart Configurator". The Smart Configurator is a graphical interface allowing to configure the pins of the MCU and it automatically generates code with adequate drivers and

initialization allowing the use of these pins through standard ANSI[4] C functions *open*, *read*, *write*, *control*. A specific identifier is given to the *open* function to allow the right driver to handle the operations. For example, *open(DEVICE_IDENTIFIER "riic1", O_RW);* is used to get a handle for the first I^2C bus.

Another feature of E2 Studio is the complete integration with the GNU ARM embedded toolchain, which is even included in the installation process. It allows to compile, run and debug the code interacting only with GUI elements. The software is distributed for Windows™and Linux.

Renesas also provides multiple sample projects for their processor family, including example usage of their proprietary drivers covering all their functionalities. Each example project comes with proper documentation and explanations. In the same idea, each automatically generated driver by the Smart Configurator comes with a PDF document describing its role, the public API, and sometimes examples of use. This is very useful as it allows to quickly get hands on with the MCU. For more advanced considerations, the complete documentation of the chip with all the details is provided as a user manual of 3581 pages.

2.3.2 ST Microelectronics

For the LSM6DSM accelerometer, a complete library is given in what they call "platform-independent standard C"[5] on the GitHub repository[6] of the device with example codes for interacting with the module. For example, the complete code enabling the orientation interrupt is provided where only the communication (platform-specific) functions need to be re-implemented.

Furthermore, the BlueNRG module also comes with platform-independent libraries and documentation. But in this case, the library was already integrated in the aRdent firmware by a previous contractor company, IOT-D[7], that did not provide proper documentation of their work.

2.3.3 Drawbacks

A first drawback of E2 Studio is its performance. It has a slow launch time, freezes sometimes when changing perspective, launching a plugin or a build, and it does not handle large files such as the single file driver for the accelerometer very well, disabling for example auto-completion, syntactic coloration or other features for them.

Second, the Smart Configurator is a required tool to get up-to-date drivers for the

[4]American National Standards Institute

[5]This is ambiguous since C comes with multiple standards. We suppose they refer to the ANSI C standard. As the compilation succeeds with the GNU ARM GCC compiler that we use without any error, we do not investigate this further.

[6]Link: `https://github.com/STMicroelectronics/STMems_Standard_C_drivers/tree/master/lsm6dsm_STdC/examples`, Accessed on 2023-03-03

[7]Website: `https://www.iot-d.com/`.

MCU functionalities. However, it overwrites all the driver files each time a small change is made only for changing the version in the header comment of the file. This makes it necessary to ignore changes in the versioning software for those files to avoid noise changes in pull requests. Another example is that it always removes the constant priorities for FreeRTOS tasks that are set in a specific generated file.

Third, the debugger integration is not always functioning properly. For example, it regularly happens to have breakpoints that are never reached using the feature directly in E2 Studio. Although, these are in fact reached by the software and we can verify that using the command-line GDB.

2.3.4 Alternatives

To overcome the performance issue, Quimesis, the contractor who worked on the firmware during the beginning of the thesis, has set up a CMake project that allows compilation of the firmware without E2 Studio, and a Docker script to allow easy compilation from any platform. This solution was preferred and we used it as soon as we merged our developments in January 2023. However, E2 Studio remains necessary to obtain new drivers and/or configure the MCU pins.

Chapter 3

User experience improvements

3.1 Screen orientation

3.1.1 Problem statement

The display module of the smart glasses is mounted on a side of a headband making the device monocular. aRdent has been designed to accommodate persons with different ocular dominance [10]. Indeed, the headband allows the glasses to be easily turned over allowing the optical module to be in front of the right eye or the left eye. Then, the accelerometer/gyroscope module allows to read the orientation of the device and to trigger an interrupt when it changes. However, the current firmware does not allow for screen orientation selection and/or change. As a result, the device is permanently using the screen orientation suited for the right eye. Thus, it is necessary to develop a solution that allows for screen orientation selection and/or change.

There are multiple ways of achieving this: Either the screen orientation is chosen at boot time and can only be changed by rebooting the device, or the accelerometer generates an interrupt for rotating the screen when it detects an orientation change. This last solution is suggested in the initial project statement and it is the most relevant one. Indeed, it offers a more interactive experience as it does not require to remove the battery to change the orientation[1].

The interrupt solution being chosen, it is also important to ensure that we address the problem of undesired changes. For example, a mechanic that works under a car could eventually have his/her head at an angle that triggers the orientation change. Again, there are two solutions to explore and discuss:

- Play with the sensibility of the detection to make undesired detection difficult while still detecting what is desired with ease.
- Offer a way to lock the orientation, via Bluetooth for example. But, in this case, it would be easier to rely on Bluetooth to directly set the orientation not use the accelerometer.

It is not necessarily needed to make a choice and both solutions can be implemented in the same device. So, further in this section, the sensibility of the detection will be fine-tuned and the design of a Bluetooth command that will allow to lock and/or set the orientation will be commented. Then, we will explore the different ways the

[1] aRdent having no usable buttons, removing the battery is the only way to reboot the device.

display can be flipped over and implement the way that is the most suited for our application.

3.1.2 Interrupt generation and handling

Establishing communication with the LSM6DSM module is straightforward given the example code[2] provided by ST Microelectronics for generating an interrupt on the pin INT1. Indeed, the code was usable as-is in the aRdent firmware. A FreeRTOS task has been created to handle the orientation and an Interrupt Service Routine (ISR) uses FreeRTOS notifications to unlock the main loop of that handling task. In that main loop, we retrieve the orientation from the device interrupt flags and perform a screen rotation accordingly.

The interrupt is detected on falling edge and has a priority of 27 and the handling task has been given a priority of 8. These priorities are higher than the OS main task which have a priority of 6, and only blinks an activity led after having initialized the system. They are also higher than the Bluetooth handling task and interrupt which also have been given a priority of 6. First, because the interrupt task is quick and should preempt any other task as we want a good response time. Then, because we do not want the task performing a screen rotation to be preempted by the blinking led task or a received Bluetooth packet. Finally, the system is not loaded at all and only a few tasks run on it so these values are not so important. Indeed, it rarely happens to have a Bluetooth packet arriving at the same time than the screen orientation is changed[3].

Detection refinement

As discussed in the introduction, small changes are made to the example code in order to limit false positives. One first change consists in enabling the 4D mode bit of the interrupt to ignore the orientation changes around the Z axis as only the X axis orientation matters in this situation: When the X axis is up, the optical module is on the right eye, and when it is down, the optical module is on the left eye. Figure 3.1 clearly shows the 3 axes of the accelerometer in the mounted product.

By default, an interrupt is generated by the accelerometer when it rotates by more than 80°around an axis (X or Y). Given that it corresponds to the maximum flexion of the cervical spine [16], this default parameter may be too restrictive. This threshold can alternatively be configured to 70, 60 or 50°. It was reduced to 60°because we observed situations were the orientation change interrupt did not trigger when it should have; for example, when the device is powered up in a position near the usage position and the angle does not change enough to trigger an orientation change letting the display in the wrong orientation.

[2]Link: `https://github.com/STMicroelectronics/STMems_Standard_C_drivers/tree/master/lsm6dsm_STdC/examples`, Accessed on 2023-03-03

[3]This would mean that the user changes the glasses orientation and interact with the paired device at the same time which is not easily feasible.

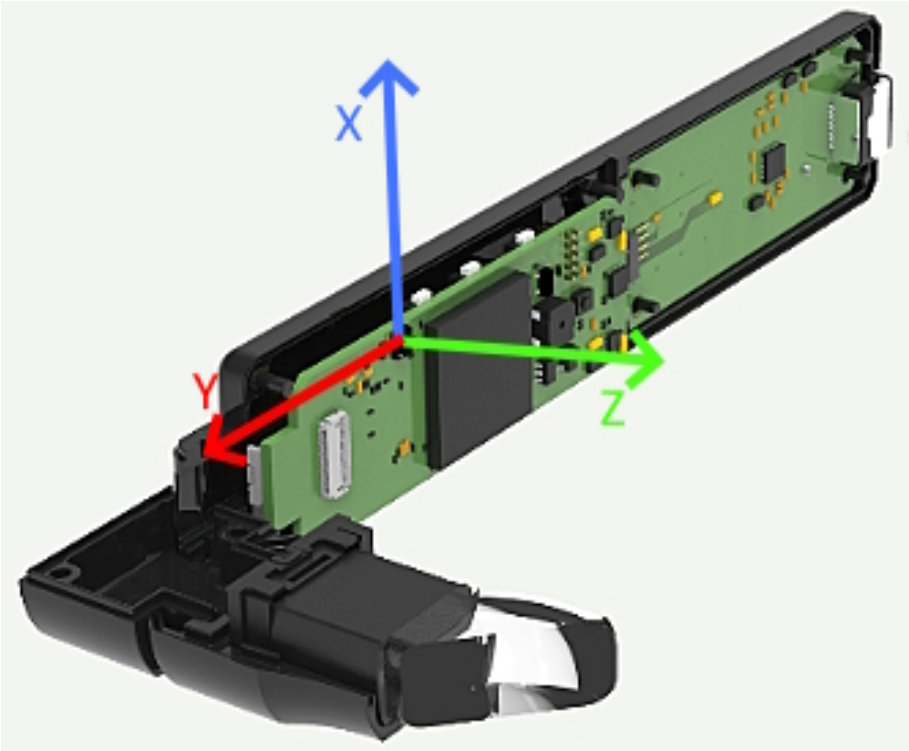


Figure 3.1: 3 Axis of the LSM6DSM in the final product.

3.1.3 Screen rotation

The interrupt handling task needs a way to perform the screen rotation.

In general, there exists several ways to change the orientation of a LCD screen for embedded devices and this can be done at different levels. For example, some LCD drivers (e.g., the ST7789) allow to mirror or rotate the image directly at the level of the display which is the most efficient way of doing it as it only requires to change a register value in the driver. In the case of our smart glasses, the driver (Kopin A913) does not allow to do such an operation so the rotation must be performed before reaching the display. Figure 3.2 shows the flow of the data before it is drawn on the screen.

Figure 3.2: From Bluetooth to LCD: The screen composition data flow

Rotating the screen using the VDC

The next possibility to rotate the screen in this hierarchy could be in the VDC (Video Display Controller), which is a driver provided by Renesas. But the documentation doesn't explain how to rotate the buffer before it is sent to the driver despite the fact that there is something about rotations in its source code:

```
typedef enum
{
    VDC_WR_MD_NORMAL = 0,          /*!< Normal */
    VDC_WR_MD_MIRROR,              /*!< Horizontal mirroring */
    VDC_WR_MD_ROT_90DEG,           /*!< 90 degree rotation */
    VDC_WR_MD_ROT_180DEG,          /*!< 180 degree rotation */
    VDC_WR_MD_ROT_270DEG,          /*!< 270 degree rotation */
    VDC_WR_MD_NUM
} vdc_wr_md_t;
```

The difficulty of reaching an expert or posting on the Renesas community forum makes it impossible to continue on this track.

Rotate the screen using the buffer

Another common way of rotating the screen is to use two mirrored buffers and switch them when the orientation changes. This is not possible with our chip because it has not enough memory for containing 3 buffers of RAW pixels. Indeed, the graphics library already works with two buffers: a working buffer where it stores the operations that are still not committed and the displayed buffer. A rough approximation is the following: the screen resolution being 854 x 480 times the 3 colors represents 1.2MB, and the MCU has a total of 4MB of RAM memory in which only 2.9MB is allocated to cached heap. So, the allocation of 3.6MB for the three buffers exceeds the capacity of the memory. Before these estimations, the solution was tested and it did not work in practice. Even though the pixel format for the buffer is optimised and stored on two bytes instead of the three from our estimation, it still does not work. Indeed, when the library is given an image to display on the screen, it needs a lot of space which exceeds the capacity of the SRAM and the system freezes in those conditions.

Rotate the screen in RGA

The RGA (Renesas Graphics Architecture) is a driver provided by Renesas that allows to use the hardware accelerated graphics capabilities of the chip to render images, shapes and perform other graphics operations such as rotation, scaling and projection. The problem with this solution is that all the elements drawn on the screen need to be stored because changing the matrix of the drawing only affects the elements that are drawn afterwards. So, a structure is created to store all drawn images on the screen, and it contains the following information:

- The filename
- The X position
- The Y position
- The width if it is resized or -1
- The height if it is resized or -1

As all elements drawn on the screen are images[4], this is enough for our intended application. Then, to perform a rotation, the screen is cleared, the drawing matrix is set to the new orientation and everything is redrawn. Below, the matrix operations for each screen orientation is given.

```
// Right eye screen orientation
R_GRAPHICS_TranslateMatrixI(&graphics, 0, FRAME_HEIGHT);
R_GRAPHICS_ScaleMatrix(&graphics, 1.0f, -1.0f);

// Left eye screen orientation
R_GRAPHICS_TranslateMatrixI(&graphics, FRAME_WIDTH, 0);
R_GRAPHICS_ScaleMatrix(&graphics, -1.0f, 1.0f);
```

In the previous code, the normal orientation (left eye) shows that the screen was already mirrored horizontally. Indeed, this code is inspired from the initialization code which already did that operation on the RGA drawing matrix.

This way of rotating the screen is fast enough. It was experimentally found that the redrawing of an entire screen containing 2500 images of 12 x 12 pixels is executed in $<$2.5s, which is less than the time needed to take the glasses of and put them back in the other position.

LCD configuration issue: Logical size different from hardware size

At this stage, another problem has been identified while testing this new rotation technique: The rotated screen was nearly 10% too high and those 10% were over the top of the screen meaning that it was an invisible overflow. The problem comes from a difference between the definition of the screen configured in the RGA library (800 x 480) and the real definition of the screen (854 x 480). When looking carefully, there were also horizontal black borders on the left and right size.

The following figures 3.3 and 3.4 represent the problematic view and the corrected view using a simple numeric pad image drawn at each corner. The pictures are blurry and not well-oriented because of the difficulty of capturing the screen with a regular smartphone objective. The top part of the figure shows the fact that the top left corner is at the right position initially, but when the screen is rotated (right eye orientation), the top left corner overflows. In the bottom part of the figure, the 4 corners are well contained in both orientations.

[4]Texts are composed using pre-rendered fonts where each character is an image.

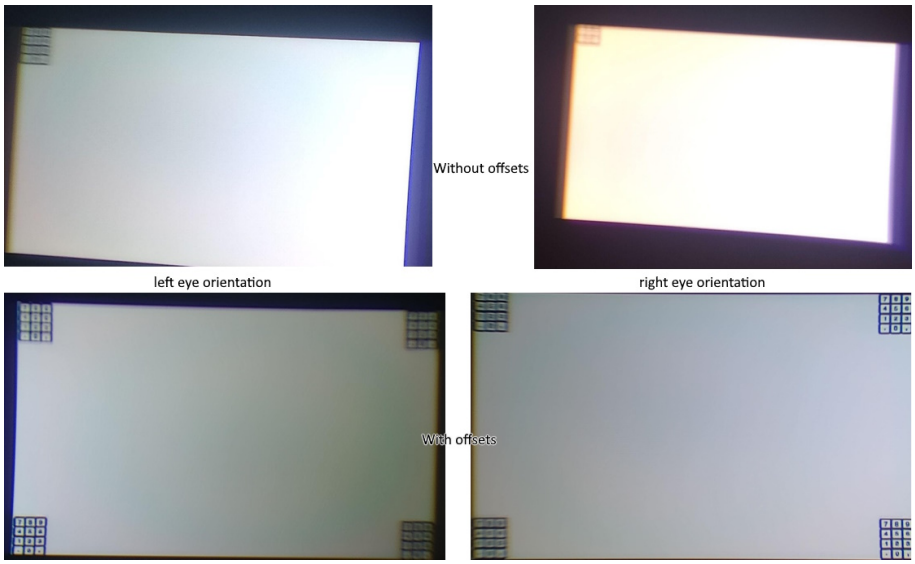


Figure 3.3: The image position on screen have been re calibrated.

In Figure 3.4, the left part shows that the screen is drawn with black borders because of the non-deforming scaling. After the resolution has been corrected to be the same in all drivers, the black borders disappeared and the screen is drawn on the whole led array.

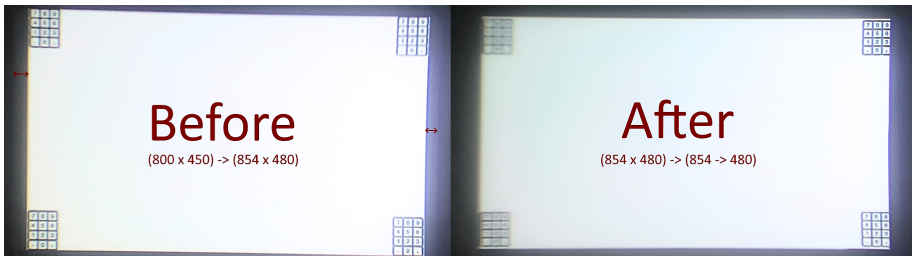


Figure 3.4: The LCD timings have been redefined to remove black borders.

This misconfiguration was induced by the fact that Renesas only provided timing configuration examples for the 800 x 480 WVGA resolution screens, and computing timings for the correct resolution required too much investigations for a prototype without bringing significant improvements. Indeed, the prototype interfaces were drawn carefully in safe zones and black borders were nearly not visible. However, as the screen orientation could change, the safe zone became different in both orientations and this problem had to be solved.

The configuration of the resolutions and LCD timings takes place in the various macros of the *LCD_config.h* file and some registers of the A913 display driver (*kopin.c*).

Renesas provides a software tool for computing the right timings called "QE for Display" [11], but this tool expects the display to be documented in a manner that is different from the provided documentation for the Kopin Golden Pearl Optical Module. Moreover, there are ambiguities that lead to make several assumptions when trying to transpose or compute information from one document to the other.

The use of various generic calculators or generators that we can find online, such as `https://arachnoid.com/modelines` (Accessed on 2023-08-20) did not work either because of incompatibility or the lack of configurability.

For those reasons, the final effective way of retrieving the right timings was by experimental trial-and-error method as there were only 6 integer macros to tune for which the range of possible values was under 20. The list of timing macros (i.e., using Renesas terminology) found to work properly are given in the list below:

- WVGA - 854x480 @ 60Hz - Pixel Clock @ 33Mhz
- The Vertical Pulse Width (VPW): 10 (clock cycles)
- The Vertical Back Porch (VBP): 35 (clock cycles)
- The Horizontal Pulse Width (HPW): 30 (clock cycles)
- The Horizontal Back Porch (HBP): 173 (clock cycles)
- The Vertical Total Period (VTP): 549 (clock cycles)
- The Horizontal Total Period (HTP): 1167 (clock cycles)

3.2 Autonomy

3.2.1 Autonomy problem statement

One major differentiation point of Get Your Way's product from its competitors is the hot-swappable battery. The first one is already implemented but the second needs some additional work. Indeed, the measured real-world autonomy of the aRdent prototype was of 2 hours and 30 minutes. The ideal objective is to make the battery last for 4 hours (i.e., the equivalent of half a day of work), so it can be swapped during a lunch break, for example. [17]

3.2.2 Autonomy improvement possible solutions

First, some suggestions have been provided by Get Your Way regarding the power optimisation of the device. The first page of the document is given in Appendix B. The idea is to use the sleep capabilities of the MCU to progressively diminish the power consumption of the device. However, the effects of this solution on the power consumption of the device are unknown. Moreover, it may be frustrating for users that the device goes to sleep and deep sleep modes in an unwanted manner.

Second, a methodology of testing needs to be set out in order to correctly assess the effects of our changes on the power consumption. For example, in an article from the Serbian journal of electrical engineering, the authors used an oscilloscope and a resistance to measure the power consumption of a different MCU in the 3 different

modes (active, sleep, and deep sleep)[5], using Ohm's law formula (U = R.I). [13] It is the most affordable method to measure current because current probes are expensive.

Finally, the most common way of reducing power consumption in computing devices is to always keep the CPU clock at the lowest possible frequency[6] and disable all unneeded peripherals. In the case of aRdent, when used, all the peripherals are needed, but the current implementation makes the processor always operate at its maximum clock of 528Mhz while it is unneeded most of the time.

3.2.3 Autonomy evaluation methodology

First, regarding the methodology for measuring power consumption gain of the various tweaks we can make, we could also use an oscilloscope or a special multimeter. However, an easier and more affordable method is to use an affordable USB ammeter as the one shown in Figure 3.5. Indeed, the battery module can be plugged into any standard USB power source and works without a battery soldered to it. The accuracy of the USB ammeter is evaluated using the measured average autonomy of the battery and computing the theoretical autonomy given the instantaneous consumption.

Figure 3.5: The USB ammeter used to evaluate the power consumption.

However, the computed charge capacity on the tool (in mAh) did not seem accurate. Indeed, it shows a consumption of 161mAh after 30 minutes when the average instantaneous current draw is of 110mA.[7] This is not a problem since we can still draw conclusions from the average instantaneous current and voltage measurements.

[5]We can observe that for their chip, the consumption is reduced nearly by a factor 2 in deep sleep mode.

[6]One can simply observe their CPU clock on their computer when performing various tasks to see that it is most of the time operating at a base minimal clock speed.

[7]This was measured on the glasses, in IDLE with the lowest clock (66Mhz). Given the instantaneous current draw, it should have consumed only 55mAh after 30 minutes.

3.2.4 Power consumption measurements and interpretation

A list of measured average instantaneous consumption in different states is given in Table 3.1. LP stands for a "Low-Power" mode offered by the display driver and the last row was measured with the peripheral clock lowered from 66MHz to 55MHz.

Clock	Display	Bluetooth	W	V	A
528MHz	off	on	0.765	5.1	0.15
528MHz	on	on	1.268	5.07	0.25
66MHz	off	on	0.561	5.1	0.11
66MHz	off	off	0.510	5.1	0.10
66MHz	on	on	1.120	5.6	0.2
55MHz	off	on	0.511	5.11	0.10
55MHz	off	on	0.561	5.1	0.11
55MHz	on (LP)	on	1.014	5.07	0.2
55Mhz	on (LP)	on	0.913	5.07	0.18

Table 3.1: Measured power consumption in Watts (= Voltage (V) x Current (A)) of aRdent glasses at various clock speeds.

The first thing we can observe is that the screen consumes about 525mW on average. Then, we can see that reducing the clock of the MCU completely with the peripheral clock to 55Mhz allows a saving of 355mW, or looking at the current, 70mA. Finally, the product was tested once with the Bluetooth turned off and we can observe a gain of 10mA. After some quick tests, reducing the brightness of the screen also allows a gain of about 10mA.

A second observation is the different voltages. For example, when the clock is at 66MHz with everything on, we can see a voltage of 5.6V which is 500mV over the average consumption of the other runs. However, there is no immediate answer to those variations and the voltages were the same even when repeating the experiment a second time. Moreover, the computed power consumption seem coherent with what we could expect (i.e., lowering the clock and disabling peripherals lowers the total power consumption). Therefore, this will not be investigated.

3.2.5 Analysis of possible autonomy solutions

First, it is legitimate to think that shutting the screen off when unneeded would allow to increase the autonomy of the glasses by a lot. However, this solution is not user friendly. Indeed, it is not possible for the user to tell if the screen is off because of a timer or because the battery is empty. This solution is not prioritized but can be kept for further improvements as it affects the user experience negatively.

Second, we can see that lowering the CPU clocks as much as possible could give the most power consumption economy after the screen solution discussed above. Moreover, there are ways to do it such that nothing changes for the end-user. Indeed, when using a regular computer, nobody is affected by the dynamically changing CPU

frequency in the background as long as the system is reactive enough. In fact, we observed that aRdent works near perfectly even with the permanent lowest clock.

Finally, we can expect very little consumption gains by disabling Bluetooth or dimming the screen. On the first hand, Bluetooth is essential to interact with the device so we can not disable it. On the second hand, dimming the screen after a long period of inactivity could improve the battery life but it may have a stressing effect on the user.

3.2.6 Dynamic CPU clock solution

This solution consists in always running the CPU with its lowest frequency except when some tasks require performance. For example, when receiving a new screen via Bluetooth. In that case, the CPU clock is pushed to the maximum during some milliseconds.

Figure 3.6 shows a Finite State Machine representing this simple solution. Most of the time, the CPU will stay in its lowest frequency operating mode. From time to time, a call to *request_ cpu_ burst* will put the CPU in its high frequency operating mode for an arbitrarily defined time. As changing the CPU frequency takes some time (several milliseconds), intermediate states are also represented.

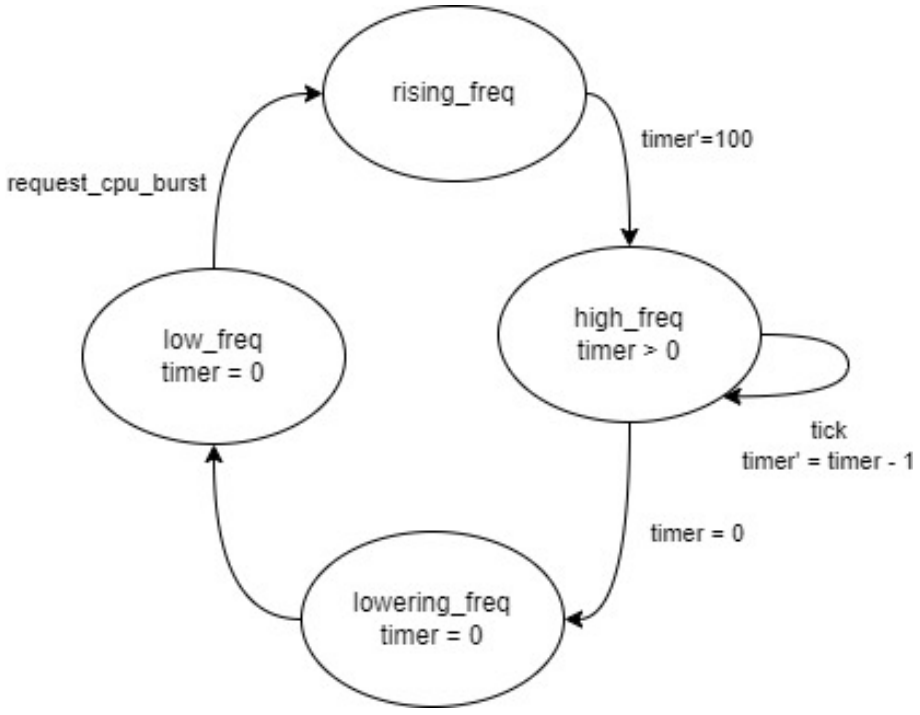


Figure 3.6: FSM of dynamic CPU power optimization.

Implementation of dynamic CPU clock

In the first place, the Smart Configurator allows the tweaking of the various clocks of the RZ/A2M MCU. The GUI of the software, which is captured in Figure 3.7, prevents us from setting incompatible clocks. In order to be able to reach the lowest

clock of 55Mhz, we have to set the main clock to 10Mhz in low mode instead of 24Mhz High-mode which will limit our maximum clock to 440Mhz instead of 528Mhz. This is not a problem as the difference between 440Mhz and 528Mhz is not noticeable while the device is in use. Moreover, it leaves some margins in case we need more performance in the future thanks to a better battery. The peripheral clock is also lowered from 66Mhz to 55Mhz and this will not be changed dynamically.

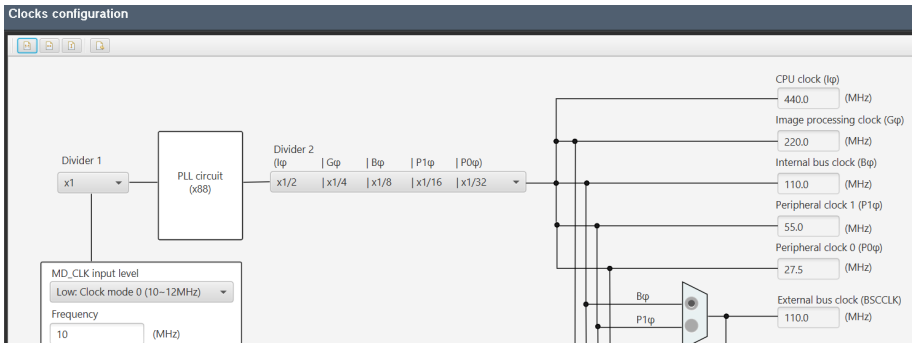


Figure 3.7: Renesas Smart Configurator clocks settings interface.

Then, the *FreeRTOSConfig.h* file needed to be modified to set the new peripheral clock and CPU clock. It is important because the system ticks relies on the peripheral clock. The periodic task that trigger the ticks was also updated to wait the appropriate amount of clock counts (550 instead of 660).

After that, the Renesas *r_ cpg* driver allows to control the CPU clock. So a task with higher priority than the *IDLE* task is created to manage the CPU frequency. There is a global variable that maintain the current state of the CPU: "low frequency", "burst requested" or "bursting". When the CPU is in "low frequency", the task is suspended and nothing happens. A call to *request_ cpu_ burst* will set the state as "burst requested" and wakes up the task which will call the appropriate driver to higher the CPU frequency and set the state as "bursting", then it will wait for 100ms giving the CPU back to the operating system. When the 100ms are elapsed, the task puts the CPU in low frequency mode and sets the global variable accordingly. If a call to *request_ cpu_ burst* is made while the previous burst is not finished, the task will simply set the global variable back to "bursting" from the new "burst requested" and wait for another 100ms.

Finally, the CPU burst request function is called from various performance requiring functions such as the Bluetooth packet handler function, the screen rotation task, and at the entry of the main function which initializes the glasses for faster boot time.

Chapter 4

The I^2C communication problem

4.1 Problem statement

During the establishment of communications with the LSM6DSM module discussed in the previous chapter, a problem has been identified on the I^2C bus. Indeed, there was no acknowledgement from the accelerometer when its address was sent by the Renesas "r_riic" driver[1].

There are several reasons that could explain the fact that the LSM6DSM does not acknowledge when it is addressed on the I^2C bus. First, the module can be defective. Second, the effective electronic signal could be different from those observed in the firmware trough the debugger. Finally, it could be an error in the PCB design or manufacturing. Each of those reasons are explored in the following sections.

4.2 Reason 1: Defective component

One possible reason for the lack of response could be that the LSM6DSM module on the tested board is either dead, malfunctioning or damaged. However, this possibility was quickly ruled out. In fact, 30 copies of the PCB have already been produced and the code has been tested on several of them, with none of the modules responding with an acknowledgement. We note the possibility of having a batch of defective accelerometers but the probability is low and the dead-end induced by this hypothesis makes us explore the other paths.

4.3 Reason 2: Unexpected electronic signals

Another possible explanation could be that the actual electronic signals sent on the lines differ from what is reported during firmware debugging. For instance, the firmware may behave as if it has sent the correct address on the I^2C bus and wait for an acknowledgement, even though nothing was actually sent.

It was possible to analyse the signal effectively sent on the bus using an oscilloscope. Two tiny vias[2] allowed to probe the two lines (SDA and SCL) that compose the bus. Figure 4.1 below shows the captured signal.

[1] The bits sent over the bus were double checked via GDB inside the driver source code and when the acknowledge is checked, an error code is returned meaning that there was no acknowledge by the module.

[2] An electronic *via* is a "plated hole" in the PCB for connecting multiple layers. [2]

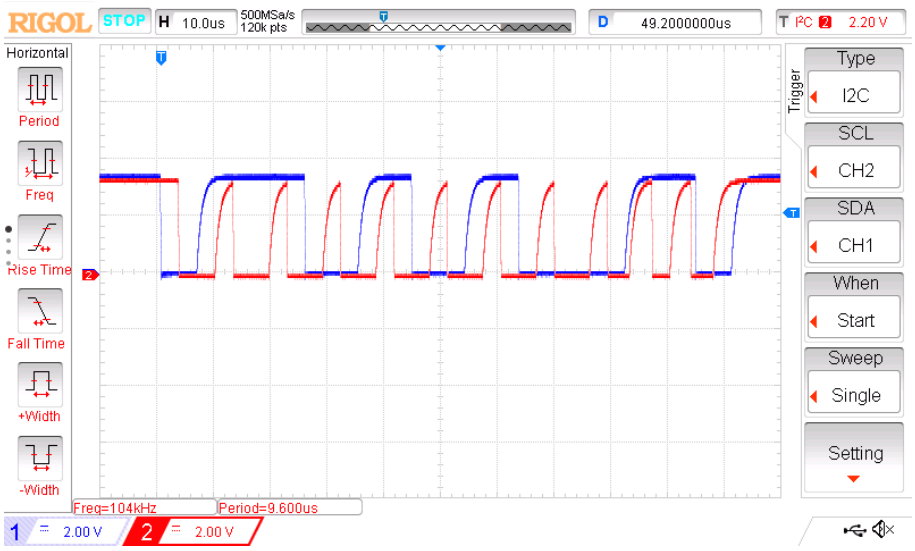


Figure 4.1: I^2C signal shows that the address is sent but not acknowledged

The signals are correct and conforming to the bits observed in the firmware through the debugger. This automatically leads us to dismiss this reason.

4.4 Reason 3: Bad design or manufacturing

The last reason explaining this problem could lie in the design of the product or it could have been a problem in the manufacturing process. Indeed, the problem could be explained by a failing soldering process, or traces that are grounded unexpectedly due to manufacturing error.

The vias allowing to probe the signals with the oscilloscope being so small, it was not obvious in the first time that the SDA and SCL lines coming from the MCU were inverted. In Figure 4.2, the documentation of the LSM6DSM pins and the PCB annotated traces design are given. The pin 13 must be the SCL line and the 14 the SDA, but in the PCB design they are inverted.

4.5 The I^2C communication solution

There can be several solutions to solve this problem. One solution is physical: The traces could be cut and cross-connected. Nevertheless, solving this problem in software would be a more cost-effective solution, if possible.

It is not possible to invert the pins in the Renesas Smart Configurator and a deep exploration of the generated code of the driver allowed to identify that the MCU uses specific registers to manage I^2C communications and there is no way to configure which pin is SDA and which is SCL as they are hard-wired in the MCU.

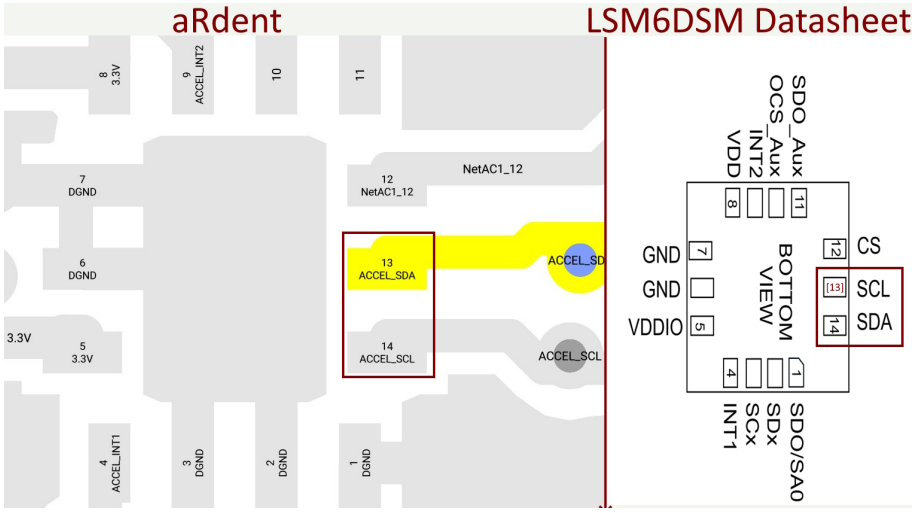


Figure 4.2: Electronic scheme design of aRdent product showing inverted pin definitions for the LSM6DSM

Therefore, the remaining software solution was to configure those MCU pins as General-Purpose Input-Output (GPIO) and control their level through code to perform I^2C operations. The pins being physically designed to be open-drain, the only configuration to be done was to set them as high outputs. A first code was quickly developed to validate the solution and it worked. Below, Figure 4.3 shows a working I^2C communication with correct acknowledgements using this solution.

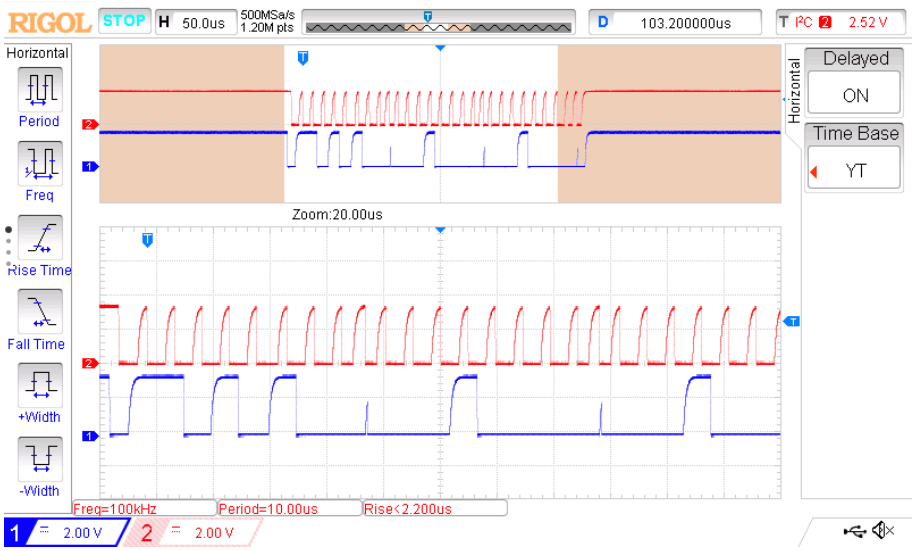


Figure 4.3: The LSM6DSM component address is sent over I^2C and correctly acknowledged

4.6 A homemade GPIO I2C driver

4.6.1 Motivations and introduction

Other issues were observed with the Renesas "r_riic" driver with the A913 display driver. Even though the A913 is correctly connected regarding the SDA and SCL pins on a different MCU I^2C bus, the I^2C communication failed sometimes due to software bugs according to GYW. Our objective was therefore to make the GPIO I^2C driver discussed in the previous paragraph as general as possible and reusable for multiple I^2C channels.

The previously developed proof-of-concept was not good enough because it used busy-waiting with loops of "nop" operations to introduce appropriate delays between clock signals. Therefore, a better solution needed to be designed.

The firmware being based on FreeRTOS, the solution is a combination of tasks and interrupts generated by the Renesas OSTM (OS Timer Manager) driver. Indeed, this driver can generate interrupts with a frequency up to the one of the peripheral clock (55 MHz) which is more than needed for the 100 kHz standard I^2C maximum rate. Moreover, it was not possible to rely on FreeRTOS delays as the smallest possible value for those is one tick which is set to one millisecond in this port of FreeRTOS.

4.6.2 Analysis and theoretical solution

First, when an I^2C operation is requested by a task, some global structures for the given bus are initialized. Those structures contain, for example, the state of the I^2C FSM, counters and buffers. The complete I^2C FSM is represented in Figures 4.4, 4.5, 4.6 and 4.8. It is organized as two encapsulated machines: one responsible for the macro operation (read or write), and the micro machine responsible for the bit-wise operations such as start, stop, transmit, receive, ack, etc.

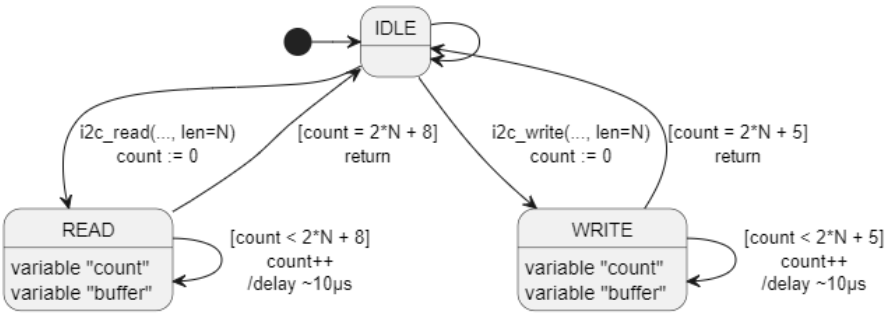


Figure 4.4: I^2C global macro FSM

Then, the task managing I^2C communications is started. This task requests the OSTM driver to generate an interrupt with a 100 kHz frequency which will unlock the execution of the FSM and it enters its main loop executing the machine. Figure 4.9 shows an UML sequence diagram that summarizes the activity for a simultaneous read and write I^2C call by two different tasks (T1 and T2).

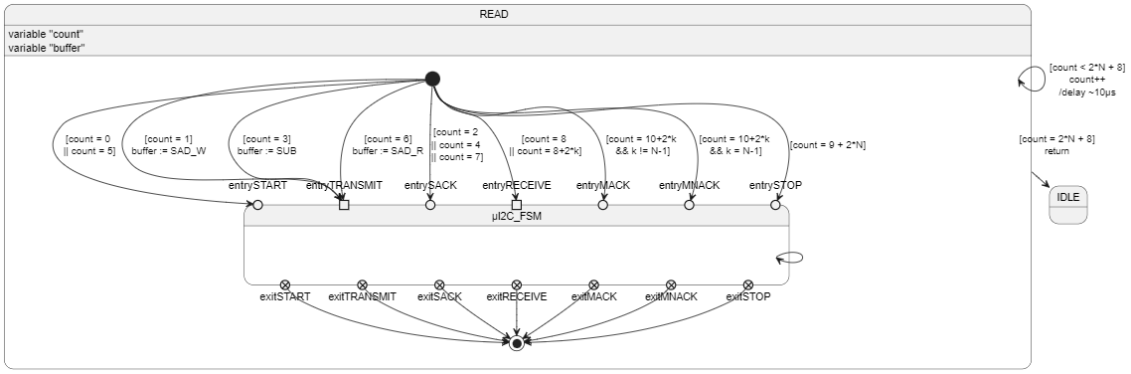


Figure 4.5: I^2C read operation FSM

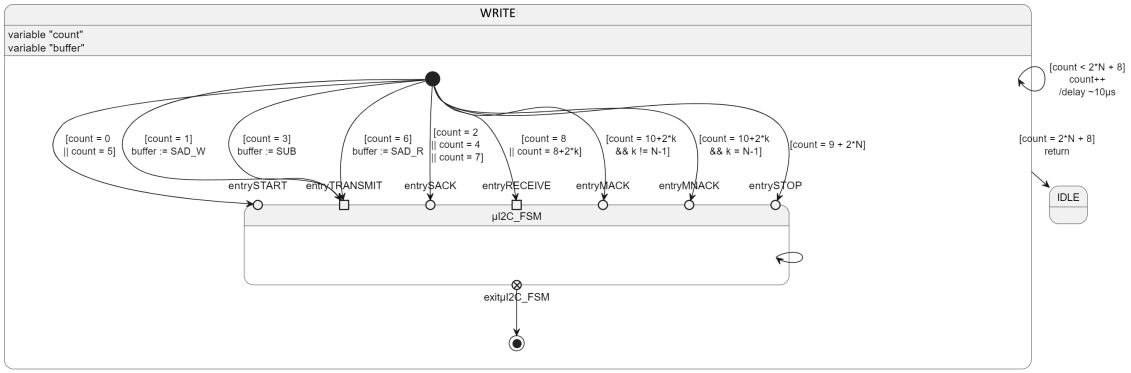


Figure 4.6: I^2C write operation FSM

The priority of our "GPIO I2C" task should finally be higher than the caller tasks in order to ensure that a task is never waiting for a task with lower priority to complete.

4.6.3 Implementation of the theoretical solution

The theoretical solution was implemented as in the design using enumerations for states and C structures for maintaining the state of each one of the 4 I^2C channels of the MCU.

However, multiple bugs were encountered at the moment of testing and they were not easy to correct given the fact that the processes were time-sensitive and split across multiple tasks. Moreover, it was urgent to find a better solution that was easier to develop and works despite giving up some features.

For those reasons, the branch of this theoretical solution was put aside and a new solution was developed. Indeed, thanks to the gained experience with the OSTM driver, it was now easy to get rid of the busy waiting in the proof-of-concept solution. Additionally, the requirement of being able to simultaneously perform I^2C communication on all the channels was relaxed to "one communication on any channel at a given time".At the moment of writing, this constraint is not affecting much the user experience. Indeed, only two channels are quite rarely used for the accelerometer and the configuration of the display driver. By consequence, the final implemented

and tested solution consists in a simple global mutex blocking simultaneous I^2C calls. The proof-of-concept solution was improved to use the OSTM driver instead of busy waiting. Finally, we rely on the MCU context switch capability for maintaining the state, counters and buffers. For example, if a screen rotation interrupt happens during an I^2C communication, the context switch will save the stack and processor registers before jumping to the Interrupt Service Routine (ISR). These saved elements are equivalent to the counters and states we represented in the designed FSM. Indeed, the call stack is equivalent to the counter and the program counter is similar to the sub-counter.

In the list below, we give an example call stack to better show the similarities:

- *i2c_read* is first called to perform a read. This is the macro operation (Figure 4.4).
- *i2c_read_byte* is called after having written the start and the slave address and register on the bus. This is the macro operation at a given counter (Figure 4.5), where the counter is equivalent to the program counter register.
- *i2c_set_scl_[high/low]* are called 8 times for reading a byte on SDA. This is the micro operation shown in Figure 4.8.
- *direct_control* is the function that interacts with the GPIO driver.

The necessity to use a mutex to avoid two I^2C communications at the same time comes from the fact that we use only one OSTM timer and it could lead to a race condition without the mutex.

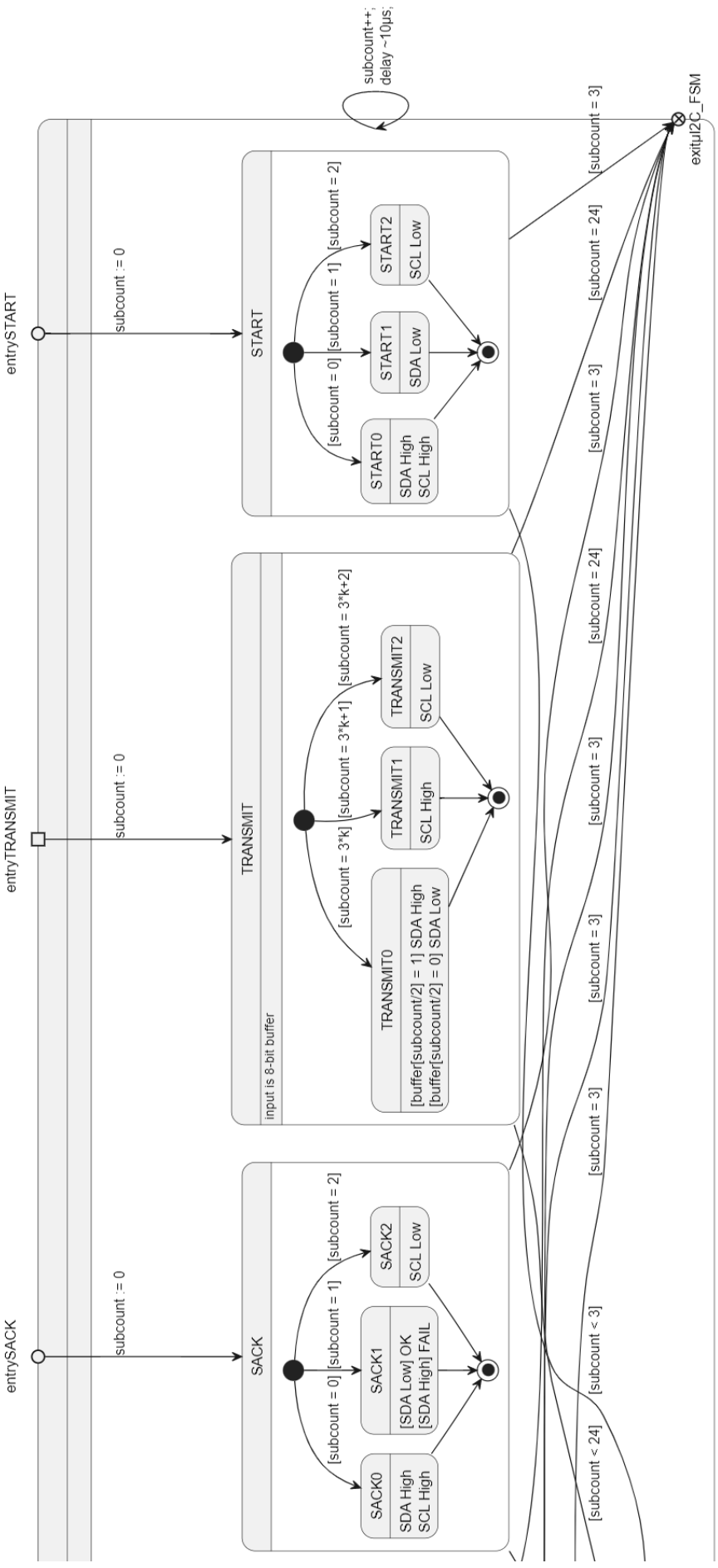


Figure 4.7: I^2C micro FSM (part1)

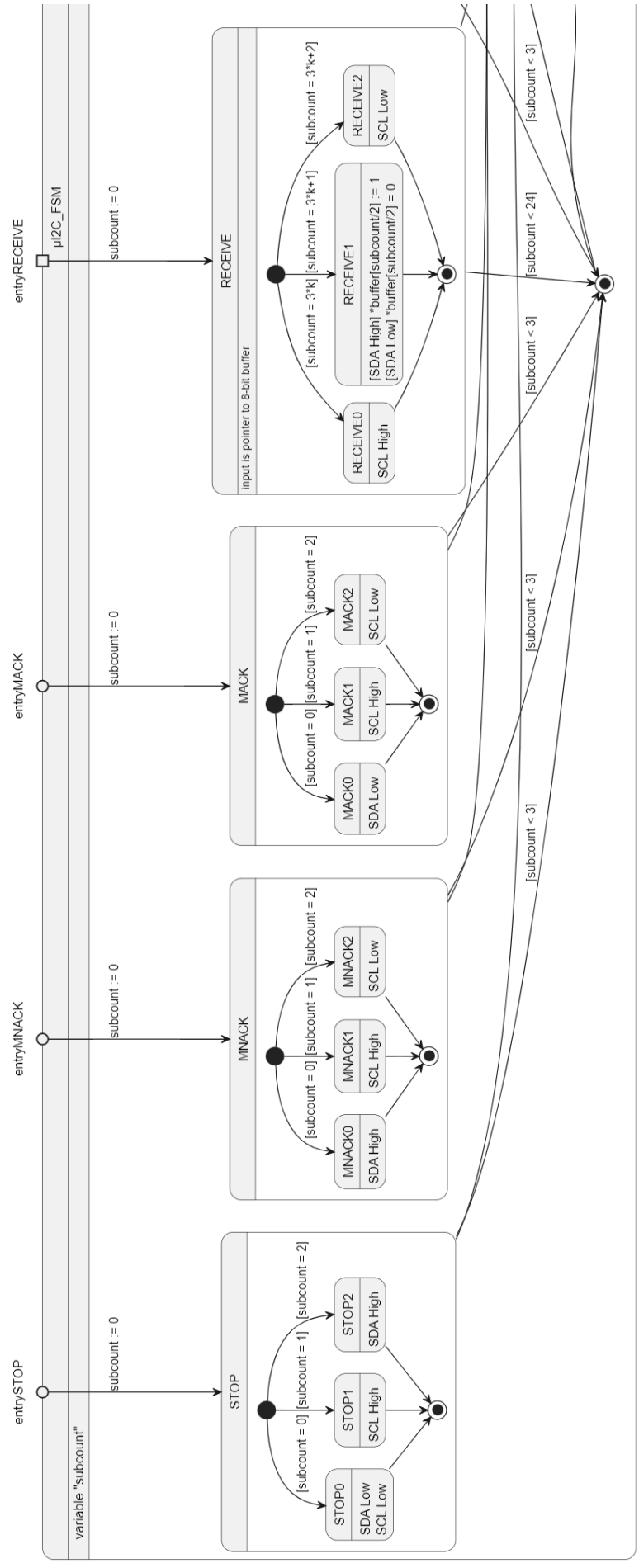


Figure 4.8: I^2C micro FSM (part2)

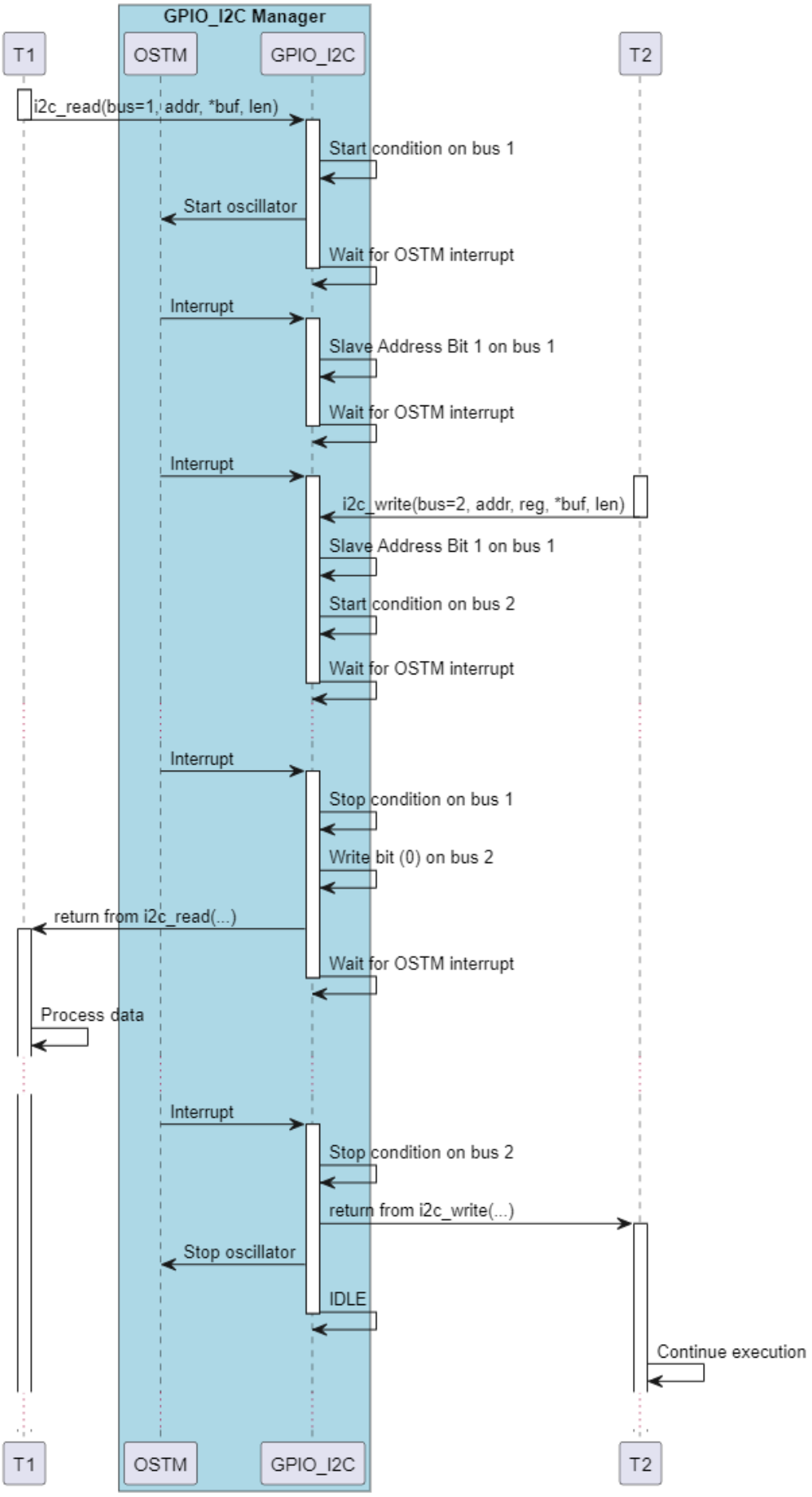


Figure 4.9: I^2C sequence diagram

Chapter 5

Security improvements

5.1 Security management in the organization

Information security management is a critical aspect of modern business operations and is governed by international standards such as ISO-27001, which provides a framework for managing and protecting sensitive information using a risk management approach [4]. However, according to Urpo Kaila and Linus Niman [5], these standards can be too complex for start-up companies like Get Your Way to easily adopt. Despite having a co-founder with a background in the domain, Get Your Way had no real plan regarding information security when I started this work. Nevertheless, the start-up company uses a lot of cloud solutions with which they share the risks. For example, Google Drive is used for storing files and that makes Get Your Way benefit from some of the information security expertise of Google for ensuring the confidentiality, integrity and availability of their data.

Therefore, a first step towards improving the security management at GYW was to apply the simplified framework proposed in the aforementioned article [5]. The deliverable of this task consisted in a first template document containing the elements suggested in the framework. The delivered version is given in Appendix A.

First, the safety of the product entirely depends on the business continuity of Get Your Way activities. This framework is a simple way of pursuing this goal. Indeed, it gives a simple and clear strategy for dealing with information security. For example, it is suggested to establish a business continuity plan which can be useful in the event of an information security incident. This will mitigate the impact of such an incident on the interest of financial institutions and investors.

Then, as the product is targeting mainly business customers, certifications may be required before the adoption of a business due to higher requirements in terms of security, for example. Given that, keeping an up to date version of the document provided in appendix A will ease the access to obtain ISO-27001 certification in the future as it is a simplified process summarizing important steps of both the ISO standard and the NIST framework [5].

Finally, the framework is not the only shield against cyber threats. Indeed, it is important to refine it on a regular basis and mitigate risks as soon as they are identified. The key to develop a safe product is the intellectual assiduity of the organization's members. That's why, for example, it is highly advised to subscribe to information security newsletters and educate the members of the organization to adopt safe reflexes.

5.2 Device security

After a first section on the security of the organization, we will now discuss the security of the device which will be assessed, and identified issues that have a major risk will be addressed. There are multiple ways that such a device can be compromised or be used as a vector of attack. For example, an attacker could update the glasses with a compromised firmware that will also infect paired devices, listen to communications, interfere with them, etc.

5.2.1 OTA security

First, the OTA capabilities implemented by Quimesis for the aRdent firmware can trigger some questions from a security point of view. Indeed, as shown in the example above, this feature can be used to compromise the device if it is not carefully implemented.

On the first hand, the contracting company thought about security when implementing the feature. Indeed, the delivery report explains in details how the update process works and what was been done for security including the process of signing the firmware and how the signature protects the device from being compromised.

On the second hand, the OTA is done through an unprotected Bluetooth connection. This is not a problem because the compiled firmware binary is already publicly available in theory for the update[1]. Moreover, an attacker who may want to interfere and modify bits of the firmware will not be able to corrupt the device as the firmware is signed and the certificate is kept in a secure bootloader which is not updatable. However, it is important that the firmware binary does not contain any hard-coded secrets and passwords. At the moment of writing, this is verified.

5.2.2 Bluetooth Low Energy security requirements

BLE will be the only protocol allowing to use the aRdent glasses. Therefore, it is important to study the security of the the protocol, what was implemented in the firmware until now and what can be improved. As always in computer security, this analysis will be carried out around the three main pillars of the domain: Confidentiality, Integrity and Availability (i.e., the CIA triad).

First, the confidentiality of the BLE communication must be ensured and it is the most critical aspect. Indeed, we can not tolerate that business secret operations are sent in the air without any sort of encryption. For example, we can imagine a situation where the glasses are used in a restaurant for displaying recipes in the eyesight of the cooker. In that case, we can not permit an intruding customer[2] to easily steal the recipes by simply eavesdropping Bluetooth communications.

Second, the integrity of the communications must be ensured too. Indeed, the whole

[1] So, it is not a security issue if someone is able to eavesdrop the firmware.

[2] It can also be the neighbouring restaurant or a pedestrian as Bluetooth can cross walls.

utility of the device is to provide an operator with precise instructions. Therefore, it is not acceptable to have undetected alterations of the instructions as they could lead to severe consequences. For example, if the glasses are used in an industrial laboratory for medicine production and the quantity of a reactive is altered, this could lead to a severe disaster.

Finally, the availability of the protocol should not be affected. The major threats to this guarantee are denial of service attacks which could affect negatively the reputation of GYW and the adoption of their product.

5.2.3 BLE Security Offerings

In this point, we will explore the security guarantees that can be offered by the BLE protocol. There are a lot of resources available to learn about the protocol security such as the official BLE Security Study Guide [1], but a good summary can be found in a blog post on medium.com by Panagiotis Antoniou[3], a computer security enthusiast.

First, the author starts by using a slightly different triad consisting in confidentiality, authentication and authorization which are all part of the confidentiality in our previous discussion. (Source: Panagiotis[3])

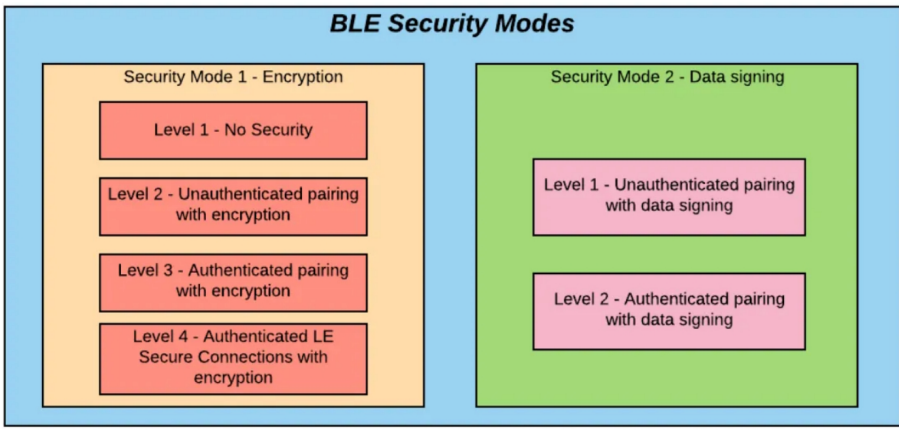


Figure 5.1: BLE security modes. (Source: Panagiotis[3])

Then, the author gives the 2 main modes and 6 associated levels of BLE security represented by the Figure 5.1. After what, he explains what keys are used, the three pairing methods (i.e., "Out Of Band (OOB)", "Passkey Entry (PE)" and "Just Works (JW)") and the three phases of the pairing process. The blog post finally ends with a list of vulnerabilities and possible threats which is an interesting resource captured in Figure 5.2. (Source: Panagiotis[3])

This blog post is a good resource but it however lacks some small details. Indeed, in order to chose one of the three pairing methods, the Bluetooth standard relies on the input and output capabilities of a device, then a matrix allows to choose the

[3]Link: `https://antonioupan6.medium.com/bluetooth-low-energy-ble-security-in-a-nutshell-5d20164cceb2`, Accessed on 2023-08-09

Vulnerability	Possible threats
Link keys such as LTK, CSRK and IRK become can be retrieved and used again	Eavesdrop attack
JW pairing method has a TK all set to zeros thus the pairing can be easily intercepted. No link key authentication.	Man In The Middle (MITM) attack
PIN pairing has only 10000 possible combinations if the PIN is four digits.	Brute Force attack
Encryption key length is very small	Brute Force attack
JW has no user or device authentication	Disclosure attack
Flood of data requests from the target device	Denial of Service (DoS) attack
An adversary can change the Bluetooth address of the device to be identical with another device which can result to merging two connections to one	Denial of Service (DoS) attack
Authenticating the device is a very simple method which involves sharing a set of keys	Man In The Middle (MITM) attack
Master key used is shared in all of the devices in a Bluetooth topology	Disclosure attack
Weak Encryption when generating a private Bluetooth Address	Disclosure attack
BLE Security Mode 1 Level 1 has no sign of defence mechanisms which correspond to either encryption or authentication.	Eavesdrop attack
There is no end-to-end security due to the fact that only some links are encrypted or authenticated.	Man In The Middle (MITM) attack
A device can be discoverable by other devices.	Disclosure attack

Figure 5.2: BLE security vulnerabilities and threats. (Source: Panagiotis[3])

pairing method. [15] The devices categorisation is given in Table 5.1 and the pairing method selection matrix in Table 5.2[4].

	No output	**Display**
No input	No input no output	Display Only
Yes/No	No input no output	Display Yes/No
Keyboard	Keyboard only	Keyboard Display

Table 5.1: Device capabilities matrix. Source: Table 10 of [15]

	Display only	**Display Yes/No**	**Keyboard only**	**No Input No Output**	**Keyboard Display**
DO	JW	JW	PE	JW	PE
DYN	JW	JW	PE	JW	PE
KO	PE	PE	PE	JW	PE
NINO	JW	JW	JW	JW	JW
KD	PE	PE	PE	JW	PE

Table 5.2: Pairing method given device capabilities. Source: Table 11 of [15]

[4]"DO" (in first row) stands for Display Only. As the row names are the same as the column names, the same holds for next rows.

5.2.4 BLE Security suited to aRdent

Given our requirements and the security capabilities of BLE discussed in the previous points, we can deduce that the security mode one level four is the best fit. Indeed, it provides encryption of the traffic which prevents eavesdroppers from easily obtain confidential information. However, it does not explicitly provide integrity protection, but this is not a problem since altering a cipher text normally results in complete corruption of the decrypted clear text and, Bluetooth packets additionally contain a CRC (Cyclic Redundant Check) field for integrity protection.

Therefore, we need to secure the pairing as it is the moment where the security mode and level is chosen by the paired devices. Given the fact that the glasses do not have NFC capability, the only remaining pairing methods are PE and JW depending on the coupled device. Indeed, it is possible that the paired device to the glasses is a simple slide-show clicker that only has two buttons and no screen. The glasses having only a screen and eventually a "Yes/No" selection using the "tap" detection capability of the accelerometer, we are in the first two columns of the Table 5.2 which are identical and imply that the coupled device have a keyboard for the PE pairing method.

In theory, we should stick with the well-known "secure by default" strategy and always use PE, keeping JW as a fallback for devices with no keyboard. This could be done, for example, by showing the pin on the glasses with a message "Tap on the glasses for unsecure pairing.". Then, if the user taps on the glasses[5], the BLE module is restarted with a JW configuration. Nevertheless, the need of entering a pin code or having to tap at each start could affect too much the user experience. Moreover, most of current customer projects are not dealing with critical secrets. As a consequence, we decided that aRdent will initially "Just Work" and it would be possible to enable the PE security after that.

The risks of using JW BLE pairing method

Using this pairing method exposes the connection to Man-In-The-Middle (MITM) attacks. [15] Indeed, there is no way of proving that the communication is end-to-end encrypted as, for example, in PE where a pin is generated on the screen of the glasses and is not transmitted over the air but encoded by the user in the paired device to use it as a temporary key. We can see PE as a simplified[6] OOB where the user is the secondary channel.

However, Man-In-The-Middle (MITM) attacks requires the attacker[7] to be close enough to the the devices so it can send faster responses than authentic devices. Moreover, it requires the attacker to be active for relaying packets from the beginning of the communication. That is why we did not consider this as a severe threat. For example, coming back to the restaurant example where a secret recipe is transmitted

[5] We can detect that thanks to an accelerometer interrupt.
[6] Because the key is shorter.
[7] Or simply the antenna of the attacker.

to the glasses, the attacker should have an antenna or be present in the restaurant at the pairing time, which could be before the restaurant even welcomes its first customers. This is not impossible, but current aRdent customers are working in secured industrial environments where intruders are normally not able to approach.

5.2.5 Implementation of BLE security for aRdent

Securing the Bluetooth pairing requires implementing some functions. The programmers guide provided with the BlueNRG module [15] explains in a very detailed way how to implement security pairing and bounding (cf. point 3.5) with example C code. Our BlueNRG chip having no computing capabilities ([14]), the keys have to be generated and persisted by the MCU.

Before creating the functions to generate and persist security keys at the first boot[8], we copy the example code in the BLE initialisation function in aRdent source code with arbitrarily chosen hard coded keys. This allows us to check if the encryption is effective and applied.

5.2.6 Testing: Eavesdropping BLE 4.2 communications

There are powerful devices that allow to eavesdrop Bluetooth communications with ease, but they are expensive. For example the Ubertooth One costs a little more than one hundred dollars at the time of writing [9]. And there are boxes capable of listening on multiple channels for tens of thousands of dollars.

An alternative solution is the Android's Bluetooth logging feature allowed by enabling Developer options. Then, as it is not possible on non-rooted devices to access the log file on the device, the command *adb bugreport dirname* is used to generate a bug report which creates a zip in the current directory containing the log in a file (on Android 12)

dirname.zip/FS/data/log/misc/bluetooth/BT/_HCI_2023_0726_133050.cfa.

This file can then be open with Wireshark or any other compatible network packets analysing tool. However, this way of capturing the traffic always gave non-encrypted clear text. It can also be tested by sending a known plain text (e.g., a text file with known content) over a secured connection (e.g., between a Windows computer and the Android phone) and checking the logs. One of the following two conclusions could be drawn from this: Either the HCI logs are never encrypted, or the Bluetooth connection between an Android phone and a Windows computer is not encrypted. We also tested it with two Android phones and the HCI log also contained the file in plain text.

To set out the real reason, we can use a USB dongle, less sophisticated than the Ubertooth presented above, and eavesdrop a Bluetooth communication. The device

[8] It must be done like that to avoid having hard-coded secrets in the firmware binary as discussed in the previous section.

[9] `Source:https://hackerwarehouse.com/product/ubertooth-one/`, Accessed on 2023-08-02.

is called *nRF52840-Dongle*[10] and it can be acquired for less than ten euros. This dongle allows to eavesdrop a communication at a time using the "nRF Sniffer for Bluetooth LE"[11] plugin in Wireshark. The result was that that it was impossible to sniff the connection between the PC and Android devices because they used the additional security of random addresses which prevents the dongle from properly sniffing the connection. However, the transfers between the glasses and the Android device were made in plain text.

5.2.7 Conclusions on aRdent BLE security

In this section, we first studied the security offered by the Bluetooth LE 4.2 specifications. Then, we have chosen the guarantees we wanted for aRdent and that were supported by our hardware. Finally, we tried to implement them using the developer guide provided by the manufacturer. However, we realized that the communications are still not encrypted and can be easily eavesdropped.

Fixing the security of the BLE pairing of aRdent requires more investigations that the time left for this Master' thesis permitted. Indeed, it would require to dive deep in the integration of the Bluetooth driver made by a precedent contractor company that did not document the code appropriately. For these reasons, this vulnerability must be reported by GYW to its customers and an effort must be made in order to correct it.

[10] Website: `https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle`, Accessed on 2023-08-20.

[11] Website: `https://infocenter.nordicsemi.com/pdf/nRF_Sniffer_BLE_UG_v3.1.pdf`, Accessed on 2023-08-20.

Chapter 6

Conclusions

6.1 Contributions to aRdent

The contributions brought by this thesis were to add some features and correct some bugs in the aRdent firmware and hardware. First, we allowed the users to use the display on their director eye thanks to the automatic rotation of the screen given the orientation of the accelerometer. Then, we fixed the display's LCD timings and resolution for a fully usable screen with no black borders. At the same time, we developed a workaround for a hardware error by the implementation of an I^2C driver that allow communications to take place even with reversed pins. Finally, we improved the security of the product starting by suggesting upgrades for the security of the organization itself, then by attempting to secure the Bluetooth connection between aRdent and the paired device.

Regarding the firmware code, these contributions were made in nine pull requests containing around one thousand lines of code written by me and more than ten thousands lines of code from third party libraries. Additionally, around 500 lines of documentation were written into markdown files to explain what was developed, how to debug and contribute to the project. The initial I^2C solution that was put apart contains a little bit over 800 lines of code. These numbers are not so big because the present work overall required more research than development. For example, calibrating the LCD timings only required to change six integer values in a C file but it took several hours of work.

6.2 Lessons learned

In this thesis, we have explored several domains such as organization security that could each be the subject of an entire thesis if studied in depth. Through this research, I have gained valuable insights into these domains and how they relate to each other. An example of such a relation is the conflict between the "secure by default" principle and the risk of non-acceptance of the technology because of the worse UX it proposes. We encountered this situation when implementing the secure Bluetooth pairing.

6.3 Future developments

This thesis also generates perspectives in each explored domain. First, the newly usable accelerometer and gyroscope can be exploited to provide more features than just changing the screen orientation. Second, the organization security documentation can be further grown to the point where Get Your Way can obtain an international certification for their security management. Finally, the Bluetooth pairing was not secured successfully. Thus, additional research can be made in that direction. Get Your Way is already developing their second iteration of the aRdent product where flaws identified in this thesis are solved and some peripherals are improved. For example, the Bluetooth module will be upgraded to a newer module supporting the version 5.0 of the protocol which is even more secure.

Bibliography

[1] Bluetooth: The bluetooth le security study guide | bluetooth technology website. https://www.bluetooth.com/bluetooth-resources/le-security-study-guide/, (Accessed on 08/20/2023) [Cited on page 35.]

[2] Butterfield, A., Szymanski, J.: A Dictionary of Electronics and Electrical Engineering. Oxford quick reference, Oxford University Press (2018) [Cited on page 24.]

[3] Davis, F.: A technology acceptance model for empirically testing new end-user information systems (01 1985) [Cited on pages V and 1.]

[4] ISO: Iso/iec 27001 and related standards — information security management. https://www.iso.org/isoiec-27001-information-security.html, (Accessed on 03/22/2023) [Cited on page 33.]

[5] Kaila, U., Nyman, L.: Information security best practices: First steps for startups and smes. Technology Innovation Management Review 8, 32–42 (11/2018 2018) [Cited on page 33.]

[6] Koelle, M., El Ali, A., Cobus, V., Heuten, W., Boll, S.C.: All about acceptability? identifying factors for the adoption of data glasses. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. p. 295–300. CHI '17, Association for Computing Machinery, New York, NY, USA (2017) [Cited on page 1.]

[7] Kopin: Golden pearl optical module. https://www.kopin.com/kopin-golden-pearl-optical-module/, (Accessed on 05/10/2023) [Cited on page 9.]

[8] Kress, B., Saeedi, E., de-la Perriere, V.B.: The segmentation of the HMD market: optics for smart glasses, smart eyewear, AR and VR headsets. In: Kazemi, A.A., Kress, B.C., Mendoza, E.A. (eds.) Photonics Applications for Aviation, Aerospace, Commercial, and Harsh Environments V. vol. 9202, p. 92020D. International Society for Optics and Photonics, SPIE (2014) [Cited on pages 1 and 2.]

[9] Microelectronics, S.: inemo inertial module: always-on 3d accelerometer and 3d gyroscope. https://www.st.com/resource/en/datasheet/lsm6dsm.pdf, (Accessed on 05/10/2023) [Cited on pages 8 and 9.]

[10] Porac, C., Coren, S.: The dominant eye. Psychological bulletin 83(5), 880 (1976) [Cited on page 13.]

[11] Renesas: Qe for display: Development assistance tool for display applications. https://www.renesas.com/us/en/software-tool/

qe-display-development-assistance-tool-display-applications, (Accessed on 03/03/2023) [Cited on page 18.]

[12] Renesas: Rz/a2m group user's manual: Hardware. https://www.renesas.com/us/en/document/mah/rza2m-group-users-manual-hardware?r=1054511, (Accessed on 05/10/2023) [Cited on page 8.]

[13] Simonovic, M., Saranovac, L.: Power management implementation in freertos on lm3s3748. Serbian Journal of Electrical Engineering 10, 199–208 (01 2013) [Cited on page 20.]

[14] STMicroelectronics: Flyer for BlueNRG Modules. Brochure (March 2020), https://www.st.com/resource/en/flyer/flbluenrgm0320.pdf [Cited on pages 8 and 38.]

[15] STMicroelectronics: PM0237 BlueNRG, BlueNRG-MS stacks programming guidelines. Documentation (March 2020), https://www.st.com/resource/en/programming_manual/pm0237-bluenrg-bluenrgms-stacks-programming-guidelines-stmicroelectronics.pdf [Cited on pages V, 8, 36, 37, and 38.]

[16] Swartz, E.E., Floyd, R.T., Cendoma, M.: Cervical spine functional anatomy and the biomechanics of injury due to compressive loading. J Athl Train 40(3), 155–161 (Jul 2005) [Cited on page 14.]

[17] Syberfeldt, A., Danielsson, O., Gustavsson, P.: Augmented reality smart glasses in the smart factory: Product evaluation guidelines and review of available products. Ieee Access 5, 9118–9130 (2017) [Cited on page 19.]

[18] Wikipedia contributors: State of charge — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=State_of_charge&oldid=1152972112 (2023), [Online; accessed 10-May-2023] [Cited on page 7.]

Appendix A

Get Your Way Security Management

Information Security @ Get Your Way

Application of a comprehensive extensible information security framework

Table of contents

Foreword

This document contains the artifacts related to information security at Get Your Way. For the moment, we use the simplified framework for information security proposed by Urpo Kaila and Linus Nyman in Technology Innovation Management Review, November 2018 (Volume 8, Issue 11). This is mandatory for being compliant with the GDPR.

Security Framework

When available time, consider using the following template:
https://wise-community.org/risk-assessment-template/

Assets (from the most valuable to the less)	Risks			Risk Owner	Mitigation/Control	Reviewed (Date & Initials)	Review Schedule
	Leak of Confidential information	Loss of Data	Service Down				
Google Drive, R&D content	Low	Low	Low	Head Security Manager	2FA? Strong Password? Service level Agreement ? Backups?	?	Annual
Bitwarden Password Manager	Low	Low	Low	Head Security Manager	2FA? Strong Password? Service Agreement , Local availability ? Encryption ?	?	Annual
Internal Management on Miro	Medium	Low	Low	Head Security Manager	2FA? Strong password? Service Agreement ? Protected Data?	?	Quarterley
Customer Database	…	…	…	…	…	…	…
…							

Protection guidelines for members

- **Password policy:** Minimum 12 characters, using at least a small case letter, a capital letter, a digit and a special character. Using words from any language dictionary as-is (eg. dog) is prohibited.
- Enforce usage of **2FA** for critical applications containing sensible information (assets).
- **Never plug an unknown USB thumb drive to your computer and personalize yours to distinguish them from mass.**
    - Eg.: https://www.youtube.com/watch?v=kfaHJwcG2mg
- Implement mandatory information security training for members about common cyber threats: Phishing, Social Engineering, safe online browsing.
- Only those who need accounts should have them. Be mindful of test accounts – make them as secure as other accounts (e.g., with strong passwords) and delete them once they have served their purpose.
- Remove accounts when no longer needed.
- Restrict rights on accounts to what is needed.
- Subscribe to a relevant cybersecurity newsletter to be alerted of new forms of threats or attack waves.

Protection guidelines for systems

- Ensure that you install security patches in a timely manner
- Shutdown all unnecessary services on your hosts. For example, shut down a local email service if you do not need it.
- Do not make your mission-critical infrastructure directly available on a public network.
- Implement layered defense. For example, do not expose confidential information directly to public networks.
- Ensure that all accounts are unique and can be connected to a person. Do not share accounts.
- Authenticate all users. Everybody must log in with a password or with a key.
- Log access and keep your logs on a separate host.
- Restrict network access with firewall rules, both on the network level and on host or service levels.

Protection guidelines for cloud-based assets

- Check what your provider promises you on security.
- Marketing materials are not enough – you should require security agreements.
- Make sure your providers have solid privacy policies.
- Check what guarantees are provided on the availability of the service and of your data.
- **Mandatory for GDRP:** Check the physical location where your data is stored, it must be in the EU for EU subjects-related data.

Protection guidelines for data

- Write down what critical data you need to be able to restore to recover your business.
- Have an automatic backup system in place.
- Test the backup system regularly, for example monthly, to ensure that the data really is being backed up and can be successfully restored.
- Check the integrity of your files and databases, too. Can you actually read what is restored?
- Mark and classify your data and your property. Write “confidential” or “internal” if the file is not public.
- Write a security policy for your staff. Company systems are intended only for business use; inappropriate use and abuse or causing harm is prohibited!

Business Continuity Plan (BCP)

Purpose and scope of the plan

This plan will allow Get Your Way to continue its activities in case of any critical hazard with clear descriptions of the information system and how it can quickly recover from any circumstance.

This plan allows to recover from the following incidents:

- Computer/data loss, ransomware, cyber attack
- Disruptions (eg. in telecommunications) due to severe weather conditions
- …

This plan can help with the first steps (max 24h) of the recovery process. Longer process should be planned with more granularity depending on the situation.

Description of Get Your Way

Description

Get Your Way information system refers to all the data of the company: R&D data, CRM data, Financial data, Accounting data.

IS Diagram

R&D Data -> GitHub
CRM Data -> Odoo
Financial Data -> Google Drive?
Accounting Data -> Google Drive?
Communications -> Gmail, Slack, Signal…

System components

Laptop of Pierre - Windows 11
Laptop of Antoine - Ubuntu
Laptop of Nicolas - MacOS

Dependencies (of other services and to other services)

Github, Google drive, Google Mail, Slack, Messenger, Odoo, BitWarden.

Contact Information

Administrative staff

Pierre J. <email perso> phone:
Antoine M. <email perso> phone:
Nicolas D. <email perso> phone:

Customers and other stakeholders

[Mettre ici contacts importants]

Subcontractors and agreements

Google: <contact?>
Github: <contact?>
[Personnes de contact des contrats en cours ici]

Risk and impact analysis

Risk analysis

See table 1.

List of related security controls

- Regularly check the logs.
- Regularly check the backups that they are recoverable.
- Check people awareness to cyber threats

Incident management guideline

Roles

Head Security Manager: In charge of cybersecurity [?]

Crisis communication

To be defined.

Contact with authorities and CSIRT teams

To be defined.

Disaster recovery plan

Computer loss

- [ ] Contact authorities
- [ ] Lock your computer from remote if possible
    - [ ] If not found in 24hours try to remotely remove the data
- [ ] Change all the passwords of the member
- [ ] Buy a replacement computer
- [ ] Recover the data from the most recent backup on the new computer

Data loss, encrypted data, ransomware

- [ ] Use the logs to try to identify the responsible/compromised account
- [ ] Lock the compromised account
- [ ] Perform virus analysis on the devices of the compromised account
- [ ] Recover the most recent backup with the data

Disruptions in telecommunications

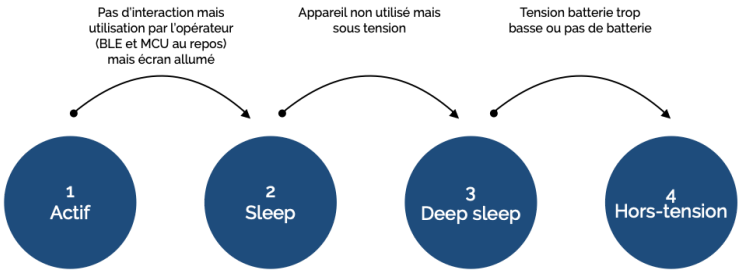
- [ ] Continue working in local or with “hard” means (pen and paper) on current activities
- [ ] Check the reason of disruptions: you could be victim of a MITM attack
- [ ] If identified a MITM attack, action your data breach communication plan. Investigate to precisely identify stolen data and alert relevant persons

Appendix B

Power optimisation suggestion

États système - lunettes aRdent

Les états possibles du système sont décrits dans l'ordre de la plus grande consommation à la plus petite consommation d'énergie.



1 - État Actif

Description globale

Fonctionnement normal avec toutes les fonctionnalités. Le MCU est occupé à traiter des données ou le module BLE reçoit des données d'affichage. C'est dans cet état que la lunette reçoit des informations et change l'écran de l'utilisateur.

Autonomie

Écran allumé et streaming vidéo BLE : minimum 2h

Composants actifs

Tous

Mesures hardware

Réalisation d'un système global low-power.

Condition de sortie

Aucune détection d'interaction dans les **30 secondes**.

- Pas d'échange via BLE
- Le MCU a fini ses tâches et l'écran reste identique