Faculté Polytechnique

Establishment of two Proof-of-Concept:

— monitoring and predictive maintenance tracking using machine learning;

— prototyping a low-cost VoIP phone using Raspberry Pi

Company internship: Company

Project Report

VIDOTTO Benoît

Under the direction of
Internship supervisor: AMAR Amar
Academic referent: MOEYAERT Véronique

Academic Year 2021-2022

Table des matières

Introduction

In order to allow Master 1 students to familiarize themselves with the life of an engineer in a company, the Polytechnic Faculty of Mons sets up internships in companies. This internship allows you to discover life in a company, the roles of the different people involved in the company, its internal operations, the different atmospheres and working methods, etc. The student, during these studies, has not had the opportunity to intervene or learn all these specificities. The internship therefore allows students who are starting their final year of study to develop their professional project and thus to launch themselves into professional life with more robust and complete preparation. In order for the internship to play a significant role in the student's training, the company must provide them with a professional project that could be created for an engineer employed in this company and carried out over 8 to 12 weeks.

The internship took place in the company See Telecom, based in Baulers and whose specialty is communication in confined environments such as tunnels or mines. As a project, the company has developed a subject called *Condition Monitoring and Predictive Maintenance based on Machine Learning* with the objective of creating an artificial intelligence whose goal is to predict possible breakdowns of the company's electronic equipment. The motivation is to reduce the costs related to the maintenance of the company's various machines.

During the realization of this project, various organizational problems and lack of resources have hampered the successful realization of this artificial intelligence. Thus, with the help and advice of the internship supervisor and the academic advisor, it was decided to move towards another more affordable project. This project is called *VoIP to audio decoder using Raspberry Pi* and consists of the creation of a prototype VoIP decoder using the multi-channel SIP protocol. This prototype will be produced using 4 Raspberry Pi and will allow the company to reduce the current costs resulting from the use of single-channel products existing on the market whose price is much higher.

This report first provides an analysis of the context of the internship, the activities and organization of the company as well as the issues and implications for the company during the internship. Then, this report provides a technical and scientific analysis of the two projects covered during the internship. The internship took place over 10 weeks, 4 of which were dedicated to the first subject while the last 6 were dedicated to the second subject.

Première partie

Internship framework and mission

Chapitre 1

Internship framework, context and issues

1.1 History and activities of the Company company

Company started in 1972 as a manufacturer of coaxial cable connectors and manufactured, among other things, connectors for radiating cables for the Eupen cable industry. A radiating cable is a perforated cable that allows radio signals to be broadcast to different locations, much like a perforated sprinkler cable can broadcast water to different locations. Company saw the use that these cables could have in confined spaces such as tunnels in order to provide radio signals. In concrete terms, an antenna outside a tunnel captures electromagnetic waves and broadcasts them via the radiating cable in the tunnel. At the same time, Company expanded its knowledge of optical fiber and, in order to improve radio wave coverage in tunnels, Company released its flagship product of the time : the SDR8 for Software Defined Radio, a device using a large amount of signal processing and allowing the signal to be repeated after attenuation in the cable. Today, Company continues to manufacture connectors for coaxial cables but the company is gradually reducing this category in favor of connectors for radiating cable. This product category has since become a small part of revenues due to Chinese competition. Recently, Company took a turn towards digital by offering software to accompany hardware equipment. The idea comes from the web giants (GAFAM) who are diversifying their operations towards digital proposals.

Company has customers all over the world : all tunnels in Belgium have radio coverage thanks to this company ; after the fire in the Mont Blanc tunnel in 1999, the company changed its slogan to "Connecting to life" [1] ; the company deals with tunnels in Vietnam, Colombia, Sweden, Denmark, Poland, etc. [2] ; the company also supplies tunnels other than road tunnels such as in mines, metro (Lille) and railway tunnels, offshore power stations and buildings ; another type of customer are the telecommunications networks dedicated to emergency services in countries such as Belgium (Astrid), the Netherlands (C2000), Germany (BDBOS), Luxembourg (Renita) and Austria (Tetron)[3].

When I arrived at Company, the focus was on a French motorway operator : a tunnel in Paris was causing some problems. Company provides Tetra (300 MHz or UHF for Ultra High Frequency) and FM (100 MHz for Very High Frequency) coverage for this client. The company supplied its digivas and digiguard products from the see smart range (IoT range) : the first allows emergency announcements to be made via radio using a switch and the second is digital surveillance management software [4].

Company is also leading the "SWITCH" project in collaboration with various Walloon companies and universities, including UMONS and Polytech, which consists of designing and developing an integrated "Smart Tunnel" solution for road tunnel constructors and operators. [5]

1.2 Department, decision levels and project integrations in the company

The research and development department is at the forefront of technology and experiments with all sorts of recent innovations in order to create the best products for the company. Currently, this department is developing a PoC (Proof-of-Concept) for the Mont Blanc tunnel between France and Italy in order to equip it with radiating cables to enable a 4G network in the tunnel, the availability of GPS signals and the support of platooning (IEEE 802.11p standard) [1].

Given that innovation is the driving force of this type of company, it is obvious that the RD department is therefore the heart of Company. Obtaining the budget necessary to complete a project is therefore one of the management's priorities and it will not hesitate to make the necessary investments if the usefulness of the project is proven. In order to obtain the various financing for the good of the 2 projects of this internship, direct or indirect interactions had to be made with the management and other financial managers. In order for a project to be financed, the RD director must judge the project useful to the company and will propose it to the financial director (Mr. Paul Goerg). If all the signals are green, a purchase order for the equipment useful to the project can then be made via the purchasing-sales manager (Ms. Dominique Dominique). The aim of the project from a management and company perspective is therefore, firstly, to study the feasibility of the predictive maintenance project in order to be able to reduce maintenance costs using the STWIN development kit and, secondly, to create a SIP decoder at a more competitive price than the solution currently used by the company.

1.3 Team Description

The internship supervisor Mr. Amar Amar is the director of the company's research and development department. A specialist in low-level programming (FPGA) and signal processing, he studied at the University of Burgundy in France and did an industrial doctorate in signal processing applied to telecommunications in a competitor company to Company. Mr. Thanh-Long Thanh-Long also provided assistance during the internship on the predictive maintenance project.

Mr. Thanh-Long studied computer science engineering in Vietnam and Australia and completed his PhD in engineering science and information technology at the Université Libre de Bruxelles. In the RD department, Mr. Thanh-Long is in charge of the software part by creating the back end part of the web interfaces of the various Company products. The subject of predictive maintenance was Mr. Thanh-Long's idea and he did some research on the subject before the intern arrived, which is why Mr. Thanh-Long will help Mr. Amar during the supervision of the first project.

During the internship, other people were able to be useful, such as the purchasing and sales manager (Ms. Dominique Dominique), some RD engineers (Mr. Marc Marc, ...) and the

1. Platooning allows a connection to be established between multiple heavy goods vehicles so that they can travel at shorter distances than under current regulations. Equipped with driver assistance and autonomous driving systems, platooning will eventually allow convoys of trucks requiring only one driver.[**TRUCKS**]

project managers who, through discussions due to the proximity in the open space, taught me various anecdotes about life in the company and about Company itself.

Chapitre 2

Project I : Condition Monitoring and Predictive Maintenance based on Machine Learning

In order to reduce maintenance costs, Company is looking to more effectively predict occasional breakdowns of its equipment. Company is therefore looking for a predictive technique for detecting breakdowns to replace its preventive technique (condition monitoring).

Condition monitoring Machine monitoring (condition monitoring) is of great importance in any type of industry, it allows to improve the quality of the product and to reduce downtime. Machine monitoring, using sensors such as temperature, pressure or movement, allows to alarm technicians and to plan maintenance (preventive maintenance).

Predictive maintenance Predictive maintenance seeks to improve machine monitoring and preventive maintenance by adding the concept of artificial intelligence, more precisely machine learning. Using these tools, predictive maintenance predicts potential breakdowns instead of preventing them, thus allowing less intensive machine monitoring but also less frequent maintenance, therefore resulting in lower costs, greater safety and fewer breakdowns. Predictive maintenance learning is done using the same sensors used for preventive maintenance and machine monitoring : pressure, temperature, movement sensors, etc.

To make it easier to create the machine learning algorithm, the company provides the STWINKT1B development kit from STMicroelectronics. It is also possible to develop artificial intelligence from scratch or from open-source examples.

The objective of this project is therefore to study the feasibility of a predictive maintenance system based on machine learning and adapted to Company equipment using the STMicroelectronics device or independently.

Chapitre 3

Project II : VoIP to audio decoder using Raspberry Pi

3.1 Changes during the internship

A change in the project topic was made during the internship. This change in content concerns the change in the internship project. The initial topic, entitled "Condition Monitoring and Predictive Maintenance based on Machine Learning", consists of using a commercially available computer card - taking advantage of the various sensors integrated into the device and the integration of artificial intelligence provided by the manufacturer - to perform predictive maintenance of electronic equipment based on artificial intelligence. During the first 4 weeks of the internship, the product was explored as a whole and it was determined that it was not suitable for the desired use. Thus, it was decided, with the agreement of the internship supervisor, the intern and the academic advisor, to branch off to another topic entitled "VoIP to audio decoder using raspberry PI".

3.2 Description

This project consists of creating a product whose ultimate goal would be to market it. The goal is to create an entity comprising 4 phones using the SIP protocol using Raspberry Pi cards (see figure 3.1). It is also necessary that calls to the Raspberry be made through a server equipped with the Asterisk program, thus allowing the created product to be directly compatible with the equipment already marketed by the company. The development of this product would allow the company to avoid using the similar TERRA-IEX product available commercially and whose costs are much higher.

In order to carry out this project, 4 Raspberry Pi, TERRA-IEX and ZYCOO X10 SIP telephones and a Company server including Asterisk software were made available.

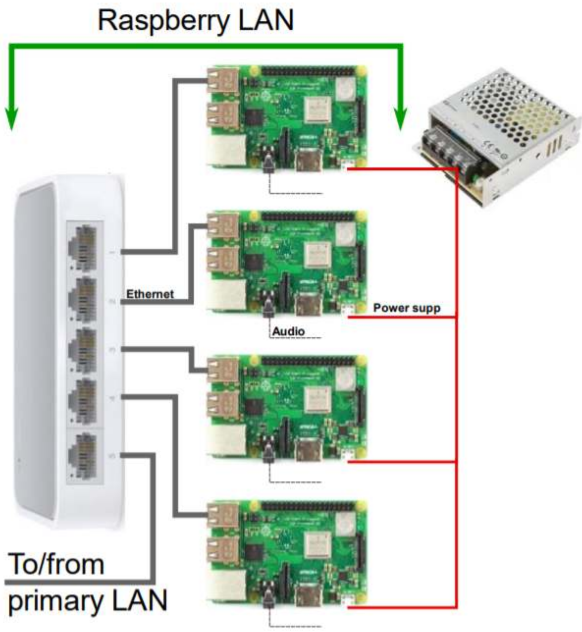


FIGURE 3.1 – Prototype diagram

Chapitre 4

Planning and feedback

The decision to change the topic was made during the 4th week with the help of Mr. Amar and Mrs. Moeyaert during a meeting reporting on past and future events. The catalysts for the change were the additional autonomy needed in the coming weeks due to Mr. Amar's departure on business and vacation. One of the reasons for changing the topic was the lack of feedback from the internship supervisors. Indeed, regardless of their will, Mr. Amar and Mr. Long are very busy in their daily tasks and spend their time "putting out fires" at different clients. Also, some problems with the progress of this project were the difficulty in obtaining data on Company's servers located at the clients. Thus, Mr. Amar, realizing his unavailability and the problems hindering my progress in this project, proposed the second topic. This second subject therefore allows me to be more independent given the familiarity of the subject and to work autonomously in the absence of the supervisor for professional reasons. Thanks to this second subject, I was able to receive various technical assistance from Mr. Amar and other colleagues present in the open space curious about the progress of the VoIP decoder as well as other advice allowing me to learn about corporate life, project management, or writing documents and preparing presentations for management.

	Week	Dates	Task
Machine learning	1	From 06/28 to 07/02	Bibliography and purchase of equipment
	2	From 5/07 to 9/07	Bibliography, experiments with existing solutions
	3	From 12/07 to 16/07	Experiments with the STWIN kit
	4	From 09/06 to 09/10	Experiments with the STWIN kit, writing of the report
Decoder VoIP	4	From 09/06 to 09/10	Bibliography, cost study and purchase of equipment
	5	From 09/13 to 09/17	Bibliography and cost study
	6	From 09/20 to 09/24	Appropriation of hardware
	7	From 09/27 to 10/1	Implementation of the software
	8	From 4/10 to 8/10	Software implementation and Tests and evaluations
	9	From 10/11 to 10/15	Tests and evaluations
	10	From 10/18 to 10/22	Writing of the report and presentation to the internship supervisor and his colleagues

Table 4.1 – Planning

The first 4 weeks were therefore dedicated to the first project involving predictive maintenance while the last 6 weeks were dedicated to the second project (see table 4.1). The predictive maintenance project having been planned over 10 weeks, this total number of internship weeks was therefore kept after the change of subject. Throughout the internship, I was able to have meetings on a few occasions to demonstrate the progress of the subject.

Deuxième partie

Scientific and technical report I
Predictive maintenance

Chapitre 1

State of the art

1.1 Definition of predictive maintenance

Through the industry revolution that is Industry 4.0 thanks to the Internet of Things IoT, more and more data is collected about the health of industrial equipment. The Internet of Things allows the creation of autonomous industrial systems. [6] present the various maintenance techniques within Industry 4.0 :

- Run-to-failure (R2F) : no maintenance, the equipment is running until it fails
- Preventive maintenance (PvM) : the equipment is periodically examined according to a schedule
- Condition-based Maintenance (CBM) : thanks to data collected by the IoT, one can analyze this data to determine when it is appropriate to carry out maintenance
- Predictive Maintenance (PdM) : Predictive maintenance provides an estimate of the health of the equipment and predicts when maintenance is required through continuous monitoring of the machine's condition and various sensors.

The goal is to categorize the different maintenance strategies and to be able to identify the most effective strategy in order to minimize the operating costs of the equipment by increasing its lifespan and reducing maintenance costs. The most effective maintenance in this case seems to be predictive maintenance. According to Baptista ([7]), the artificial intelligence approach seems to be more effective than the statistical approach in predictive maintenance. The artificial intelligence approach is done through machine learning, it is one of the most powerful tools when it comes to developing intelligent predictive algorithms for various applications. ML approaches have the ability to process high-dimensional and multi-variable data, and to extract hidden relationships in data in complex and dynamic environments in data in complex and dynamic environments ([7]). However, this artificial intelligence can have difficulty processing asymmetric data : in fact, every day, there are many more "non-failures" than failures. IBM Code uses the SMOTE algorithm to overcome this problem. Susto et al [8] draw up a model composed of multiple classifiers to manage this asymmetry in the data. They demonstrate that the use of multiple Support Vector Machine classifiers in parallel guarantees better performance compared to K-nearest-neighbours in terms of costs. [7] and Sustainability have studied the various scientific works in order to draw up the advantages and disadvantages of the different machine learning techniques, their results, challenges and opportunities in the context of predictive maintenance. They show that the most used methods are, from the most used to the least used, Random Forest, Artificial Neural Networks (i.e. ANN, CNN, LSTM and deep learning), Support Vector Machine (SVM) and k-means clustering. Furthermore, [7] emphasizes that to create an effective predictive maintenance algorithm, it is necessary to have implemented R2F or PvM techniques in the past in order to be able to collect the data.

Overall, the benefits of predictive maintenance are :

- Improving worker safety and the environment
- Increased Reliability
- Increased availability

— Improving product quality
— Reduced parts and labor costs
— Less waste in terms of raw materials and consumables, such as lubricants.
— Energy savings due to slower machines (e.g. alignment claimed in some cases 3

Also, the non-refutable/non-negligible advantages can be :

— Return on investment : 10 times;
— Reduction of maintenance costs : 25 % - 30 % (maintenance costs representing 15 % - 70 % of the total production cost, considerable savings can be achieved thanks to PdM);
— Fault Elimination : 70 % - 75 %;
— Downtime Reduction : 35 % - 45 %;
— Production Increase : 20 % - 25 %.

Finally, different techniques for detecting anomalies in equipment for predictive maintenance exist, such as vibration, oil, thermal, acoustic or other analyses ([9]).

1.2 Evaluating the performance of a machine learning model

Google [10] provides an introductory course for people who have basic knowledge of algebra and Python programming. This course explains how to evaluate the performance of a machine learning model. These performance evaluations are accuracy, precision and recall, and the ROC curve and AUC, the area under the ROC curve.

Accuracy refers to the proportion of correct predictions made by the model. This can be defined as :

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{VP + VN}{VP + VN + FP + FN} \tag{1.1}$$

where VP = True Positives, VN = True Negatives, FP = False Positives, and FN = False Negatives.

Precision can answer the question “How many of the selected items are relevant ?” while recall can answer the question “How many of the selected items are relevant ?”. Precision and recall can be expressed as :

$$\text{Precision} = \frac{VP}{VP + FP} \qquad \text{Reminder} = \frac{VP}{VP + FN} \tag{1.2}$$

Precision and recall are dependent on the classification threshold, that is, the threshold that determines when an item is considered a 1 or a 0 (example : a mail spam filter outputs a score of 0.6. Is it spam or not ?).

The Receiver Operating Characteristic (ROC) curve is a graph that plots the true positive rate (TPR = VP / (VP + FN)) versus the false positive rate (FPR = FP / (FP + VN)) values for different classification thresholds. The Area Under the ROC Curve (AUC) is the area below the ROC curve. This metric provides an aggregate measure of performance across all possible classification thresholds.

Chapitre 2

Equipment Overview

STWINKT1B, also known as STEVAL-STWINKT1B (STEVAL is short for "ST evaluation" because it is an evaluation tool and "STWINKT" is short for ST Wireless Industrial Node Kit), is a development kit created by ST that aims to simplify the prototyping and testing of advanced industrial IoT applications such as than machine monitoring and predictive maintenance. [11] Included in the kit are a plastic holder, a battery, a programming cable to connect the STLink to the STWIN, the STLink-V3Mini and the STWINKT1B itself visible in figure 2.1.

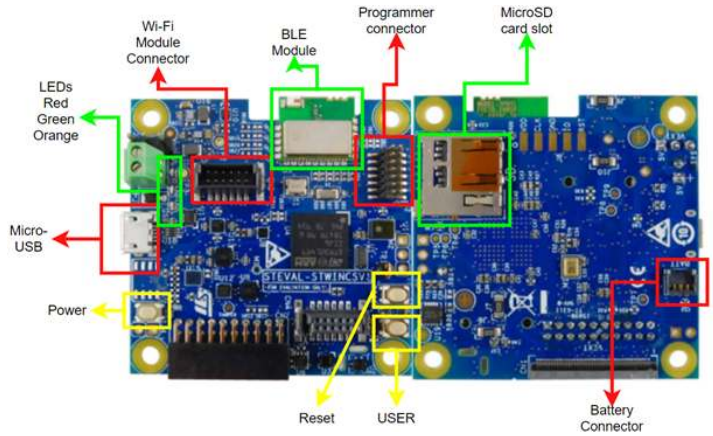


Figure 2.1 – STWINKT1B and its interfaces

The STWINKT1B has sensors of a humidity and temperature sensor, a MEMS (Micro Electro Mechanical Systems) pressure sensor, a digital temperature probe, a 3D accelerometer and 3D Gyroscope, a high-performance, ultra-low-power accelerometer, a 3-axis magnetometer, an analog MEMS ultrasonic microphone and a digital MEMS microphone. With the help of these sensors, the main functionality of the STWIN is to be a wireless sensing platform of multiple sensors and implementing vibration monitoring and ultrasonic detection. These sensors are exploited in the use of "Function packs", the subject of the next chapter.

Since STWIN does not have a Wi-Fi antenna, it is possible to add the STWINWFV1 expansion card to obtain this functionality.

Chapitre 3

Function Packs

In order to allow the user to become familiar with the card, ST provides different "function packs" which are guides created to highlight the different features of STWIN. The different function packs useful for this application are FP-AI-NANOEDG1, FP-IND-PREDMNT1, FP-SNS-DATALOG1.

3.1 NanoEdge (FP-AI-NANOEDG1)

FP_AI_NANOEDG1 is a feature pack that uses machine learning through the third-party program NanoEdge AI Studio from the French company Cartesiam. This program allows the user to do machine learning without any skills in data science, artificial intelligence and even programming [12]. In addition, each step of creating machine learning from STWIN is precisely documented in the getting started guide provided by ST, including information such as the software and hardware required and how to use the feature pack. In order for the artificial intelligence to learn the normal vibration pattern, the STWIN must be held in this vibration pattern during training. Thus during training, the STWIN was held in the hand, so the natural movement of the arm represents the normal vibration pattern.

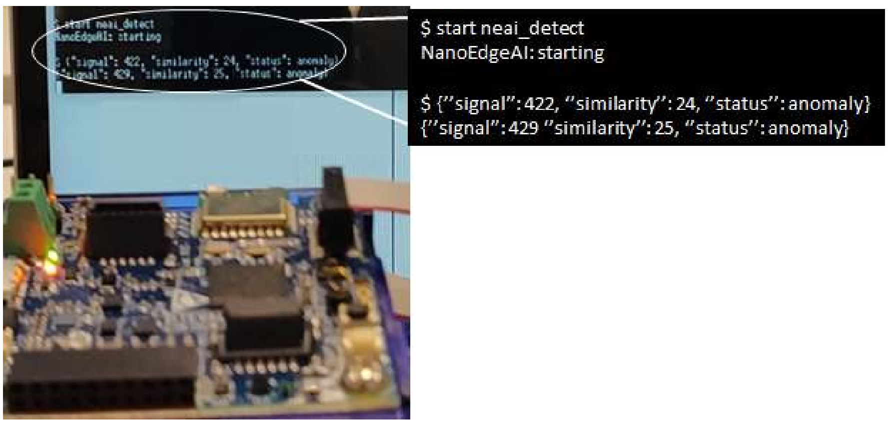


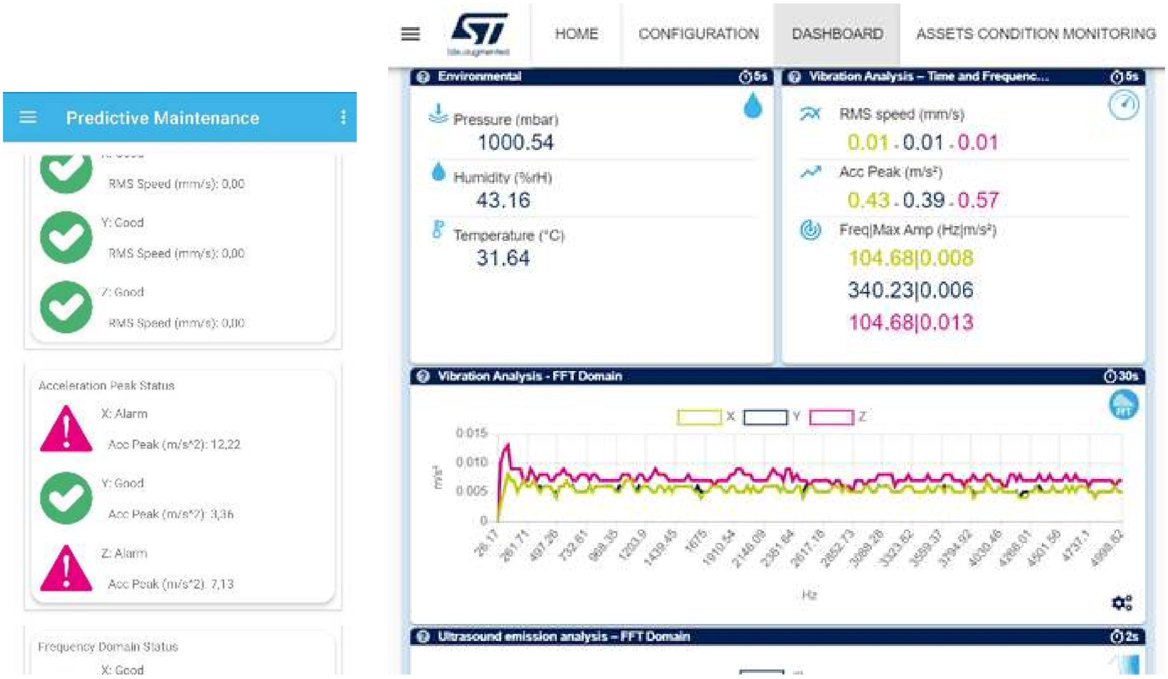
FIGURE 3.1 – AI test to detect vibration anomalies

Figure 3.1 is a snapshot of a video in which the AI test is shown. The figure shows the STWIN in the foreground and the console in the background displaying an anomaly message when the STWIN deviates from the vibration pattern learned by the AI. Since the normal vibration pattern is the natural movement of the arm, a hand shake is enough to create an abnormal vibration.

3.2 Predictive maintenance (FP-IND-PREDMNT1)

The FP-IND-PREDMNT1 predictive maintenance functional pack makes use of all sensors to gather information from these sensors in the form of a dashboard. This dashboard can be accessed in 2 different ways : the implementation for BLE using the ST BLE Sensor application and the predictive maintenance dashboard via the STWINWFV1 Wi-Fi module and cloud services, i.e. via a web browser.

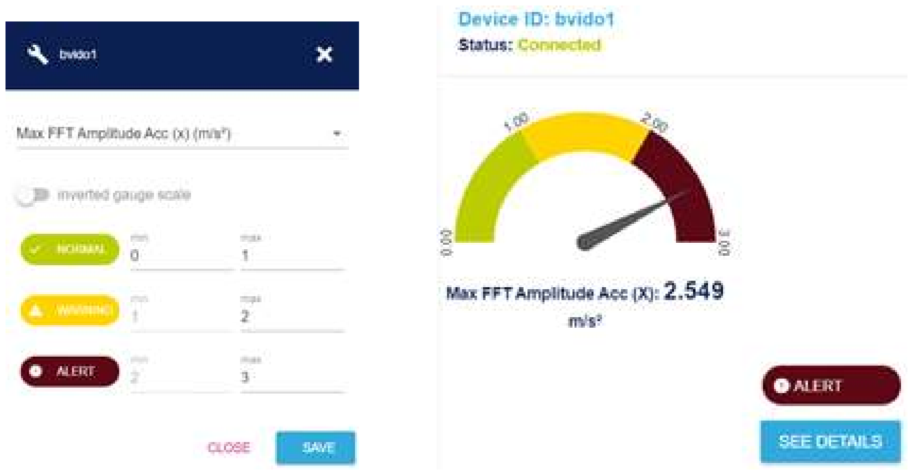
ST BLE Sensor When programmed correctly, the STWIN sends information to an Android or iOS phone equipped with the ST BLE Sensor app via BLE, Bluetooth Low Energy. The app shows the predictive maintenance statuses for the different sensors on the computer board, graphs showing the data from each sensor, an FFT amplitude graph and the connection status and battery level, all in real time. A screenshot of the app is available in figure 3.2 (a) and shows the home page when an anomaly appears from a vibration perspective via the device's accelerometer sensor.



(a) ST BLE Sensor (b) Web browser

FIGURE 3.2 – Predictive Maintenance Pack Dashboard

Web Browser Using the STWINWFV1 Wi-Fi expansion module, it is possible to connect the STWIN to ST servers via the Internet. Once the connection is established, the STWIN communicates the information from the different sensors via the MQTT protocol and the dashboard then becomes available to the user. The dashboard shows in real time the values of different STWIN sensors and associated graphs as seen in figure 3.2 (b). It is also possible to define thresholds on the desired sensors present on the board such as temperature, pressure, humidity, acceleration (along the axes) and noise sensors. In figure 3.3 (a), a threshold has been defined on the x-axis of the accelerometer. In figure 3.3 (b), an alarm has been activated when the STWIN accelerometer has exceeded the defined threshold. If the device location has been set manually, it is also possible to locate the STWINs in the world and thus determine where the alarms are located.



(a) Threshold on the accelerometer (b) Threshold exceeded alarm

FIGURE 3.3 – Defining a threshold on the web dashboard

3.3 Data Logging (FP-SNS-DATALOG1)

The FP-SNS-DATALOG1 function pack allows to record the data of the STWIN sensors in .dat files encoded in ASCII to be readable by any human being. This function pack offers 2 different ways to record the data : either via command line when the STWIN is connected via micro-usb, or via the ST BLE Sensor application. The data retrieval by command line is done via a Python script created by ST and the terminal emulator recommended by ST. It is thus possible to retrieve the complete data of each sensor in a few commands. To retrieve data via the ST BLE Sensor application, the user must first select the desired sensor, the .dat file will then be saved on the STWIN micro-SD card. Finally, this function pack does not mention it but it is also possible to retrieve data via the predictive maintenance table on web browser.

Troisième partie

Scientific and Technical Report II Voice over IP using Raspberry Pi

Chapitre 1

Hardware

1.1 Equipment Overview

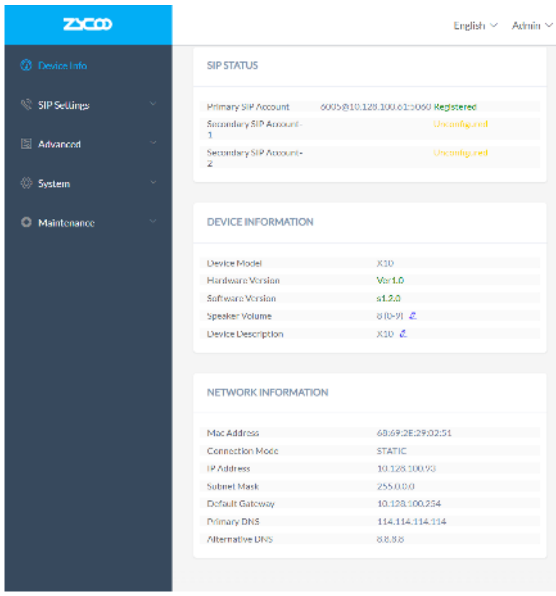
In order to have the best value for money for our SIP phone, different SIP phones existing on the market will be compared : ZYCOO X10, Terracom TERRA-IEX and the Raspberry Pi.

1.1.1 ZYCOO X10

ZYCOO is a developer and manufacturer of IP telephony devices whose customers mainly consist of small and medium-sized businesses. The company has enabled more than 100,000 customers in 100 different countries to develop their own VoIP communications solutions [13].

The ZYCOO X10, with its full name ZYCOO X10 SIP paging gateway, is a multifunctional SIP-compatible IP audio device that allows its users to convert audio streams from SIP or IP PBX systems into sounds such as background music or intercom. The device has multiple features such as a built-in amplifier, support for different HD audio codecs (G.722, G.711u, G.711a and G.729), power supply via the Ethernet port (PoE, Power over Ethernet), compatibility with third-party SIP servers, etc.[14] The X10 also has a graphical interface, accessible on any web browser using the device's IP address, on which all parameters can be modified, visible in figure 1.1(b). In figure 1.1(a) there is a photo of the device. The ZYCOO X10 costs between €123.6 ([15]) and €202.1 ([16]) excluding VAT.

(a) X10 (front and back)



(b) Graphical interface

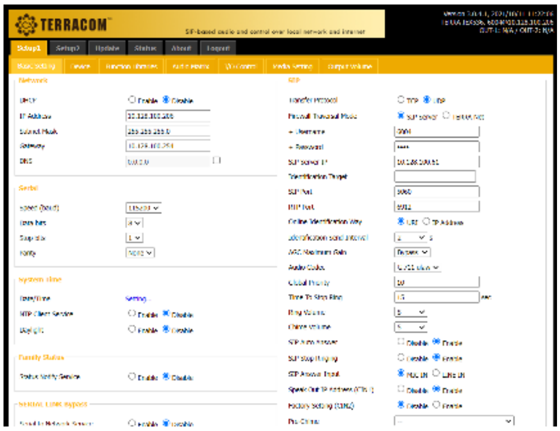
FIGURE 1.1 – Zycoo X10

1.1.2 Terracom TERRA-IEX

Terracom is a unit of PAX ProAV Group, a group of various brands and a global manufacturer of audio technologies and services. The group started as Penton, manufacturing loudspeakers for the Public Address market[1] Over the years, the group acquired various brands, which led to its name change in 2014. Each brand deals with a category of products such as voice alarms (Ateïs), intercom systems (Magellan), audio conference systems (Xavtel) and acoustic solutions (Messenger Pro) [18]. Terracom is the VoiceOverIP branch of the group, which therefore deals with equipping places such as airports, churches or hotels with IP telephony devices configurable via a standard web browser. Terracom is an international brand with customers all around the world from small to large [19].

The TERRA-IEX is a 2-channel audio over IP encoder/decoder that works as an audio source for an IP network using the SIP protocol, among others. The device has multiple features such as support for different HD audio codecs (G.722, G.711u, G.711a, G.729, G.726-32, G.727-32 and PCM16K), power supply via the Ethernet port (PoE, Power over Ethernet), compatibility with third-party SIP servers, support for the SNMP (Simple Network Management Protocol) protocol or recording and streaming of audio content. Just like the ZYCOO X10, the TERRA-IEX has a similar graphical interface visible in figure 1.2(b). The device costs between €500 ([20]) and €700 ([21]) and is visible in figure 1.2(a).

(a) TERRA-IEX (front and back)



(b) Graphical interface

FIGURE 1.2 – Terracom TERRA-IEX

1.1.3 Raspberry Pi

The Raspberry Pi Foundation is a charity whose goal is to bring the power of computing and digital manufacturing into the hands of everyone around the world. To do this, the charity created the Raspberry Pi, the first model of which was released in 2012, a cheap mini-computer the size of a credit card whose operating system is based on the Linux operating system. The Raspberry Pi benefits from a very large community that creates all sorts of open source

1. A Public Address refers to a device comprising a microphone and loudspeakers used for broadcasting (such as to an audience in an auditorium) [17].

projects based on this mini-computer such as NAS (Network Attached Storage), robots or a prosthetic arm [22].

There are several generations of the Raspberry Pi available for purchase, the most recent being version 4. For this project, version 3B+ was chosen : probably due to the current semi-conductor crisis, this model is the only one that could be found, regardless of the supplier, with an industrial grade. The penultimate version of the mini-computer was therefore purchased at a slightly higher price than normal but still cheaper than the latest version : Company thus acquired 4 Raspberry Pi industrial grade 3B+ for the total price of €138.88 excluding VAT, or €34.72 per Raspberry Pi. It should be noted, however, that this project could probably be carried out with older generation Raspberry Pis, thus lowering the cost of the finished product.

The Raspberry Pi 3B+ is therefore a computer board with a 64-bit quad-core ARM processor running at a frequency of 1.4 GHz, 1 GB of SDRAM, a dual-band Wi-Fi antenna at 2.4 or 5 GHz and Bluetooth 4.2/BLE. The board has an HDMI port, 4 USB ports, 40 GPIO pins, a micro-SD card reader, a micro-USB port for power, an Ethernet port and an audio jack port [23]. As this computer board is sold as a mini-computer, Raspberry Pi is supplied with an operating system called Raspbian : this is a free operating system based on the Debian operating system, itself based on Linux, and optimized for the Raspberry Pi [24]. Because of this dependence on the free Linux operating system, many programs and methods initially created for Linux work on Raspberry. Using the Twinkle program, the Raspberry Pi is able to support the following audio codecs for the SIP protocol : G.711a, G.711u, GSM, Speex (8, 16 and 32), iLBC and G.726 [25]. A representation of the Raspberry Pi 3B+ is available in figure 1.3.

FIGURE 1.3 – Raspberry Pi 3B+

1.1.4 Network Switch

In order to connect all the devices on a single ethernet cable, it is necessary to use a network switch, also known as a network switch, which will share the uplinks and downlinks from each port to all the other ports.

The network switch chosen for this project is the D-LINK DES-105/E. As shown in the figure 1.4, this network switch has 5 ports, allowing 4 SIP phones connected to their own ports to be connected to the network via an Ethernet cable connected to the 5th port. This equipment cost €23.14 excluding VAT.

FIGURE 1.4 – D-LINK DES-105/E

1.1.5 Food

To power the 4 SIP phones using a single power outlet, an AC/DC power supply is required. In order to be powered at low workloads and without USB devices connected to the board, the Raspberry Pi 3B+ need a power supply of 1.9 to 2.1 W : that is to say a voltage of 5 Volts and an amperage of about 400 mA [26]. Thus, the power supply needed to power 4 Raspberry Pi 3B+ must be able to provide a voltage of 5 volts and an amperage of 1.6 amps, or a power of 8 watts.

The choice was thus made on the RS-50-5 model from Mean Well (figure 1.5) which provides 50 watts at 5 volts and 10 amps. This amperage at first glance high allows the Raspberry not to be limited to a low load and to add other components if necessary. The power supply was thus acquired at the price of €20.16 excluding VAT.

1.1.6 Complete prototype

Figure 1.6 shows the whole prototype with in the background, the PC calling and doing the tests, and in the foreground from left to right, the power supply (USB power supply during the tests), the 4 Raspberry Pi, and the network switch. In this picture, the communication is done and the 4 Raspberry Pi are in group call (conference) with the calling PC.

1.2 Cost study

The goal of this project is to create a device with 4 audio outputs/inputs connected to each SIP phone. It is therefore necessary to have 4 SIP phones. In addition, in order for them to be connected to the network, it is more practical to connect everything using a single Ethernet cable, hence the use of the network switch. It is also more practical to connect each device

FIGURE 1.5 – Mean Well RS-50-5

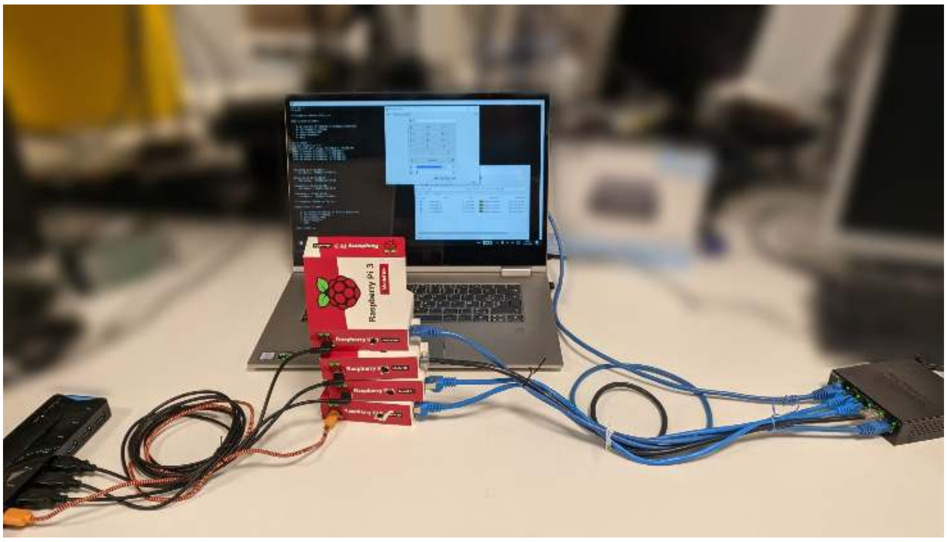


FIGURE 1.6 – Photo of the working Raspberry Pi prototype

using a power supply in order to give them the necessary energy. In this section, the costs of the 3 proposed architectures will be compared : Raspberry Pi, TERRA-IEX and ZYCOO X10.

Raspberry Pi As explained in the subsection **??**, the power supply chosen to power the 4 Raspberry Pi while having a sufficient safety margin is a power supply that provides 50 watts at 5 volts. In addition, the power supply must also power the network switch : the D-Link DES-105/E requires a power supply of 5 V at 0.55 A and therefore integrates perfectly with the chosen power supply. In total, the installation therefore costs €182.18 : 4 x €34.72 for the four Raspberry Pi 3 B+ ; €23.14 for the network switch ; and €20.16 for the power supply.

TERRA-IEX The TERRA-IEX requires a 24 volt power supply. The power supply used for the Raspberry is therefore no longer compatible but the price for this type of power supply remains similar to the 5 V power supply. To power the network switch, however, it is necessary to convert the 24 V voltage from the power supply into 5 V using a 24 V - 5 V DC/DC converter which costs around €5. Finally, taking into account this additional component, the cost of the power supply comes to around €26. According to these two sources ([20], [21]), the TERRA-IEX costs between €500 and €700 : for the cost estimate, the average of these two values will be retained, i.e. €600. In total, the installation therefore costs €2443.3 : 4 x €600 for the four TERRA-IEX ; €23.14 for the network switch ; and €26 for the power supply.

ZYCOO X10 The ZYCOO X10 requires a 12 V power supply. For the same reasons explained in the previous paragraph, the power supply — taking into account the 12 V - 5 V DC/DC converter — costs about €26. According to these two sources ([15], [16]), the ZYCOO X10 costs between €123.6 and €202.1 : for the cost estimate, the average of these two values will be retained, i.e. €162.85. In total, the installation therefore costs €694.7 : 4 x €162.85 for the four ZYCOO X10s ; €23.14 for the network switch ; and €26 for the power supply.

Conclusion Given the comparison table 1.1, it seems obvious that the prototype of 4 SIP phones composed of Raspberry is much more profitable than the prototypes composed of ZYCOO or TERRA : in fact, the Raspberry prototype is 13.4 times cheaper than that of TERRA and 3.8 times cheaper than that of ZYCOO.

	Raspberry Pi 3 B+	TERRA-IEX	ZYCOO X10
Device x 4	138.88	2400	651.40
Network Switch	23.14	23.14	23.14
Power supply	20,16	26	26
Total	€182.18	€2443.30	€694.70

Table 1.1 – Comparative table of costs according to different SIP phones

Chapitre 2

Software and tools

This chapter presents the different software and tools used to carry out this project.

Twinkle Due to the dependency of the Raspbian operating system on Linux, most software created for this free operating system is compatible with the Raspberry Pi. Twinkle is a good example : this program was initially created for Linux but works perfectly with the Raspberry Pi. Twinkle was the software chosen among others to use a Raspberry Pi as a SIP phone. One of the great qualities of this software is the possibility of using it through command lines, thus allowing to use the Raspberry Pi as a SIP phone without having to connect an external screen to the mini-computer. [25]

Asterisk Asterisk is a free open source Linux software designed to be used on servers to transform them into a communication server. Company already uses this open source software on its Digiguard servers to communicate on the network via a VoIP application. Asterisk can be used to equip servers with IP PBX systems [1], VoIP gateways, conference servers or others. This software is very well known for an IP communication solution and is used by a large number of companies. [28]

MicroSIP MicroSIP is the equivalent of Twinkle for the Windows operating system : it is a softphone for Windows that allows you to make high-quality VoIP calls via SIP from PJSIP (a set of protocols that allow the use of SIP). MicroSIP supports a large number of audio codecs that are useful for our project.

Spyder Spyder is an open-source development environment for the Python programming language : it is a software that allows you to compile the codes of this programming language, explore the variables present in the codes, display the generated images, debug the codes or receive help on the programming language [29]. The Python programming language is a very popular high-level language due to its ease of use that, with the help of the multiple different open-source libraries created by passionate users, offers endless possibilities in the field of programming and computing.

PuTTY PuTTY is an SSH client for Windows that allows you to communicate via command line with a compatible device via its IP address, in this case, the Raspberry Pi. It is actually a terminal emulator, meaning that through this program, you access the Raspberry Pi's command line terminal via a computer other than the Raspberry. [30]. PuTTY can also be used with the Digivas server that has the Asterisk software.

FileZilla While PuTTY allows you to communicate with the Raspberry Pi over the network, it is not possible to transfer files to and from the Raspberry Pi. To overcome this problem, FileZilla uses the File Transfer Protocol (FTP) to transfer files with the device [31]. Like

1. A PBX (private branch exchange) is a telephone switching system that redirects calls from one station to another. [27]

PuTTY, filezilla can be used for this project with both the Raspberry Pi and the Digiguard server.

VNC Viewer While PuTTY allows access to the Raspberry Pi terminal and FileZilla allows access to files, the VNC Viewer software allows access to the mini-computer's graphical interface without the need for office automation equipment such as a screen, keyboard and mouse [32]. Natively, the operating system has this software, making it an easy and quick solution to access the Raspberry Pi's graphical interface.

Raspberry Pi Imager The Raspberry Pi does not come natively with Raspbian OS in its storage memory. In reality, you have to add a micro-SD card containing Raspbian OS to the Raspberry Pi. This operating system must be formatted on the micro-SD card via the Raspberry Pi Imager software. Raspberry Pi Imager allows you to write any operating system compatible with the Raspberry Pi to the selected micro-SD card, the default operating system being Raspbian OS. [33]

Win32 Disk Imager Win32 Disk Imager is a software that allows you to create an ISO file from a storage device. Thus, the software allows you to clone the installation of a Raspberry Pi, including all the third-party files included in it, into a single file, to be subsequently rewritten on another additional Raspberry Pi. [34]

Advanced IP Scanner In order to access the Raspberry Pi or the Asterisk server, it is necessary to know their IP addresses. Advanced IP Scanner allows you to find the IP addresses of the different servers. This is a Windows software that scans the network of different devices connected to the same network. Once the network is scanned, the software offers a list of all the devices classified by IP address as well as their name and manufacturer, thus making it easier to identify them. [35]

SoX - Sound eXchange SoX is a command line utility that allows you to generate sounds based on the chosen frequency, sample rate, etc. [36] This program is useful for creating sounds that can be used to qualify the quality of SIP phones.

Wireshark Wireshark is a network protocol analyzer widely used in the world : this program allows you to analyze the packets that pass through the computer's Internet network. [37]

Chapitre 3

Configurations

3.1 Asterisk

Asterisk is a very complete telecommunications software, using its own programming language and whose complexity and scope allows some authors to create software help books. It is interesting to analyze the files that are important to the installation of Asterisk, such as some files with the extension .conf, which allow you to determine the different settings and options of the software. Each .conf file is located in the same directory at the path */etc/asterisk*.

asterisk.conf This asterisk.conf file contains the Asterisk folder options and assignments. The settings used here are the default settings as seen in the Asterisk documentation [38]. In order to avoid corrupting the installation, it is best to avoid modifying this file.

extension.conf The extension.conf file is used to determine the behavior to be performed when a SIP phone calls a certain number. For example, according to the 3.1 listing, calling the number 100 will play the message "Hello World" and then hang up (lines 1 to 5) while a call whose called number has 4 digits will redirect the call to the SIP user of the same number (line 6) defined in pjsip.conf.

```
1 exten = 100,1,Answer()
2 ;same = n,Wait(1)
3 same = n,Playback(hello-world)
4 same = n,Hangup()
5
6 exten => _XXXX,1,Dial(PJSIP/${EXTEN})
7 same => n,Hangup()
```

Listing 3.1 – extensions.conf

pjsip.conf SIP users are defined in pjsip.conf. The Company server is configured for 20 users in total : Alice, Bob, 5001 to 5010 and 6001 to 6010. This configuration file defines for each user its name and password. In addition, some users inherit settings from a higher class such as the allowed audio codecs (default codec is G.711u), the transport protocol (default protocol is UDP) or the maximum number of contacts. A complete example of pjsip.conf with explanations is available in the appendix D.

3.2 Windows computer serving as SIP phone

Leaving aside the process of setting up the Raspberry Pi and the tests to be performed, the only program to install to make the computer a VoIP phone is microSIP. Once microSIP is installed and opened, it is necessary to define the SIP user parameters that are compatible with the users specified in the pjsip.conf file of the Asterisk server as seen in the section 3.1. Figure 3.1 shows the parameters to be encoded where 6006 (Account Name, Username and

Login) and the associated password are defined in pjsip.conf and 10.128.100.61 (SIP Server, SIP Proxy and Domain) is the local address of the Asterisk server. Once the username is defined, microSIP makes a registration request to the Asterisk server. If the user is available and the password is correct, Asterisk registers the IP address to be able to redirect calls to the defined user.

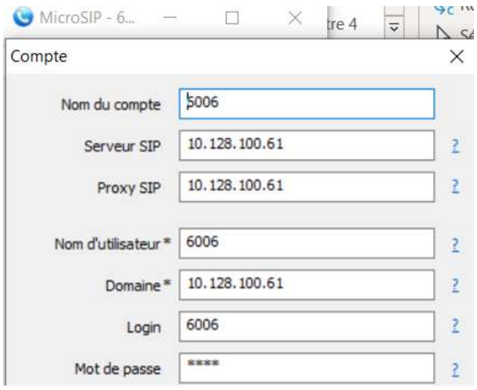


FIGURE 3.1 – microSIP Settings Window

3.3 Raspberry Pi

3.3.1 Installation

The complete configuration of the Raspberry Pi from a blank installation is long and contains multiple steps that are described in the appendix C. However, this configuration can be done very quickly using the .iso image file that reproduces the installation of a Raspberry Pi already configured for this application. To do this, simply insert a micro-SD card into the computer, obtain the .iso file containing the appropriate installation and write it to the micro-SD card using the Win32 Disk Imager software described in the chapter 2. This micro-SD card can finally be inserted into the Raspberry Pi to complete the installation.

In order to change the SIP phone settings as explained in the next section, you must first access the Raspberry Pi via its IP address. The IP address can be retrieved using the Advanced IP Scanner program. Advanced IP Scanner is a fairly simple program : the computer and the Raspberry Pi just need to be connected to the same network and the program will display the IP addresses of all the devices on the network after launching the scan. Once the IP address has been retrieved, it can be entered into PuTTY to launch an SSH session. Once the SSH session is open, PuTTY will ask the user for the Raspberry Pi username and password : by default, these are *pi* and *raspberry* respectively.

3.3.2 VoIP Configuration

Once the board is configured to be a SIP phone, it remains to modify the SIP user settings of the mini-computer. As a reminder, the application chosen to transform the Raspberry Pi into a SIP phone is Twinkle because of its ease of integration with the operating system and the support of the command line interface. Just like microSIP, once the username is defined, Twinkle makes the registration request to the server and, if the request is granted, Asterisk associates the username with the IP address of the Raspberry.

The first option to change these settings is to edit the Twinkle configuration files using PuTTY on the Raspberry Pi in the *.twinkle/* directory. However, the process of making such changes is cumbersome and can lead to syntax errors in the configuration files which could then corrupt the Twinkle installation.

When opening the Raspberry Pi terminal via PuTTY, a message is displayed explaining that certain commands allow you to modify Twinkle's settings. The second option is to use the Raspberry Pi's *UserEditor* command. This command guides the user to modify SIP settings such as the Asterisk server address or the username and associated password through different menus, offering considerable ease compared to the first option. Using this command actually leads to a Python script that was created for this project : for this option to be accessible, it is therefore essential that the Raspberry Pi has been installed using the .iso file mentioned above.

In order to avoid using PuTTY, it is possible to use the *Wizard.exe* file — or the Python script on which this .exe program was created — which offers the same possibilities as the *UserEditor* command but without going through the Raspberry terminal via PuTTY. To create this file in the Python programming language, the Paramiko library was used and allows to create an SSH connection as PuTTY would do using the IP address and the identifier and password of the Raspberry Pi. Also, another Python script exists on the Raspberry that allows to respond to requests sent by the *Wizard* file : for example, when modifying the SIP user, the *Wizard* file asks the Raspberry Python file what the current username is and *Wizard* displays the result of the operation to the user. As with the previous option, using this program requires certain files to be present on the Raspberry Pi and is only guaranteed to work when the Raspberry Pi has been installed with the appropriate .iso file.

Finally, the last option is to use the VNC Viewer program which allows access to the Raspberry's graphical interface if the function is enabled in the Raspberry settings, which is not the case by default. Furthermore, enabling this option could provide an additional access door for a malicious individual to modify the Raspberry Pi.

3.3.3 Security

As mentioned in the previous paragraph, using the VNC Viewer program could be a security hole. Furthermore, although the main argument for using Twinkle is the support of the command line interface, for the program to be launched at Raspberry startup, it must be launched via its graphical interface. This workaround implies that the graphical interface starts without having to enter the Raspberry username and password. In order to avoid possible abuse, it is therefore recommended to disable the HDMI port of the Raspberry Pi to prevent anyone from accessing it. This parameter change is done using the *HDMI* command similar to *UserEditor* in the previous section, or directly via the *Wizard* file. It is also necessary to change the Raspberry username and password according to Company's wishes.

3.4 Create a call

To create a call from the calling computer to the called Raspberry, simply call the SIP username defined in the Raspberry when the Raspberry and the computer are connected to the same network as the Asterisk server. Once the call is initiated, it is directed to Asterisk which redirects the call to the IP address associated with the called username. Figure 3.2 shows a call via microSIP with the 4 Raspberry whose usernames were retrieved using their IP address on the *Wizard.exe* program.

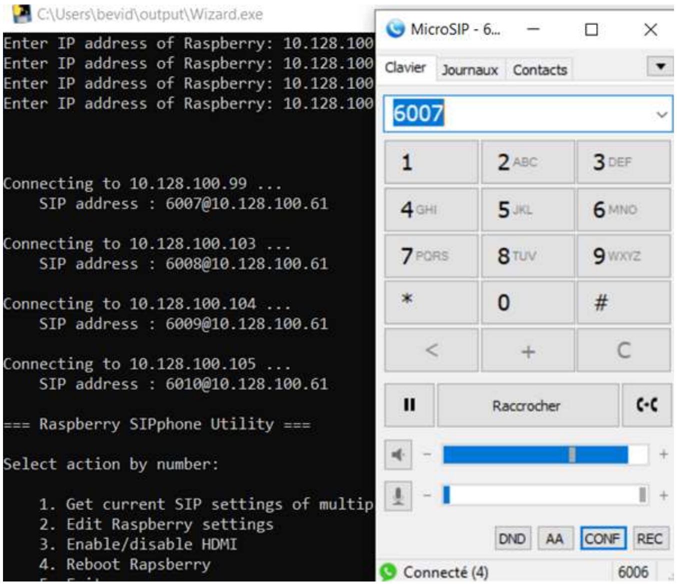


FIGURE 3.2 – Conference between the calling computer and the 4 Raspberry Pi

Chapitre 4

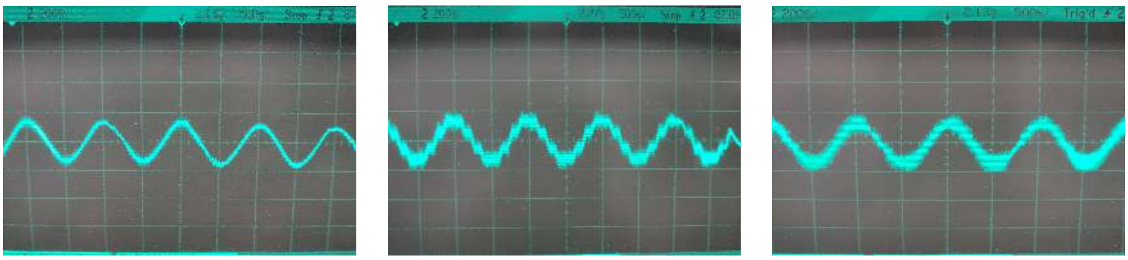
Results

4.1 Test protocol

In order to test the audio quality of SIP phones, it is necessary to analyze the difference between an audio file as it is sent and the same audio file as it is received. Latency is also a parameter that can be calculated. In order to take these two parameters into account, a Python script was created : using the PyAudio library, this script plays the audio file to be analyzed, calculates the time needed to receive the played audio file back, records the result and, using the Matplotlib and SciPy libraries, displays the Fast-Fourier Transform (FFT) of the received audio file compared to the sent audio file.

To send the audio file to the SIP phone, microSIP was configured to accept the Windows virtual microphone which allows to send the sound played by Windows itself (Realtek(R) Stereo Mix). To record the audio coming from the SIP phone, a jack cable with an adapter allowing the audio present on the L and R terminals of the jack cable to be redirected to the AUX terminal of the jack cable was used.

The tests were performed from a Windows computer using the microSIP softphone, an Asterisk server, and a SIP phone, i.e. the ZYCOO X10, the Terracom TERRA-IEX, or the Raspberry Pi with the Twinkle softphone. The volume of each device was normalized with an oscilloscope : Figure 4.1 shows the normalized signals of each device when a continuous 1000 Hz sound is played over VoIP. Raspberry Pi is at maximum volume, TERRA-IEX is at -10 dB volume, and ZYCOO is at level 7 on a scale of 9. Also, the computer performing the tests was set to a volume of 20 % to avoid saturation.



(a) Raspberry (b) TERRA-IEX (c) ZYCOO X10

FIGURE 4.1 – Outgoing signals from SIP phones at normalized volume on oscilloscope

The tests were performed with the G.711 (µ-law) audio codec because it is the default codec of the Company server. This codec is a standard of ITU (International Telecommunications Union) published in 1972 which supports a sampling rate of 8 kHz over a bandwidth of 300 to 3400 Hz [39]. A network analysis using the Wireshark program confirms that the packets sent have a sampling rate of 8 kHz .

4.2 Tests

4.2.1 Latency

Latency is of particular importance for an audio application like this : a latency of 30 ms is sufficient to be detected by humans [**Latency**]. This is why the Python script created to record the received audio and plot the FFTs includes a latency measurement. This measurement is the time difference on the computer performing the tests between sending the audio file to the SIP phone and receiving it from the same phone. The final measurements determining the latency for each device are the mean and standard deviation of 3 concurrent tests. For the Raspberry Pi, the test was performed on the 4 Raspberry Pis available and therefore there are 12 measurements instead of 3. The results of this test are shown in the table 4.1.

	Raspberry Pi (combined)	TERRA-IEX	ZYCOO X10
Average (µs)	522.28	465.42	742.48
Standard deviation (µs)	12.10	37.76	49.21

Table 4.1 – Means and standard deviations of the latency of each device after 3 trials (in µs)

In the table 4.1, the TERRA-IEX offers the best latency while ZYCOO X10 offers the worst latency. The Raspberry Pi has a latency just between the other 2 devices. However, the Raspberry Pi is superior to the other 2 on the standard deviation : this result could be biased by the larger number of tests for the Raspberry Pi but, individually, the standard deviations of the Raspberry Pi are between 1.67 µs and 21.16 µs, therefore always lower than the second lowest standard deviation. The ZYCOO X10 remains in last place for the standard deviation, preceded by TERRA-IEX.

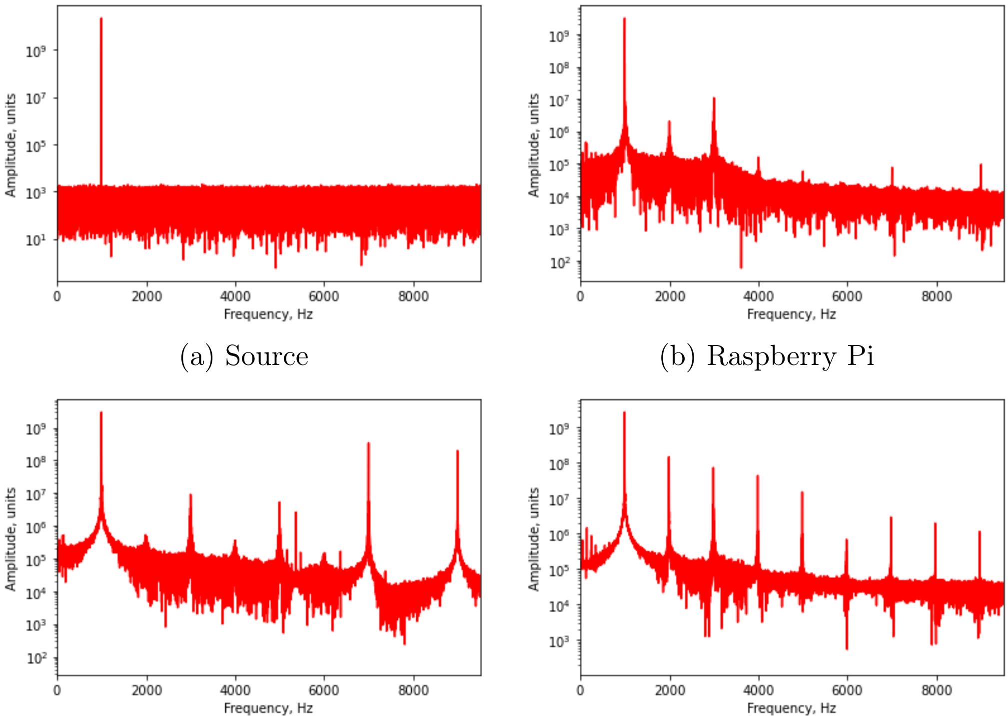
4.2.2 Audio Quality

The analysis of the signals used to normalize the volume of each device already allows us to qualify the audio quality : the signal sent to the SIP phones is a 1000 Hz sinusoid, so a noise-free signal should be a perfect sinusoid. Looking at the figure 4.1, the device that comes closest to this description is the Raspberry Pi. The sinusoid is also visible on the other 2 devices but the line is bolder, indicating a higher noise level.

The analysis of the FFT graph of a fixed frequency sound also allows to qualify the audio quality. Only the fundamental frequency is visible on the FFT graph of such a sound. On the other hand, when the transmission quality decreases, harmonics appear. The test consists of sending to the different devices a signal with a frequency of 1 kHz at a sampling rate of 8 kHz carried out using the SoX software. The FFT graphs of this test are shown in figure 4.2. Since the results of the tests carried out on each Raspberry Pi are very close to each other, it was decided to display the results of only one Raspberry Pi out of the four.

Figure 4.2 shows the results of sending a 1 kHz signal sampled at 8 kHz ((a) Source) to the 3 different SIP phones. The Raspberry Pi offers the best audio quality according to this test because the FFT graph only shows a single low amplitude harmonic at 3000 Hz. The ZYCOO X10 offers the second best audio quality : the device shows lower amplitude harmonics than the TERRA-IEX.

Since the chosen audio codec offers a bandwidth of 300 to 3400 Hz, analyzing a sweep of all frequencies over this bandwidth is a good test. Audio quality is better when the frequency



(a) Source (b) Raspberry Pi

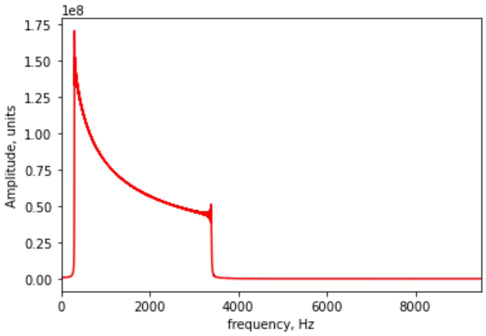
(c) TERRA-IEX (d) ZYCOO X10

FIGURE 4.2 – FFT plots of a 1 kHz signal sent to each SIP phone

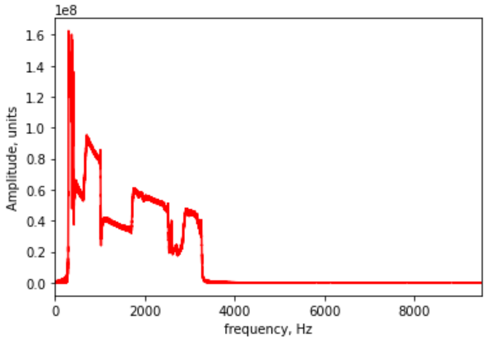
curve is closer to the reference and the energy is distributed only in the analyzed bandwidth. For this test, a 30-second audio file sweeping the frequency band from 300 to 3400 Hz and created with the SoX software was sent to the SIP phones. The results shown in FFT graphs are shown in figure 4.3.

The results of this test in figure 4.3 show that the Raspberry Pi fits the reference curve the least well. On the other hand, the Raspberry Pi distributes the least energy outside the considered bandwidth. TERRA-IEX fits the reference curve better but distributes the most energy outside the considered bandwidth compared to the other devices : the cause seems to be a parasitic carrier at 4000 Hz. ZYCOO X10 distributes the energy less and fits the reference curve better.

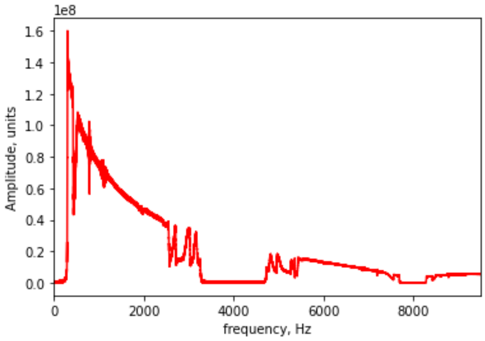
Finally, listening to the audio content coming out of the SIP phones allows us to assess the audio quality : on this test, the Raspberry Pi is superior, then comes the ZYCOO X10 and finally the TERRA-IEX whose audio quality suffers from a large amount of noise. This listening test by ear is in agreement with the other tests : the Raspberry has the best audio quality (low amplitude harmonics during the 1 kHz test and no energy distribution outside the considered bandwidth during the sweep test), then comes the ZYCOO X10 and finally the TERRA-IEX (high amplitude harmonics during the 1 kHz test and a lot of energy distributed outside the considered bandwidth during the sweep test).



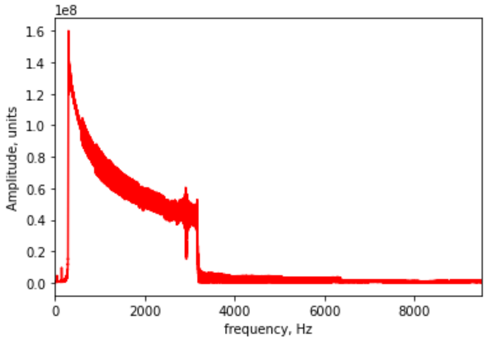
(a) Source



(b) Raspberry Pi



(c) TERRA-IEX



(d) ZYCOO X10

FIGURE 4.3 – 300 Hz to 3400 Hz

Chapitre 5

Prospects for improvement

In order to help the company continue the project if desired, various improvement perspectives can be presented. Figure 5.1 shows the top view of a box in which the network switch and power supply are arranged vertically and the Raspberry Pi are stacked. The box would therefore adopt minimum dimensions of 15x15x15 centimeters (without taking into account the inelasticity of the cables) and would have 6 different ports : a power port, an Ethernet port and 6 jack ports. Ideally, the box would also have provisions for mounting it in server cabinets.

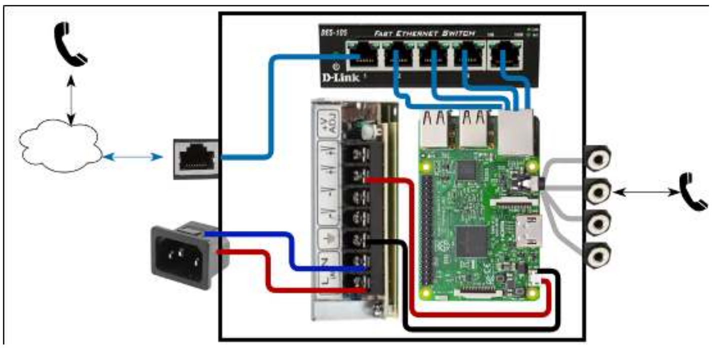
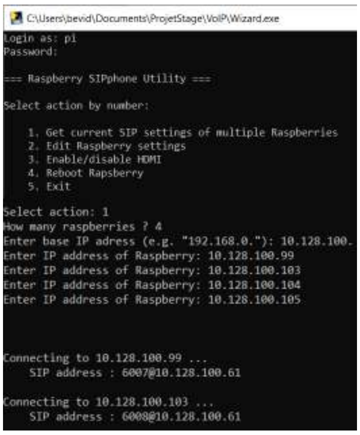


FIGURE 5.1 – Box adapted to the complete prototype

Figure 5.2 shows useful interfaces to conveniently change Raspberry Pi settings remotely on an external computer. Figure 5.2(a) shows a basic inline code interface already made demonstrating the feasibility of the dummy web GUI located in figure 5.2(b) whose design was inspired by existing Company products.



(a) Basic interface

Images/webInterface1.png

(b) Proposed graphical interface

FIGURE 5.2 – Graphical interfaces

Conclusion

The possibilities offered by ST through its STEVAL-STWINKT1B development kit were explored through the different function packs FP_AI_NANOEDG1, FP-IND-PREDMNT1 and FP-SNS-DATALOG1. The predictive maintenance function pack (FP-IND-PREDMNT1) does not contain predictive maintenance but rather preventive maintenance and therefore allows to monitor the sensors incorporated in STWIN from anywhere in the world on different platforms, the data logging function pack (FP-SNS-DATALOG1) retrieves data from all sensors on the microSD card or on a connected computer, and the NanoEdge function pack (FP_AI_NANOEDG1) can create artificial intelligence for predictive maintenance using vibration sensors.

The latter is the most promising feature pack for predictive maintenance : it can work with vibration data where it can learn vibration patterns. However, this predictive maintenance based on NanoEdge artificial intelligence is limited to the sensors present on the STWIN module. The STWIN therefore does not fit with the project defined by Company : none of the sensors present on the STWIN are compatible with predictive maintenance for the company's electronic equipment.

The last possibility to carry out this project would therefore be to create an artificial intelligence from scratch using the alarm data already recorded to perform preventive maintenance. On the other hand, the absence of scientific literature on predictive maintenance applied to this equipment suggests that the creation of such artificial intelligence is premature for Company. Thus, in order to carry out this internship, it was decided to modify the subject of the internship and to branch off on the subject VoIP to audio decoder using Raspberry Pi.

In the audio quality tests, the Raspberry Pi stood out from its two competitors while its latency is in the average of the tests carried out on the 3 devices. The Raspberry Pi is many times cheaper than its competitors in the creation of a functional prototype. For all these reasons, the Raspberry Pi is a very good candidate for the commercialization of such a device. It is nevertheless necessary to pay attention to security in order to prevent an outside person from taking control of the Raspberry Pi. In addition, the prototype could be improved by creating a secure web interface to control the computer board remotely : the Python script *Wizard* demonstrates the feasibility of such an interface. The .iso image file allows a quick installation of programs on the computer board. Finally, the entire prototype — the 4 Raspberry Pi, the network switch and power supply — could be conveniently arranged in a box in order to allow its commercialization.

Bibliographie

[1] Company. *About Us*. URL : https://www.Company.com/about-us (visité le 14/07/2021).

[2] Company. *LATEST WORKS*. URL : https://www.Company.com/latest-works (visité le 14/07/2021).

[3] Company. *Public Safety*. URL : https://www.Company.com/products/public-safety (visité le 14/07/2021).

[4] Company. *Company releases its new software suite*. URL : https://www.Company.com/news/311-see-telecom-announces-software-suite (visité le 14/07/2021).

[5] Logistics in wallonia. *SWITCH | Logistics in Wallonia*. Mai 2021. URL : https://www.logisticsinwallonia.be/fr/projets/switch (visité le 25/10/2021).

[6] Zeki Murat Çinar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael et Babak Safaei. « Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0 ». In : *Sustainability* 12.19 (2020). ISSN : 2071-1050. DOI : 10.3390/su12198211. URL : https://www.mdpi.com/2071-1050/12/19/8211.

[7] Thyago P. Carvalho, Fabrízzio A. A. M. N. Soares, Roberto Vita, Roberto da P. Francisco, João P. Basto et Symone G. S. Alcalá. « A systematic literature review of machine learning methods applied to predictive maintenance ». In : *Computers & Industrial Engineering* 137 (2019), p. 106024. ISSN : 0360-8352. DOI : https://doi.org/10.1016/j.cie.2019.106024. URL : https://www.sciencedirect.com/science/article/pii/S0360835219304838.

[8] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone et Alessandro Beghi. « Machine Learning for Predictive Maintenance : A Multiple Classifier Approach ». In : *IEEE Transactions on Industrial Informatics* 11.3 (2015), p. 812-820. DOI : 10.1109/TII.2014.2349359.

[9] Sule Selcuk. « Predictive maintenance, its implementation and latest trends ». In : *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture* 231.9 (2017), p. 1670-1679. DOI : 10.1177/0954405415601640. eprint : https://doi.org/10.1177/0954405415601640. URL : https://doi.org/10.1177/0954405415601640.

[10] Google Developers. *Machine Learning Crash Course | Google Developers*. URL : https://developers.google.com/machine-learning/crash-course (visité le 07/07/2021).

[11] STMicroelectronics. *STEVAL-STWINKT1B - STWIN SensorTile Wireless Industrial Node development kit and reference design for industrial IoT applications - STMicroelectronics*. URL : https://www.st.com/en/evaluation-tools/steval-stwinkt1b.html (visité le 14/10/2021).

[12] STMicroelectronics. *FP-AI-NANOEDG1 - Artificial Intelligence (AI) condition monitoring function pack for STM32Cube - STMicroelectronics*. URL : https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/fp-ai-nanoedg1.html (visité le 14/10/2021).

[13] Zycoo. *About Us | Zycoo*. URL : https://www.zycoo.com/company/about-us/ (visité le 08/10/2021).

[14] ZYCOO. *X10, ZYCOO SIP Paging Gateway*. 2020. URL : https://zycoodocs.s3-ap-southeast-1.amazonaws.com/datasheet/X10_SIP_Paging_Gateway_Datasheet_v1.2.1.pdf (visité le 08/10/2021).

[15] VoIPdistri.com. *Zycoo X10 SIP PA Gateway (G.722, OPUS, G.711 Codec), 148,27 €*. URL : https://voip.world/navi.php?a=3054&lang=eng (visité le 08/10/2021).

[16] businesscom. *Zycoo X10 SIP PA gateway to a normal environment www.businesscom.cz*. URL : https://www.businesscom.cz/detail/1111/zycoozycoo-x10-sip-pa-brana-do-bezneho-prostredi.html (visité le 08/10/2021).

[17] Bruce Borgerson. *Is it P.A. or SR ?* URL : https://web.archive.org/web/20150520193257/http://mylocalnews.com/pa-system/is-it-p-a-or-sr/ (visité le 11/10/2021).

[18] PAX ProAV Group. *PAX ProAV Group*. URL : http://www.paxproavgroup.com/ (visité le 08/10/2021).

[19] PAX ProAV Group. *About Us - SIP-Based Audio Distribution over IP - TERRACOM*. URL : http://www.paxproavgroup.com/terracom/about?l=en_US (visité le 08/10/2021).

[20] AVW Broadcast Pty Ltd. *AVW : Broadcast, RF and Audio Visual Equipment Wholesalers*. URL : https://www.avw.com.au/catalog/iptv-digital-signage-digital-matv/TERRA-IEX_1terracom-terra-iex-2-channel-audio-over-ip-encoder-decoder (visité le 08/10/2021).

[21] ITDealer. *Product Details : "ATEÏS Terra-IEX decoder Wired Black" | ITDealer.com*. URL : https://www.itdealer.com/product/view/1464358 (visité le 08/10/2021).

[22] Raspberry Pi Foundation. *Raspberry Pi Foundation - About Us*. URL : https://www.raspberrypi.org/about/ (visité le 08/10/2021).

[23] Raspberry Pi Foundation. *Buy a Raspberry Pi 3 Model B+ – Raspberry Pi*. URL : https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/ (visité le 08/10/2021).

[24] Raspbian. *FrontPage - Raspbian*. URL : https://www.raspbian.org/ (visité le 08/10/2021).

[25] Michel de Boer. *Twinkle - SIP softphone for Linux*. URL : https://mfnboer.home.xs4all.nl/twinkle/index.html (visité le 09/10/2021).

[26] Shahriar Shovon. *Raspberry Pi 3 Power Requirements*. URL : https://linuxhint.com/raspberry_pi_3_power_requirements/ (visité le 11/10/2021).

[27] Cisco. *Qu'est-ce qu'un PBX ? Définition - Cisco*. URL : https://www.cisco.com/c/fr_fr/solutions/small-business/resource-center/collaboration/what-is-a-pbx.html (visité le 11/10/2021).

[28] Sangoma Technologies. *Get Started *Asterisk*. URL : https://www.asterisk.org/get-started/ (visité le 11/10/2021).

[29] Spyder. *Home — Spyder IDE*. URL : https://www.spyder-ide.org/ (visité le 11/10/2021).

[30] Simon Tatham. *Download PuTTY - a free SSH and telnet client for Windows*. URL : https://www.putty.org/ (visité le 11/10/2021).

[31] *FileZilla - The free FTP solution.* URL : https://filezilla-project.org/ (visité le 11/10/2021).

[32] RealVNC. *Télécharger VNC Viewer depuis Raspberry Pi | VNC Connect.* URL : https://www.realvnc.com/fr/connect/download/viewer/raspberrypi/ (visité le 11/10/2021).

[33] Raspberry Pi Foundation. *Raspberry Pi OS – Raspberry Pi.* URL : https://www.raspberrypi.com/software/ (visité le 11/10/2021).

[34] Win32 Disk Imager. *Win32 Disk Imager Download (Latest Stable Version).* URL : https://win32diskimager.download/ (visité le 11/10/2021).

[35] Famatech Corp. *Advanced IP Scanner – Téléchargez un analyseur de réseau gratuit.* URL : https://www.advanced-ip-scanner.com/fr/ (visité le 11/10/2021).

[36] Patrick Michaud. *SoX - Sound eXchange | HomePage.* URL : http://sox.sourceforge.net/ (visité le 12/10/2021).

[37] Wireshark. *Wireshark · Go Deep.* URL : https://www.wireshark.org/ (visité le 13/10/2021).

[38] Sangoma Technologies. *Asterisk Main Configuration File - Asterisk Project - Asterisk Project Wiki.* URL : https://wiki.asterisk.org/wiki/display/AST/Asterisk+Main+Configuration+File (visité le 11/10/2021).

[39] W3TEL. *G.711.* URL : https://www.w3tel.com/voip/g711.html (visité le 12/10/2021).

Appendices

Annexe A

Getting Started STWINKT1B

This appendix was created during the internship to help the internship supervisors to update themselves in order to better supervise me during my absence intended for the second exam session. This appendix was to be accompanied by a .zip file which is mentioned several times. In addition, the appendix has multiple internet links which are in bold in order to distinguish them.

A.1 Introduction

STWINKT1B, also known as STEVAL-STWINKT1B, is a development kit created by ST that aims to simplify the prototyping and testing of advanced industrial IoT applications such as as condition monitoring and predictive maintenance. ST provides some examples on how to use this development kit by the through feature packs : each feature pack demonstrates different features of the STWINKT1B. In each pack of functions, ST provides comprehensive getting started guides to simplify prototyping for the user. This report will add information in parallel with the guides provided by ST : all functional packs have have been explored in detail, the problems due to lack of information are reported and advice is given to facilitate the use of all the features of the STWINKT1B.

A.2 Available resources

All information collected was found on the website of ST. However, for ease of reference, each document has been downloaded in the attached .zip file in the folder *INTERNSHIP21AI.* The .zip file includes the *Documentation* folder, which includes the data sheets and quick start guides, the file *Functionpacks*, which must be downloaded from the website ST, and the *results* of these functional packs. Overall, all useful documents have been downloaded in .zip files, but for ease of use, the links in the document are will direct you to download them from ST. You will need to also download the functional packs, the links of which are **FP-AI-NANOEDG1**, **FP-IND-PREDMNT1**, **FP-SNS-DATALOG1**.

A.3 Computer hardware

A.3.1 Kit Containers

The plastic cases included in the box are separated into two parts : the smaller one is separated from the main one in order to add magnets to attach the card to equipment.

Also included are a battery, a programming cable to connect the STLink to the STWIN, the STLink-V3Mini and the STWINKT1B itself. Images of each are available on pages 2 and 3 of this **document** and assembly is described on page 23.

A.3.2 Sensors

STWIN includes some sensors :

— Humidity and temperature sensor (HTS221)
— MEMS pressure sensor (LPS22HH)
— Digital Temperature Probe (STTS751)
— 3D Accelerometer and 3D Gyroscope (ISM330DHCX iNEMO IMU)
— High-performance, ultra-low-power accelerometer (IIS2DH)
— 3-axis magnetometer (IIS2MDC)
— Analog MEMS Ultrasonic Microphone (IMP23ABSU)
— MEMS digital microphone (IMP34DT05)

A.3.3 Interfaces and Connectivity useful

The STEVAL-STWINKT1B card, where STEVAL is short for "ST evaluation" because it is an evaluation tool and "STWINKT" is the abbreviation of ST Wireless Industrial Node Kit, has various connectors, buttons and LEDs. Those that deserve to be mentioned for this guides are :

— the battery connector ;
— Micro-SD card slot ;
— USER, RESET and POWER buttons ;
— Red, orange and green LEDs. The red LED indicates the status of the Battery : When the card is connected by USB, the battery is on charging and the indicator light is flashing. The green light is an indicator activity. The orange light indicates the presence of the SD card : if the SD is not present, the LED flashes ;
— Male extension connector : to connect the Wi-Fi module STWINWFV1, not included in the box ;
— Programmer Male Connector : For connecting the STLink-V3Mini with the programming cable, both included in the box of the STWINKT1B ;
— BlueNRG-M2SA : BLE (Bluetooth Low Energy) module.
— Micro-USB connector.

Figure A.1 illustrates all the interfaces that will be used throughout along this guide.

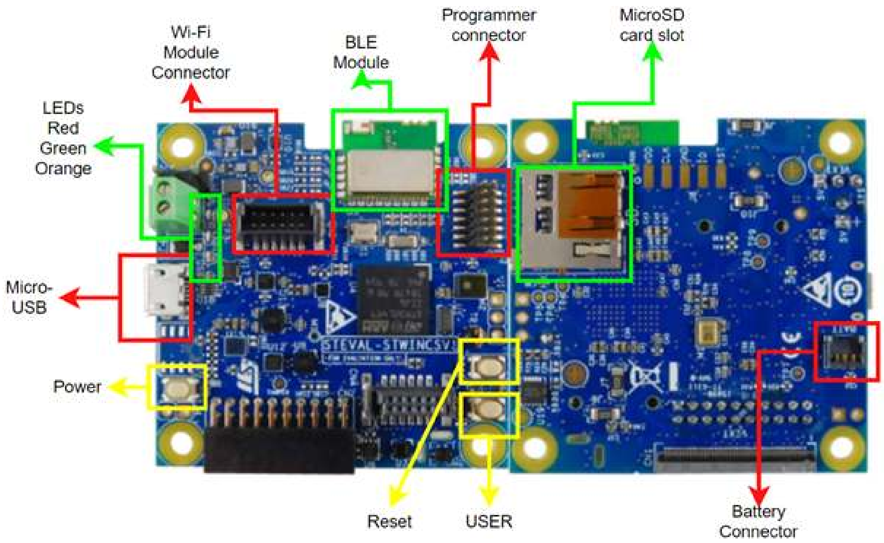


FIGURE A.1 – STWINKT1B and useful interfaces

A.4 Connecting to the computer

To connect the card to the computer, two microUSB cables are necessary : one connected to the STWIN, the other connected to the STLink. The procedure is as follows :

1. Connect STWIN to STLink by programming cable via the programmer connector (figure A.1).
2. **First** connect the **STWIN** to the computer using of a micro-USB cable.
3. Then connect the STLink to the computer via the other cable micro-USB.

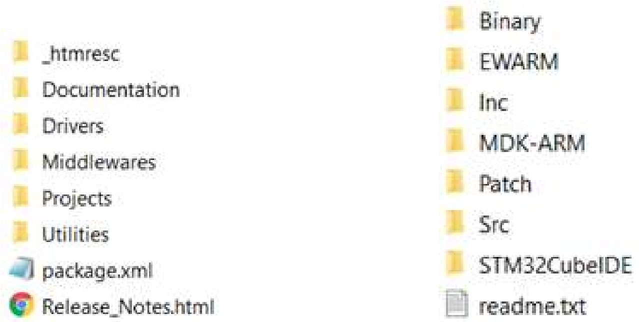
Once this is done, STLINK_V3M should appear in File Explorer's list of devices and drives of Windows.

A.5 Feature Packs

ST offers different feature packs (**MCU & MPU Embedded Software**) for STWIN with guides that allow the user to become familiar with the map. In this guide, we will focus on predictive maintenance scenarios with the following packs :

— **FP-AI-NANOEDG1**,
— **FP-IND-PREDMNT1**,
— **FP-SNS-DATALOG1**.

The typical folder structure of these packs is available in figure A.2(a).



(a) Feature packs folder structure (b) "Projects" folder structure

FIGURE A.2 – Folder structure

The folders we are interested in are the *Projects* and *Utilities*. The structures of the *Utilities* and *Projects* are different depending on the feature pack. However, Some folders in projects have a particular structure like This is shown in figure A.2(b) where the most useful folder will be *Binary*. This folder contains the binary file which is of importance special to program the STWIN according to the functional pack. In each functional pack, this binary file will have to be "dragged and deposited" in STLINK_V3M in order to program the card.

ST provides other functional packs but they lack interest. FP-CLD-AZURE1 and FP-CLD-AWS1 functional packs are basically the same as the predictive maintenance dashboard but through other interfaces supported by Azure and AWS. The pack functional X-CUBE-MEMSMIC1 uses the add-on module of STEVAL-STWINMAV1 microphone for capturing sound from microphones.

A.5.1 NanoEdge Functional Pack (FP_AI_NANOEDG1)

FP_AI_NANOEDG1 is a feature pack that uses machine learning automatic thanks to the company's third-party program NanoEdge AI Studio French Cartesiam. This program allows the user to make Machine Learning without any data science skills, in artificial intelligence and even in programming.

Each step of creating machine learning from STWIN is precisely documented in the

Section 5 of the Getting Started Guide requires the use of data created from section 4. Sample data are available for the early stages of library generation, as explained in the guide (section 5.1.3). However, the fourth step requires additional data to validate and test the library. However, the guide does not specify the data required for this step. I think the data to use is the data generated in section 4. If this data is not available (faulty SD card reader for example), you can use the Datalog functional pack FP-SNS-DATALOG1 example data in the case *STM32CubeFunctionPack_DATALOG1_V1.1.0\Utilities\STWIN_acquisition_examples.*

As a result, the STWIN should be able to learn the model of normal vibration as seen in this **video** and detect the abnormal vibration pattern. In my tests, the card was able to detect normal vibrations that occur occur when holding the STWIN in hand and to detect when the card is shaken intensively. My test parameters via the Tera Term console are available in figure A.3. A video demonstrating The hand vibration test is available in the .zip file under the *Results.* folder

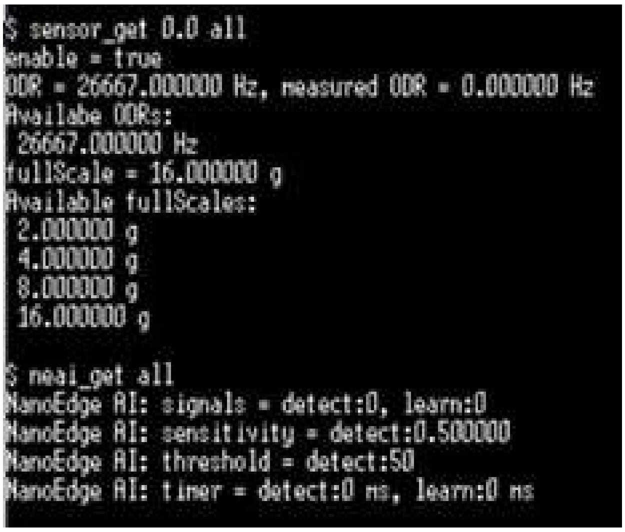


FIGURE A.3 – Screenshot of Tera Term showing the settings of setup for hand vibration test.

As explained in section 4.2.1, the supported sensors for predictive maintenance using AI are achieved by the command "$ sensor_info". The supported sensors are then the accelerometer and the gyroscope.

A.5.2 Predictive Maintenance Feature Pack (FP-IND-PREDMNT1)

Unlike FP_AI_NANOEDG1, ST does not provide such a guide complete than the latter for FP-IND-PREDMNT1. Instead, a document is available in the *Documentation* folder of the .zip file. This document is called ***um2566-getting-started-with-the-stm32cube-function-***

pack-for-multisensor-nodes-with-signal-processing-to-enable-predictive- maintenance-stmicroelectronics.pdf and includes getting started guides for various ST products. Obviously, the one we are interested in here is the platform STEVAL-STWINKT1 presented in section 4, page 59. The functional pack for STWIN is divided into two parts, the implementation for BLE and the predictive maintenance dashboard via Wi-Fi module **STWINWFV1** and cloud services.

BLE application

The application **STBLESensor** required for this feature pack is available for **iOS** and **Android**. BLE predictive maintenance function is not difficult to implement in place : just program the STWIN with the binary file *Predictive_ Maintenance_ BLE* included in the folder *Projects* and connect the card with your phone by Bluetooth and via the app. In the *Projects* folder, you you will find different names of MCUs : ours is *STM32L4R9ZI-STWIN.* In this folder you will find the structure usual files as shown in Figure 3. It should be noted that a reprogramming of the bootloader may be necessary as mentioned in the file *um2566....* To do this, the file binary included in \Utilities\BootLoader\STM32L4R9ZI must be dragged and dropped into STLink_V3M. The binary file needed to program the card with the BLE program is then the one whose name contains the attribute "*BL*".

Therefore, screenshots of the STBLESensor application are available in the *Results* folder. The application shows the predictive maintenance statuses for different sensors on the map, graphs showing data from each sensor, a FFT amplitude graph and connection status and level of the battery.

Maintenance Dashboard predictive

To connect the STWIN to the predictive maintenance dashboard, even though this is done via the AWS cloud, there is no need to have an AWS account but an ST account is required. The dashboard has its own **page** on the ST website and is accessible through this **link**.

After creating your ST account, you need to add another device : in the configuration tab, click on the green + button at the far right of the screen. The device name must be a name that is not yet assigned to any device created in the world. After creating your device, save the zip file and save the IoT address in a .txt file.

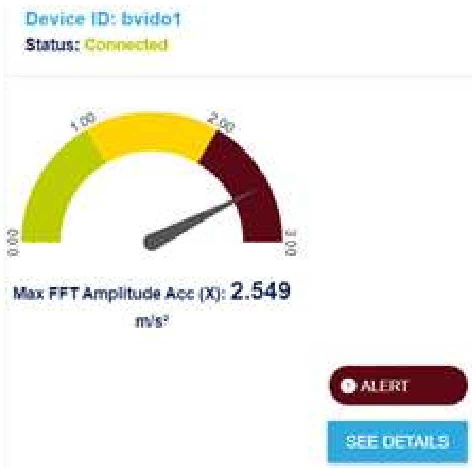
In section 4.4.2.2 of this **document** (um2566), configure the STWIN through Tera Term by following the steps. At some point you will be asked to enter the SSID of the Wi-Fi you are using. you want to connect the STWIN (step 10). To avoid any incompatibility with Company guest Wi-Fi, I configured my own Wi-Fi via an access point on a computer. In step 11, it You are prompted to enter the AWS IoT Core access point. This address is the one you saved in a .txt file previously. To paste the address into Tera Term, copy it and right click in the Tera Term window. Steps 12 and 13 ask you to enter the certificates. To do this, unzip the .zip file downloaded from Dashboard and drag and drop the file corresponding when necessary (Root CA = root.ca.pem, Security certificates = xxx.cert.pem, private key = xxx.private.key). After all these steps, the MQTT connection will be activated and the STWIN will be available via the dashboard as shown in figure A.4.

The dashboard is supposed to have been designed for maintenance predictive but only preventive maintenance has been implemented. figures A.5, we can see how it works : define thresholds on the desired sensors present on the map such as the temperature, pressure,

FIGURE A.4 – Predictive Maintenance Dashboard

humidity, acceleration (along axes), noise and effective speed - in figure A.5(a), the x-axis of the accelerometer has been defined -, then, monitor the sensors on the figure A.5(b) (here, the acceleration is high enough to trigger an alert). If the device location has been set manually, you can see where the alarms are located in the world. The dashboard allows you to download all the data up to the last 6 months.

(a) Threshold on the accelerometer



(b) Threshold exceeded alarm

FIGURE A.5 – Defining a threshold on the web dashboard

A.5.3 High-speed Datalog function pack (FP-SNS-DATALOG1)

THE

In order to use the machine learning part of the sensor ism330dhcx, you can find the .ucf configuration files necessary in \Utilities\STWIN_config_examples\UCF_examples.

No results were recorded for this section because it is simply collected data and acquisition examples are available in the *Utilities* folder. In addition, learning automatic does not show any results in the application.

USB Mode - Line of command

Section 2.1 of the "Getting Started" document explains how to use this feature pack via C++ script. I failed to do it work with this script but I used the Python script as explained in section 2.6.1 on page 31. The two Python scripts more useful are *hsdatalog_cli.py* which is the command line to record the data, and *hsdatalog_plot.py* to plot the data that has been recorded. To plot all data, you can add "-s all" to the end of the command, as explained in page 34.

BLE application

Data logging can also be done through the application (section 2.3 of the document). After connecting your phone and the STWIN via Bluetooth, you can choose which sensor to use and the data will be saved in a .dat file on the microSD card of the STWIN.

A.6 Tera Term

Tera Term is a terminal emulator used to communicate with the card via computer. It is quite difficult to read without the parameter appropriate : to improve this, in the configuration settings, set the font to *consoles* and its size to 10, then save the configuration and overwrite the existing one.

Possible commands to use are available in the **section 4.3** of the FP_AI_NANOEDG1 user manual and in figure A.6.

A.7 Troubleshooting

If the STWIN no longer appears to respond and/or the green LED is not on/flashing, it is good to reload the bootloader on the card. To do this, simply drag and drop the configuration file binary located in FP-IND-PREDMNT1 as explained previously in the BLE Application section.

A.8 Conclusion

Throughout this guide we have explored the possibilities available by ST through its STEVAL-STWINKT1B development kit. The possibilities are separated into different function packs : FP_AI_NANOEDG1, FP-IND-PREDMNT1 and FP-SNS-DATALOG1. PREDMNT allows you to monitor sensors incorporated in STWIN from anywhere in the world on multiple devices, DATALOG records data from all sensors on the microSD card or on a connected computer and NANOEDG can create artificial intelligence for maintenance predictive using vibration sensors. NANOEDG is the pack most promising feature for predictive maintenance : it can working with vibration data where it can learn from vibration models. The next step in this project is to explore NANOEDG further to find cases where STWIN could be useful in predicting maintenance of Company's Digiguard.

Command name	Command string	Note
CS1 - Generic Commands		
help	help	Lists all registered commands with brief usage guidelines. Including the list of applicable parameters.
info	info	Shows firmware details and version.
uid	uid	Shows STM32 UID.
date_set	date_set <date&time>	Sets date and time of the MCU system.
date_get	date_get	Gets date and time of the MCU system.
reset	reset	Resets the MCU System.
start	start <"datalog", "neai_learn", or "neai_detect" >	Starts an execution phase according to its execution context, i.e. datalog, neai_learn or neai_detect.
stop	stop	Stops all running execution phases.
datalog_set	datalog_set timer	Sets a timer to automatically stop the datalogger.
datalog_get	datalog_get timer	Gets the value of the timer to automatically stop the datalogger.
CS2 - PdM Specific Commands		
neai_init	neai_init	(Re)initializes the AI model by forgetting any learning. Used in the beginning and/or to create a new NanoEdge AI model.
neai_set	neai_set <param> <val>	Sets a PdM specific parameters in an execution context.
neai_get	neai_get <param>	Displays the value of the parameters in the execution context.
neai_save	neai_save	Saves the current knowledge of the NanoEdge AI library in the flash and returns the start and destination address, as well as the size of the memory (in byte), used to save the knowledge.
neai_load	neai_load	Loads the saved knowledge of the NanoEdgeAI library from the flash memory.
CS3 - Sensor Configuration Commands		
sensor_set	sensor_set <sensorID>.<subsensorID> <param> <val>	Sets the 'value' of a 'parameter' for a sensor with sensor id provided in 'id'.
sensor_get	sensor_get <sensorID>.<subsensorID> <param>	Gets the 'value' of a 'parameter' for a sensor with sensor id provided in 'id'.
sensor_info	sensor_info	Lists the type and ID of all supported sensors.
config_load	config_load	Loads the configuration parameters from the `DeviceConfig.json` and `execution_config.json` files in the root of the microSD™ card.
CS4 - File System Commands		
ls	ls	Lists the directory contents.
cd	cd <directory path>	Changes the current working directory to the provided directory path.
pwd	pwd	Prints the name/path of the present working directory.
cat	cat <file path>	Display the (text) contents of a file.

FIGURE A.6 – Tera Term Commands

Annexe B

Learning Machine Learning Applied to Predictive Maintenance

Introduction This appendix was created during the internship in order to help the internship supervisors to get up to date in order to better supervise me during my absence intended for the second exam session. In addition, the appendix has multiple internet links which are in bold in order to distinguish them.

In order to teach machine learning from the beginning, Google has put set up a free machine learning crash course on its Google Developers Platform. The **course** would last 15 hours, but it is possible to complete it faster by bypassing the videos and revealing the answers earlier. The course is available at this **link**.

To be more specific about machine learning for the predictive maintenance, a useful resource is this **GitHub** where an IBM software engineer shows what he learned during his 4 years of work on machine learning. He created a notebook Python with comments on every manipulation it performs. Some of the topics presented in Google's crash course are found in this example. The author also wrote an article about this work on Medium. The GitHub is available **here** and the article on Medium is available **here**.

More resources for predictive maintenance Python examples for machine learning include (in order of relevance/relevance) : **PySurvival**, **LSTM**, **LittleHurt**, **Unikie**.

Annexe C

Raspberry Pi Setup Steps

C.1 Configure SD Card

Raspberry imager Follow this tutorial.

Enable ssh Once the card is activated, 2 partitions will be created on the card. In the partition called *boot,* you need to create a file called *ssh.* This file must not have an extension.

C.2 Controlling the Raspberry

Connectivity The Raspberry will be controlled using an SSH client, for example PuTTY. To do this, connect the Raspberry to a power supply via its port microUSB/USB Type C, and an Ethernet cable to the corresponding port. The Ethernet cable can be connected either to a network on which the PC required for control, either directly to the said PC.

Software If the Raspberry is connected directly to the PC via Ethernet cable : On the PC that will control the Raspberry, open PuTTY. In PuTTY, the Host name should be *raspberrypi.local* and port *22.* The connection can then be opened.

If the Raspberry is connected via network : use an IPScanner to find the address of the Raspberry card. Then enter this address in Putty.

C.3 Getting rid of the repeating audio message

When there is no connection to the HDMI connector, the Raspberry broadcasts a message at regular intervals to activate the screen reader. To disable it, run this command : *sudo mv /usr/share/piwiz/srprompt.wav /usr/share/piwiz/srprompt.wav.bak.*

C.4 Install Twinkle

sudo apt-get install twinkle

An internet connection is required to run this command. folder where Twinkle configuration files are located is */home/pi/.twinkle*

C.5 Configure the .cfg file

The file name is the **user** name that will be set in the GUI Twinkle. For simplicity, the file name will therefore be the same as the name username (user_name) in the configuration file. Write to the **user**.cfg file which is in the box below. Do not are required as user_name and user_domain for creating a very basic user (not our case here). To work with the Company Asterisk server, UiD must be 500X or 600X and the sip.foo.bar should be

10.128.100.61 (local Asterisk server address). The command to access the **user**.cfg file is *sudo nano /.twinkle/**user**.cfg*

```
user_name=**UiD**
user_domain=**sip.foo.bar**
user_display=**Your Name**
user_organization=
auth_realm=
auth_name=**UiD**
auth_pass=**Password**

# SIP SERVER
outbound_proxy=**sip.foo.bar**
all_requests_to_proxy=no
registrar=**sip.foo.bar**
register_at_startup=yes
registration_time=3600

#RTP AUDIO
codecs=g711a,g711u,gsm
ptime=20
dtmf_payload_type=101
dtmf_duration=100
dtmf_pause=40
dtmf_volume=10

# SIP PROTOCOL
hold_variant=rfc3264
check_max_forwards=no
allow_missing_contact_reg=yes
registration_time_in_contact=yes
compact_headers=no
use_domain_in_contact=yes
allow_redirection=no
ask_user_to_redirect=yes
max_redirections=5
ext_100rel=supported
referee_hold=no
referrer_hold=yes
allow_refer=yes
ask_user_to_refer=yes
auto_refresh_refer_sub=no

# NAT
nat_public_ip=
#stun_server=**sip.foo.bar**:10000
```

```
# TIMERS
timer_noanswer=30
timer_nat_keepalive=30

# ADDRESS FORMAT
display_useronly_phone=yes
numerical_user_is_phone=no

#RING TONES
ringtone_file=
ringback_file=

# SCRIPTS
script_incoming_call=
```

C.6 Allow auto-answer

For twinkle to automatically pick up the call it receives, you must launch twinkle via the command line using *twinkle -c*, then run the command *auto_ answer -a* on to enable auto-answer for the active user. To choose the user, run the command twinkle -c -f **user**.cfg replacing **user** with the user desired.

C.7 Start Twinkle when turning on the Raspberry

Set default user when starting twinkle Edit the twinkle.sys file using the *sudo nano command /home/pi/.twinkle/twinkle.sys*. In this file, below the line *# Startup*, add *start_ user_ profile=* followed by the user name defined in the previous step according to the name of the corresponding file. For example, if the file created in the previous step is 6005.cfg, the line to add in twinkle.sys will then be *start_ user_ profile=6005*.

Create the startup.py file in /home/pi/.twinkle Use the command *sudo nano /home/pi/.twinkle/startup.py* to access startup.py :

The content to put in it is as follows :

```
import os
os.system('twinkle')
```

Enable autostart and create the twinkle.desktop file Below are the commands to use to access the twinkle.desktop file.

```
mkdir /home/pi/.config/autostart
nano /home/pi/.config/autostart/twinkle.desktop
```

The contents of the twinkle.desktop file are as follows :

```
[Desktop Entry]
Type=Application
Name=Twinkle
Exec=/usr/bin/python3 /home/pi/.twinkle/startup.py
```

Enable desktop autologin in raspi-config sudo raspi-config

In raspi-config, navigate to *1 System Options > S5 Boot/Auto Login > B4 Desktop Autologin*, enable auto-login according to what is displayed on the screen, press finish in the main menu and accept the reboot.

Twinkle does not accept to be used via command line in the background (auto deregistering of the SIPphone). Thus, among the possibilities of Starting a program when the Raspberry Autostart starts is the most appropriate because it allows you to launch the Twinkle GUI after the raspberry desktop graphical environment is launched, which explains why it is necessary to enable autologin on the desktop.

Disable HDMI port (optional) The Raspberry desktop being in autologin, the entirety of the Raspberry is accessible by anyone who connects a display to the HDMI port. So, from a security point of view, it may be necessary to Disable the HDMI port. Disabling the HDMI port also helps prevent inefficient use of energy.

The command to disable the HDMI port is */usr/bin/tvservice -o*. For To re-enable the HDMI port, the command is */usr/bin/tvservice -p*. For the HDMI port is always disabled every time the card is turned on, it just add the line */usr/bin/tvservice -o* in rc.local using from *sudo nano /etc/rc.local*.

HDMI.py and HDMI.sh can also be used, by putting bash */home/pi/.twinkle/HDMI.sh* in the rc.local file (Command : *sudo nano /etc/rc.local*).

C.8 Menu to modify user settings

Copy the files 'UserEditor.py' and 'UserEditor.sh' into */home/pi/.twinkle*. For the UserEditor to be displayed at the cmd login of the map, add the line bash */home/pi/.twinkle/UserEditor.sh* in *sudo nano /etc/bash.bashrc.*

C.9 Creating commands to facilitate access to UserEditor/HDMI menus

Follow the instructions in this tutorial.

In bash.bashrc, the instructions to add are *printf '\n========================\n Instructions \n\nType \"HDMI\" to enable/disable the HDMI port\nType \"UserEditor\" to change user/SIP settings\n\n========================\n'*.

Annexe D

pjsip.conf

This appendix explains the different options for creating a user on Asterisk using explanations provided in the pjsip.conf file in table D.1.

pjsip.conf	Comments
;===TRANSPORT	
[transport-udp]	
type=transport	
protocol=udp	
bind=0.0.0.0	Defines an IPV4 address
;===SOFTPHONE TEMPLATE	Simple comment for readibility
[ipfone](!)	Declaration of the template ipfone
type=endpoint	All ipfone are endpoints from a
point of view	architecture (proxy). An endpoint is
essentially a profile for the configuration of a SIP	
	endpoint such as a phone or remote server.
allow=!all,ulaw	Audio/telephony codecs to allow (here G.711 µ-law)
	Allow=!all is the equivalent of disallow=all.
context=from-internal	Dialplan context for inbound sessions
transport=transport-udp	Definition of the protocol defined
	in the associated section above
direct_media=no	Determines whether media may flow directly
	between endpoints.
send_pai=yes	Send the P-Asserted-Identity header
refer_blind_progress=no	Whether to notify all the progress details
	on blind transfer
rtp_timeout=2	Delay without RTP to consider channel as dead
[auth](!)	Holds the options and credentials related to
	inbound or outbound authentication
type=auth	
auth_type=userpass	Set to userpass to read password. Other option is
	md5
[aor](!)	Tells Asterisk where an endpoint can be contacted.
	Without it, the endpoint can't be contacted
type=aor	
max_contacts=3	Max number of contacts that can bind to an AoR
remove_existing=yes	New contacts replace existing ones.
;===EXTENSION 6001	
[6001](ipfone)	Declaration of the user 6001 based
	on the ipfone template

outbound_auth=6001	Authentication object used for outbound requests
auth=6001	Refers to auth section below
aors=6001	AoR(s) to be used with the endpoint (refers to section below)
allow_transfer=yes	Determines whether SIP REFER transfers are allowed for this endpoint
context = from-internal	
[6001](aor)	
[6001] (auth)	
type=auth	
auth_type=userpass	
password=6001	
username=6001	
;nat=yes	

TABLE D.1: pjsip.conf and explanatory comments