

1. INTRODUCTION

I have completed my second and final mandatory internship in my academic career in a company named SAIT which is a sub-company of the famous cybersecurity company Securitas. This internship lasted almost two months, and started on 16th of July 2018 and ended on 7th of September 2018. The office was situated in Zellik, Belgium. In this two month duration, I have worked as an engineer intern and developed a software for their main product the SAIT CST-300X Transmitters.

In the second part of this report, I will give information about the company such as its history, their main products, projects and etc. In the third part of this report, I will clearly explain the software that I have developed, what tools I had used, what programming language(s) I have used and many more. My source codes are also included. Finally in the last section I will conclude the report and analyze my work, I will criticize both myself and the company and state what I have learned this summer.

1.1 Overall Goal

My overall goal in this summer practice was to link the academic teachings that I have learned from my university courses and put these theoretical teachings into practice in a professional working environment as a true engineer. I would also learn to contribute to the society by producing or creating new products to ease people's daily lives as engineers do. Also this internship was in Europe so I wanted to experience working abroad and to leave a good impression for my future career.

1.2 Expected Learning Outcomes

My initial expectation from this internship was to test my programming skills that I have learned in my programming courses such as CMPE 112 and CMPE 211. Also Java was the programming language that I was most comfortable with. I wanted to change that and learn a new programming language or develop my skills in C, C++ or HTML.

Secondly, after this internship I am expecting to fully understand how to solve problems using the methodology that I have learned in my courses such as algorithms and discrete math, how to approach a real-life problem in a professional manner and efficiently solve it in a certain amount of time.

Also I expect to see how a professional software company works as a whole and to witness how computer engineers work and produce products during this internship. Additionally, I wanted to see a working environment where people from different backgrounds and ethnicities have to work together to earn their keep. Besides these reasons, I also wanted to understand the importance of professional and ethical responsibility during my internship.

Perhaps the thing that I wanted to experience the most was working in another country, away from my comfort zone. I believe that this would help me both in my academic and future career.

Lastly, I expect to test my communication skills and techniques that I have learned through my communication courses which is my secondary field to efficiently communicate with other interns, workers, engineers and etc.

2. COMPANY INFORMATION

2.1 General Information About the Company

SAIT is a sub-company of the famous world-wide cyber-security company “Securitas” which develops long range transmitters with low power consumption which makes them cost effective. SAIT is a Belgian company that goes back to the 1900s. SAIT is the acronym of “Société Anonyme Internationale de Télégraphie” which can roughly be translated as “Anonymous Society of International Telegraphy”.

2.1.1 Historical Background

The history of SAIT dates back to 1901 when a group of Belgian entrepreneurs got together to find a solution for the communication problem with the colonies of Belgium in Africa. With the help of Leopold II who was the current Belgian King and

Guglielmo Marconi who was the inventor of the wireless telegraph they founded C.T.S.F. (Compagnie Générale de Télégraphie sans Fil - General Telegraphy Company).

Like most European companies CTSF had troublesome times during the World Wars, especially the second one. But after the WWII during the rebuilding of western Europe CTSF played a crucial role. It changed its name to SAIT and worked with French companies to implement the communication system to trains and trams.

In the 90s the company changed ownership, names several times and also merged with other companies from Holland and Norway. In November 2014 SECURITAS, has taken over the SAIT in Belgium in order to acquire a broad range of technology solutions for the Belgium market and in other territories. Since July 2016, SAIT (Belgium) and SECURITAS have been merged together.

2.1.2 Location(s)

The company is situated in Zellik, a small town in the Flemish area of Belgium north of its capital Brussels. The office can be seen in figure 1 below.



Figure 1: SAIT-Zenitel Office Building in Zellik Research Park, Belgium

2.2 Products, Services & Projects

As mentioned before SAIT develops long-range transmitters for ground to air (aeronautical), land to land and shore to ship (maritime) platforms. Their main products are the MF/HF transmitters which come in three options according to the power consumption. 1kW, 5kW and 10kW. Because of this they come in different

sizes. The sizes and the weights are 133 x 483 x 500 mm and 50kg for the CST3001 (1kW), 1200 x 600 x 900 mm and 200kg for the CST3005 (5kW) and 2000 x 600 x 900 mm and 300kg for the CST3010 (10kW). So we can say that the CST3001 transmitter is no bigger than a desktop computer and the CST3010 is almost as big as an average sized refrigerator.

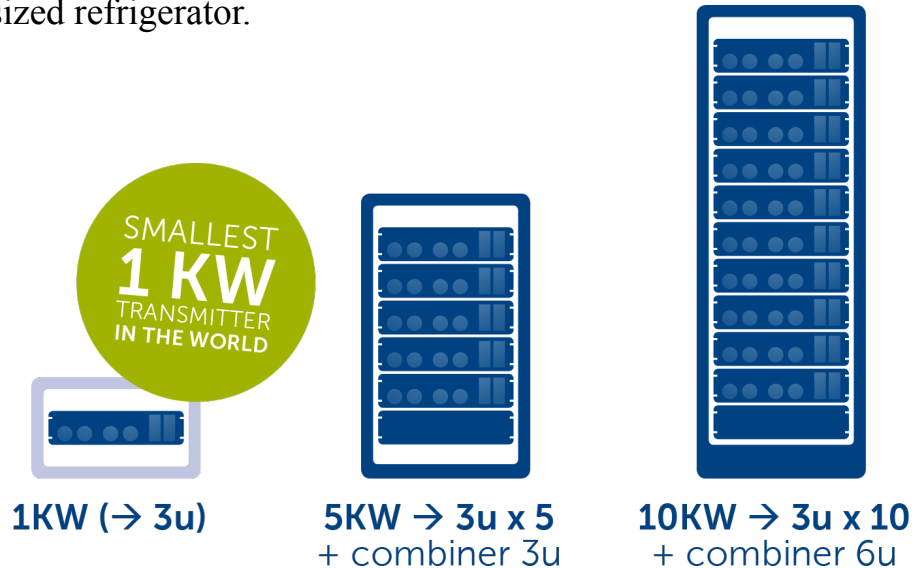


Figure 2: CST3001, CST3005 & CST3010 Transmitters size comparison

The CST3001 (1kW) transmitter consisted of a single panel. The CST3005 (5kW) transmitter consisted of five panels and a combiner unit. Finally the CST3010 (10kW) transmitter consisted of ten panels and a single combiner unit. These can be seen in the figure above. Every panel consisted of two PS (power supply) units, four PA (power amplifier) units and a single DFS (dynamic frequency selection) unit. Every one of them has different types of values.

The main purpose of these transmitters are to send voice and data through different platforms and to long distances. Some of the applications are;

- Digital Selective Calling (DSC) [ASN: Appel Sélectif Numérique (French)]
- Navigational Telex (NAVTEX)
- Selective Calling for air traffic control (SELCAL)
- Meteorological information for aircraft in flight (VOLMET)
- Voice communication (SSB)
- Morse (A1A)
- Data communication (F1B and other modes)

Transmitters are not a new invention nor it is a complex device. So why SAIT is so important in this field? So important that it also has NATO contracts. Currently NATO uses the CST3001 and CST3005 transmitters in different bases in the world. Some of the unique features of the CST3000 series transmitters are;

- IP-connectivity (local and remote control)
- SNMP support
- Websocket
- User friendly interface
- Full solid-state design
- Easy maintenance (BITE interface)
- DSP-based exciter
- Extremely compact
- Modular and highly reliable

One of the most used feature of the transmitters is the Websocket web application. This allows users to examine the products constantly changing values easily because it is a user friendly interface which is very simple to understand and use. After a brief instruction anyone can use it. The task given to me was related with this application so during my internship I have used this application a lot.

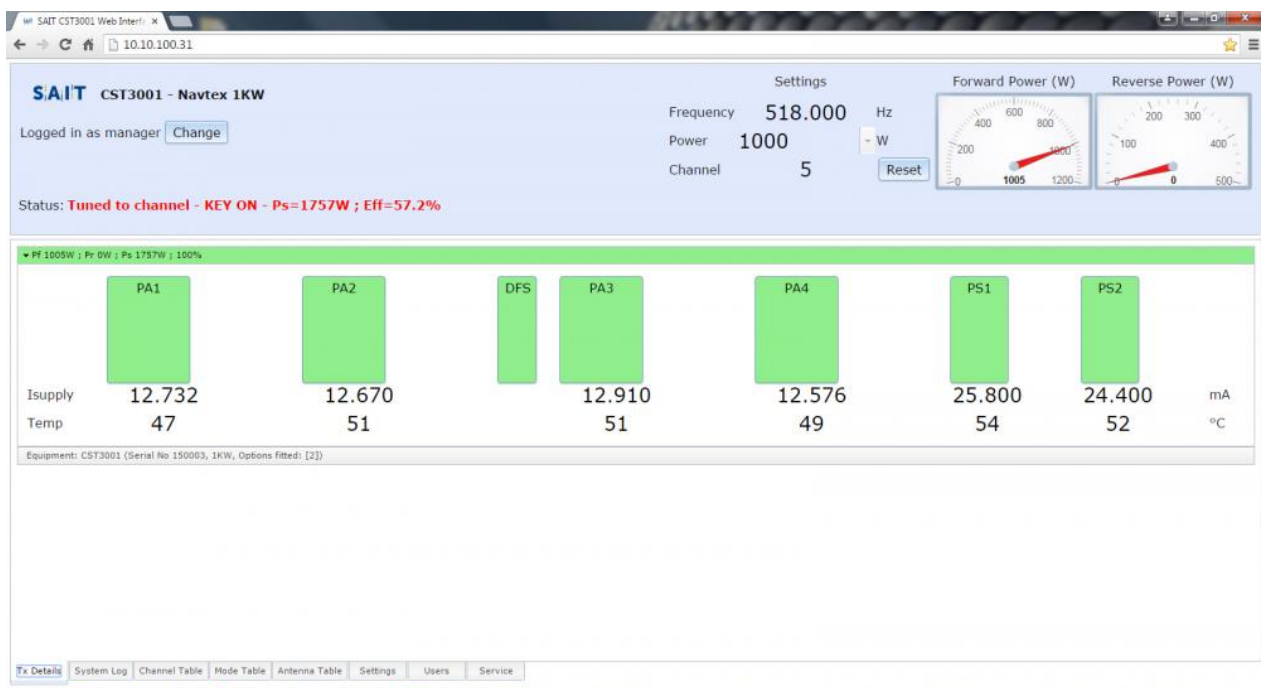


Figure 3: Screen shot from the Websocket web application which shows a CST3001 Transmitter when transmitting

2.3 Working Environment

The office of SAIT was located in Zellik which is a small town north of Brussels like I have mentioned in the previous section. SAIT shared the building with other companies, some minor startup companies, some agencies which were all a part of SECURITAS. SAIT occupied a wing of the building which had a storage room to store the hardware, a loading garage, a testing room and a room for the team.



Figure 4: Employees of SAIT

The team consisted of only six employees. It had a manager Patrick Delanghe, a project manager Arif Punjwani who was also my supervisor (arif.punjwani@securitas.be), a test engineer Koen Verdijck who also designed the 1kW version of the transmitter, two consultants Wim Casteleyn and Erkan Belek and a part timer. Some of them are shown in figure 4. I was the only intern they had.

Like all employees I was given a laptop, a desk, a monitor and a mouse. However I asked for a keyboard too because the laptop's keyboard was a Dutch keyboard so many of the letters were in different places I asked for a QWERTY keyboard in addition. We used key cards when entering the building and the office.

The security was not as highly prioritized as it was in Milsoft which was my first internship company.

The working hours were also very flexible compared to Milsoft. In Milsoft I had to work a 8-hour shift but in SAIT sometimes they allowed employees to work from home and also no one, even the manager interfered with anyones arrival and departure hours. The traffic of Belgium is an important reason for this, however, it is not the only reason. I will explain this reason in the conclusion part of my report.

3. ASSIGNED TASK(S)

During the initial days of my internship I was introduced to the other employees, given a laptop and briefed by the project manager about my task. In this meeting the project manager Mr. Punjwani explained to me what they are expecting from me. He explained the requirements of the software they want from me and the functionalities and asked for my opinions. They initially gave me six to eight weeks to finish the task and I completed the project in seven weeks. The final tests and presentations lasted for a week.

3.1 Work Description

As I have mentioned before the values of the transmitters can be seen in the web interface application which they have designed prior to my internship. The application was suitable for most of the browsers, but it was best suitable with Chrome, Firefox and Safari. In this web application users could log-in and see the real time values of the components of the transmitters. For example, a user could see the temperature of each component or the current that flows through it and so on.

The problem was that to compare the values, an employee must always be present and online. He/she must be online 24/7 and note every value change. Because this was not a cost-effective nor an efficient way, they asked me to develop a program that will retrieve the data from the transmitter, store them somewhere and, if needed, the user will search for a specific time interval and print it into the screen in an organized way.

3.2 Knowledge Integration

In this internship I had the chance to test myself in the field and to put what I have learned from my courses into use in a professional environment. For example the programming courses like C, Java I and Java II majorly helped me in this project because those courses were the foundations of my theoretical knowledge. Also I have used the sql and database knowledge which I have learned from my relational databases course. One other significant course was the computer graphics course which introduced me with the Microsoft Visual Studio and C#/C++ programming languages. Throughout this internship I have realized that the material I have learned from my courses are beneficial and actually helpful in my professional working career. In addition, during this internship I have learned how to design a GUI in Visual Studio, how to develop a handshake protocol between a computer and a server and etc. Before this internship I had the basic theoretical knowledge that was needed to get accepted for this internship which I gained from my school courses. In this internship I tried to use these theoretical knowledge in the project that was given to me.

Also as a secondary field I am taking communication courses, besides the computer engineering related courses. I have also taken communication courses that helped me to improve both my daily and professional communicational skills. I used the methods that I have learned in my introduction to communication course (COM 250) to ensure an effective communication with the SAIT team and other employees in the company. I used these methods especially during meetings and presentations.

3.2.1 Main Problems Encountered

The main problem about the project and perhaps the most difficult part that I encountered was the part where I had to establish a connection between the transmitter and my computer. Because I had no experience in that field, I worked in the hardware level just a couple of times and those were for university laboratories which were much easier.

Another problem was that Mr. Punjwani who was my supervisor and the

project manager whom did not help me throughout the initial stages. If he had helped me during the planning and initial stages maybe I would not have done the mistake that I will mention in the following sections. Nevertheless, I believe that this mistake helped me in the long term by giving me a unique experience. In my opinion one cannot be flawless in the beginning; however, mistakes help one become perfect.

One other major problem was the cultural and language barrier. Because I did this internship in another country, I needed to adapt to their standards. Although they all knew English, most of the time they spoke Dutch among themselves and this was sometimes problematic in the daily life but not in the professional way. Because they always spoke English with me during meetings and face-to-face encounters. The only problem was Mr. Punjwani's accent. Because he was a bit hard to understand.

3.2.2 Learned Tools, Programs, Languages & etc.

Throughout this internship although I did not use it in my final product, I had the chance to work on Microsoft's Visual Studio. My only experience with this environment was in Erasmus in the Computer Graphics course. However, during this internship I learned that it allowed developers to create a variety of things. For example, I mostly used it to create a GUI.

Also because I used Visual Studio, I used C# which I had no prior experience in. However, I came to realize that C# and Java are like the different sides of the same coin. They are very similar and so it was no problem to me to adapt to it.

I also used Visual Studio's database system to create and connect a database with my GUI. I already knew how to connect a database but I realized that the method I knew was harder than this one. One important thing that I have learned in this internship was about transmitters. So besides the software I learned about a hardware component. I have learned to use the company's CST300X model transmitters and also the web socket application. Besides these technical things I have also experienced working abroad although it was a short time.

3.3 Work Done

Throughout this internship I have developed a software that can connect with the CST300X transmitters and retrieve its json formatted data, stores them in a local text file in string type and prints the data to the console with a simple table-like way according to the users time input. The software consisted of two parts; the “capture” part retrieves and stores the data and the “query” part prints the data to the screen after filtering. To develop this application I have used Java programming language, the Eclipse IDE and a CST3001 for testing. Although I started working with C# programming language and used Visual Studio as my compiler, I finished my task using Java and Eclipse. Also I used Microsoft sql server as well at one point but that lead to no where. To efficiently describe the work that I have done I divided my progress into sections. All of my source code can be found in the appendix part of this report.

3.3.1 Initial Work & Plans

On my first day, I attended a meeting with the project manager Mr. Punjwani who briefed me about the requirements of the software. After that the test engineer Mr. Verdijsk explained the main functionalities and properties of the transmitters. Later Mr. Punjwani helped me to connect my computer with the transmitter using their own web application.

The storing and printing parts of the application did not worry me. However the first part which was retrieving the real-time data from the transmitter seemed a bit frightening to me. So, I made my first and perhaps the most crucial mistake, I skipped the first part and started from the end.

As I did in my last internship, I started researching about the company, its products, transmitters, how to connect with them and many more. I tried some simple testing programs and after the long research period I started to work on my application with the way that I saw fit.

I wanted a GUI(Graphical User Interface) for my software and because of this, I used Visual Studio and C# since it is very easy to create a GUI with them. Later on I planned to store the data that I will retrieve in a database so I used Microsoft's sql

server. Since I was using another Microsoft product (Visual Studio), it was much easier to connect these two when compared to ORACLE's MySQL.

3.3.2 Unsuccessful Attempts

Like I have mentioned in the previous section I started my development phase with a very significant mistake. Because of this mistake, I wasted two weeks of my time and I had to start all over again.

I designed a GUI with C# on Visual Studio which had a search functionality, a help button which opened a window that showed a manual for the application, a close button and some comboboxes for the filtering. When I completed the GUI I started working on the database. Initially I tried Microsoft Access but I couldn't find a free version so I worked with Microsoft sql Server. I created three tables. However, at one point I realized that the database will not be effective for the data and also I did not have any connection with the data at that time. So, I deleted all of my work and started from scratch.

3.3.3 Connecting With The Transmitter

After starting fresh, I returned to the research phase again. After my second research and help from Mr. Punjwani I managed to connect and retrieve data in real-time with a simple program. However, I decided to switch to Java rather than C# since I was more comfortable with it. So, I installed a compiler named Eclipse.

I converted the connection program to Java from C# and wrote a simple handshake protocol that gets an IP address as an argument and tries to connect with that domain. If connection is successful a message is printed, if not an error message is printed. Later I updated this simple handshake protocol and created a program which gets an IP address as an argument (in this case the transmitter's IP address) and prints every message that the domain sends when connection is established. With this test I saw that the values were stored in json format so I would have to write a method which converts json to normal values such as string, integer, character and so on. So at this point my program could connect with the transmitter and get its data in real-time.

3.3.4 Capture & Store Data

In this phase, I wanted to add a method that would literally “capture” the coming data and store them into a local file with a unique name. For this method I used Java’s file and buffer libraries. As a result, I have managed to write a class which creates a directory and a uniquely named file and stores the constantly coming data in that uniquely named text file. I chose the date and time as the unique name of the file. This part was not challenging because in my programming courses I have done very similar tasks a lot, especially in the C Programming course. So, at this point my program could establish a connection with the transmitter and store the constantly flowing data to a uniquely named text file which is created inside a user-specified directory.

3.3.5 Find & Print Data

This phase was easier to work on because in the previous steps I needed to test with the transmitter every time. Since I couldn’t take the transmitter home, I always had to be present at the office and I couldn’t work in the weekends. In this phase I already collected my data so someday I worked from home and it was a very effective way to increase efficiency in my opinion.

For this part I have read the documentation of the values and learned the codings of the values and wrote a method which converts them from json to strings. I also added a simple time filtering method that takes two arguments from the user, a start time and an end time. If no argument is entered, the program will show all the data. I also prepared a specific printing style so that it would be more understandable. I did not have the time to develop a GUI like I had wanted but the main functionalities worked and I tried my best to show the data as user friendly as possible. At this point my software worked properly so I moved on to the final tests.

3.3.6 Last Minute Changes

As I mentioned above at this point the program worked perfectly, however, it worked perfectly for the CST3001 (1kW) model. I did not test it on the other two

models. So I asked to test on those and I came across a problem. The 1kW transmitters had a single panel which had two PSs, four PAs and a DFS. On the other hand, the 5 kW had five panels and a combiner unit and the 10 kW had ten panels and a combiner unit. So when I ran my program for the 5 kW, it skips the first four panel and just prints the fifth panels information. This was because of a loop mistake.

After solving this logical error, I encountered another. This time the visualization of the printed data was changed, so I had to write it again for it to appear in a more table-like manner and of course more neat.

3.3.7 Final Version

The final version of the “SAIT Capture & Query” application consisted of nine classes. To run the application the user connects the transmitter to his/her computer via ethernet cable or if it is in a secure network, Wi-Fi. After that, the user launches the application. There are two modes of the application; a capture mode and a query mode.

To run the capture mode, the user must type “capture”, a space and the IP address of the transmitter which he/she wants to connect with. The program will pass these to the software as arguments and do the required operations. For example “capture 1.1.1.1” is a valid capture mode command. This will connect the transmitter with the computer, open a directory and under that directory a uniquely named text file and write the coming data from the transmitter to that text file. Additionally, I have added a third optional argument. If the user wants to collect the data for a certain amount of time, s/he can add a third argument such as “120” which will allow the program to collect data for only 120 seconds.

To run the query mode, the user must type “query”, a space and the table acronym of which they want to display the information of. The program will pass these to the software as arguments and do the required operations. For example, “query sm” is a valid capture command. There are four different tables which store different values of the units; Slow Monitoring(sm), Fast Monitoring(fm), Current Config(cc) and Hardware Config(hc). This command will start the query mode,

search and find the sm values and print them in a specified way to the screen. In addition, I also added two more optional arguments. If the user wants to give a specific interval, not to display the whole data, s/he can simply add a start and end time as third and fourth arguments.

3.4 Work Plan

	Week 1	Week 2	Week 3	Week 4
Monday	Meet the team, Get company laptop	GUI design	Database Design	Start over; New plan
Tuesday	Meeting with Mr. Punjwani and Mr. Verdijsk	GUI design	Database Design	Connection with Transmitter(C#)
Wednesday	Work on CST3001	GUI design	Database Design	Connection with Transmitter(Java)
Thursday	Work on CST3001	GUI design	Connect database with GUI	Meeting with Mr. Punjwani and Testing
Friday	Work on CST3001 and WebSocket Application	GUI design	Connect database with GUI	Capture Data design
Saturday	-	-	Start Over	-
Sunday	-	-	-	-
	Week 5	Week 6	Week 7	Week 8
Monday	Capture & Store Data	Print Data	Find Data	Final Version Test
Tuesday	Capture & Store Data	Print Data	Test & Debug	Presentation to Mr. Verdijsk
Wednesday	Capture & Store Data	Test & Debug	Final Version Testing	Documentation
Thursday	Test & Debug, Presentation	Find Data	Test with CST3005 and debugging	GOOD-BYE
Friday	Test & Debug	Find Data	Test with CST3005 and debugging	Formality Paper Work
Saturday	-	-	-	-
Sunday	-	-	-	-

Figure 5: Weekly Plan

The first week I spent my time with settling in and working on the company's main product the CST300X transmitters, especially the CST3001, and the web-socket application which the transmitters are used with. Also I attended meetings and planned my software.

The second week I have created a suitable GUI for my software like mentioned in previous sections. On the third week I have worked on the database and tried to connect the database with the GUI. At this point I have realized that this course of action was wrong so I have decided to start over.

On the fourth week I started over, planned a new course of action. I finally managed to create a handshake protocol using C# but converted to Java later on and started the capture part of my application which is mentioned in section 3.3.4.

On the next week I finished the capture & store part of my application and finished its testing and debugging. On the sixth week I finished the find & print part of my application and its tests.

On week seven I tested my final version on the CST3005 and did the necessary modifications on my code so it will properly work on all three versions of the transmitters.

Finally on the last week I prepared a documentation and made a demonstration to Mr. Verdijck who was the testing engineer.

3.5 Evaluation & Impact Analysis

The software I have developed throughout this internship will be quite beneficial to the company in the future. Like I have mentioned before the main problem which required this application was that to check problems occurred on the transmitters, an employee must be in front of the screen looking at the web-socket application 24/7. Since most of the customers did not use this method, when a problem occurs they can not specify which component of the transmitter had a problem and when.

My capture & query application solves this problem by storing the data in a local directory with their date and when a problem has occurred simply the user can enter a time interval and check the values easily in a more readable way to find the

problematic component and the date which the problem occurred. If it is something that the customer can not handle, they can send a message to SAIT stating the problem. This program will ensure that both of the customers' and SAIT's time will not be consumed for nothing.

The version I have given to the team is currently very raw, maybe later on it can be modified and updated. Perhaps they can add a simple GUI, which I intended in the beginning, to be more user-friendly.

4. CONCLUSION & ANALYSIS

The second mandatory internship in my academic career was completed with success in my opinion. Both the team and I were happy at the end of the internship because they got the software which they needed, and I gained experience which I needed. I believe I have managed to achieve my overall goal which was to link my academic knowledge with real-life problems and put my knowledge into a professional work.

When compared with my other mandatory internship at Milsoft, it was more beneficial and fun for me. In Milsoft, because of security reasons the interns were not given any important work. The work that I have done in Milsoft was quite simple when compared to this project. However, because I was less experienced it was difficult for me back then. One major reason for that was I was left alone to complete my work, and did not receive any help from my supervisor in Milsoft. On the other hand, in SAIT, although I did not have the help I wanted, the team did not give me the feeling that I was alone or helpless. Also the working hours in Milsoft was fixed, 8:00 to 17:00. On the other hand, in SAIT they give me the option to select my working hours and even sometimes worked from home. In my opinion to give the employees this kind of freedom helps increase the work efficiency, at least mine increased.

SAIT offers their interns a very different approach, they treat them as if they are real engineers and give them actual projects which they will use in the future like mine. I highly recommend that SAIT will continue with this procedure in the future

with their other interns. This method allows students to feel like actual engineers. Also this feeling gives them a sense of responsibility more than regular students. For example at one point I thought I could not finish the project in time and it helped me to work even more. Sometimes I even worked in the weekends as well. So I can say that SAIT provides a working and social environment that makes you belong. You never feel as an outlander. On the other hand I recommend that the TED University engineering department should include more laboratory hours rather than lectures so the students can learn about the topic and practice it more. For example every class should be like the CMPE 112 and 211 courses.

To sum up, my second and last mandatory internship, which I completed in SAIT, was a unique experience which certainly will help me in the future. Although this internship helped me get closer to my goals, there were things that I wished went differently. However, in overall this internship was essential for my future professional career and my current academic career. Also it was a bonus that this internship was in a foreign country, it helped me gain the experience which I needed to pursue my goal which is to work in Europe or perhaps the USA.

REFERENCES

- [1]: Figure 1. Retrieved from http://www.keyestate.be/en/item_zenitel-researchpark-zellik_482.aspx?pid=102
- [2]: Figure 2 & 3. Retrieved from <http://www.saitsystems.be>
- [3]: Company information retrieved from <http://www.saitsystems.be>