ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Proyecto Fin de Carrera

“Investigación para la mejora de los actuales sistemas de sincronización en la esgrima”

AUTOR: Pedro Fidel Espín López

DIRECTORES: Frank Greig, Francisco Javier Garrigós

Junio / 2012

Autor	Pedro Fidel Espín López
E-mail del Autor	pedrofi.espin@gmail.com
Director(es)	Frank Greig, Francisco Javier Garrigós
E-mail del Director	javier.garrigos@upct.es
Codirector(es)	
Título del PFC	Investigación para la mejora de los actuales sistemas de sincronización en la esgrima
Descriptores	

Resumen

Este proyecto trata de hacer un estudio de los actuales sistemas de puntuación en la esgrima para posteriormente proporcionar el diseño de un completo prototipo de sistema que es capaz de mejorar en ciertos aspectos a los actuales.

El diseño del prototipo está dividido en tres partes, la creación de un protocolo que sirva de apoyo a la comunicación entre nodos, el diseño del cableado físico comprendido entre el equipo del esgrimista y la construccion de una sencilla interfaz gráfica que haga posible el manejo del sistema sin necesidad de tener conocimientos de este.

Titulación	Ingeniería Técnica de Telecomunicación, especialidad Telemática
Intensificación	
Departamento	Electrónica, Tecnología de Computadoras y Proyectos
Fecha de Presentación	2012

Descripción

Gracias al rápido y eficaz crecimiento de la tecnología en los últimos años, el campo de aplicación de los microcontroladores se ha visto incrementado sustancialmente hasta abarcar distintas áreas que en un principio no eran de su competencia. Una de estas áreas es el deporte de competición en donde una “máquina” es capaz de darnos unos valores de precisión y rapidez imposibles para el ser humano.

El mundo de la esgrima como deporte de alta competición, e incluso a niveles de entrenamiento, ha visto pronto como gracias a la tecnología disponible se podía conseguir una gran precisión en la determinación de los puntos o tocados que hacía de este deporte un deporte más justo.

Desde la introducción de la tecnología en la esgrima, las mejoras de ésta se han centrado únicamente en la mejora de la precisión del sistema de detección de tocados manteniendo una idea siempre fija, que el esgrimista debía ir siempre conectado al sistema mediante un cable. Sin embargo en los últimos años y gracias al gran avance en el área de las tecnologías *wireless* algunos prototipos de sistemas “sin cables” han empezado a aparecer. Este proyecto trata de formar parte de estos estudios y contribuir en el cambio tecnológico hacia una esgrima sin cables.

A la hora de proveer algo que sea de utilidad a este campo, el principal objetivo de este proyecto es producir un prototipo *wireless* de un sistema de puntuación de esgrima, en concreto para la disciplina de florete, haciendo uso de los dispositivos EZ430-CC2500, una herramienta de desarrollo *wireless* propiedad de Texas Instruments.

Para conseguir este objetivo, ha sido diseñado un sistema lógico que da solución al problema de la detección de los diferentes tocados completamente adaptado al equipo del esgrimista. Se ha conseguido un protocolo fiable y preciso con una precisión de 2 milisegundos. Por último también ha sido producida una interfaz gráfica que permite la comunicación del sistema con un PC con sistema operativo Windows.

En conclusión, se ha conseguido un prototipo completo de un sistema de puntuación para la disciplina de florete basado en componentes de bajo coste y consumo obteniendo una gran precisión.

Índice

Resumen en Castellano del Proyecto Fin de Carrera

1 Introducción

El propósito de esta introducción es presentar las ideas generales del proyecto, especialmente el "cómo" y el "por qué" de una forma breve e introductoria. Inicialmente se hará un breve resumen de la historia de la esgrima que llevará al por qué de la necesidad de este proyecto mostrando los objetivos principales de éste. Finalmente serán presentados los caminos a seguir para llevar a cabo esos objetivos mostrados.

1.1 Breve Recorrido por la Historia de la Esgrima

Debido al propósito del proyecto, es de suma importancia hacer una corta descripción de lo que el deporte de la esgrima ha representado en la historia y de cómo ha evolucionado hasta lo que actualmente se conoce como esgrima moderna. Toda la información aquí descrita, ha sido recogida del artículo "Fencing history" de Gérard Six publicado en el sitio web de la Federación Internacional de Esgrima [1].

Según Gérard Six para establecer el inicio de la esgrima propiamente dicha, la esgrima como deporte relacionado de cierta manera con la esgrima actual, hay que remontarse al siglo XVI, con el auge de los duelos de arma blanca en toda Europa. Hasta ese entonces se tienen referencias de algún tipo de esgrima muy primitiva en el 1190 a.C. en Egipto o la inclusión de un deporte similar en los Olimpiadas en la Antigua Grecia allá por el 776 a.C. Pero no es hasta el surgimiento de las armas de fuego y su introducción en el mundo bélico cuando los duelos a espada se convierten en un verdadero deporte, muy similar a lo que hoy en día conocemos como esgrima moderna.

A lo largo del siglo XVI las armas utilizadas para este tipo de duelos fueron evolucionando y perfeccionándose hasta llegar finalmente a la conocida como espada ropera, una espada de hoja recta y larga con forma similar al "florete" o a la "espada", ambas actualmente usadas en la esgrima moderna. Al mismo tiempo, en España, Italia y Francia, empezaron a verse publicados ciertos estudios sobre el arte de la esgrima incrementando la popularidad de este "deporte" y elevándolo a su vez al nivel de "arte".

Pronto un nuevo tipo de arma llamada "florete" surgió, a la vez que poco a poco se introducían nuevas reglas a los ya conocidos duelos enfocadas a hacer de estos duelos un verdadero deporte. Diferentes prototipos de máscaras fueron introducidas consiguiendo de esta manera reducir la peligrosidad de los tocados. Y rápidamente la esgrima pasó a constituirse como deporte, donde la puntuación, el número de tocados válidos, era más importante que la técnica o belleza de los movimientos. Es a partir de este momento donde se introduce el término esgrima moderna para referirse a este tipo de esgrima cuyo principal objetivo es sumar más tocados válidos que el rival.

Con la llegada de la esgrima moderna, este deporte, solo en sus disciplinas de florete y sable, fue incluido en los Juegos Olímpicos de Atenas de 1896. Unos años más tarde, en 1913, la FIE (Federación Internacional de Esgrima) fue constituida. Finalmente, fue en 1936 cuando un aparato eléctrico de señalización de tocados fue aceptado por la FIE, haciendo de la esgrima un deporte más justo, a la vez que restaba movilidad a los esgrimistas debido a los aparatosos cables que los unían al aparato.

A lo largo del siglo XX, se han ido introduciendo mejoras sustanciales en el campo de la señalización de los tocados en la esgrima, pero siempre manteniendo el concepto de tener al esgrimista o tirador unido mediante un cable a al aparato de señalización.

1.2 Principales Objetivos del Proyecto

En la sección anterior los beneficios obtenidos gracias a la introducción de nuevos métodos eléctricos a la hora de señalizar los tocados en la esgrima han sido brevemente descritos. La introducción de estos nuevos métodos dotó a la esgrima de una nueva dimensión, una dimensión más justa y precisa donde la responsabilidad de juzgar la materialidad de un tocado no recaía sobre una sola persona. Sin embargo, la adopción de estas nuevas tecnologías tiene su coste. Los sistemas usados a lo largo del siglo XX estaban basados en la idea de conectar los tiradores al aparato de señalización mediante un cable [1]. Los resultados fueron una serie de sistemas realmente precisos, pero que al mismo tiempo hacían a los tiradores perder movilidad.

La ingeniería eléctrica y electrónica ha experimentado un gran crecimiento desde la introducción de ésta al mundo de la esgrima, intentando siempre acercar este deporte a lo que fue en sus orígenes.

Este hecho, acercar la esgrima a sus orígenes, ha sido el objetivo de muchos grupos de investigación y compañías dedicadas al mundo de la esgrima en los últimos años, y también, el objetivo de este proyecto. Este proyecto quiere contribuir al mundo de la esgrima diseñando el prototipo completo de un sistema de puntuación *wireless* aplicado en este caso a la disciplina del “florete”, con la principal idea de conseguir una buena precisión usando componentes de bajo coste. Una descripción gráfica del sistema se muestra en la Figura 1.

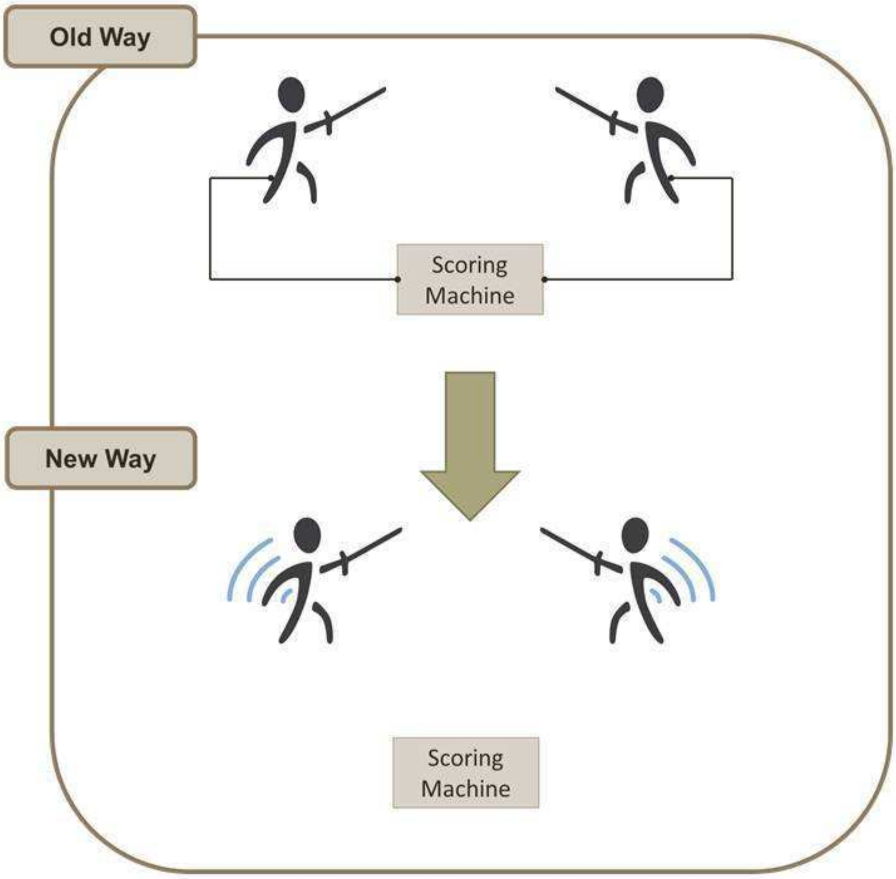

Figura 1: Descripción Gráfica de los Objetivos del Proyecto

1.3 Cómo los Objetivos Propuestos Serán Conseguidos

La manera de conseguir los objetivos propuestos en la sección anterior será produciendo el prototipo de sistema de puntuación haciendo uso de nodos *wireless*, dando solución a la sincronización entre tiradores, a la forma en la que el florete es conectado al nodo del tirador, y a la interfaz que el juez controlará en un combate real.

A la hora de conseguir una correcta sincronización entre esgrimistas, la etapa más compleja del proyecto, una nueva capa por encima de los protocolos *wireless* existentes será diseñada. Esta capa, completamente adaptada a los propósitos del proyecto, será responsable de conseguir la precisión buscada a la vez que es capaz de controlar el tráfico producido entre nodos.

Se espera también del proyecto un completo esquema de cómo el florete debe ser conectado al nodo del tirador. El sistema proveerá un esquema físico y lógico del florete y el chaleco del tirador que se encargará de diferenciar perfectamente entre los dos tipos de tocado, el tocado válido y el tocado no-valido.

En último lugar, es necesario que el sistema sea fácil de manejar, de manera que se abstraiga de todo el proceso técnico para facilitar la labor al juez o encargado de llevar la puntuación en un supuesto combate real. Este proyecto incluirá un pequeño y sencillo software ejecutable bajo un sistema operativo Windows que permitirá tener control total de la puntuación mediante la pulsación de unos sencillos botones.

Finalmente como resultado obtendremos el prototipo completo de un sistema de puntuación de esgrima que cubrirá la mayoría de aspectos de un sistema real.

2 Resultados

Los resultados obtenidos por este proyecto son varios y están divididos según los componentes del proyecto a los que hacen referencia.

En primer lugar es necesario hablar del protocolo de comunicación y sincronización diseñado sobre el protocolo de Texas Instruments SimpliciTI. Se ha conseguido diseñar un protocolo apropiado para la aplicación que da solución original a los problemas de sincronización entre nodos. Este protocolo permite una comunicación sin colisiones, una manera lógica de medir el tiempo real de combate y un error máximo de sincronización entre dispositivos de 0.18 milisegundos.

En segundo lugar, el diseño de un sistema lógico integrado en el equipo del esgrimista para detectar los tocados puede ser considerado un logro importante. El sistema es capaz de detectar y diferenciar todos los tocados posibles, tales como un golpe en un área válida, un área no válida, o en el florete del adversario, mediante un sistema de cables conectados entre el nodo del esgrimista y su equipo.

Finalmente es importante presentar los resultados del sistema completo. El sistema diseñado ofrece una solución *wireless* al problema de la esgrima "cableada". Los logros más importantes de éste son sus componentes de bajo coste y bajo consumo, y la precisión de 2 milisegundos conseguida, mayor que la exigida por la FIE para este tipo de dispositivos [17].

3 Conclusiones

Como ha sido explicado en la introducción de esta memoria, este proyecto tiene como objetivo mejorar aspectos de los sistemas *wireless* de puntuación en la esgrima y ayudar en el desarrollo de este campo.

Después de los resultados obtenidos se puede decir que ciertos aspectos de estos sistemas de puntuación han sido mejorados. Sin embargo, es importante advertir que todavía queda mucho camino que recorrer en el campo de la esgrima *wireless*.

El desarrollo de nuevas herramientas *wireless*, tales como el kit EZ430-RF2500, proporciona a estos proyectos la posibilidad de mejorar aspectos como la precisión o el consumo, que ayudarán a llegar al principal objetivo de este campo: construir un correcto, seguro y preciso sistema de puntuación *wireless* que finalmente sea capaz de reemplazar para siempre a los antiguos sistemas cableados.

Este proyecto ha intentado aportar un grano de arena más en la consecución de este objetivo desarrollando el completo prototipo de un sistema centrado en los aspectos de bajo consumo, bajo coste y alta precisión.

Para conseguir este objetivo, ha sido diseñado un sistema lógico que da solución al problema de la detección de los diferentes tocados completamente adaptado al equipo del esgrimista. Se ha conseguido un protocolo fiable y preciso con una precisión de 2 milisegundos. Por último también ha sido producida una interfaz gráfica que permite la comunicación del sistema con un PC con sistema operativo Windows.

En conclusión, se ha conseguido un prototipo completo de un sistema de puntuación para la disciplina de florete basado en componentes de bajo coste y consumo obteniendo una gran precisión.

Proyecto Fin de Carrera Original

An investigation into the improve of the actual fencing timing systems

Pedro Fidel Espín López

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
Computer Systems and Networks

School of Computing

April, 2012

Authorship Declaration

I, Pedro Fidel Espín López, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed:

Date:

Matriculation no:

Data Protection Declaration

Under the 1998 Data Protection Act, The University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Abstract

As a result of the improvement in all the areas of technology, its application field has been growing faster and faster. Sports are part of that field, and particularly fencing as one of the most technology influenced sport.

All these improvements in the fencing field have always pointed to the accuracy and fairness of the systems used, having the fencer obligatory connected to the scoring machine through a wire. However, in the last years and with the help of the development in the wireless technology area several system prototypes have started to appear. This report tries to be part of these studies and contribute in the technologic change from the wired fencing to the wireless fencing.

In order to provide something useful to this field, the main objective of this project is to produce a wireless based prototype of a complete foil fencing scoring system using the EZ430-CC2500 wireless development tool of Texas Instruments.

To get these objectives it has been designed a properly logic system which gives a solution to detect the different hits totally adapted to the fencer's equipment. It has been achieved a reliable and accurate protocol with an accuracy of 2 milliseconds. It has also been produced a graphic interface which can communicate a Windows PC with the microcontrollers.

In conclusion it has been achieved a whole prototype of a foil fencing system made with low-cost and low-power components and obtaining a great accuracy.

Table of contents

List of Figures

List of Tables

Acknowledgements

This project could not have been finished without the regular advices of my supervisor, Frank Greig. I am very grateful for his assistance in the critical moments of the project.

Furthermore, I would like to thank my proof reader and friend Maria Marset for the invaluable help offered during the realisation of the dissertation.

The project could not have successful without them.

Introduction

The purpose of this introduction is to present the general ideas of this project, especially the “why” and the “how” as brief as possible. Initially it will make a brief journey through the history of fencing, then it will link with the aims and the objectives of this project, and finally it will end explaining the way followed to get these objectives.

Brief overview of the history of fencing

Due to the scope of this project, the starting point chosen to describe the importance of the history of fencing in this section is going to be the rebirth of the thrusting fencing in the XVIth century. All this information was collected from the Master Gérard Six article “Fencing History” published in the International Fencing Federation in its website [1].

Several authors do mention about a young and primitive fencing in the High Egypt in 1190 BC, the inclusion of a fencing games in the Olympics Games in the Antique Greece in 776 BC and the relation of the Roman gladiators games and Middle Age duels with the modern fencing. However is with the emergence of the gunpowder when the fencing duels start to become into a true sport, very similar to modern fencing.

Along the XVIth century the swords used in the duels were evolving until finally coming to the “rapier”, a slender and sharply pointed sword with a form very similar to the actual épée or foil. At the same time, several studies of the art of fencing started to be published in Spain, Italy and France increasing the popularity of this sport called “art” by most of fencers and masters.

Soon a new type of sword called foil arises, and the street duels become into a true sport with its own rules. Different prototypes of masks were introduced in order to reduce the hazard of the games. Fencing evolved from “art” to “sport”, a sport where the score, the number of hits, were more important than the technique. So it is now, in this moment of the history when the term “modern fencing” can be used to make reference to the fencing sport.

With the coming of the modern fencing, the fencing sport, only in its foil and sabre form, was included in the Olympic games of Athens in 1896. Several years later, in 1913, the FIE (International Fencing Federation) was formed. And finally, the electric apparatus of signalisation of hits was adopted by the FIE in 1936, making fencing fairer, but at the same time making the fencers lose mobility.

Along the XX^{th} century, fencing was introducing changes in its rules, but always keeping the concept of fix the fencer to the scoring machine.

Aims and objectives

In the section above the benefits of using electrical ways to signal the duels have been shown. The introduction of a scoring system based in the new electric technologies gave to this sport a new dimension, an accurate and fair dimension. However, adopting these new technologies had a cost. The systems used along the XX^{th} century were based in the idea of connect the fencers to the scoring machine through a cable [1]. The results were a really accurate system which at the same time made lose the full mobility to the fencers.

Electric and electronic engineering have had a big evolution since the beginning of the XX^{th} century, and is natural try to find a proper solution to convert the world of fencing into a sport closer to its origins.

This fact has been the aim of many investigation groups and fencing companies the last years and also, the aim of this project. This project intends to contribute to the world of fencing, designing a prototype of a complete wireless based scoring system, applied in this case to the foil discipline, with the main idea of getting a good accuracy, using low cost components. A graphic description of the system is shown in Figure 1.

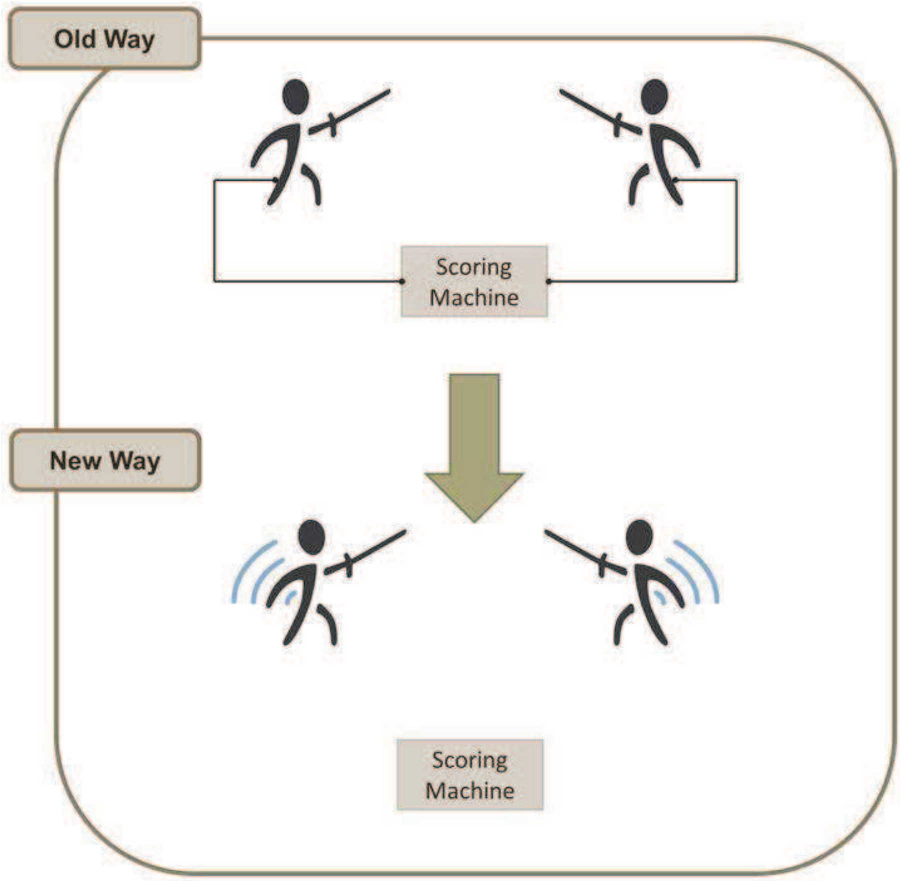

Figure 1: Transition from the old way to the new way

How the aims and objectives will be achieved

The way of getting the objectives shown in the section above is producing a prototype of a system using wireless nodes; giving solutions for the synchronisation of the fencers, the way the foil is wired to the fencer node, and the interface the referee is going to control in a real match.

In order to get the synchronisation between fencers, the main milestone of the project, a new layer over the existing protocols will be designed, one layer completely adapted to the purpose of the project. This layer will be responsible of getting the accuracy searched and be able to manage the traffic between nodes without interferences.

A complete schema of how the foil is going to be connected to the fencer node is also expected. The system will provide a physical and logical schema to design the foil and the vest in order to perceive the two different types of hits, the valid touch and the non-valid touch.

In last place, an easy way to control the system is also necessary. This project will include a small piece of software which allows have control of the whole system by simple button presses.

Finally we will have as result a complete prototype of foil scoring system which will cover most of the aspects of a real system.

1 An evaluation of fencing scoring systems

This chapter has the aim to provide an overview of the different fencing scoring systems and analyse the advantages and disadvantages of them.

The first section of this chapter is dedicated to the field of the wired systems. This concept of system has been used in all fencing competitions from its insertion into fencing in the 30s until the appearance of the wireless scoring systems the first years of this century [1]. The section provides an overview of these systems explaining how they work and analysing their strengths and weaknesses.

The second section is focused into the concept of the new wireless systems, how have they tried to solve the problems associated to “wired fencing” and which is the state of the projects ongoing.

1.1 Wired scoring systems

The electrical wired scoring systems are based on the idea in which the competitors are wired up to an electrical scoring system. This kind of fencing links the competitor’s weapons with a score board. Sensors in the end of the foils are activated when depressed against the opponents valid target area. Hits to the target area are recorded by the special vest jackets worn by the competitors. Containing hundreds of fine wires within its construction, the lame is an electrical conductor.

A wire runs from the Foil tip all the way up the Fencer’s arm, underneath their glove. Continuing to the lame vest, where the opponent’s hits are recorded, the signal then continues to a box with two lights. One Fencer will have a green light and a white light, the other a red light and a white light.

The colored lights (green and red) activates when a valid hit is recorded. The white light illuminates with non-valid touches [2].

In the following figure it can be seen a description of a patented scoring system made in 2003.

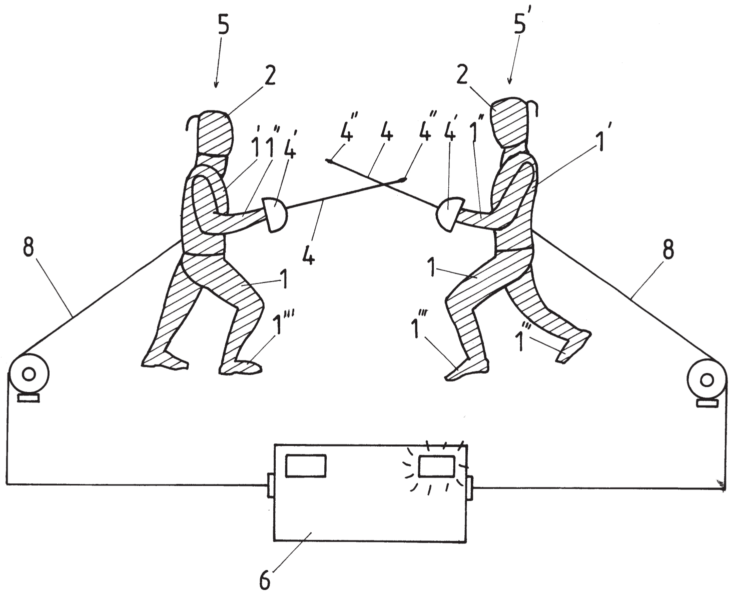

Figure 2: Patent for a wired scoring machine

This system, invented by Frédéric Marciano, is a good example of a modern wired system based on reels [3]. Fencers are linked to the scoring machine through a cable and with the help of a mechanical reel system. This aspect of the scoring system does not allow a full mobility of the fencers and it is at this point where the wireless fencing research starts.

1.2 Wireless scoring systems

Several attempts exist to develop wireless scoring systems for fencing. One of these wireless attempts was developed by a group of engineering students at Simon Fraser University in British Columbia [4]. The wireless system is based on removing the reel system. This system, which was developed only for the epee discipline, uses radio type circuits to replace the conventional reels. When a hit is made, a button on the end of an epee sword is depressed, completing a circuit. Full circuit sends a signal to a small transmitter at the back of the scoring competitor. The judge's box receives this signal confirming the hit and giving properly the point.

Another design of scoring system uses simple "buzz boxes" having compact battery-powered devices that signal touches with a light or a buzzer. But these devices, which are available from various sources, have limited functionality. They cannot distinguish between targets, or distinguish the timing of hits, and do not work with sabre at all.

Another wireless scoring system encountered involves the use of a touch detector unit that is worn by each fencer. The touch detector operates by providing signal indications of fencing touches or hits which uniquely identify each fencer and which signify valid and invalid hits. The hits can be wirelessly transmitted to receiving apparatus coupled to the scoreboard or other scoring equipment [4].

As it can be appreciated, there are good attempts to build a proper system in this field. However, none of these attempts are concluded and are far to be a good solution acceptable by the International Fencing Federation.

One of these attempts, called SG31, was finally accepted by the International Fencing Federation and it was tested on the 2004 Athens Olympic Games [5]. The SG31 is based in the infrared communication to register the different hits. The device can record up to 99 scores for each player, uses RS422 connectors for general purpose computer communications and its price is around $2400 [5]. It is a reliable device but too expensive to be used in local training schools.

Another of these devices was also accepted by the International Fencing Federation. The device called Wireless 2000 and developed by STM, introduced new features such as lights on the helmet of each fencer, and was tested in sabre and foil discipline on the 2008 Beijing Olympic Games [6]. A “starter set” kit of this device for one foil fencer can cost around $1000 [6], still a high price for a single person who wants to try this sport.

2 The design of a wireless based scoring system for foil fencing

This chapter is entirely dedicated to the design process of the scoring system. The design of the system is completely determined by the scoring rules in foil fencing, being necessary a first brief section to explain concepts such as a “valid-hit” or what the valid target area is exactly.

Following the brief explanation of foil fencing rules, the description of the design stage comes. The design stage is divided into another two sections comprising the entire designing task.

The first of this section is called “The design of a valid logic system within the fencer equipment able to differentiate the different possible hits” and comprises the part of the system between the tip of the foil, the vest, and the input ports of the microcontroller. The objective of this part of the system is to design a logical solution able to differentiate between a “valid hit” and a “non-valid hit” and send that information to the microcontroller.

The second design section called “The design of a reliable and accurate protocol able to allow communication between nodes keeping them synchronised” comprises the part of the system between the fencer nodes and the static system. This section describes how a protocol has been designed in order to establish a synchronisation between fencers and manage the communication with the static node.

Finally, to end with this chapter, a summary section called “The operation of the whole system” has been included. This section has the purpose of giving an explanation of the system operation, making use of the components explained before. This section will include flow diagrams in order to clarify the different processes.

2.1 A brief overview of the fencing rules in its foil discipline

As it has said above, it is essential to know some basic aspects in foil fencing in order to make this project more understandable. The vocabulary used here will help in the posterior lecture of the technical sections.

The foil is the lightest and more flexible weapon in fencing, and is comprised by the blade and the hilt. The hilt consists of a metal guard and a grip. The blade adds a tip at the end in order to prevent hazard and help with the hit detection. In Figure 3 a modern foil is shown.

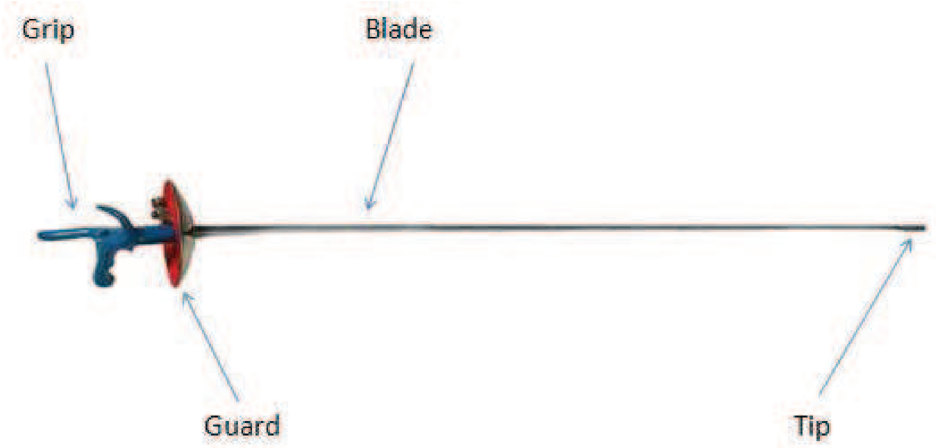

Figure 3: Description of a competition foil

Each fencing discipline has its own weapon, and also has its own rules. An example of these different rules is the target area. The target area is the zone of the fencer body in which a hit of the opponent makes a point for the fencer who does the hit. In the foil discipline, the target area is the torso, from the shoulders to the groin, front and back. The area does not include head, neck, arms and legs. The Figure 4 shows the target area (in dark grey) for the foil discipline.

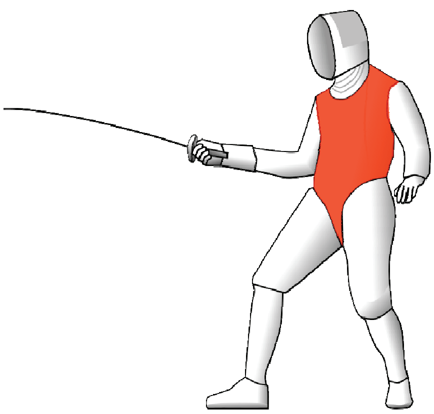

Figure 4: Target area in foil fencing

To conclude this brief section, the two different types of hits should be explained. As it has been said above, when a fencer hits with his weapon in the target area of the opponent he makes a point. In foil fencing this can only happen if the hit is done with the tip of the foil, and that is what this project noun a “valid hit”.

However, in foil fencing another different hit has importance, the “non-valid hit”, which occurs when a fencer hits his opponent out of the target area. This hit has a deeply impact on the game because when it is made the scoring system has to detect it and the game should be stopped.

In summary, a good scoring system has to be able to detect the hits made with the tip into the target area, out of the target area and signal them correctly.

2.2 The design of a valid logic system within the fencer equipment able to differentiate the different possible hits

As it has been said previously, the purpose of this section is to provide the description of the system designed between the fencer device and the fencer equipment (foil and vest), and explain how it works.

It is also important to say that this stage of the design will not be implemented due to the lack of the necessary hardware such as foils and vests.

2.2.1 The description of the physical schema of the system

The system is designed with the aim of creating a way to detect valid, non-valid and other hits, such as hits in the guard, as simple and effective as possible.

Before starting with the description of the cabling, it is necessary to know that the foil and the vest are both conductive surfaces.

The physical schema of the system is very simple. The fencer’s equipment is connected to the fencer’s device through a set of four pins provided by the microcontroller. The different pins are called P1.1, P1.2, P1.3 and Vcc. In that way each pin is connected at the same time with different parts of the fencer equipment.

P1.1 is configured like an input connected directly to a button in the tip of the foil. That input is, at the same time, configured with a pull down resistor to settle the input value to “zero”. That button on the tip, when pressed, makes contact with the rest of the foil and changes its state. The purpose of this pin is to be able to know when the fencer has touched “something”.

All the foil excluding the tip will be at the same time connected to the Vcc output of the fencer device.

P1.2 is also configured like an input with a pull down resistor (so, its normal state is “zero”) and is connected to the tip of the foil without any button.

In parallel with P1.2 there is a cable connected to the Vcc output in the fencer device in order to create a return circuit when the tip of the foil touch a conductive surface.

Finally, the conductive vest is connected to the input P1.3, also configured with a pull down resistor and with the purpose of detect when the fencer has been touched.

The system has been designed with the aim of be simple, however, due to the number of connections, parts of the equipment and pins of the microcontroller used it can be hard to understand. The Figure 5 describes the physical connections of the system.

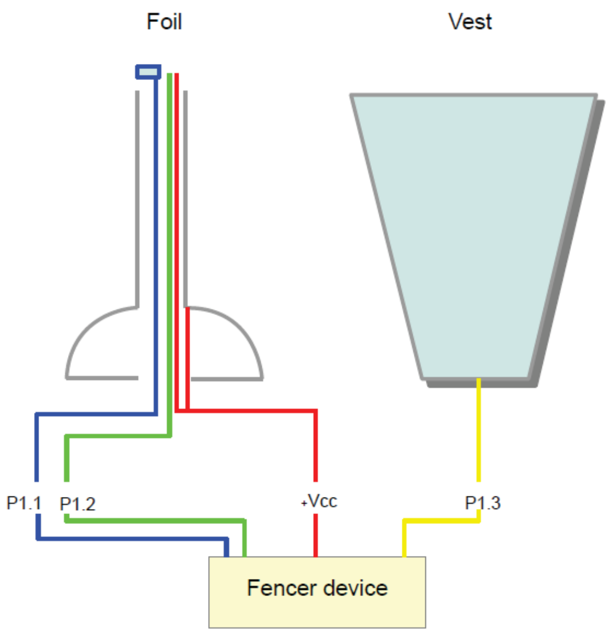

Figure 5: Cabling of the hit detector system

2.2.2 How the system works

Once the physical schema of the system has been explained, it is necessary to explain how it works and the different possible scenarios of a real fencing match.

Firstly, the initial state of the inputs which are all connected to pull down resistors is “zero”. When the fencer touches something with the tip of the foil and with the strength enough to take the button of the tip to make contact with the rest of the foil, the P1.1 change its state to “one”. This is the first step to recognise a hit.

After that, two different scenarios can happen. The first of them occurs if the fencer hits a conductive surface (the vest or the opponent’s foil). In this scenario the contact of the conductive surface against the tip of the foil closes the circuit between the red and the green cable of the last figure (P1.2 and Vcc), giving a “one” in the P1.2 input and at the same time, if the conductive surface is the vest, it also gives a “one” in the P1.3. The Figure 6 shows the behaviour of the system in this scenario.

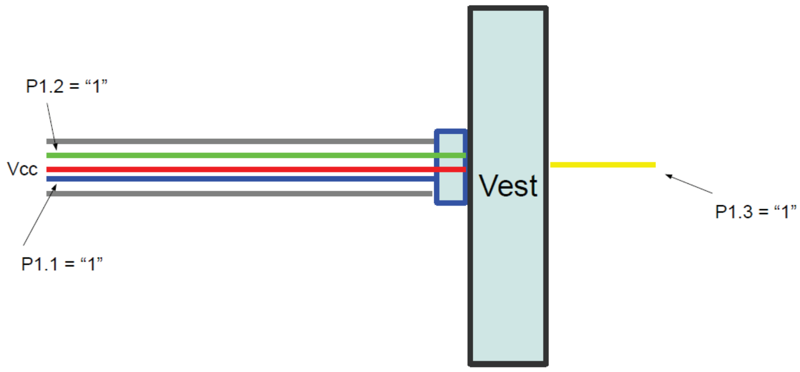

Figure 6: Foil doing a valid hit

The second scenario occurs if the fencer hits on a non-conductive surface such as the rest of the opponent body. In this case, the circuit between the green and the red cable will not be closed, keeping the low state in the P1.2 input as the Figure 7 shows.

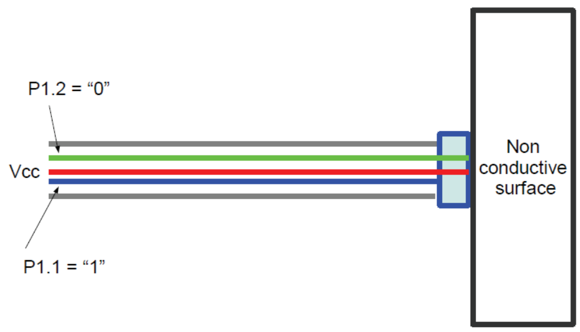

Figure 7: Foil doing a non-valid hit

Until now, It has only been described how the system looks and what response gives to different scenario, but, how does the fencer device interpret all these ones and zeros?

In order to clarify what these signals mean, the Table 1 with the different possible inputs has been created. It will help to understand the behaviour of the system. The table contains four columns. The first of them describe the possible values for the P1.1 input (tip button/blue cable) of the fencer who makes the hit. The second is the

P1.2 input (tip/green cable) of the fencer who makes the hit. The third is the P1.3 input (vest/yellow cable) of the opponent, the fencer who receives the hit.

Table 1: Logic values of the hit detector system

P1.1 (tip button /blue cable)	P1.2 (tip/green cable)	P1.3 (vest/yellow cable)	Description
0	**0**	**0**	**Normal State**
1	**0**	**0**	**Non valid hit**
1	**1**	**0**	**Hit on the foil**
1	**1**	**1**	**Valid hit**

Therefore, the steps realised by the fencer device to recognise a hit are the followings:

1 - Watch value of P1.1

- P1.1=0. Do nothing. Normal state.

- P1.1=1. Something touched. Go to step 2.

2 - Watch the value of P1.2.

- P1.2=0. Non-valid hit. Inform.

- P1.2=1. Foil or vest touched. The responsibility of differentiate between this two relies on the opponent device.

At the same time, the steps realised to recognise when the own fencer has been touched are the followings:

1 - Watch the value of P1.3

- P1.3=0. Do nothing. Normal state.

- P1.3=1. I have been touched. Valid hit.

The software part of this logic solution will be explained in the next section of the chapter.

2.3 The design of a reliable and accurate protocol able to allow communication between nodes keeping them synchronised

As the introduction of this chapter has explained, this section has the aim of describe the solution created to manage the message exchange and to deal with the synchronisation problems.

In order to deal with these problems, a protocol though to work over Texas Instruments protocols has been designed. The explanation of the solution will be divided in several subsections from highest to lowest level of abstraction in order to describe it as clear as possible.

2.3.1 Overview of the service

The protocol designed has the aim of providing a correct, accurate, and wireless environment for the practice of foil fencing which means that the protocol will be totally adapted to rules and objectives of fencing.

The protocol difference between two types of the devices: the fencer device (represented by Figure 8) and the scoring system (represented by Figure 9).

Figure 8: Representation of the fencer device

Figure 9: Representation of the scoring system

The fencer device will have the responsibility of keeping its time “up to date”, to be able to differentiate between valid and non-valid hits and inform the scoring system when one of this hits happens.

The scoring system has the control of the entire system and has the responsibility of keeping fencer’s devices synchronised, be able to receive properly the messages from the fencer devices and interpret them correctly in order to give the appropriate result in a fencing match, in other words, know “who hits first”.

2.3.2 The environment of the protocol

The protocol is thought to work in a particular scenario. This scenario is fixed and consist of two fencers, each one fitted with his own fencer device, and a PC fitted with the scoring system and controlled by a referee. The fixed distance from the apparatus to the fencer’s court is 2 metres [7].

Therefore the work scenario is basically formed by two mobile fencer devices, each one carried by one fencer, exchanging messages with a static scoring system placed within 2 metres from them. The Figure 10 shows the scenario.

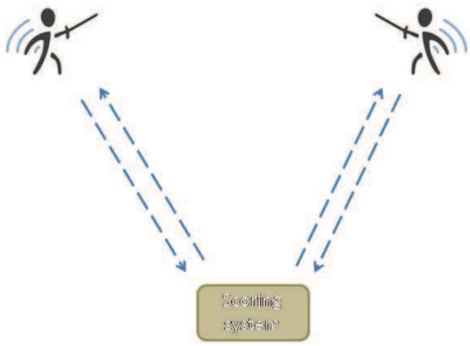

Figure 10: Environment of the protocol

Another important restriction of the scenario is the frequency used by the two fencer devices. Both fencer devices have to share the medium, same frequency; this means that collisions can happen. During the tests of the systems and as the chapter 4 “Test and measures of the prototype” shows, these collisions appear in some scenarios of the exchanging message process, this situation made necessary to redesign the system in order to prevent these collisions. The solution will be explained in the next section.

2.3.3 Rules of service

2.3.3.1 The concept of “splitting the time”

The discussion about how to measure the time in synchronisation protocols has been faced several times in the last years. The way to interpret the time, in a continuous way or in a discrete way, is crucial for the development of new synchronisation protocols.

This project has decided to use a discrete approximation of the time in order to reduce the collision chance and have more control about the possible errors of synchronisation.

This idea was inspired by the famous Aloha protocol. This protocol shares several scenario features with the project protocol such us the use of a unique frequency for all the communications. The first version of Aloha protocol was quite simple and did not implement the slot mechanism. In order to reduce the collisions and have more control about when the packets are sent, a new version of the Aloha protocol was released. This version included a slot mechanism which only allows to send a packet at the start of a "slot" [8].

The reason to implement this mechanism into the protocol is double. In one hand the static device gets a way to keep the fencer devices synchronised through sending a synchronisation packet, advertising them of the beginning of a "slot". In the other hand the mechanism provides the designer an easy way to identify the maximum error which in no case can be higher than the time between "slots".

This protocol splits the time two times depending on the type of device. Firstly, under the point of view of the static device, the scoring system, there will be only "blocks" or "slots" of 50 ms each. These slots have the aim of give to the scoring system a way to resynchronise the fencer devices in order to keep the synchronisation between them.

Secondly, the fencer devices will split the time into blocks of one millisecond. In that way, the protocol will be able to get more accuracy when a time stamp is required. This method only allows the fencer devices to send packets at the beginning of each one millisecond "slot". The following Figure 11 shows how this "splitting" mechanism works.

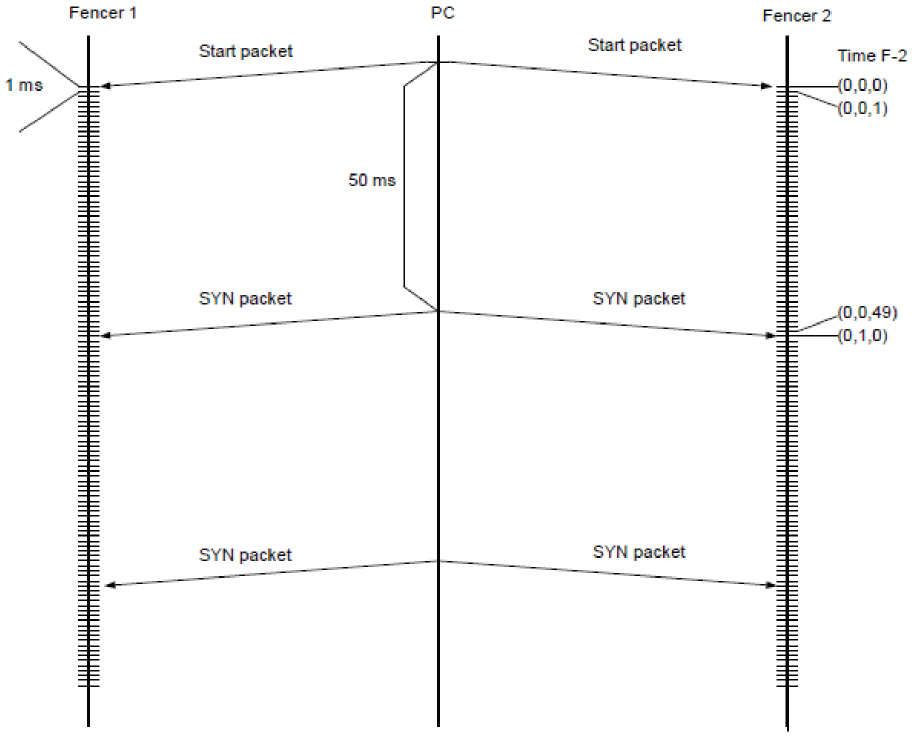

Figure 11: Synchronisation mechanism used by the protocol

The mechanism presents a good way to control the synchronisation and measure the time between devices. However, the reader can ask himself why these values have been chosen. The process to select the right values to make the protocol work in the most efficient way is very complex and requires a proper math study and several tests to achieve them. As the introduction of the project says, the aim is to help those different ongoing works in this area, and it is not to produce a product which can face the actual systems in the market. Taking these considerations, it has been decided not to spend time and efforts trying to find these perfect values and chose them taking into account more simplistic features.

Regarding the value of the 50 milliseconds, the lower we take the value, the more accurate the system will be. However, if a very low value is taken, the project has the risk of flooding the network too much and disturbs the exchange of messages. Taking this into account, the value was fitted in 50 milliseconds.

The 1 millisecond value had to be a value much higher than the maximum synchronisation error in a 50 millisecond period (obtained during the tests stage and explained in the test chapter), and low enough to give the accuracy required.

2.3.3.2 How the time is measured

The control of the time is crucial for the project in order to differentiate the winner in the case of a double hit.

The concepts discussed previously can confuse the reader regarding the way the protocol manages the time. In order to explain this, the hardware restrictions have to be taken in consideration. The project, as way to implement this system, makes use of a microcontroller which does not allow the programmer to manage the data as he likes. The microcontroller used, which will be explained in the next chapter, works (simplifying the explanation) with variables of 8 and 16 bits. In order to have a more flexible way to control the time, the protocol will work with 3 variables of 8 bits each one. In this way the protocol uses three variables with a maximum value of 255 (2^8-1) each one.

The protocol interprets the time in a different way the humans do. It forgets about minutes and hours, and it controls time making the fencer device responsible to increase separately the three variables. The first variable will be increased each 1 millisecond and will be restarted when the synchronisation packet comes each 50 millisecond. The second variable will be increased when the synchronisation packet comes. Finally, the third variable will be increased when the second variable overflows, this third variable will give a larger temporal space during the operation of the system. The maximum time the system can work well before the reset of the time value is calculated in the next equation.

$$\begin{aligned} maximum\ time &= 255 * 255 * 50 = 3251250\ milliseconds \\ &= 54\ minutes\ and\ 11\ seconds \end{aligned}$$

As the maximum duration of a fencing round is 3 minutes [9] and the time values are restarted at the end of each round or when a hit is made, the maximum time is enough to the project purpose.

2.3.3.3 *Each fencer device has its own slot to send packets*

Another crucial feature of the protocol which was ignored in the early times of the design stage is the assignation of each slot to each fencer device. This means that each fencer device can only send a packet at the beginning of its slot, not any slot, being the slots properly distributed between the fencer devices.

This idea arose later, during the tests and measures stage, when a piece of software was built to simulate a double hit during exactly the same millisecond in both fencers. The test gave as result that when this scenario happens, a collision between devices occurred and the scoring system was not able to detect the packets properly.

In order to solve that, a design solution was thought. To prevent the system from the collisions, each fencer will have its own slot. Therefore when a packet requires to be sent, a fencer device will have to check the actual slot, see if the slot allows it to send, and if not, wait until the next slot and repeat the process. The Figure 12 explains briefly the decision mechanism.

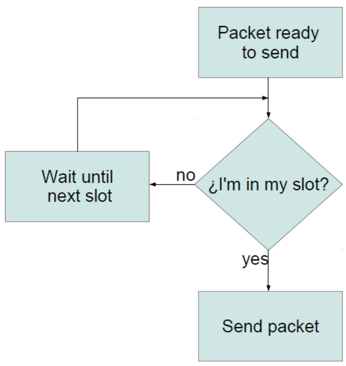

Figure 12: Decision mechanism used by the fencer device to send a packet

The way to distribute the slots between the fencer devices has been designed taking into account fairness and especially the time a sender needs to send a packet. The measures made to know this time are explained in chapter 5, and give as result a time between 1 and 2 milliseconds. Consequently the time assigned by the protocol to each fencer device to send a packet was fixed to 2 milliseconds. This means that the protocol will give 2 slots to the first fencer device to use it or not, then the next 2 slots will be given to the second fencer device and it should decide use it or not, and in this way the process repeat its operation.

The protocol does not work if the fencer devices are not forced to send at the beginning of the first slot assigned to them. Otherwise a collision could arise.

The Figure 13 shows in a cycle of the scoring system (50 milliseconds) how the slots are divided into the fencer devices. The slots coloured in dark grey represents the slots assigned to the first fencer and the light grey slots represents the slots assigned to the second fencer.

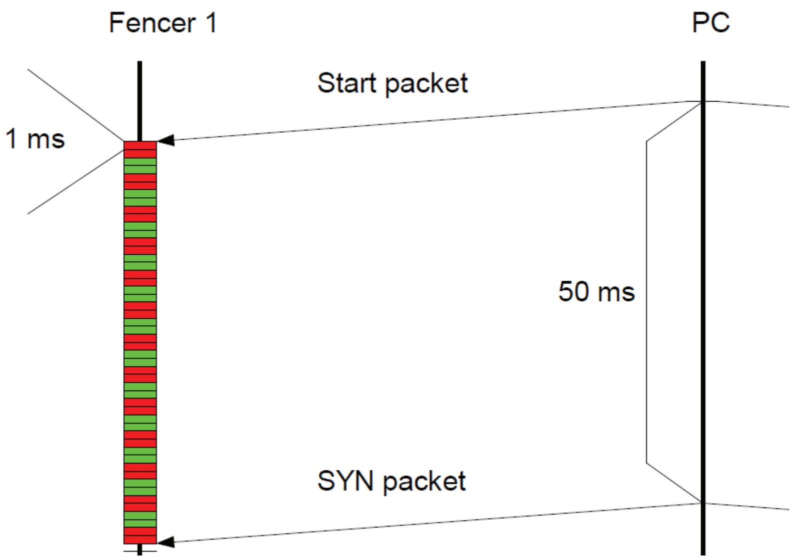

Figure 13: Distribution of the 1 ms slots

2.3.3.4 How the scoring system receive all these packets

Until now, this section has described how the protocol operates always from the side of the fencer devices, how this protocol allows the fencer devices keep its time synchronised and send packets without the fear of a collision.

In this subsection, how the protocol works under the point of view of the scoring machine will be explained.

As has been said before, the responsibility of keeping the time "up to date" lies on the fencer devices. Therefore, the scoring machine, during his normal operation has not need to implement a way to measure real time, except for a little clock which notifies the system each 50 milliseconds to send the synchronisation packets.

Once this has been explained, a new challenge in the side of the scoring machine arises. How the scoring machine is able to receive the packets from the fencer devices? What it has to do when a packet is received?

In order to give an answer to these questions a solution for this problem was thought. The scoring machine, during the normal state, is open to receive any packet. When a packet is received, the scoring machine analyses it and if it is a valid packet, it decides to open a temporal window to watch if a double hit has been produced. If the temporal window comes to its end, the point will be given according to the packet that has come. If within the temporal window another packet comes, it will be analysed and the hit times will be compared in order to differentiate a single hit from a double one. The Figure 14 describes correctly the operation of this solution.

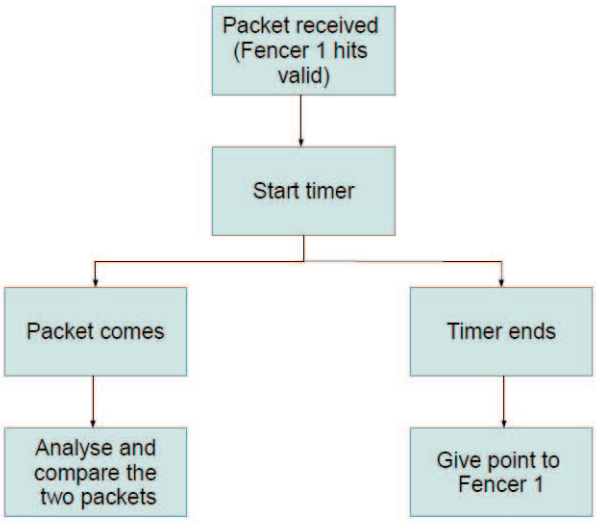

Figure 14: Mechanism implemented in the scoring machine to manage a double hit

To end with this section, it is necessary to explain which value has been taken for the timer explained in the figure above. The value was fixed taking into consideration the worst possible scenario in the case of a double hit. This scenario happens when the fencers are resynchronised; in that moment the protocol can assign to the same fencer two slots of 2 milliseconds each, this is the result of the fact that the synchronisation cycle is composed of 50 slots of 1 millisecond which means 25 priority slots of two milliseconds, and 25 is not an even number. Therefore, in order to prevent that a double hit is not registered correctly, the temporal window should be fixed to 4 milliseconds as the lowest value. A value of 5 milliseconds was decided to be implemented in order to give more time to receive the packets. In any case, the value could be higher without affecting the system's behaviour because the time stamps have been taken on the fencer device side.

2.3.4 Message structure used by the system and coding used

As it has been explained before, this protocol has been thought to work over SimpliciTI low-power RF network protocol created by Texas Instruments [10]. A deep description of this protocol and how it was used to incorporate it in the project will be provided in the next chapter as part of the implementation stage.

To explain the format used in the messages exchange it is necessary to explain that this message structure is embedded into the payload of the SimpliciTI packet format as well as a TCP segment is embedded into the payload of an IP packet [11].

The message structure is composed by two main fields. The first field is an 8 bit field and will make reference to the type of message used. This field can take 4 values as the table 2 explains. The second field is the data field and contains the data required to make the system works. This field can take from 0 bits to 24 bits (3 subfields of 8 bits each).

Table 2: Description of the different type of packets

Type field value in decimal	Used by	Description
1	Fencer devices	This is a valid packet
2	Fencer devices	This is a non-valid packet
3	Scoring machine	This is a synchronisation packet
4	Scoring machine	This is a reset packet

In case of sending a valid packet, this field will take 24 bits and will be divided into 3 subfields of 8 bits each. This field will be used to send the value of the time stamp.

As it was explained in the previous section the time is measured using three variables of 8 bits each, therefore each variable will be stored in a subfield of the data field from lower to higher precision.

If the fencer device needs to send a non-valid packet, a time stamp is not required. Consequently, the data field will take 0 bits.

From the scoring machine side, whether it is a synchronisation packet or a reset packet, this field will always take 0 bits. Furthermore, to send this type of message the sending operation will have to be changed to a broadcast sending in order to allow the fencer devices to receive the packet at the same time.

The Figure 15 describes the composition of the different types of packets.

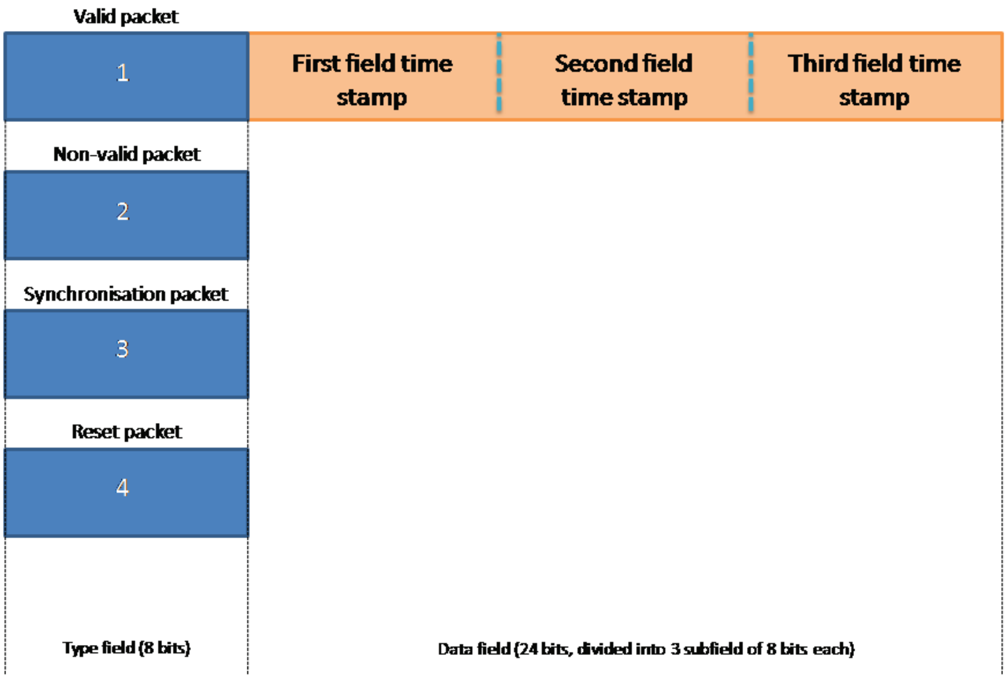

Figure 15: Graphic representation of the packets used

2.4 The operation of the whole system

After explaining the different components and features of the system, the lector can have some doubts about the behaviour of the whole system. In this section the flowcharts of the fencer device and the scoring system with a brief description of them will be included.

These flowcharts are focused in the messages exchange, synchronisation and point decision and they ignore the network operations.

2.4.1 Flowchart and explanation of the fencer device behaviour

The design of the fencer device is based in "sleep" and "wake up" when there is something to do. The Figure 16 shows this.

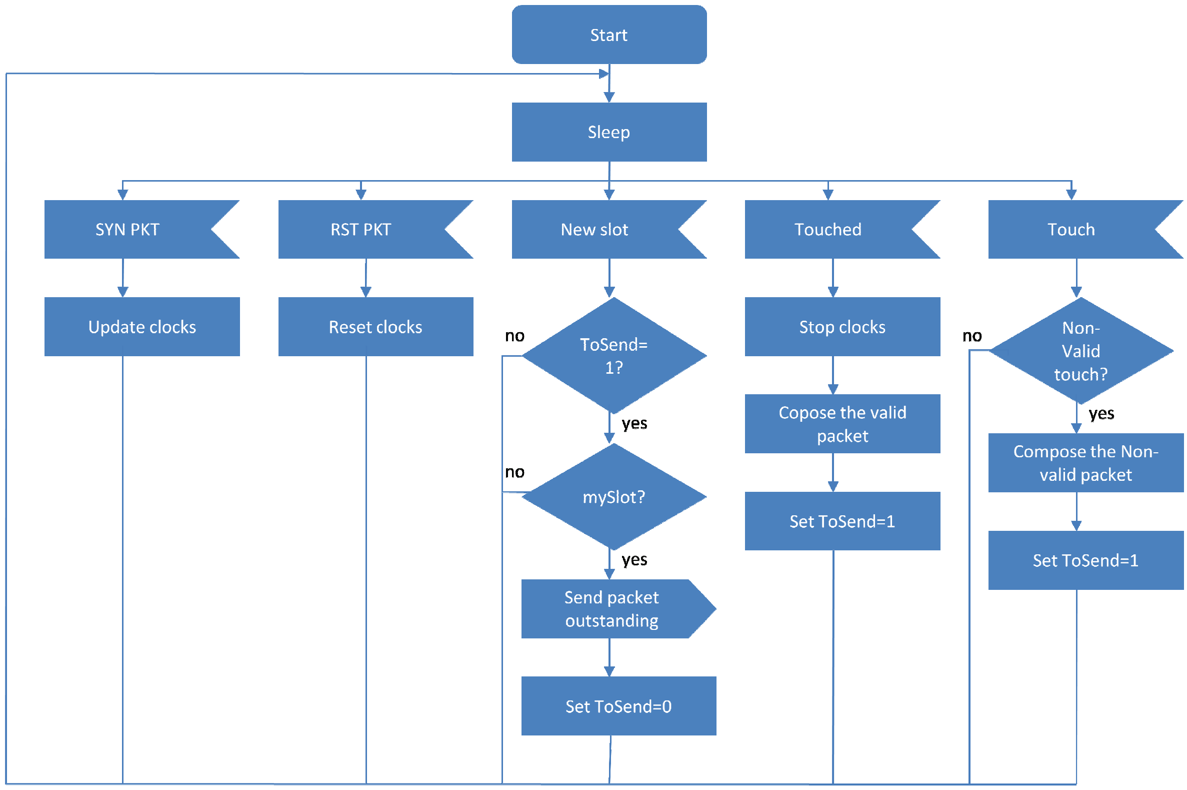

Figure 16: Flowchart of the fencer device

The flowchart shows that the central point of the fencer device operation is the sleep stage. As a battery powered device, save energy is crucial to be able to keep running the device for as long as possible.

The device wakes up from this state when some event occurs. The most frequent event will be the coming of a new slot. This event comes each 1 millisecond as it has been explained in previous sections. After that, the fencer device will check if there is some packet ready to send analysing the value of the variable `ToSend`. If there is nothing to send, the system will go back to the sleep state again, however, if there is something to send the system will check if the actual slot allows the device to send and will send the packet if the answer is "yes". Otherwise the system will go back to sleep again.

The system can also wake up because of the coming of a packet. In this case, if the packet is a SYN packet, the system will update its clocks; whereas that if the packet is a RST packet the system will reset them. After that the system will entry again in the sleep state.

Finally, the system can also be awakened because the fencer has touched something or has been touched on the vest. In the case of being touched on the vest the system will stop the clocks, it will compose a valid packet filling each field with its right value and it will mark a variable called `ToSend` to notify that there is a packet waiting to be sent. When, otherwise, the fencer has touched something, firstly the system will check if the tip of the sword has hit a non conductive surface or not. If it is a non conductive surface then the hit will be "non-valid" and the system will compose the non-valid packet and will mark the variable to notify the outstanding packet. If it is a conductive surface the system can not differentiate between the vest or the guard and it has to go back to the sleep state and leave this work to the opponent device.

2.4.2 Flowchart and explanation of the scoring system behaviour

The flowchart of the scoring system can be more confusing. It is based also in a waiting state, where it waits to receive any packet or the timer interruption. The Figure 17 shows that.

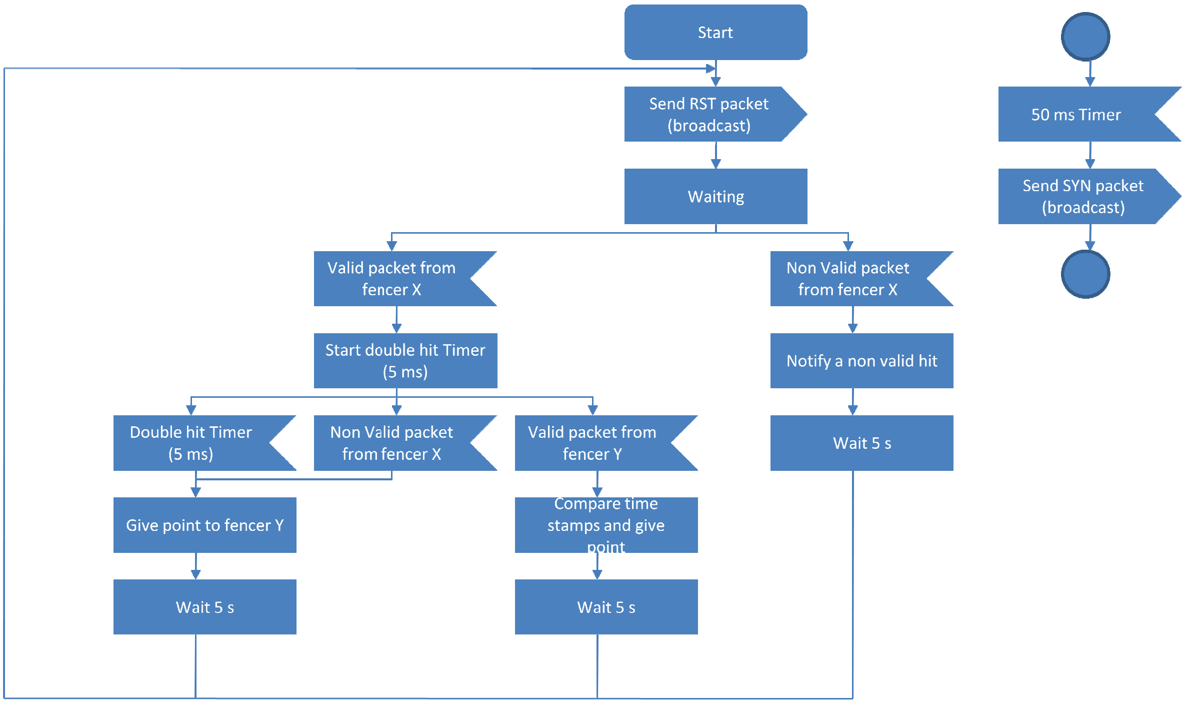

Figure 17: Flowchart of the scoring system

As the flowchart shows, the first action the system realises is to send a broadcast RST packet in order to advertise the fencer devices of the beginning of a round. After that the system enters into a waiting state where it waits for a packet.

Two different types of packet can come, a valid packet or a non-valid. When a valid packet comes, the system will start a 5 milliseconds timer in order to give the chance to the other fencer to send a packet. If the timer comes to its end and no packets have arrived, a point will be given to the opponent of the fencer who sent the packet. This action will be repeated if within these 5 milliseconds a non-valid packet comes. Otherwise, if a valid packet comes from the other fencer the system will have to analyse the clocks inside the packets and give the point to the fencer with the lower time stamp, or give a double hit in the case of the same time stamp. After all these processes the system will wait for 5 seconds and return to the start of the flowchart.

The processing when a non valid packet comes and the system is in the waiting state is easier and will only consist of notifying the non valid packet and wait 5 seconds. After that the system will return to the start of the flowchart.

Finally, the process of the right corner indicates that in any moment when the 50 milliseconds timer comes to its end, the system will send the SYN packet to keep the devices synchronised.

3 The implementation of a wireless based scoring system prototype into the ez430-RF2500 wireless nodes

This chapter has the aim of going deeper in the development of the project and explain all the real components used to implement the system.

The chapter will be divided into 5 sections. All these sections have the aim to explain each component or tool used in the project with the exception of the last section. This last section called "Overview of the whole system" will show how these components are related to form the whole system.

3.1 The EZ430-RF2500 wireless development kit

The EZ430-RF2500 kit is a set of tools composed by the hardware necessary to build a wireless based, low-power system [12]. The Figure 18 shows a picture of all the components included in the kit. The kit used is composed by:

- Three eZ430-RF2500T wireless target boards.
- One eZ430-RF USB debugging interface.
- One MSP430 Development Tool CD-ROM containing documentation and new development software for eZ430-RF2500.
- Two AAA battery packs with expansion board.

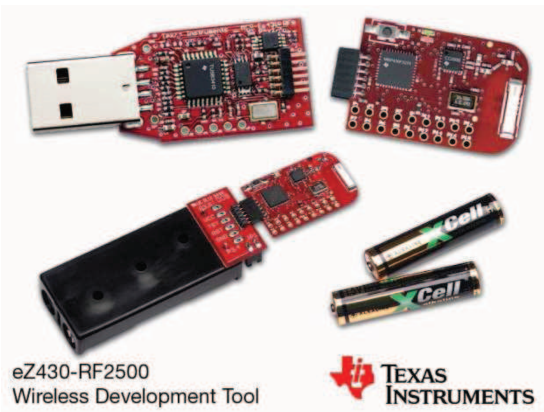

Figure 18: EZ430-RF2500 Wireless Development kit

In order to describe the main features of this kit, it is remarkable the inclusion of a USB interface which make the communication with the wireless target board easier.

It has been very helpful that the wireless target board had more used pins in an accessible way in order to simplify the connections with peripherals. These pins have been used to connect the foil and vest cables explained in the chapter two.

Each EZ430-RF2500 card includes at the same time these components [12]:

- CC2500 - 2.4 GHz, ISM band multi-channel low power transceiver.
- MSP430F2274 ultra-low-power MCU.
- 2 LEDs.
- 1 pushbutton.

The following Figure 19 shows the components of the card.

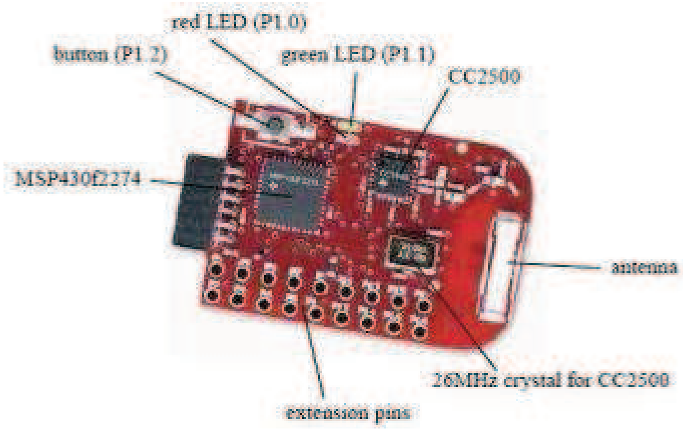

Figure 19: Description of the components included in the EZ430-RF2500 card

Each card can be connected to the battery expansion board or to the USB interface to work. The USB interface lets the programmer send and receive data from the PC to the board using the UART msp430 application.

The UART is the base of the communication system. It is located on the board and its main task is to handle the interrupts of the devices connected to the serial port.

Within the card, the most important components are the microcontroller and the transceiver.

3.1.1 Transceiver CC2500

The CC2500 is a low-cost 2.4 GHz transceiver that was designed for very low-power wireless applications. The modem supports different modulation formats and has a configurable data rate up to 500 kBaud. It is able to store data, work with in a packet level and know the quality of the channel that it is being used.

The main operating parameters of the CC2500 can be controlled by a SPI interface. It uses a FIFO policy to control data and it is also used for being implemented together with a microcontroller and other passive components [13].

In the Figure 20 it is shown the pins and connections of the CC2500.

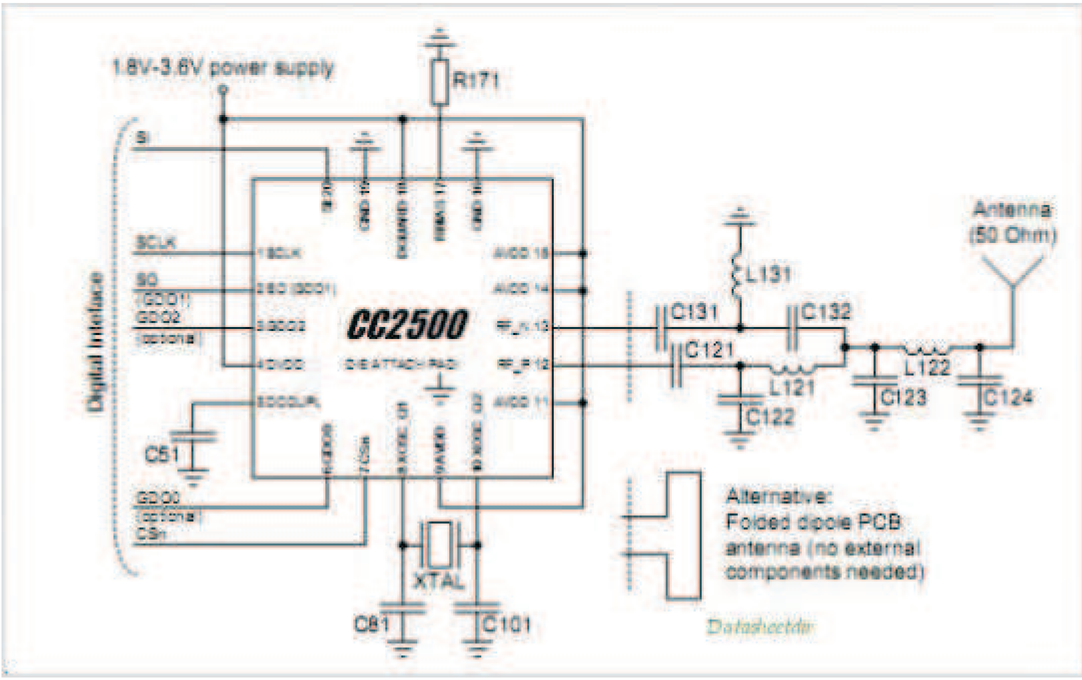

Figure 20: Connections of the CC2500 transceiver

The CC2500 transceiver has different states to realise the different tasks. These states are:

- Send.
- Receive.
- Idle. Ready to receive but saving energy.

- Sleep. All the transceiver components off. Sometimes the wake up process can consume more energy than keep it "Idle".

The main features of the transceiver are:

- High sensitivity (-104 dBm at 2.4 kBaud, 1% packet error rate).
- Low current consumption (13.3 mA in RX, 250 kBaud, input well above sensitivity limit).
- Programmable output power up to +1 dBm.
- Programmable data rate from 1.2 to 500 kBaud.
- Frequency range: 2400 - 2483.5 MHz.

3.1.2 Microcontroller MSP430F2274

This microcontroller is part of the ultra low power family MSP430 [14] released by Texas Instruments. These devices can be used to produce a vast variety of applications. The architecture combined with its five low power operating modes makes this device ideal for the application that this project is intending to build.

This device is characterized by a 16 bits RISC CPU, 16 bits registers and constant generators. The digitally controlled oscillator (DCO) allows a wake-up from low power modes to active modes in less than 1µs.

Other features of the MSP430x22x series are:

- An ultra-low-power mixed signal microcontroller with two built-in 16-bit timers.
- A universal serial communication interface.
- 10-bit A/D converter.
- Two general-purpose operational amplifiers in the MSP430F22x4 devices (which is used in this project).
- 32 I/O pins.

3.2 SimpliciTI network protocol

The CC2500 transceivers make use of Texas Instruments SimpliciTI protocol for the radio communication.

SimpliciTI is a low power RF protocol able to compose networks up to 30 nodes. It is "open-source" software, making it ideal for any low cost project. Due to its ability to remain sleep for long time intervals, is a perfect protocol to be used in low power applications. [10]

SimpliciTI has been designed looking for an easy implementation, using the minimum resources from microcontrollers. Is for this reason and for its low power and low cost design, this protocol has been chosen to build this project.

SimpliciTI classifies the devices inside its network as End Device (ED), Access Point (AP) and Range Extender (RE). In this project the Range Extender will not be used.

The Access Point is the one which creates the network and establishes a connection with the End Devices, marking them with an identifier. The Access Point also stores the messages, make decisions about them, and is responsible of maintain the network operative.

The aim of the End Devices is to search an operating network and collect data to send it immediately to the Access Point.

The SimpliciTI protocol is fully integrated into the EZ430-RF2500 kit and provides several example applications [10].

The project network has been implemented as the Figure 20 shows.

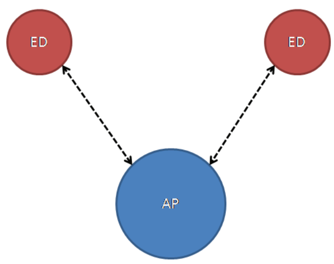

Figure 21: Design of the project network

The system will be composed by two End Devices and One Access point. The two fencer devices will be configured to act as End Devices, this means that their task will be to collect data (signals from the foil) and send it to the Access Point.

The scoring machine will be, at the same time, configured as an Access point. It will have to receive the signals and manage the communication.

3.3 Code Composer Studio

Code Composer Studio is an integrated development environment (IDE) for Texas Instruments embedded processor families. It provides several tools used to develop and debug embedded applications [15].

Code Composer Studio is based on the Eclipse open source software framework.

This IDE was used to implement and debug the program designed on the EZ430-CC2500 devices through the USB interface.

3.4 Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment created for Windows operating systems. It supports several programming languages such as Visual C++, Visual C# or Visual Basic .NET.

With the help of the C++, a high level object oriented programming language; a developer can create a whole desktop application easily [16].

In this way, these tools were used to create a whole graphic interface to communicate the human with the scoring machine. The interface notifies the user of the points scored by the fencers and allows the human to have control of the punctuation.

3.5 Overview of the whole system

This section has the aim of giving a global view of the whole system once the different components and tools of the system have been explained.

In order to clarify the implementation of the system the Figure 22 was made. It shows the project scenario with the real components added.

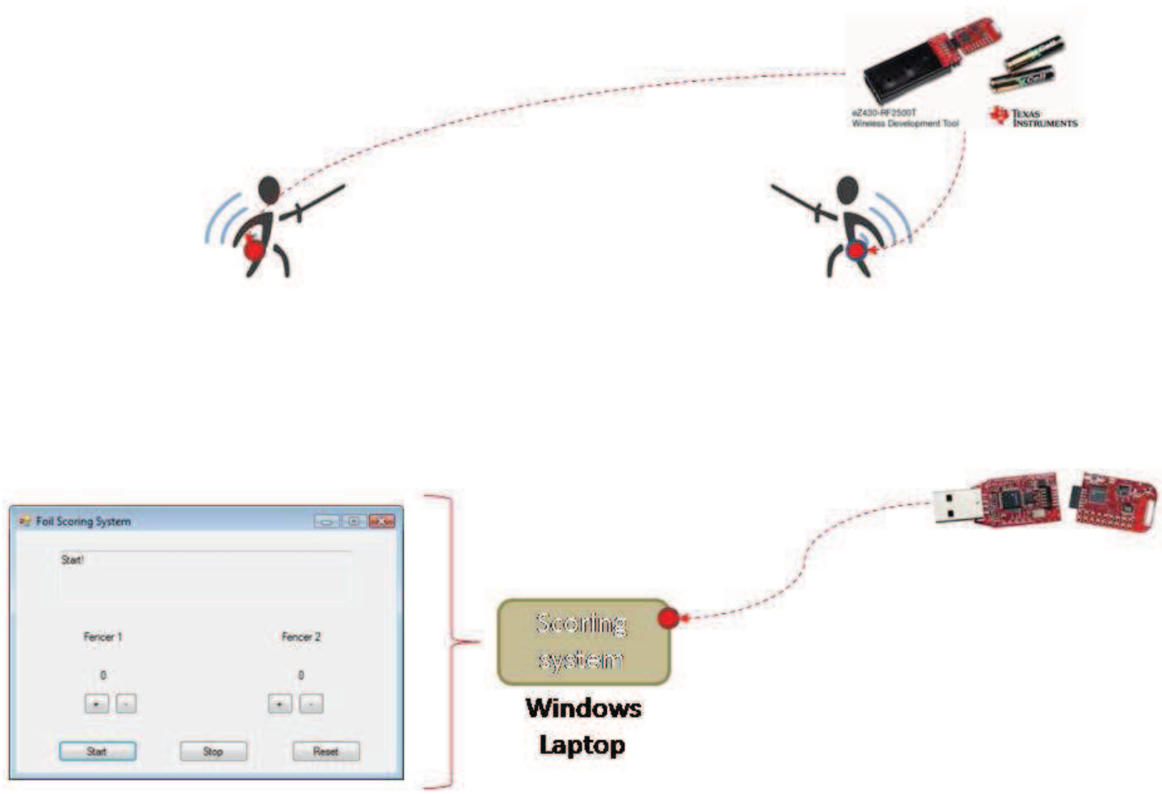

Figure 22: Project scenario with real components

As the figure shows, the basic scenario is composed by the two fencers and the scoring machine.

The two fencers will have to carry with them one battery powered EZ430-RF2500 node each one. This node can be placed in the belt of the fencer.

The scoring machine will be a Windows laptop which will have plugged another EZ430-RF2500 node, but this time through an USB interface. At the same time, the laptop will be running the desktop application created with Microsoft Visual Studio which will help the referee to control the combat. This application is able to communicate with the node connected through the USB interface.

4 Tests and measures of the prototype

This chapter has the aim of describing the different tests made to the system and knowing how the system responds to these tests.

The system will be divided into three sections. Each section involves a type of test made to the system and the results of this test.

The first test made and probably the most important was the measure of the time difference between the fencer devices clocks in order to know the maximum error assumed.

The second test consisted of conducting simulations of different double hit scenarios. This test has the aim of knowing when the system is able to detect a double hit.

And finally the last test consisted on proving if the system presents problems working during a long time, and if there were packets lost.

4.1 The measure of the time difference between fencer devices clocks

As it has been said this test has the aim of measuring the time difference between fencer devices. Knowing this time difference is crucial for the development of the project because it is a value to know the synchronisation of the system.

The tool used to find this time difference was a logic analyser connected to the computer and a modification of the nodes code in order to toggle an output when a new "slot" comes, this means that the output will be toggled each millisecond in the fencer devices and each 50 milliseconds in the scoring machine.

The Figure 23 is a capture of the analyser which shows exactly one cycle of the scoring machine (colour red, channel 0) and 50 cycles of the fencer devices (colours orange and yellow, channels 1 and 2).

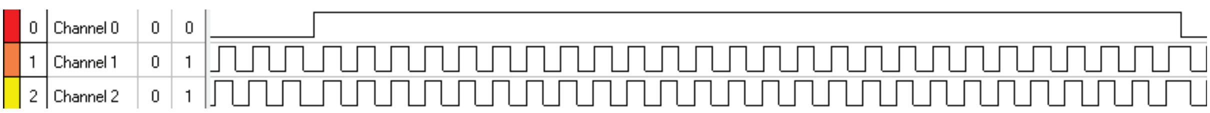

Figure 23: Capture of the logic analyser showing 1 cycle of the scoring machine and 50 cycles of the fencer devices

Analysing the figure a good synchronisation between fencers can be appreciated and how a 50 milliseconds cycle of the scoring machine is composed of 50 cycles in the fencer devices clocks. The figure also shows when an synchronisation packet comes (signal toggled in the channel 0).

However, the most interesting part of the capture is the “synchronisation zone” because it comprises two crucial cycles, the last and the first cycle.

The Figure 24 shows through a zoom of the last capture the “synchronisation zone”.

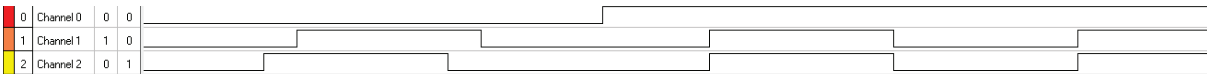

Figure 24: Capture of the logic analyser showing the last cycle of the fencer devices

Focusing on the first cycle after a synchronisation packet comes, analysing the figure the reader can see that it exists a time difference between the coming of a synchronisation packet and the start of the first fencer cycle. This is due to the transmission and processing time of the packet. However, in both fencer devices this time is exactly the same and does not affect the synchronisation between devices.

The other crucial zone is the last cycle of the fencer devices, just before the synchronisation packet comes. After many experiments, and before configuring the system to the 1 and 50 milliseconds values, a difference in the speed of the clocks was found. One device was faster than the other, so a solution was searched adjusting the time to the 1 and 50 milliseconds values. The result is shown in the figure above. A time difference exists, but this time difference is in its maximum value (the last cycle), 0.18 milliseconds.

This result had some effects over the final project. The error is small, but it can happen. If a double hit is produced in that error window, for example where the red line in the Figure 25 shows, one device will send a time stamp of (x,x,48), while the other device will send a time stamp of (x,x,49) being both hits made at the same time.

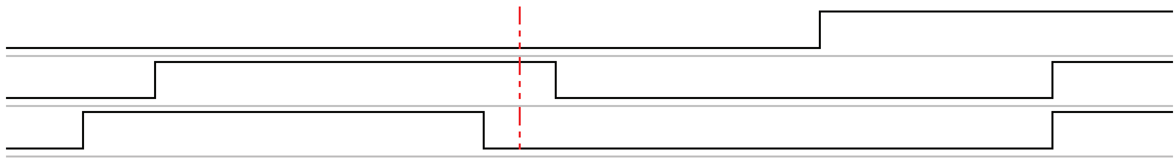

Figure 25: Capture of the logic analyser showing the moment where an error can occur

To solve that, the system was configured to give a double hit or double point when it detects a 1 millisecond difference between time stamps. Knowing this, it can be said that the accuracy of the system is 2 milliseconds. When a double hit is made with a time difference higher than 2 milliseconds the system will be able to detect who scored first, otherwise, the system will notify of a double hit.

4.2 The response of the system to several double hit scenarios

As a result of the previous test and as the last paragraph of the previous section says, the accuracy of the system is fixed to 2 milliseconds. Once that value was known, the system was tested in order to check if the system worked with that accuracy.

To check that, several simulations of different double hit scenarios were made. The program in the devices was modified in this way: it had to run until the timer comes to a pre-configured value and then send a valid hit packet.

The pre-configured values in the two devices were:

1. Both devices with the same value, fixed to 5 seconds.
2. One device configured with a value of 5 seconds and the other one with a value of 5 seconds plus 1 millisecond.
3. One device configured with a value of 5 seconds and the other one with a value of 5 seconds plus 2 milliseconds.
4. One device configured with a value of 5 seconds and the other one with a value of 5 seconds plus 3 milliseconds.

These tests were made and the behaviour of the system expected was that the system could be able to give the point to the first fencer in the 3rd and 4th tests and give a double hit in the 1st and 2nd tests.

It is important to say that during the execution of the 1st test, a design problem was found, when both devices tried to send the packet at exactly the same time, both packets collided and were not received by the scoring machine.

This problem was solved giving different slots to each device to send them packets as it has been explained in the design chapter.

After solving that, each test was made several times and the system answered properly the 100% of the times, confirming the 2 milliseconds accuracy of the project.

4.3 The response of the system when it runs for a long time

The last test made was to leave the system working for an hour, at the same time the fencer devices were configured to send a packet each 5 seconds and notify if a packet were not sent properly. The scoring system was also configured to count how many packets were received.

The result of this test was that the system does not present any problems during the operation and all the packets were sent by the fencer devices and received by the scoring machine.

Results

The results obtained from this project are many, and are divided into the different components of the project.

Firstly it is necessary to write about the communication and synchronisation protocol designed over the SimpliciTI proprietary protocol. It has been achieved a proper protocol which provides an original solution to the problem of the devices synchronisation. This protocol provides a way to communicate without collisions, a logical way to measure the time and a maximum synchronisation error of 0.18 milliseconds between devices.

Secondly, the design of the logic system inside the fencer equipment to detect hits is an important achievement as well. The system provides a way to differentiate a hit in the vest, the sword and a non-valid target with some cables connected to the fencer device.

Finally it is necessary to present the result of the whole system. The whole system, explained in the last section of the implementation chapter, offers a wireless solution to the problem of the wired fencing. The most important achievements of this system are the low-cost components used, such as the EZ430-RF2500, the low power consumption of these nodes and the accuracy of 2 milliseconds provide, better than the accuracy required by the FIE [17].

Conclusions

As it was said in the introduction chapter this project has the aim to improve some aspects of the actual wireless scoring systems and help in the development of this field.

After describing the project it can be said that some aspects of the wireless systems have been improved. However, it can be affirmed that it still exists a long way to go in the wireless fencing field.

The development of new wireless tools such as the EZ430-RF2500 nodes provides to this field the possibility of improving aspects such as the accuracy or the power consumption which will help to get the final challenge: a properly, secure and accurate wireless based system which finally can be able to replace the old wired system.

This project has intended to help with that task developing a complete prototype of a wireless based foil fencing system focused on the low-cost, the low-power and the accuracy.

In order to get these achievements it has been achieved a properly logic system which gives a solution to detect the different hits totally adapted to the fencer's equipment. It has been achieved a reliable and accurate protocol with an accuracy of 2 milliseconds. It has been produced a graphic interface which can communicate a Windows PC with the microcontrollers.

In conclusion it has been achieved a whole prototype of a foil fencing system made with low-cost and low-power components getting a great accuracy, higher than the expected by the FIE [17].

Future work

The wireless fencing field is an arising study area with a long way to go. Several researches exist in this area but the final products released do not satisfy the necessities of the International Fencing Federation to replace the wired system.

The current researches cover several technologies to do fencing wireless such as RFID, Bluetooth, optical technologies and so on. There is still indecision about which technology can be the best for these systems and future researches could clarify the advantages and disadvantages of them and adopt new technologies to this field.

Furthermore, some future researches may cover the aspects associated to the security of these wireless systems, one of the main reasons for the reject of these systems by the International Fencing Federation. A member of the public can flood the air traffic knowing the frequency of the system, even being able to produce fake packets in order to change the results of a match. The development of this area is sensitive because the improvement in the security of a system normally affects the accuracy and agility of it, crucial aspects in fencing.

Nowadays there is a great opportunity for any company of this field to build a system which can be used as the main system in wireless fencing. This system must have a good accuracy, a good security mechanism, low power consumption and a low cost version for the training schools.

Evaluation of achievement

The first point to be critical with the project is the scope of the project and the choice of focusing only the foil discipline. This decision has made possible to give more time to the timing work per se, but with a better organisation of the available time the work could have covered at least one discipline more.

The second point is about the research stage of the project. Due to the confidentiality of the ongoing projects carried by private companies, it has been very hard to have access to technical documents of actual scoring systems. This fact has made difficult the comparison between the prototype designed and the projects out in the market.

It is also necessary to talk about the security problem of these scoring systems. As it has been said in the previous chapter, security is still an important issue to solve in these wireless systems. The project has created an entire prototype of a wireless scoring system which is reliable, accurate and low cost; however it does not implement any security mechanism. This fact made the prototype suitable for a training environment but impossible to be implemented in official competitions without a proper security mechanism.

Another aspect to be improved is the implementation of the logic designed to detect the hits within the foil. This system has been designed properly thinking in a real implementation but due to the lack of hardware and probably time, the system was not tested.

Regarding the synchronisation protocol, the author thinks that it has been done a good work with it and it has been obtained a good accuracy, but maybe that accuracy could have been better with a deeper study of the variables used. The reason of not going deeper was mainly the lack of time.

In conclusion, the project has lights and shadows, but it is thought that it can be a good first step in the creation of a bigger and more ambitious product.

References

1. Master Gérard Six. Fencing History: brief history of a technique, of a science, an art, a sport and (or) a method of education [journal article online]. [Cited 2012 April 20]. Available from: http://www.fie.ch/download/en%20bref/en/Histoire%20escrime-ANG.pdf
2. Danielle Dames. An overview on the scoring of foil fencing. Helios [Internet]. 2009 May 9 [cited 2012 April 20]. Available from: http://www.helium.com/items/1443207-an-overview-on-the-scoring-of-foil-fencing
3. Marciano, Frédéric. System of hit detection and signaling for fencing. United States: Computer Masters International (Besancon, FR); 2003 Dec 30. Report No.: 6669601
4. Joseph Anthony Alvarez, Michael Victor Cristobal, Marvin Trevor Gamalinda, Anthony Malino and Joseph Miguel. Design of a Wireless Scoring System for Fencing Using RFID Technology. International Journal of Sports Science and Engineering. 2008; 2(2):79-85.
5. Alex Langseth, Dagmawi Hailemariam, Raeey Regassa, Rebika Yitna, Shakeela Bader. Wireless Fencing Scoring System [Senior Design Project Proposal]. Atlanta, Georgia: Georgia Institute of Technology; 2007.
6. STM-fencing. History of creation StM wireless fencing system and basic historical marks [Internet]. 2009 [cited 2012 April 20]. Available from: http://stm-fencing.com/content.php?pid=5&cid=10&
7. British Fencing. FIE Technical Rules [Internet]. 2010 [updated 2011 December]. Chapter 3 Field of play; p: 11-12 [cited 2012 April 20]. Available from: http://www.britishfencing.com/uploads/files/book_t_2012-feb-05.pdf
8. Aloha protocol [Internet]. [Cited 2012 April 20]. Available from: http://www.laynetworks.com/ALOHA%20PROTOCOL.htm
9. British Fencing. FIE Technical Rules [Internet]. 2010 [updated 2011 December]. Chapter 5 Fencing, Duration of the bout; p: 30-32 [cited 2012 April 20]. Available from: http://www.britishfencing.com/uploads/files/book_t_2012-feb-05.pdf
10. Texas Instruments. SimpliciTI Compliant Protocol Stack [Internet]. [Cited 2012 April 20]. Available from: http://www.ti.com/tool/simpliciti

11. IP Datagram Encapsulation [Internet]. The TCP/IP guide. [Updated 2005 September 20; cited 2012 April 20]. Available from: http://www.tcpipguide.com/free/t_IPDatagramEncapsulation.htm
12. Texas Instruments. MSP430 Wireless Development Tool [Internet]. [Cited 2012 April 20]. Available from: http://www.ti.com/tool/ez430-rf2500
13. Texas Instruments. CC2500 [Internet]. [Cited 2012 April 20]. Available from: http://www.ti.com/product/cc2500
14. Texas Instruments. MSP430F2274 [Internet]. [Cited 2012 April 20]. Available from: http://www.ti.com/product/msp430f2274
15. Texas Instruments. Code Composer Studio (CCStudio) Integrated Development Environment (IDE) v5 [Internet]. [Cited 2012 April 20]. Available from: http://www.ti.com/tool/ccstudio
16. Microsoft. Microsoft Visual Studio 2010 [Internet]. [Cited 2012 April 20]. Available from: http://www.microsoft.com/visualstudio/es-es
17. International Fencing Federation. New Rules for the convention weapons: foil and sabre [Internet]. 2004, August [cited 2012 April 20]. Available from: http://www.fie.ch/download/rules/en/semi-foil%20and%20sabre.pdf

Appendix 1 Project Overview

Appendix 2 Second Formal Review Output

Appendix 3 C Code of the Fencer's device

```c
#include "bsp.h"
#include "mrfi.h"
#include "nwk_types.h"
#include "nwk_api.h"
#include "nwk_frame.h"
#include "nwk.h"
#include "bsp_leds.h"
#include "bsp_buttons.h"

#include "app_remap_led.h"


//Prototypes
void toggleLED(uint8_t);
void startTimerA();
void resetTime();
void sendTouched();
void sendNonValid();
static uint8_t sCB(linkID_t);
void configurePorts();
static void linkTo(void);


static uint8_t state = 0;
static linkID_t sLinkID1 = 0;

//variables to measure the time
static uint8_t tics3 = 0;
static uint8_t tics2 = 0;
static uint8_t tics = 0;

static uint8_t msg_send[4];

static uint8_t enviar=0;
static addr_t lAddrAux;


//Inputs
#define TIP BIT2
#define TIP2 BIT0
#define JACKET BIT1

//Tipe of touch
#define VALID 1
#define NONVALID 2

//States
#define JOINING 1
#define WAITING 2
#define SENDING 3


#define SPIN_ABOUT_A_SECOND   NWK_DELAY(1000)
#define SPIN_ABOUT_A_QUARTER_SECOND   NWK_DELAY(250)

#define MILISEC_TIC 1                       //from 1 to 4000 ms
#define CICLES_TIC 1000*MILISEC_TIC

#define TEST_DELAY 1

/* How many times to try a Tx */
```

```c
#define MISSES_IN_A_ROW  100

void main (void)
{
  BSP_Init();
  BSP_ENABLE_INTERRUPTS();

  //configuring the address of the device
  addr_t lAddr={0x80,0x56,0x34,0x12};
  SMPL_Ioctl(IOCTL_OBJ_ADDR, IOCTL_ACT_SET, &lAddr);

  state = JOINING;

  while (SMPL_SUCCESS != SMPL_Init(sCB))
  {
    toggleLED(1);
    toggleLED(2);
    SPIN_ABOUT_A_SECOND;
  }

  /* LEDs on solid to indicate successful join. */
  if (!BSP_LED2_IS_ON()) toggleLED(2);
  if (!BSP_LED1_IS_ON()) toggleLED(1);


  /* Unconditional link to AP which is listening due to successful join. */
  linkTo();

  while (1); //Unreachable statement
}

static void linkTo()
{
  /*Preparing the end device to listen the AP too*/
  SMPL_Ioctl( IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_RXON, 0);

  /* Keep trying to link... */
  while (SMPL_SUCCESS != SMPL_Link(&sLinkID1))
  {
    toggleLED(1);
    toggleLED(2);
    SPIN_ABOUT_A_SECOND;
  }

  /* Turn off LEDs. */
  if (BSP_LED2_IS_ON()) toggleLED(2);

  if (BSP_LED1_IS_ON()) toggleLED(1);

  while (1)
  {

      configurePorts();

      for(;;){
            LPM1; //Sleep...
      }
  }
}

void sendTouched(){
```

```
      /* Building the message to send */
      msg_send[3] = tics3;
      msg_send[2] = tics2;
      msg_send[1] = tics;
      msg_send[0] = VALID;

        state=SENDING;
        enviar=1;
}

void sendNonValid(){

      /* Building the message to send */
      msg_send[3] = 0;
      msg_send[2] = 0;
      msg_send[1] = 0;
      msg_send[0] = NONVALID;

        state=SENDING;
        enviar=1;
}


void toggleLED(uint8_t which)
{
  if (1 == which)
  {
    BSP_TOGGLE_LED1();
  }
  else if (2 == which)
  {
    BSP_TOGGLE_LED2();
  }
  return;
}


static uint8_t sCB(linkID_t lid)
{
      uint8_t     *msg=NULL;
      uint8_t           len;


      if (state == JOINING){
          state = WAITING;
          return 0;
      }

      else if(state!=JOINING && lid==SMPL_LINKID_USER_UUD){
            SMPL_Receive(SMPL_LINKID_USER_UUD, msg, &len);
            if(msg[0]==0){

                  tics3=0;
                  tics2++;

                  if (tics2==50) {
                        tics2=0;
                        tics++;

                  }
```

```
				P2OUT ^=BIT3;
				startTimerA();

			}
			else if(msg[0]==1){
				tics3=0;
				tics2=0;
				tics=0;

				startTimerA();
				if (BSP_LED1_IS_ON()) BSP_TOGGLE_LED1();
			}

			return 0;
		}

		else if (state == WAITING){



			return 0;

		}

		return 0;
}

void startTimerA(){

	TACCR0 = CICLES_TIC; // Upper limit of count for TAR
	TACCTL0 = CCIE; // Enable interrupts on Compare 0
	TACTL = MC_1|ID_3|TASSEL_2|TACLR; // Set up and start Timer A

}


void configurePorts(){

	P1IE = TIP;
	P1IES = TIP;

	P2REN |= JACKET | TIP2;
	P2OUT |= JACKET | TIP2;
	P2IE |= JACKET;
	P2IES |= JACKET;

	P2DIR |=BIT3;
}

#pragma vector = TIMERA0_VECTOR
__interrupt void TIMERA0_ISR (void) // ISR for TACCR1 CCIFG and TAIFG
{

	tics3++;

	if(enviar==1){
		SMPL_Ioctl(IOCTL_OBJ_ADDR, IOCTL_ACT_GET, &lAddrAux);
		if(lAddrAux.addr[0]==0x80) {
			if(tics3%4==0){
				SMPL_Send(sLinkID1, msg_send, sizeof(msg_send));
				BSP_TOGGLE_LED1();
```

```
                        enviar=0;
                        state=WAITING;
                }
            }
            else {
                if(tics3%4==2){
                        SMPL_Send(sLinkID1, msg_send, sizeof(msg_send));
                        BSP_TOGGLE_LED1();
                        enviar=0;
                        state=WAITING;
                }
            }

        }

        P2OUT ^=BIT3;

}

#pragma vector = PORT1_VECTOR
__interrupt void PORT1_ISR (void)
{
        if(!(P1IN & TIP) && state==WAITING){

            if((P2IN & TIP2) && state==WAITING){
                sendNonValid();

            }
        }

        do {
            P1IFG = 0; // Clear any pending interrupts ...
        } while (P1IFG != 0);

}



BSP_ISR_FUNCTION( BSP_GpioPort1Isr, PORT2_VECTOR )
{


        if(P2IFG & JACKET){
        //I've been touched valid


            if(state==WAITING) {

                sendTouched();

            }

            P2IFG &= ~JACKET;
        }

        MRFI_GpioIsr();
}
```

Appendix 4 C Code of the Scoring Machine

```c
#include <string.h>
#include "bsp.h"
#include "mrfi.h"
#include "bsp_leds.h"
#include "bsp_buttons.h"
#include "nwk_types.h"
#include "nwk_api.h"
#include "nwk_frame.h"
#include "nwk.h"
#include "virtual_com_cmds.h"

#include "app_remap_led.h"


//Prototypes
void toggleLED(uint8_t);
void startTimerA();
void stopTimerA();
void startTimerB();
void stopTimerB();
void sendBeacon();
static void processMessage(linkID_t, uint8_t *, uint8_t);
void processValidPacket(linkID_t);
void processNonValidPacket(linkID_t);
void givePointTo(uint8_t);
void giveNonValid(uint8_t);
void processDoubleTouch();
void stopReceiving();
void startReceiving();

/* callback handler */
static uint8_t sCB(linkID_t);


/* reserve space for the maximum possible peer Link IDs */
static uint8_t NUM_MAX_CONNECTIONS = 2;
static uint8_t state = 0;

//tests values
static uint8_t errors = 0;
static uint8_t beacons = 0;
static uint8_t beacons2 = 0;
static uint8_t msgs = 0;

static linkID_t sLID[2] = {0,0};
static uint8_t  sNumCurrentPeers = 0;


//struct where the fencers will be stored
typedef struct{
      linkID_t link;
      uint8_t msg_rcv[4];
}Fencer;

static Fencer fencers[2];


/* work loop semaphores */
static volatile uint8_t sPeerFrameSem = 0;
static volatile uint8_t sJoinSem = 0;
```

```c
#define SPIN_ABOUT_A_QUARTER_SECOND   NWK_DELAY(250)

//States
#define JOINING 0
#define WAITING 1
#define ONE_TOUCH 2
#define ZERO_TOUCH 3
#define ONE_NON_VALID 4
#define ZERO_NON_VALID 5
#define RESET 6

//Fencers colours
#define GREEN 0
#define RED 1

#define VALID 1
#define NONVALID 2


#define MILISEC_DOUBLETOUCH 5                    //from 1 to 4000 ms
#define CICLES_DOUBLETOUCH 12*MILISEC_DOUBLETOUCH

#define MILISEC_TIC 50                     //from 1 to 4000 ms
#define CICLES_TIC 12*MILISEC_TIC


void main (void)
{
  bspIState_t intState;

  BSP_Init();
  BSP_ENABLE_INTERRUPTS();

  P2DIR |= BIT3;

  COM_Init();

  SMPL_Init(sCB);

  /* green and red LEDs on solid to indicate waiting for a Join. */
  if (!BSP_LED2_IS_ON()) toggleLED(RED);

  if (!BSP_LED1_IS_ON()) toggleLED(GREEN);


  state = JOINING;

  /* main work loop */
  while (sNumCurrentPeers<NUM_MAX_CONNECTIONS)
  {
    /* Wait for the Join semaphore to be set by the receipt of a Join frame
from a
     * device that supports an End Device.
     */
    if (sJoinSem && (sNumCurrentPeers < NUM_MAX_CONNECTIONS))
    {
      /* listen for a new connection */
      while (1)
      {
        if (SMPL_SUCCESS == SMPL_LinkListen(&sLID[sNumCurrentPeers]))
```

```
        {
          fencers[sNumCurrentPeers].link=sLID[sNumCurrentPeers];
          break;
        }

      }

      sNumCurrentPeers++;

      BSP_ENTER_CRITICAL_SECTION(intState);
      sJoinSem--;
      BSP_EXIT_CRITICAL_SECTION(intState);
    }
  }

  if (BSP_LED2_IS_ON()) toggleLED(RED);

  if (BSP_LED1_IS_ON()) toggleLED(GREEN);


  NWK_DELAY(500);

  sendBeacon();
  startTimerB();

  state = WAITING;


  while(1){



    if(state==RESET) {

      NWK_DELAY(1000);
      startReceiving();
      sendBeacon();
      startTimerB();
      state=WAITING;
    }

    /* Have we received a frame on one of the ED connections?*/

    if (sPeerFrameSem)
    {
      uint8_t     msg[4], len, i;

      /* process all frames waiting */
      for (i=0; i<sNumCurrentPeers; i++)
      {
        if (SMPL_SUCCESS == SMPL_Receive(sLID[i], msg, &len))
        {

          if(sLID[i]==fencers[GREEN].link){
                  fencers[GREEN].msg_rcv[0]=msg[0];
                  fencers[GREEN].msg_rcv[1]=msg[1];
                  fencers[GREEN].msg_rcv[2]=msg[2];
                  fencers[GREEN].msg_rcv[3]=msg[3];
          }

          else if(sLID[i]==fencers[RED].link){
```

```
                fencers[RED].msg_rcv[0]=msg[0];
                fencers[RED].msg_rcv[1]=msg[1];
                fencers[RED].msg_rcv[2]=msg[2];
                fencers[RED].msg_rcv[3]=msg[3];
        }


        processMessage(sLID[i], msg, len);

        BSP_ENTER_CRITICAL_SECTION(intState);
        sPeerFrameSem--;
        BSP_EXIT_CRITICAL_SECTION(intState);

      }
    }
  }
 }
}

void toggleLED(uint8_t which)
{
  if (0 == which) BSP_TOGGLE_LED1();

  else if (1 == which) BSP_TOGGLE_LED2();

  return;
}

/* Runs in ISR context. Reading the frame should be done in the */
/* application thread not in the ISR thread. */
static uint8_t sCB(linkID_t lid)
{

  if (state == JOINING )
  {
      sJoinSem++;
  }

  else if (state==RESET){
  }

  else
  {
    sPeerFrameSem++;
  }

  return 0;
}

static void processMessage(linkID_t lid, uint8_t *msg, uint8_t len)
{
  uint8_t type=msg[0];

  if (state==WAITING)
  {
      if(type==VALID) processValidPacket(lid);
      else if (type==NONVALID) processNonValidPacket(lid);

  }
  else if (state == ONE_TOUCH){
      stopReceiving();
```

```
    if(type==VALID) {
      if(lid==fencers[1].link){
          processDoubleTouch();
      }
    }

    else if(type==NONVALID) givePointTo(1);
  }

  else if (state == ZERO_TOUCH){

      stopReceiving();
    if(type==VALID) {
      if(lid==fencers[0].link){
          processDoubleTouch();
      }
    }

    else if(type==NONVALID) givePointTo(0);

  }

  else if(state == ONE_NON_VALID){

      stopReceiving();
      if(type==NONVALID) {
            if(lid == fencers[0].link) giveNonValid(3);
      }
    else if(type==VALID) {
        if(lid == fencers[1].link) givePointTo(0);
    }


  }
  else if(state == ZERO_NON_VALID){

      stopReceiving();
      if(type==NONVALID) {
            if(lid == fencers[1].link) giveNonValid(3);
      }
    else if(type==VALID) {
        if(lid == fencers[0].link) givePointTo(1);
    }

  }

  return;
}

void processValidPacket(linkID_t lid){
      if(lid==fencers[0].link) state=ONE_TOUCH;
      else if(lid==fencers[1].link) state=ZERO_TOUCH;
      startTimerA();
}

void processNonValidPacket(linkID_t lid){
      if(lid==fencers[0].link) state=ZERO_NON_VALID;
      else if(lid==fencers[1].link) state=ONE_NON_VALID;
      startTimerA();

}
```

```c
void processDoubleTouch(){

	uint8_t *green_time=fencers[GREEN].msg_rcv;
	uint8_t *red_time=fencers[RED].msg_rcv;


	if(green_time[1]>red_time[1]) givePointTo(RED);
	else if (green_time[1]<red_time[1]) givePointTo(GREEN);

	else{
		if(green_time[2]>red_time[2]) givePointTo(RED);
		else if (green_time[2]<red_time[2]) givePointTo(GREEN);
		else{
			//givePointTo(3);
			if(green_time[3]>red_time[3]){
				if ((green_time[3]-red_time[3])>1)givePointTo(RED);
				else givePointTo(3);
			}
			else if (green_time[3]<red_time[3]){
				if                              ((red_time[3]-
green_time[3])>1)givePointTo(GREEN);
				else givePointTo(3);
			}
			else{
				givePointTo(3);
			}
		}
	}
}


void sendBeacon(){
	uint8_t i=0;

	if(state==RESET) i=1;

	while    (SMPL_SUCCESS    !=    SMPL_Send(SMPL_LINKID_USER_UUD,    &i,
sizeof(i))){
		errors++;
	}
	beacons++;
	if(beacons==0) beacons2++;
	P2OUT ^= BIT3;
	startTimerB();
}

void givePointTo(uint8_t fencer){

	if (state==RESET) return;
	state=RESET;
	stopReceiving();


	if(fencer==0){
			TXString( "Punto1\n", 7 );
			BSP_TURN_ON_LED1();
			NWK_DELAY(500);
			BSP_TURN_OFF_LED1();

	}
```

```
	if(fencer==1){
			TXString( "Punto2\n", 7 );
			BSP_TURN_ON_LED2();
			NWK_DELAY(500);
			BSP_TURN_OFF_LED2();

	}

	if(fencer==3){
			TXString( "Punto3\n", 7 );
			BSP_TURN_ON_LED1();
			BSP_TURN_ON_LED2();
			NWK_DELAY(500);
			BSP_TURN_OFF_LED1();
			BSP_TURN_OFF_LED2();

	}


}

void giveNonValid(uint8_t fencer){

	if (state==RESET) return;
	state=RESET;
	stopReceiving();

	if(fencer==0){
			TXString("Non1\n", 5);
			BSP_TURN_ON_LED1();
			NWK_DELAY(500);
			BSP_TURN_OFF_LED1();

	}
	else if(fencer==1){
			TXString("Non2\n", 5);
			BSP_TURN_ON_LED2();
			NWK_DELAY(500);
			BSP_TURN_OFF_LED2();

	}
	else if(fencer==3){
			TXString("Non3\n", 5);
			BSP_TURN_ON_LED1();
			BSP_TURN_ON_LED2();
			NWK_DELAY(500);
			BSP_TURN_OFF_LED1();
			BSP_TURN_OFF_LED2();
	}

}

void stopTimerA(){
	TACCTL0 &= ~CCIE;
}

void startReceiving(){
	SMPL_Ioctl( IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_RXON, 0);

}
```

```
void stopReceiving(){
      SMPL_Ioctl( IOCTL_OBJ_RADIO, IOCTL_ACT_RADIO_RXIDLE, 0);
      stopTimerA();
}

void startTimerA(){

      BCSCTL3 |= LFXT1S_2;                          // LFXT1 = VLO
      TACCTL0 = CCIE;                               // TACCR0 interrupt enabled
      TACCR0 = CICLES_DOUBLETOUCH;                               // ~ 1 sec
      TACTL = TASSEL_1 | MC_1 | TACLR;                      // ACLK, upmode

}


void startTimerB(){

      BCSCTL3 |= LFXT1S_2;                          // LFXT1 = VLO
      TBCCTL0 = CCIE;                               // TACCR0 interrupt enabled
      TBCCR0 = CICLES_TIC;                            // ~ 1 sec
      TBCTL = TBSSEL_1 | MC_1 | TBCLR;                      // ACLK, upmode

}




#pragma vector = TIMERA0_VECTOR
__interrupt void TIMERA0_ISR (void) // ISR for TACCR1 CCIFG and TAIFG
{
      if(state==ONE_TOUCH){
            givePointTo(1);

      }
      else if(state==ZERO_TOUCH){
            givePointTo(0);
      }
      else if(state==ONE_NON_VALID){
            giveNonValid(1);
      }
      else if(state==ZERO_NON_VALID){
            giveNonValid(0);
      }




}

#pragma vector = TIMERB0_VECTOR
__interrupt void TIMERB0_ISR (void) // ISR for TACCR1 CCIFG and TAIFG
{
      sendBeacon();

}

BSP_ISR_FUNCTION( BSP_GpioPort1Isr, PORT2_VECTOR )
{

  MRFI_GpioIsr();
}
```

Appendix 5 C++ Code of the Graphic Interface

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication3
{
    public partial class Form1 : Form
    {
        int points1 = 0;
        int points2 = 0;
        public String a;
        public String info = null;


        public Form1()
        {
            InitializeComponent();
            label1.Text = points1.ToString();
            label2.Text = points2.ToString() ;
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            //label1.Text = "sseaasda";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //String a;

            if (!serialPort1.IsOpen) serialPort1.Open();

            textBox1.AppendText("Start!\n");

            //cont++;
            //label1.Text = cont.ToString();
        }


        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.AppendText("Stop!\n");
            if (serialPort1.IsOpen) serialPort1.Close();
        }

        private          void          serialPort1_DataReceived(object          sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {


            if (!serialPort1.IsOpen) serialPort1.Open();
            a = serialPort1.ReadLine();

            if (a.Equals("Punto2"))
            {
                points2++;
                info = "Touch done by 2";
```

```
        }
        else if (a.Equals("Punto1"))
        {
            points1++;
            info = "Touch done by 1";
        }
        else if (a.Equals("Non1")) info = "Non Valid touch done by 1";
        else if (a.Equals("Non2")) info = "Non Valid touch done by 2";
        else if (a.Equals("Non3")) info = "Non Valid touch done by both";

        this.Invoke(new EventHandler(processData));


    }

    public void processData(object sender, EventArgs e)
    {
        //textBox1.AppendText(a+"\n");
        label1.Text = points1.ToString();
        label2.Text = points2.ToString();
        textBox1.AppendText(info+"\n");
    }

    private void label1_Click(object sender, EventArgs e)
    {

    }

    private void button3_Click(object sender, EventArgs e)
    {
        points1 = 0;
        points2 = 0;
        label1.Text = points1.ToString();
        label2.Text = points2.ToString();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        points1++;
        label1.Text = points1.ToString();
        label2.Text = points2.ToString();
    }

    private void button5_Click(object sender, EventArgs e)
    {
        points1--;
        label1.Text = points1.ToString();
        label2.Text = points2.ToString();
    }

    private void button7_Click(object sender, EventArgs e)
    {
        points2++;
        label1.Text = points1.ToString();
        label2.Text = points2.ToString();
    }

    private void button6_Click(object sender, EventArgs e)
    {
        points2--;
        label1.Text = points1.ToString();
        label2.Text = points2.ToString();
```

```
        }

        private void pictureBox2_Click(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

    }
}
```