# Sistema de transmisión de mensajes con cifrado *lightweight* como proyecto en asignatura de Máster

Carlos J. Jiménez, Carmen Baena, Manuel Valencia
Dpto. de Tecnología Electrónica, Universidad de Sevilla /
Instituto de Microelectrónica de Sevilla (CSIC)
Sevilla, España
cjesus@us.es

Juan M. Fernández, Alejandro Moreno
Universidad de Sevilla
Sevilla, España

***Abstract*—Las asignaturas de Máster deben ser muy prácticas, aplicando los conocimientos a problemas concretos. En este trabajo se presenta como trabajo de una asignatura de Máster la realización de un diseño de comunicaciones seguras utilizando un bus SPI. El sistema utiliza un cifrador de flujo para cifrar y descifrar la información y permite el envío de mensajes de longitud aleatoria. Además mediante el uso de CRCs se comprueba que el mensaje no ha sido alterado.**

***Keywords— Comunicaciones seguras; bus serie; información cifrada; docencia en másters.***

## I. INTRODUCCIÓN

Este trabajo se enmarca dentro de la asignatura "Diseño de Sistemas Digitales sobre FPGAs" del Master de Instalaciones y Productos de la Escuela Politécnica Superior, de la Universidad de Sevilla [1]. Este Máster se ofrece, principalmente a los alumnos de las cinco titulaciones de la Escuela Politécnica Superior (Ingeniería en Electrónica Industrial, Ingeniería Eléctrica, Ingeniería Mecánica, Ingeniería Química Industrial e Ingeniería en Diseño Industrial), pero también está abierto a alumnos de otras titulaciones.

El programa de este Máster tiene como objetivo fundamental formar profesionales capaces de integrar conceptos y técnicas científicas para plantear y resolver problemas ingenieriles, basándose en el uso del método científico, el razonamiento analítico, sintético y deductivo. Además se pretende formar a profesionales que deseen extender sus competencias y experiencias hacia nuevas temáticas interdisciplinares mediante una capacitación experta en interpretación, valoración y selección de alternativas ingenieriles.

Como objetivos específicos, el alumno, al terminar sus estudios deberá ser capaz de:

- Diseñar y gestionar el ciclo de vida de productos, instalaciones industriales y sistemas electrónicos con criterios de sostenibilidad.

- Modelar y simular productos, instalaciones industriales y sistemas electrónicos.

- Generar, desarrollar e implementar soluciones innovadoras de productos, instalaciones industriales y sistemas electrónicos.

- Optimizar instalaciones y productos, mejorando su eficiencia.

- Desarrollar proyectos emprendedores en el sector de producto e instalaciones derivados de nuevas ideas de negocio e innovaciones.

El Máster se estructura en un bloque común, a cursar por todos los estudiantes y tres bloques de intensificación entre los que el alumno deberá escoger uno de ellos. Los bloques son todos de 24 créditos.

Bloque Común. En las materias del bloque se pretende que el alumno obtenga una formación avanzada para el modelado y la simulación soportado con TIC´s. Incluye contenidos obligatorios y comunes de la titulación del Máster. Este bloque tiene seis materias con contenidos instrumentales y básicos para introducir a los estudiantes en el ámbito del Diseño y Desarrollo de Productos e Instalaciones Industriales. En este bloque se presta una especial atención a los fundamentos científicos y las tecnologías de procesos.

Bloque de Diseño y Desarrollo de Productos. A cursar en su totalidad por los alumnos que deseen obtener esta especificación. En él se estudian los aspectos estéticos del diseño industrial, se introduce al alumno en los métodos de diseño y fabricación, y se le hace una presentación de los materiales utilizados en el diseño y desarrollo de productos, principalmente.

Bloque de Instalaciones Industriales. A cursar en su totalidad por los alumnos que deseen obtener esta especificación. Se analizan las instalaciones electrónicas, de automatización, térmicas y energéticas, las hidráulicas, las química-ambientales y las instalaciones eléctricas.

Bloque de Diseño y Aplicación de Sistemas Electrónicos Industriales. A cursar en su totalidad por los alumnos que deseen obtener esta especificación. Se proporciona al estudiante una formación especializada en el campo de los equipos y sistemas electrónicos que se aplican en la industria

tanto desde la perspectiva de la renovación y adaptación de las industrias a las nuevas tecnologías como también, al diseño, aplicación, integración y desarrollo de productos electrónicos de automatización y control industrial.

La formación del Máster Universitario en Instalaciones y Diseño de Productos se completa con la realización de Prácticas en Empresas (3 créditos) y el Trabajo Fin de Máster (9 créditos).

La asignatura en la que se enmarca este trabajo pertenece al bloque de especialización de "Diseño y Aplicación de Sistemas Electrónicos Industriales". Este bloque cuenta, junto con la asignatura Diseño de Sistemas Digitales sobre FPGAs, con las asignaturas Diseño de Sistemas Inteligentes para el Procesado de Datos, Diseño y Gestión de Redes Industriales, Instrumentación con Redes de Sensores, Robótica, Inteligencia y Percepción y Computadores Empotrados para Sistemas de Tiempo Real. Todas las asignaturas son de cuatro créditos ECTS.

Los objetivos fundamentales de la asignatura que nos ocupa son el aprendizaje de metodologías de diseño usando el lenguaje VHDL y dispositivos FPGA, y viene a cubrir la carencia en diseño electrónico usando lenguajes de descripción de hardware que existía en los antiguos títulos de Ingeniería Técnica y que sigue existiendo en los actuales títulos de Grado en Ingeniería Electrónica Industrial. La asignatura, con una orientación eminentemente práctica, consta de clases teóricas, laboratorios y un trabajo final. En los siguientes apartados de esta comunicación se presentan con detalle los objetivos y contenidos de la asignatura Diseño de Sistemas Digitales sobre FPGAs, una breve introducción sobre comunicaciones cifradas empleando cifradores de flujo y una descripción detallada del trabajo realizado y de los resultados obtenidos.

## II. Diseño de sistemas digitales sobre FPGAs

El objetivo principal de esta asignatura consiste en presentar las metodologías de diseño de sistemas electrónicos digitales complejos sobre dispositivos programables, empleando lenguajes de descripción de hardware y herramientas de síntesis a nivel RT. Esta metodología no sólo comprende aspectos relacionados con el diseño en todos sus niveles (desde la concepción abstracta hasta su implementación en dispositivos e introducción en un sistema), sino que también cubre todos los aspectos relativos a la verificación y test de los diseños (desde el nivel funcional hasta el test de las implementaciones).

En su docencia se usan aplicaciones que ilustren de forma práctica el diseño a todos sus niveles. Entre las aplicaciones tienen especial relevancia aquellas susceptibles de ser utilizadas en entornos industriales, como el control, las comunicaciones seguras, etc. Pero también se pueden incluir otras aplicaciones que por su novedad y uso puedan considerarse interesantes.

Las implementaciones de los diseños se realizan en tecnologías FPGA, que permiten explorar la metodología propuesta y también realizar un prototipado rápido de sistemas digitales complejos.

La asignatura se imparte en diez sesiones de dos horas de duración cada una. En cada una de estas sesiones hay una parte dedicada a explicación teórica (que no debe llegar a la hora) y otra parte dedicada a la realización de prácticas por parte de los alumnos. La última de las sesiones se dedica a la presentación y defensa, por parte de los alumnos, del trabajo realizado al amparo de esta asignatura.

Los contenidos de la asignatura se estructuran en cuatro grandes bloques temáticos:

Bloque 1: Introducción al diseño de sistemas digitales, tecnologías de los dispositivos, metodología de diseño, herramientas de CAD y tecnologías FPGAs.

Bloque 2: Lenguaje de descripción de hardware VHDL, uso de HDLs en el diseño digital, construcciones básicas del lenguaje VHDL y simulación funcional.

Bloque 3: El lenguaje VHDL para síntesis, principales limitaciones de síntesis, descripciones combinacionales, descripciones secuenciales síncronas y consideraciones temporales.

Bloque 4: Realización de un diseño de complejidad media. Presentación del trabajo a realizar: problema a resolver, objetivos a conseguir, medios y resultados.

## III. Comunicaciones cifradas utilizando cifradores de flujo

Actualmente, las comunicaciones digitales requieren una adecuada seguridad en la transmisión de datos. La Criptografía es la ciencia que protege la información transmitida frente a personal no autorizado y proporciona técnicas, mecanismos y herramientas para ofrecer en redes abiertas una comunicación privada segura. Con toda seguridad, toda información que fluya por una red deberá ser cifrada y descifrada, por lo que para asegurar unas transferencias seguras, será necesario incorporar diseños que contengan funciones criptográficas en los distintos dispositivos que participen en la transmisión de datos.

Los aspectos más importantes para proteger la información que se transmite son: la confidencialidad (información secreta entre personal autorizado), autenticidad (la información procede de alguien del grupo autorizado) e integridad (la información es protegida frente a ataques maliciosos).

Los algoritmos criptográficos se clasifican en algoritmos simétricos, basados en clave privada, y algoritmos asimétricos basados en un par de claves, una pública y la otra privada. En general, en ambos mecanismos, se utilizan los mismos algoritmos para el cifrado y descifrado de la información. Los circuitos que implementan algoritmos de clave privada consumen muchos menos recursos que los circuitos que implementan algoritmos de clave pública. En este trabajo nos centramos en criptografía basada en clave privada. Hay dos tipos de cifradores simétricos: los cifradores de bloques y los cifradores de flujo. Los primeros cifran bloques de datos de longitud fija, mientras que los de flujo lo hacen sobre datos de longitud variable. En este trabajo utilizaremos cifradores de flujo, cuyas implementaciones hardware se basan en la utilización de registros de desplazamiento con realimentación no lineal. Aunque ambos tipos de cifradores presentan el

problema de distribuir la clave entre emisor y receptor, los de cifradores de flujo tienen la ventaja de que son rápidos y poseen una arquitectura simple que facilita la transmisión de un gran volumen de datos, por lo que su uso es muy adecuado en sistemas de baja complejidad y en los de fuertes restricciones respecto al consumo de potencia.

En este contexto, la Unión Europea lanzó un proyecto conocido como eSTREAM [2], [3] para seleccionar propuestas de cifradores de flujo tanto en software como en hardware. Entre estos últimos, se encuentra el cifrador Trivium [4], uno de los tres finalistas, que ha sido el empleado en este trabajo.

### *A. Especificaciones del cifrador Trivium*

El cifrador Trivium es un circuito síncrono que, a partir de una clave *K* de 80 bits y de un vector de inicialización *IV* también de 80 bits, es capaz de generar una secuencia de bits de forma pseudo-aleatoria (*Keystream)* de hasta $2^{64}$ bits. El mensaje cifrado se obtiene mediante la operación XOR entre la secuencia *Keystream* y el mensaje sin cifrar (Figura 1). Su arquitectura consiste en un registro de desplazamiento cíclico de 288 bits acompañado de lógica combinacional (AND, OR, XOR) para proporcionar las realimentaciones. Este registro de desplazamiento se divide a su vez en tres registros de desplazamiento de longitudes diferentes: 93 bits, 83 bits y 111 bits respectivamente. La figura 2 ilustra la estructura del Trivium, con sus tres registros de desplazamiento así como la forma de generar tanto los bits de realimentación como el bit de salida.

En el funcionamiento del Trivium, inicialmente existe una fase de inicialización del estado interno del Trivium. Para ello se carga la clave secreta *K* y el vector *IV* en el Trivium y opera 4 veces 288 ciclos de reloj para la actualización del estado interno del cifrador antes de empezar a generar una cadena de bits *keystream* válida.

### *B. Protocolo de comunicaciones*

El objetivo de este trabajo es comunicar entre sí dos sistemas independientes y autónomos, uno que actúa como emisor y otro que actúa como receptor, pero utilizando una transmisión de información cifrada.

Los requisitos que tiene que cumplir el protocolo de comunicaciones han de ser los siguientes: debe ser sencillo, puesto que únicamente se desea enviar mensajes de un maestro a un esclavo. Además ha de permitir la transmisión de mensajes de longitud arbitraria de forma serie. Por otro lado, como la aplicación está pensada para ser usada en distancias cortas, el mecanismo de comunicación puede ser síncrono. Es decir, el emisor genera la señal de reloj usada por el receptor.

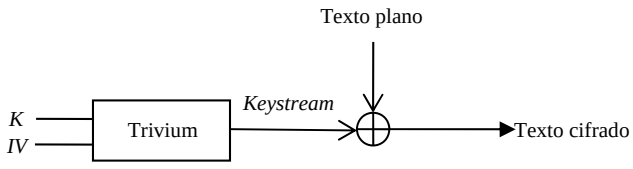


Fig. 1: Esquema funcionamiento de un cifrador de flujo.

Entre los mecanismos de comunicación que cumplen con estos requisitos se ha seleccionado un protocolo SPI [5].

Se trata de un protocolo que realiza la transmisión de información serie entre un dispositivo *Master* y uno *Slave* en ambos sentidos. La sincronización y transmisión de los datos se desarrolla por medio de 4 señales:

SCK: reloj común para ambas partes y generado por el *Master*.

MOSI: línea de datos de *Master* a *Slave* (*Master Out Slave In*).

MISO: línea de datos de *Slave* a *Master* (*Master In Slave Out*).

SS: (Selección de *Slave*) Se usa para indicar al *Slave* que se va a iniciar la comunicación.

Este tipo de comunicación también puede desarrollarse con varios *Slaves*. En ese caso, el módulo *Master* tendrá una línea SS para cada uno de los ellos y será una de estas líneas la que se active según con qué dispositivo *Slave* quiera comunicarse.

La información es transmitida sincronizada con el reloj de forma que en cada ciclo se transmite un bit. Existe posibilidad de enviar los bits sincronizados con el flanco de subida o bajada del reloj (polaridad) y también elegir un flanco activo para capturar los bits por parte del *Slave* (fase de la transmisión). La forma de sincronización en la que se transmite queda configurada mediante una serie de bits que el *Master* envía y el *Slave* reconoce.

En general, se trata de un protocolo de comunicación que requiere un hardware muy simple, una única señal de reloj y transmite mensaje de longitud arbitraria siendo por tanto muy apropiado para comunicaciones de corta distancia.

Como se explica en el siguiente apartado, este protocolo ha sido adaptado a los requerimientos concretos del sistema que aquí se presenta.

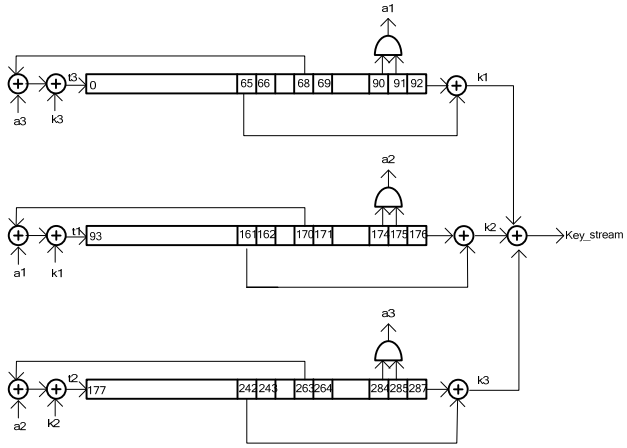


Fig. 2: Representación esquemática Trivium.

## IV. DESCRIPCIÓN DEL TRABAJO REALIZADO

El trabajo realizado ha consistido en la transmisión serie de mensajes cifrados mediante el cifrador de flujo Trivium utilizando un protocolo SPI. El sistema cuenta con dos bloques: un emisor y un receptor. Ambos bloques deben realizar primeramente la carga de la clave y del vector de inicialización del Trivium, y hacer funcionar a éste durante 1152 ciclos de reloj antes de poder estar en condiciones de enviar y recibir mensajes. Una vez realizada esta inicialización, el funcionamiento es como sigue: el emisor tiene un mensaje para enviar, comprueba si el receptor está disponible para recibir el mensaje, en cuyo caso lo irá cifrando y enviando. El receptor por su parte al recibir el mensaje lo irá descifrando y almacenando. Para hacer más atractivo para los alumnos este trabajo y también para que la verificación sea más sencilla, ambos bloques emplean visualizadores 7 segmentos para mostrar los mensajes. Al ser síncrono el protocolo de comunicaciones, el emisor genera también la señal de reloj del receptor. Con objeto de asegurar que el mensaje no ha sido alterado durante la transmisión, tras el envío del mensaje se enviará, también cifrado, el código CRC correspondiente al mensaje cifrado. El sistema se ha montado experimentalmente utilizando placas de desarrollo de Xilinx. A continuación se detallan las funciones a realizar por cada uno de los bloques.

El bloque emisor tiene que realizar las siguientes funciones:

- Generar el mensaje a enviar.
- Visualizarlo en el visualizador 7 segmentos.
- Cifrarlo con el cifrador de flujo Trivium.
- Generar un código para la detección de errores.
- Enviar mensaje y CRC mediante un protocolo de comunicaciones serie síncrono.

El bloque receptor tiene que realizar las siguientes funciones:

- Recibir y descifrar el mensaje.
- Almacenarlo en unos registros y visualizarlo en el visualizador 7 segmentos.
- Comprobar, mediante la igualdad de los códigos CRC generado y recibido, que el mensaje no ha sido alterado.

Los objetivos de aprendizaje que se pretenden alcanzar con este trabajo son varios:

- Trabajar con un protocolo de comunicaciones, que aunque sencillo, permita la transmisión de información de forma serie, síncrona y con longitud variable.
- Trabajar con información cifrada y en concreto con la forma de funcionar de cifradores de flujo.
- Utilizar códigos CRC para la detección de errores.
- Aprender la metodología de diseño y verificación mediante un caso práctico.
- Aprender el manejo de las herramientas de Xilinx.
- Montar y comprobar de forma experimental el funcionamiento del circuito.

### *A. Descripción de los bloques del emisor*

El emisor cuenta con los siguientes bloques:

a. Generador aleatorio de mensajes

Los mensajes a enviar son generados de forma aleatoria y mostrados en cuatro visualizadores 7-segmentos. La restricción a cuatro dígitos se impone porque la placa que se va a utilizar para su implementación tiene incorporado cuatro visualizadores. Sin embargo, los mensajes podrán tener una longitud variable de forma que su visualización pueda realizarse con uno, dos, tres o los cuatro dígitos. Eso significa una longitud binaria de 4, 8, 12 ó 16 bits.

La generación aleatoria es tanto para el contenido como para la longitud del mensaje. El mensaje generado será almacenado en un registro de datos (de 16 bits) y su longitud en otro registro de 2 bits. La generación de un nuevo mensaje se hará cuando se active una señal (se implementará mediante un pulsador de la placa).

b. Visualizador 7 segmentos

Un segundo bloque tomará los datos almacenados en el registro de datos y los mostrará en el visualizador 7 segmentos. Este bloque hará uso del dato almacenado en el registro de longitud para encender sólo los visualizadores correspondientes al mensaje a transmitir. La forma como se haga esta visualización depende de la placa a utilizar. En nuestro caso se ha realizado para la placa Basys2 de Digilent.

c. Cifrador Trivium

Para cifrar los mensajes se ha utilizado el cifrador de flujo Trivium. Este cifrador requiere para su funcionamiento:

- Una clave y de un vector de inicialización, ambos de 80 bits. Tanto la clave como el vector de inicialización están fijos en el código VHDL.
- La clave y el vector de inicialización se cargarán en el momento de recibir la alimentación, pero hay que esperar que transcurran 1152 ciclos de reloj antes de poder generar datos válidos.

d. Generación de CRC

Un bloque se encargará de la generación de códigos para la detección de errores. Estos códigos son importantes tanto para la detección de errores en la transmisión, como la detección de posibles ataques que pueda sufrir el sistema mediante cambios malintencionados de los mensajes. Como los mensajes que se transmiten son de una longitud muy corta, se ha escogido un CRC de tan sólo 4 bits. Sin embargo este mecanismo es fácilmente ampliable a más bits si se quisiera realizar otra implementación con mensajes más largos.

e. Control del envío de mensajes

El emisor contará con un bloque de control del envío de mensajes. Para ello tendrá que gobernar el funcionamiento del Trivium y generar las señales del protocolo de comunicaciones. Tomará los datos a enviar del registro de datos, y su longitud del registro de longitud.

Además este bloque realizará el control de la señal de reloj del Trivium, de forma que se avance un ciclo de reloj por cada dato que se quiera enviar. Si no se quieren enviar datos, la señal de reloj deberá permanecer deshabilitada.

Otras funcionalidades:

- Al recibir un pulso en una entrada de inicio, procederá a cifrar y enviar los datos (mensaje y CRC).
- Tendrá una señal de *reset* asíncrono.
- El mensaje a transmitir será la XOR de cada bit del mensaje plano (sin cifrar) con el bit generado por el Trivium.

*B. Descripción de bloques del receptor*

El receptor cuenta con los siguientes bloques:

a. Visualizador 7-segmentos

Este bloque coincide con el bloque del emisor. Tomará los datos almacenados en el registro de datos y los visualizará en los visualizadores 7-segmentos. Este bloque hará uso del dato almacenado en el registro de longitud para visualizar sólo los datos correspondientes al mensaje recibido.

b. Cifrador Trivium

Este bloque se usará para el descifrado de los datos, pero tendrá el mismo comportamiento que en el emisor. Al igual que en el emisor, el control del funcionamiento del Trivium se hará desde el bloque de control.

c. Comprobación CRC

Antes de dar por bueno el mensaje recibido deberá comprobar el CRC recibido. Si éste es correcto entonces se dará por bueno.

d. Control de recepción de mensajes

El receptor contará con un bloque de control de la recepción de mensajes. Para ello tendrá que gobernar el funcionamiento del Trivium e interpretar las señales del protocolo de comunicaciones.

La señal de reloj será la que provenga del emisor y deberá controlar la señal de reloj del Trivium para que sólo se genere cuando haya que descifrar un dato.

Otras funcionalidades:

- Deberá proporcionar una señal indicando que está disponible para recibir datos.
- Tendrá una señal de *reset* asíncrono.
- Cuando reciba los datos los irá descifrando y almacenando en un registro de datos. Además, al terminar almacenará la longitud del mensaje recibido en un registro de longitud.

*C. Protocolo de comunicaciones*

Para la comunicación de los mensajes se ha utilizado un protocolo serie síncrono que fuera lo más sencillo posible. Para ello se tomó como base un protocolo SPI, pero simplificándolo. Como la comunicación se pretendía que fuera únicamente de un emisor a un receptor, se suprimió la opción de transmitir mensajes a varios receptores y también se suprimió la posibilidad de transmitir datos de receptor a emisor. Sin embargo, como el receptor tiene que realizar la inicialización del Trivium, se decidió añadir una señal que permitiera decirle al emisor si el receptor estaba preparado para recibir mensajes. Tampoco se han implementado opciones de configuración de flancos de reloj para la generación y recepción de los mensajes. Los mensajes se generan con el flanco de subida del reloj y se muestrean en el receptor con el flanco de bajada.

Con todo esto, para el protocolo de comunicaciones el emisor genera una señal de reloj, una señal que indica el inicio de la transmisión del mensaje y una señal para los datos y el receptor genera una señal indicando si está listo para recibir un mensaje. La tabla 1 esquematiza y muestra los nombres utilizados por estas señales.

TABLA 1: DESCRIPCIÓN DE LAS SEÑALES ENTRE EMISOR Y RECEPTOR

| **Emisor** | | **Receptor** | |
|---|---|---|---|
| clk | Señal de reloj del sistema | | |
| send | Señal que activa el inicio de los datos. Cuando esté a '1' significa que se están enviando datos. | ready | Señal que indica que está preparado para recibir datos. Cuando esté a '1' significa que puede recibir datos. |
| data | Señal que contiene el mensaje enviado | | |

.

La figura 3 muestra un diagrama temporal con los momentos en los que se activan las distintas señales en la transmisión de un mensaje. El procedimiento para el envío de un mensaje es el siguiente: el emisor comprueba si la señal *ready* está en '1'. Si es así activa la señal *send* y comienza el envío de datos. Cuando la se ha enviado el último dato vuelve a poner *send* a '0'. Mientras está recibiendo datos, el receptor pone la señal *ready* a '0', indicando así que está recibiendo datos. Una vez que termina la recepción la señal *ready* vuelve a estar a '1'.
:

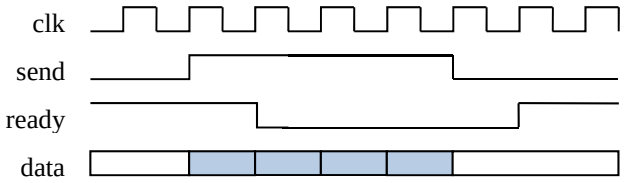


Fig. 3: Diagrama temporal de una transmisión.

*D. Desarrollo del trabajo*

Este trabajo fue desarrollado por dos alumnos, uno de ellos se encargó del diseño del emisor y otro del diseño del receptor, aunque compartieron el diseño de aquellos bloques comunes a ambas partes. Tras la entrega de las especificaciones por parte del profesor, se tuvieron tres sesiones de seguimiento, en las que se solventaron las dificultades que los alumnos se fueron encontrando, se cerraron pequeños detalles de especificaciones que no habían quedado claros en el documento original y se planificaron las pruebas de verificación a realizar sobre los

diseños, tanto para cada diseño de forma individual como para los dos diseños de forma conjunta.

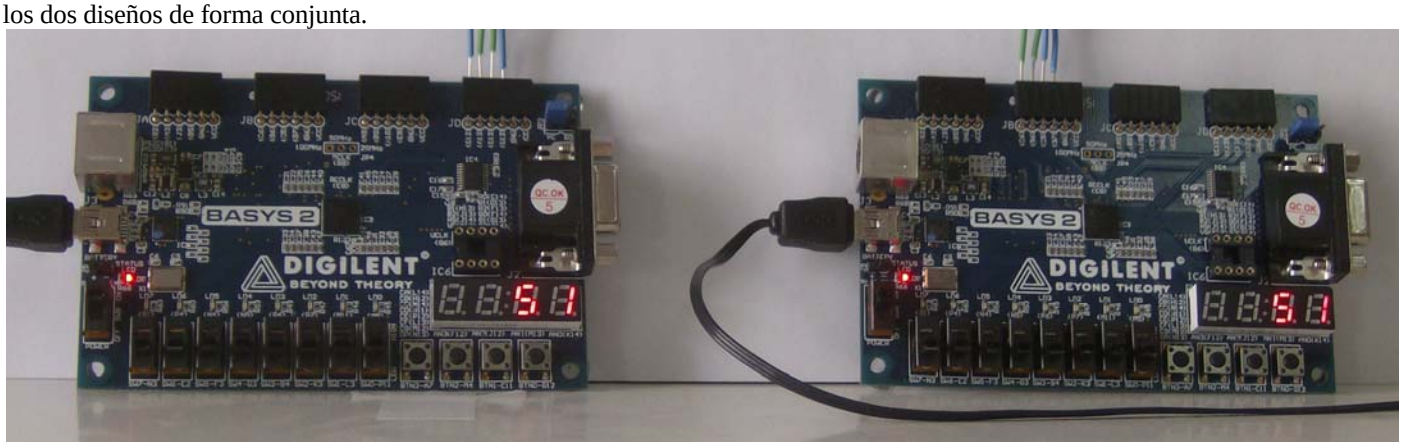


Fig. 4: Fotografía del montaje experimental.

Finalmente, en una última sesión, se pasó a la implementación de los diseños en dos placas Basys2 de Digilent y la prueba final de interconectarlas para el envío y recepción de diferentes mensajes. Se verificó que el sistema funcionó correctamente, comprobando en los visualizadores siete segmentos de cada una de las placas el mensaje enviado y el mensaje recibido. La utilización de este mecanismo, aunque muy limitante en el número de bits del mensaje a transmitir, posibilitó realizar la comprobación experimental del funcionamiento de forma muy visual y sin necesidad de utilizar equipos adicionales de laboratorio (en concreto de analizadores lógicos). La figura 4 muestra una fotografía del montaje. La placa de la izquierda se corresponde con el emisor y la de la derecha con el receptor. Los visualizadores de ambas placas muestran el mismo valor, lo que significa la correspondencia entre mensaje enviado y mensaje recibido.


## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto de investigación CITIES (TEC-2010-16870) del Ministerio de Ciencia e Innovación.


## V. CONCLUSIONES

En esta comunicación se ha presentado una experiencia de trabajo en una asignatura de Máster. En primer lugar se ha hecho referencia al Máster en el que se inserta la asignatura y se han descrito los objetivos y la metodología seguida por dicha asignatura. El trabajo que han realizado los alumnos les ha permitido, no sólo enfrentarse a una aplicación cercana a un caso real y a unas herramientas comerciales, sino que también les ha permitido adquirir una serie de destrezas adicionales: se han enfrentado a un documento de especificaciones, teniendo que cerrar aquellas que no quedaron bien especificadas en el documento original. También han tenido una experiencia de trabajo en equipo, pues aunque cada alumno se encargó de una parte del diseño, tuvieron que estar coordinados en todo momento para conseguir que los datos fuesen transmitidos de forma correcta.

La experiencia fue valorada muy positivamente por el profesor y por los alumnos y se alcanzaron todos los objetivos propuestos, pues el sistema funcionó a la primera.

## REFERENCIAS


[1] http://www.us.es/estudios/master/master_M066

[2] eSTREAM: ECRYPT Stream Cipher Project, http://www.ecrypt.eu.org/stream/

[3] New Stream Cipher Designs. The eSTREAM Finalists. Matthew Robshaw Olivier Billet (Eds.). Springer 2008.

[4] C. De Canniere y B. Preneel, “Trivium, A Stream Cipher Construction Inspired by Block Cipher Design Principles”, eSTREAM, ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream/papersdir/2006/021.pdf

[5] Motorola Inc., “SPI Block Guide V03.06,” February 2003.