ANTEPROYECTO DE TRABAJO FIN DE GRADO

(Modelo TFG-5)

Convocatoria ordinaria ☑	Convocatoria extraordinaria ☐

A CUMPLIMENTAR POR EL ESTUDIANTE			
Nombre y apellidos:	Mario Rubio Asensio	**DNI:**	
Dirección:		**CP:**	
Ciudad:		**Provincia:**	
E-mail:		**Teléfono:**	

Título del TFG: Analizador USB			
Modalidad		**Orientación**	
General ☐	Especifico ☑	Tareas de desarrollo ☐	Ejercicio de la profesión libre ☑

DATOS DEL DIRECTOR
Nombre y apellidos: Francisco Moya Fernández

Palabras clave: USB, bus, FPGA, analizador, Verilog, WireShark.

BREVE DESCRIPCIÓN DEL TFG

Antecedentes y objetivos:

Desde el momento de su lanzamiento en la última década del siglo pasado, el bus de comunicación USB *(Universal Serial Bus)* se ha ido proclamando como el bus comercial más conocido y usado.

Una de las gran ventajas que trae consigo la implementación de este bus, a parte de la sencillez general de uso, es la gran versatilidad que puede proporcionar, por eso, no es de extrañar que hayan surgido una considerable cantidad de aplicaciones , tales como:

- Dispositivos de interfaz humana (ratones, teclados, etc..).
- Dispositivos de almacenamiento masivo *"USB-MSC"* (*pendrives*, USB a SATA, etc..).
- Herramientas de adquisición de datos y comunicación (adaptadores de USB a Serie o USB a WiFi, etc..).

Debido a todo lo anterior, sería de gran interés y utilidad disponer de un analizador, que de forma pasiva pueda captar la trama de comunicación que se transmite por el bus, enviarla a un equipo, y posteriormente analizarla para su uso en depuración o seguridad.

Resultados esperados:

El resultado de este trabajo se pretende dividir en dos grupos totalmente diferenciados, en el primero se tratarán elementos a nivel hardware y comunicación entre dispositivos, mientras que en el segundo se trabajará con el tratado y análisis de los resultados del primer grupo.
Cabe destacar que durante la totalidad de este trabajo prevalecerá el uso de software libre.

1. En primer lugar, se espera poder capturar y transmitir a un equipo tramas provenientes de un bus USB, para ello:
   - Utilizando un FPGA, concretamente el modelo ICE40HX1K [1] de la empresa Lattice, se generará un sintetizado a partir del lenguaje de descripción de Hardware Verilog [2][3][4] que contenga toda la lógica para la captación de tramas, independientemente del tipo (*Low-Speed, Full-Speed, etc...)* [5][9][12].
   - Se implantará una librería escrita en lenguaje C/C++ que permita comunicar la plataforma de captación anterior con un equipo (como puede ser una Raspberry Pi [7]).

2. A partir de una trama USB obtenida de cualquier método, tanto por el método anteriormente descrito, como a partir de medios externos, se pretende poder trabajar sobre ella pudiendo integrar los siguientes aspectos.
   - Capacidad de almacenar la trama en archivos de capturas, tal como *pcap* [8].
   - Creación de un disector funcional para el analizador de paquetes WireShark [6].
   - Plataforma de análisis de dispositivos de interfaz humana (HID), tales como Keylogger o seguidor de puntero de ratón [11].

Temporización:

Tal como se ha comentado en los *"Resultados esperados"*, este proyecto se puede separar en dos grupos. Ambos, a su vez, pueden dividirse en varios apartados.

1. Captura y transmisión.
   - ***Diseño de método de transmisión de la trama a un equipo.***
     Para poder llevar un control adecuado, se necesita en primer lugar poder implementar una transmisión básica de información entre el *FPGA* y el equipo de análisis. Esta método de comunicación se prevee que se implemente en **dos semanas**, pudiendo añadir pequeñas funcionalidades en el transcurso del siguiente apartado según se necesite.

   - ***Implementación básica de un método de sincronización y captura del bus USB utilizando un FPGA.***
     Este apartado se puede considerar como el de mayor importancia en este grupo, por eso, se plantea un periodo de realización de **un mes y medio.**

   - ***Librería que permita obtener y utilizar la trama transmitida según el método anterior.***
     Al depender este apartado de los otros dos anteriores, se pretende desarrollar a la par que el resto, añadiendo funcionalidades a media que se necesiten. Antes de la finalización de este grupo, se pretende añadir **una semana** extra para depurar, mejorar y limpiar el código implementado en la librería.

   En total se pretende trabajar en este grupo un total de **dos meses y una semana.**

2. Procesado de la trama.
   - ***Utilizando la librería del grupo anterior, ampliarla para poder guardar la trama en un archivo de fácil utilización, como puede ser PCAP.***
     Existe multitud de recursos y librería útiles [13] con los que partir, por tanto, no se plantea un extenso periodo para el desarrollo de este apartado, pudiendo ser este de **dos semanas**.

   - ***Creación de librería de análisis básico de la trama.***
     Es en este apartado donde se debe implementar todo el análisis de la trama USB, por tanto, es en el que se debería emplear la mayor cantidad de tiempo, siendo esta de **un mes y medio**.

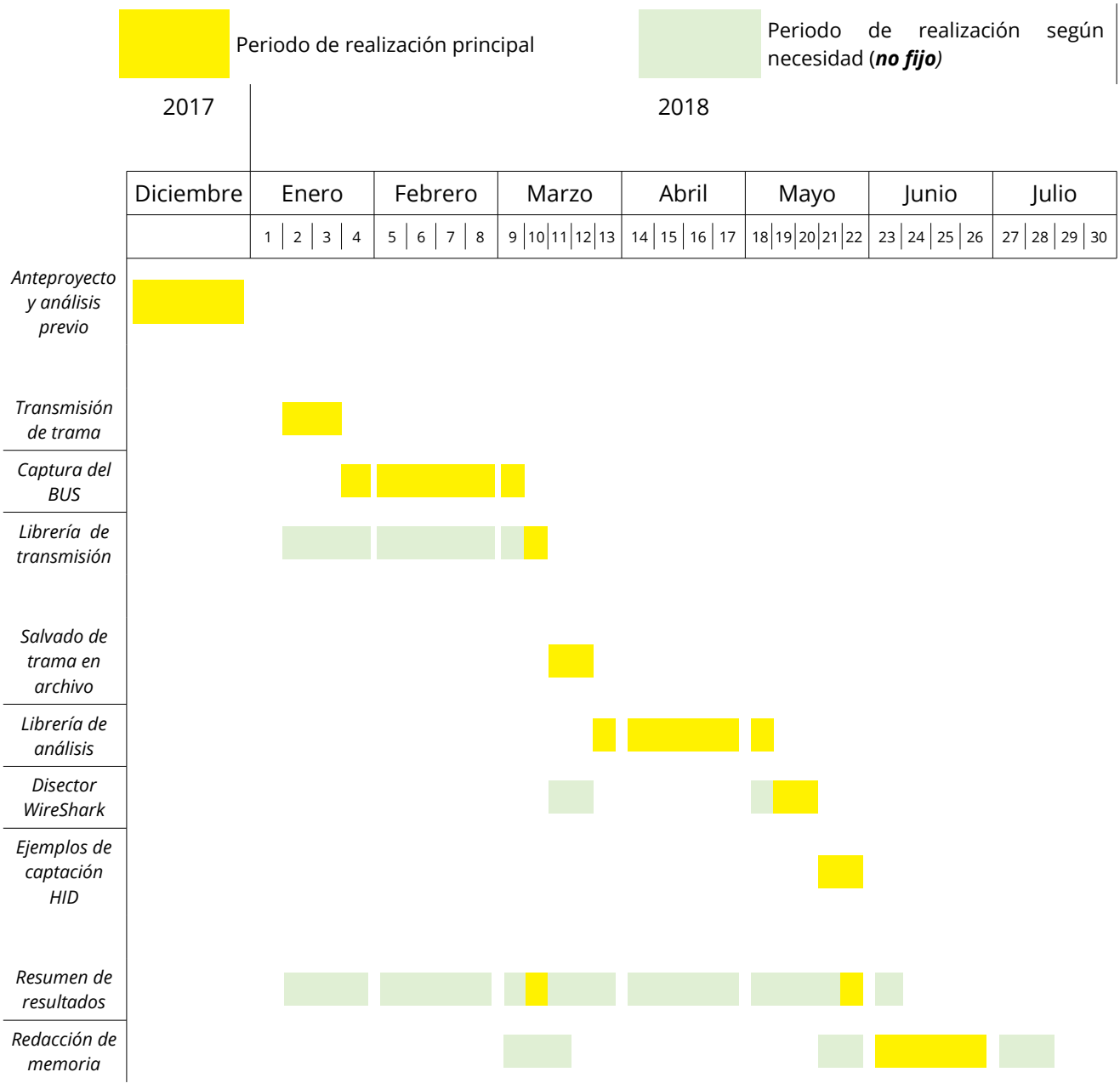
- ***Utilizando la guía de desarrollo propuesta por WireShark [6], crear un disector que puede procesar en cierta media la trama dentro dicho programa.***
  De la misma manera que en le primer apartado, existe una multitud de recursos con los que partir, por tanto, se propone su realización en **dos semanas**.

- ***Utilizando la librería creada o el disector anterior, poder analizar transmisiones de dispositivos de interfaz humana (HID [10]) para poder capturar pulsaciones de teclado o movimientos de ratón [11].***
  En este apartado se expondrán varios ejemplos de uso del trabajo desarrollado, lo que supondrá una menor cantidad de tiempo a utilizar, siendo esta de **dos semanas**.

En total se pretende trabajar en este grupo un total de **tres meses.**

A parte del tiempo anteriormente utilizado, también se prevee utilizar **un mes** para la redacción del documento final, así como todos los recursos y ayudas necesarios para ello.

En la siguiente tabla se muestra una distribución temporal más detallada



Bibliografía:

1. iCE40 LP/HX Family Data Sheet - Lattice Semiconductor – Marzo 2017 (Versión 3.3) - http://www.latticesemi.com/view_document?document_id=49312

2. Lattice ICE Technology Library - Lattice Semiconductor - Marzo 2015 (Versión 2.9) - http://www.latticesemi.com/~/media/LatticeSemi/Documents/TechnicalBriefs/SBTICETechnologyLibrary201504.pdf

3. Tutorial de FPGA utilizando lenguaje descriptivo Verilog - Juan Gonzalez-Gomez (Obijuan) – Noviembre 2015 - https://github.com/Obijuan/open-fpga-verilog-tutorial/wiki

4. Verilog HDL Quick Reference Guide - Stuart Sutherland – 2001 - http://sutherland-hdl.com/pdfs/verilog_2001_ref_guide.pdf

5. USB made simple - MQP Electronics Ltd – 2008 - http://www.usbmadesimple.co.uk/

6. Adding a basic dissector - Ulf Lamping, Luis E. Garcia Ontanon, Graham Bloice – diciembre 2014 (revisión 1.1) - https://www.wireshark.org/docs/wsdg_html_chunked/ChDissectAdd.html

7. Introducción a Raspberry Pi - Francisco Moya Fernández – Enero 2017 - https://franciscomoya.gitbooks.io/taller-de-raspberry-pi/content/es/index.html

8. PCAP next generation file format specification - M. Tuexen, Ed., Muenster Univ. of Appl. Sciences, F. Risso, Politecnico di Torino, J. Bongertz, Airbus DS CyberSecurity, G. Combs, Wireshark, G. Harris – 2017 - https://github.com/pcapng/pcapng

9. USB Complete (2nd Edition) – Jan Axelson – 2004

10. Device Class Definition for Human Interface Devices (HID) V1.11- USB Implementers Forum, Inc. - Junio 2001 - http://www.usb.org/developers/hidpage/HID1_11.pdf

11. USB-based attacks - Nir Nissim,Ran Yahalom,Yuval Elovici - 2017 - https://doi.org/10.1016/j.cose.2017.08.002

12. USB in a nutshell - Craig Peacock – 2010 - http://www.beyondlogic.org/usbnutshell/usb1.shtml

13. Awesome pcaptools - caesar0301 - 2015 - https://github.com/caesar0301/awesome-pcaptools

Vº Bº y firma del DIRECTOR *Firma del ESTUDIANTE*

Toledo, a ___ de _____ de 20___