



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

**PROGRAMA DE MAESTRIA Y DOCTORADO EN
INGENIERIA**

**CONSTRUCCIÓN DE UN AUTOCORRELADOR
POR ABSORCIÓN DE DOS FOTONES PARA
LA CARACTERIZACIÓN DE DISPOSITIVOS
ÓPTICOS ULTRARAPIDOS**

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

MAESTRO EN INGENIERIA

INGENIERÍA ELÉCTRICA - INSTRUMENTACIÓN

P R E S E N T A :

Ing. Pablo Castro Marín

TUTOR:

Dr. Jesús Garduño Mejía

Centro de Ciencias Aplicadas y Desarrollo Tecnológico
(CCADET)

México, D. F. Diciembre 2012



**CENTRO DE CIENCIAS APLICADAS Y
DESARROLLO TECNOLÓGICO**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Roberto Ortega Martínez
Secretario: Dra. Martha Rosete Aguilar
Vocal: Dr. Jesús Garduño Mejía
1er. Suplente: Dr. Alejandro Farah Simón
2do. Suplente: Dr. Antonio A. Rodríguez Rosales

Lugar o lugares donde se realizó la
Tesis: México, Distrito Federal

TUTOR DE TESIS:

Dr. Jesús Garduño Mejía

FIRMA

Dedicatoria

A mis Padres, por todo el apoyo incondicional, y la fuerza que me han enseñado a tener ante cualquier adversidad.

A mi familia y amigos por las porras, apoyo y regaños que me han dado para seguir avanzando.

Agradecimientos

Se le agradece el apoyo del Técnico Jorge Hernández Cervantes del taller mecánico del CCADET por la elaboración de los componentes opto-mecánicos.

Se agradece el financiamiento y apoyo en becas a:

DGAPA-UNAM. PAPIIT proyectos:

IN113809,

IB101212,

IACOD-I1100611,

IN104112,

BID-UNAM 98-B3-C-DGP-L0034-1077,

CONACyT proyecto:

CB 131746

Contenido

Resumen.....	1
Abstract.....	1
Introducción.....	2
Motivación y Objetivos	2
Capitulo 1. “Descripción del dispositivo y de sus componentes”	3
Descripción de Autocorrelador.....	3
Descripción del “Interferómetro de <i>Michelson</i> ”	4
Descripción del “Fenómeno <i>TPA (two photon absorption)</i> ”	5
Capitulo 2. “Propuesta de desarrollo tecnológico”	9
Capitulo 3. Protocolos de comunicación e Interfaces (USB)	10
Antecedentes.....	10
Conectores y Cables	11
Estándares de voltajes y corrientes del USB 2.0	14
Velocidades	15
Protocolos.....	15
Como se conforman las tramas usadas en el USB2.0	16
Tipos de paquetes (<i>Packet ID</i>)	17
Tipos de <i>pipes</i>	19
Tipos de <i>endpoints</i>	20
Descriptores.....	20
Desarrollo de la interfaz y protocolo de comunicación <i>USB2.0</i> para Autocorrelador por Absorción de Dos Fotones (TPA) portátil	21
Capitulo 4. Sistema Opto-Electro-Mecánico.....	23

4.1.- Bloque I “Línea de Retardo”	23
a) Descripción	24
b) Diagrama	26
c) Componentes Utilizados	27
4.2.-Bloque II “Sensor y Tratamiento de Señal” – TPA –	27
a) Descripción	28
b) Diagrama	29
c) Componentes Utilizados	30
4.3.- Bloque III “Interferómetro <i>Michelson</i> ”	30
a) Descripción	31
b) Prototipo.....	31
c) Componentes Utilizados	32
4.4- Bloque IV “Muestreo de Señales” –DAQ–	32
a) Descripción	32
b) Diagrama	33
c) Componentes Utilizados	34
4.5.- Bloque V “Software Interfaz Hombre-Máquina”	34
a) <i>Firmware DAQ (Pic18F4550)</i>	35
b) <i>Firmware VCO (Pic18F4550)</i>	36
4.6.- Sistema para la caracterización de dispositivos Ópticos ultrarrápidos	36
4.7.- Interfaz y comunicación del dispositivo con la computadora	37
4.8.- Descripción del <i>software</i>	38
4.9.- Calibración.....	42
Capitulo 5. Resultados	45

Resultados experimentales.....	45
Capitulo 6. Conclusiones.....	50
Conclusiones	50
Trabajo futuro	51
Apéndice	A
Apéndice A. Generación de pulsos ultracortos y mode-locking.....	A
A) Descripción de “Pulso Ultracorto”	A
B) Mode-locking.....	B
Apéndice B. Filtros frecuenciales	D
Apéndice C. PWM (Pulse-width modulation) [7].....	E
Apéndice D. CSV (comma-separated values)	F
Apéndice E. Acoplamientos.....	G
Apéndice F. Análisis monetario de los materiales usados	H
Apéndice G. Desarrollo matemático de la correlación.....	J
Anexos	K
<i>Anexo A. Firmware</i>	K
(a) DAQ.....	K
(b) VCO.....	L
Anexo B. Drivers	M
(a)DAQ.....	M
Anexo C. Software	P
(a) Variables usadas	P
(b) Obtención de datos (conexión con USB).....	Q
(c) Llenado de las matrices	R

(d) Tamaño del pulso	T
(e) Trigger	T
(f) Encendido y apagado del Background worker.....	U
Anexo D. Contribuciones	V
Índice de Figuras.....	X
Índices de ecuaciones.....	¡Error! Marcador no definido.
Capítulos.....	¡Error! Marcador no definido.
Apéndices	¡Error! Marcador no definido.
Índice de Tablas	Z
Bibliografía	AA
Hojas de especificaciones	AA

Resumen

En este trabajo se presenta el diseño y construcción de un autocorrelador de intensidad de segundo orden tipo *tpa* (*two-photon-absorption*), con etapa de adquisición de datos, interface para *windows* y control de componentes móviles integrados, en un tamaño y costo accesibles para la caracterización temporal de pulsos ultracortos de femtosegundos.

Los elementos que componen el instrumento son: un autocorrelador colineal tipo michelson, un fotodiodo operando bajo el principio de absorción no lineal, una tarjeta *daq* (*data acquisition*) y una línea de retardo física basados en una conexión *usb 2.0*.

Se busca que el diseño final sea un instrumento en el cual el manejo y calibración sea sencilla y amigable para el usuario.

Abstract

1

This thesis presents the design and construction of a second-order intensity autocorrelator *tpa* (*two-photon-absorption*), with data acquisition stage, interface for windows and mobile components integrated into a control, accessible size and cost for the temporal characterization of ultrashort pulses in the femtosecond time regime.

The elements that compose the instrument are: a collinear autocorrelator michelson type, a photodiode operating under the principle of non-linear absorption, a *daq* (*data acquisition*) and a physical delay line based on a *usb 2.0* connection.

The final design is an instrument in which the management and calibration is user friendly.

Introducción

“An experiment is a question which science poses to Nature, and a measurement is the recording of Nature's answer.” — Max Planck, Scientific Autobiography and Other Papers, 1949.

Los pulsos ultracortos se le han dado diferentes aplicaciones, en la física, química, medicina, telecomunicaciones, investigación de materiales, entre otros. Esta rama de la óptica ha crecido en los últimos años, debido a que estos eventos intensos y breves, logran facilitar la generación de efectos ópticos no lineales.

En el Capítulo 1 se realiza una descripción del dispositivo y en el Capítulo 2 se muestra la propuesta que se realizó. En el Capítulo 3 se explica el protocolo de comunicaciones usado (*USB 2.0*). En el Capítulo 4 se muestra el diseño, desarrollo e implemento del instrumento, el hardware, las capacidades del *software*, y algunos aspectos que fueron necesarios tomar en cuenta, para tener una medición veraz. Por la parte del Capítulo 5 y 6 se muestran algunas pruebas experimentales que se realizaron, las conclusiones y el trabajo futuro que se puede realizar con este dispositivo.

2

Motivación y Objetivos

Este trabajo está enfocado a la construcción de un dispositivo que permita la caracterización de pulsos ultracortos. Un instrumento que permita al usuario una fácil, sencilla y precisa visualización de los mismos, con una sencilla forma de alineación, un costo accesible para construirse.

Lo que me impulso a generar este instrumento fue el deseo de conocer más a fondo los instrumentos ópticos, al igual que diseñar e implementar un instrumento útil para la investigación.

Capítulo 1. “Descripción del dispositivo y de sus componentes”

Debido a que la electrónica actual no logra realizar medidas más allá del orden de picosegundos, y la necesidad de medir los pulsos ultracortos (Apéndice A, (a)), se han utilizado diferentes tipos de técnicas, como lo es la autocorrelación de segundo orden, que representa ser el método más sencillo y práctico para la caracterización de dispositivos ópticos, el cual se decidió utilizar para este proyecto.

Descripción de Autocorrelador

Para generar una señal de autocorrelación necesitamos primero saber que es una autocorrelación. Una correlación es una de las formas de cuantificar el grado de interdependencia o similitud entre dos señales. En el caso específico de una autocorrelación, es el grado de interdependencia de la señal, con la señal misma [1]. En pocas palabras:

$$R_f(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} F^*(t - \tau) F(t) dt \quad \text{Ecuación 1.1}$$

donde T es el periodo de la señal y τ es un retardo.

Ahora, la forma física de realizar una autocorrelación, es utilizando un arreglo de un *interferómetro de Michelson*. Para el retardo τ se emplea una línea de retardo en uno de los brazos del interferómetro. Y para la integral se recurre a un elemento no lineal, el cual genera una autocorrelación de segundo orden (la cual se observa en la Ecuación 1.7)

Si se tiene un espectro con un ancho de banda Δf , el mínimo ancho temporal, en FWHM¹, que se puede tener de un pulso (sin aberraciones, chirp² o GVD³) [2], suponiendo un pulso gaussiano, es según la relación:

¹ Full Width Half Maximum

² También llamado gradiente frecuencial o dispersión espacial.

$$\Delta f \approx \frac{0.441}{\Delta \tau} \quad \text{Ecuación 1.2}$$

A diferencia de la autocorrelación de primer orden, también llamada autocorrelación de campo [3], Ecuación 1.1, de la cual obtenemos información equivalente a la que proporciona un espectrómetro, la de segundo orden, mostrada en la Ecuación 1.3, nos ofrece información más útil como por ejemplo la presencia de chirp en el pulso (solo la presencia, no es posible cuantificarlo) o bien la estimación del ancho del pulso por medio de la autocorrelación de intensidad a través del efecto no lineal. Una limitante de esta técnica es la falta de información para extraer la fase de una dimensión (“*One-Dimensional Phase Retrieval*”), para esto se utilizan otras técnicas, como el FROG y SPIDER [3].

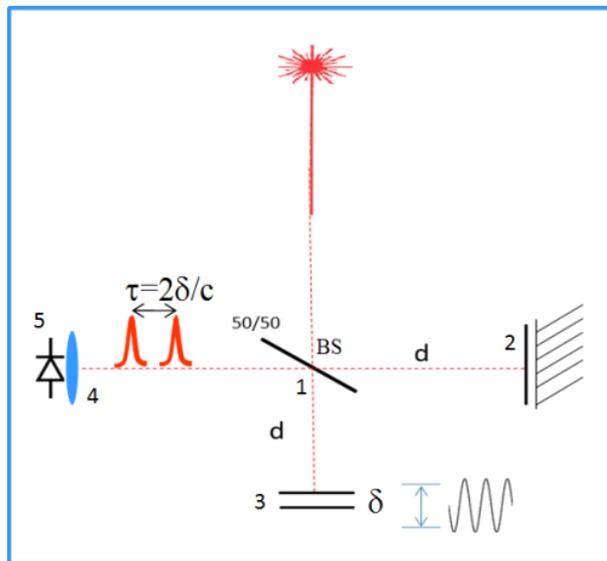


Figura 1.1 Autocorrelador de segundo orden (TPA)

(1) Divisor de haz; (2) Brazo fijo; (3) Línea de Retardo variable; (4) Lente de Enfoque; (5) Fotodiodo

Descripción del “Interferómetro de *Michelson*”

El interferómetro de Michelson (Figura 1.1) es un arreglo óptico, el cual consiste en dividir el haz de luz en dos, y, en uno de los “brazos” tener un espejo

³ group velocity dispersion

fijo, el cual forma la función sin retraso $F(t)$, en el otro brazo se colocará un espejo móvil, el cual genera la función con retardo $F(t \pm \tau)$. En este caso, el arreglo que se busca es un arreglo colineal, lo que significa, es que en la división del haz de luz, regresara por el mismo camino óptico original, esto con el fin de generar interferencia, que luego será medida por un sensor (en este caso un fotodiodo).

Descripción del “Fenómeno *TPA (two photon absorption)*”

El fenómeno de absorción de dos fotones (TPA por sus siglas en inglés), es un proceso de 3^{er} orden (debido a que se utiliza el 3^{er} orden de susceptibilidad dieléctrica $\chi^{(3)}$) [4]. Lo que pasa en este fenómeno, es que la energía de dos fotones logra excitar un electrón en la banda de valencia del material incidente y hace que salte a la banda de conducción (Figura 1.2), generando una fotocorriente correspondiente a la energía de una longitud de onda de la mitad de la original.

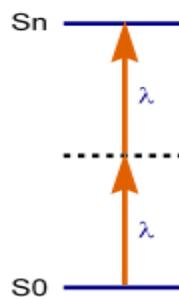


Figura 1.2 Absorción de dos fotones, S0 banda de valencia y Sn banda de conducción

Y se comporta de la siguiente manera:

$$S^{(2)} \approx A \int_{-\infty}^{\infty} [\alpha I + \beta I^2] dt; \quad \text{Ecuación 1.3}$$

Donde $S^{(2)}$ es la función de autocorrelación de segundo orden, A es una constante, α representa el coeficiente de absorción lineal y β el coeficiente de absorción no lineal, y la I representa la intensidad de la señal [2].

En este caso en particular se escogió un sensor semiconductor en el cual, el coeficiente de absorción no lineal fuera significativamente mayor al coeficiente lineal, ello para poder aproximar nuestra Ecuación 1.3 a la Ecuación 1.4 [5]

$$S^{(2)} \approx A\beta \int_{-\infty}^{\infty} [I^2] dt \quad \text{Ecuación 1.4}$$

y teniendo en cuenta que (donde E_0 representa el campo eléctrico de la señal)

$$I = |E_0(t) + E_0(t + \tau)|^2 \quad \text{Ecuación 1.5}$$

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} |E_0^2(t) + E_0^2(t + \tau) + 2E_0^2(t)E_0^2(t + \tau)|^2 dt \quad \text{Ecuación 1.6}$$

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} \{(a) + (b) + (c) + (d)\} dt. [3] \quad \text{Ecuación 1.7}$$

(Desarrollo del paso entre las ecuación 1.4 y 1.7 está en el en el Apéndice G)

Cada elemento de esta función tiene un significado:

- Autocorrelación

$$4E_0^2(t)E_0^2(t + \tau) \quad (a)$$

- Interferograma modificado de $E_0(t)$ (ω representa la frecuencia generada por la relación entre la línea de retraso y la señal interferométrica)

$$4[E_0^3(t)E_0(t + \tau) + E_0(t)E_0^3(t + \tau)] \cos(\omega\tau) \quad (b)$$

- Interferograma del segundo armónico de $E_0(t)$

$$2E_0^2(t)E_0^2(t + \tau) \cos(2\omega\tau) \quad (c)$$

- Constante

$$E_0^4(t) + E_0^4(t + \tau) \quad (d)$$

Realizando un análisis, y considerando un traslape cero ($\tau \rightarrow \infty$), los términos cruzados dejan de existir dejándonos

$$S(\infty) = 2A\beta \int_{-\infty}^{\infty} [I(t)^2] dt \quad \text{Ecuación 1.8}$$

Y este lo comparamos cuando el traslape es máximo ($\tau \rightarrow 0$),

$$S(0) = 16A\beta \int_{-\infty}^{\infty} [I(t)^2] dt \quad \text{Ecuación 1.9}$$

Lo cual nos deja una relación de contraste de

$$\frac{S(0)}{S(\infty)} = \frac{16A\beta \int_{-\infty}^{\infty} [I(t)^2] dt}{2A\beta \int_{-\infty}^{\infty} [I(t)^2] dt} = \frac{8}{1} \quad \text{Ecuación 1.10}$$

Llamándose “autocorrelación de segundo orden”.

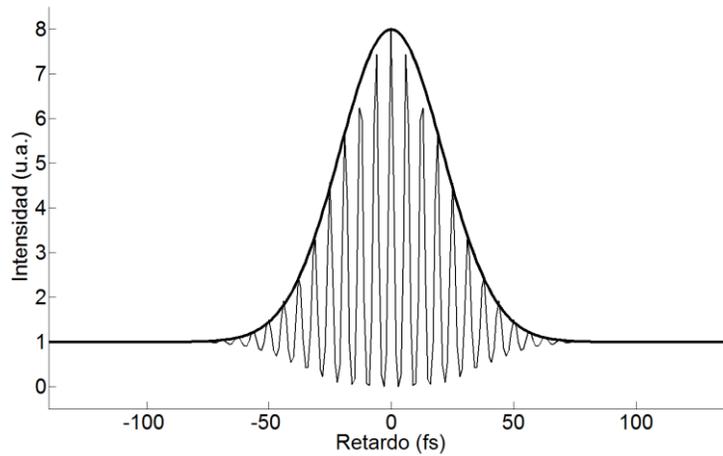


Figura 1.3 Autocorrelación de segundo orden, franjas interferométricas y su comparación con la autocorrelación de intensidad.

7

Ahora, si el sistema no tiene la capacidad de responder a las franjas interferométricas, debido a que el sistema de adquisición (sensor y/o transductor a impulsos eléctricos) es lento, conforme a los términos (b) y (c) de la Ecuación 1.7, se promediarían a cero y la relación de contraste cambiaría significativamente a

$$\frac{S(0)}{S(\infty)} = \frac{6A\beta \int_{-\infty}^{\infty} [I(t)^2] dt}{2A\beta \int_{-\infty}^{\infty} [I(t)^2] dt} = \frac{3}{1} \quad \text{Ecuación 1.11}$$

Conocida como “autocorrelación de intensidad”.

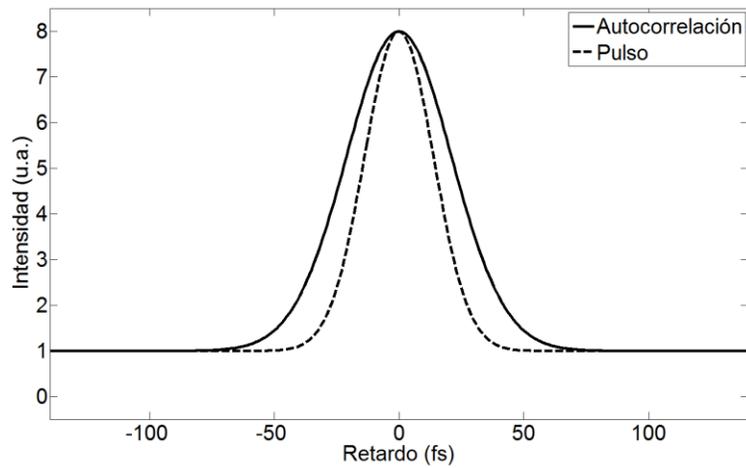


Figura 1.4 Autocorrelación de intensidad y su comparación con el pulso generado por el cual se obtiene su autocorrelación.

En la Figura 1.4 se muestra la comparación entre el pulso gaussiano que genera la autocorrelación y la misma autocorrelación, y el factor de relación que existe entre ellos es de $\sqrt{2}$.

Capítulo 2. “Propuesta de desarrollo tecnológico”

En el presente proyecto de tesis se plantea el desarrollo de un instrumento modular de fácil manejo, conectividad a través de una computadora sin periféricos adicionales, de fácil alineación, y reparación o remplazo de componentes y compatibilidad, es decir, sin la necesidad de tener que cambiar todo el dispositivo para aplicar mejoras.

A continuación, se analizarán todos y cada uno de los componentes de este dispositivo, los cuales dividiremos en bloques.

1. Bloque I “Línea de Retardo”
2. Bloque II “Sensor y Tratamiento de Señal” – *TPA* –
3. Bloque III “Interferómetro *Michelson*”
4. Bloque IV “Muestreo de Señales” –*DAQ*–
5. Bloque V “*Software* Interfaz Hombre-Máquina”

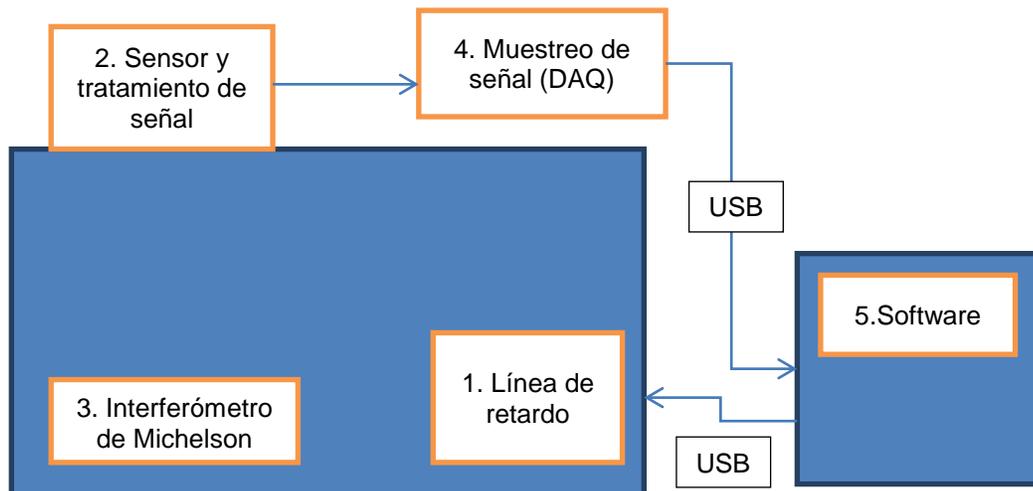


Figura 2.1 Diagrama por bloques del instrumento

Capítulo 3. Protocolos de comunicación e Interfaces (USB)

Antecedentes

Las características fundamentales y deseables para cualquier instrumento hoy en día son: su compatibilidad, facilidad de conectividad y eficiencia de comunicación con cualquier computadora o dispositivo portátil personal (tableta, teléfono, etc.). De esta forma la correcta elección de los puertos o interfaces de comunicación representan un aspecto fundamental en el diseño del instrumento.

En 1995, las compañías *Compaq*, *Digital Equipment Corp.*, *IBM PC Co.*, *Intel*, *Microsoft*, *NEC*, y *Northern Telecom* optaron por unirse y generar una interfaz de uso general, a la cual denominaron *Universal Serial Bus – USB* -, presentándola al mundo un año después (*USB 1.0* el 15 de enero de 1996).

Esta nueva interfaz corregía algunos errores que tenían las interfaces, que le precedían (puerto serial, paralelo, PS/2, etc.), al igual que mejoraba la velocidad de transferencia de datos (1.5Mbps para *USB 1.0* y 12Mbps para *USB 1.1*), un cable estándar genérico y capacidad de manejar hasta 127 dispositivos (simultáneos) por puerto *USB*, tecnología *Plug-and-Play* y *Hot-Plug* que le permite conectar el periférico y usarse sin necesidad de reiniciar la computadora. La primera computadora que uso un *USB* fue la *iMac G3*, usando la interfaz *USB* para conectar un teclado y un *mouse*.

Años después (27 de abril del 2000), se decidió publicar una revisión del estándar, generando así el *USB 2.0*, con el cual aumentaron la velocidad (480Mbps) y el rendimiento de su original, incorporan una negociación, antes no utilizada, que hace que sea compatible con su predecesor y permite al dispositivo usar la velocidad común más alta entre los dispositivos. Con lo anterior, permitieron que no hubiera cambios en los conectores.

Siguiendo en la línea de tiempo, el 17 de Noviembre 2008 nace el estándar de *USB 3.0*, protocolo que conserva las características y ventajas que ofrece su predecesor (*USB legacy*), siendo útil con los mismos conectores y permitiendo

usar la velocidad común más alta. Una de las principales mejoras que tiene con respecto al *USB 2.0* es la velocidad de transferencia, que es aproximadamente 10 veces mayor (*5.0Gbps*).

Conectores y Cables

Existen varios conectores *USB*, la diferencia que se encuentra básicamente es la configuración física de los conectores.

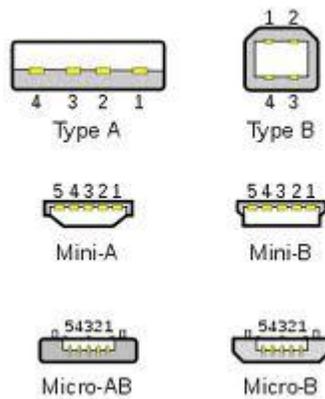


Figura 3.1 USB 1.0, 1.1 y 2.0

En caso de la Figura 3.1, se utiliza la Tabla 3-1 y la Tabla 3-2.

Pin	Nombre	Color del cable
1	VCC	Rojo
2	D-	Blanco
3	D+	Verde
4	GND	Negro

Tabla 3-1: *USB A y USB B*

Pin	Nombre	Cable
1	VCC	Rojo
2	D-	Blanco
3	D+	Verde
4	ID	-
5	GND	Negro

Tabla 3-2: Mini-A, Mini-B, Micro-AB y Micro-B

El Pin 4 (ID), a veces no se conecta, se conecta a tierra (GND) o se utiliza como indicador de conexión.

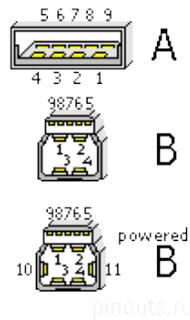


Figura 3.2 USB 3.0

Pin	Nombre	
1	Vcc	
2	D-	
3	D+	
4	GND	
5	StdA_SSRX-	<i>SuperSpeed receiver</i>
6	StdA_SSRX+	<i>SuperSpeed receiver</i>
7	GND_DRAIN	
8	StdA_SSTX-	<i>SuperSpeed trasmitter</i>
9	StdA_SSTX+	<i>SuperSpeed trasmitter</i>

Tabla 3-3: USB3.0 A

Pin	Nombre	
1	Vcc	
2	D-	
3	D+	
4	GND	
8	StdA_SSRX-	<i>SuperSpeed receiver</i>
9	StdA_SSRX+	<i>SuperSpeed receiver</i>
7	GND_DRAIN	
5	StdA_SSTX-	<i>SuperSpeed trasmitter</i>
6	StdA_SSTX+	<i>SuperSpeed trasmitter</i>
10	DPWR	
11	DGND	

Tabla 3-4: *USB3.0 B y Bpowered*

Un compromiso importante que se debe cumplir y respetar para la correcta operación, eficiencia y velocidad de transferencia de datos en los dispositivos USB son las características de los cables a emplear.

Los Cables que se deben de usar, en caso del *USB 2.0*, son:

*Cable Shielded*⁴:

-Datos: *28 AWG Twisted* Alimentación: *28 AWG - 20 AWG non-Twisted*

*Cable non-Shielded*⁵:

-Datos: *28 AWG non-Twisted* Alimentación: *28 AWG – 20 WG non-Twisted*

⁴ Cable utilizado para altas velocidades (*high-speed*)

⁵ Cable utilizado para bajas velocidades (*full-speed*)

AWG	Diámetro (mm)	Longitud máxima
28	0.321	0.81m
26	0.405	1.31m
24	0.511	2.08m
22	0.644	3.33m
20	0.812	5.00m

Tabla 3-5: Características según el American wire gauge

Resistencia del aislante a 20°C: mínimo 100MΩ/Km

Damping factor impedance: 90Ω/Km ±7%

Estándares de voltajes y corrientes del USB 2.0

El voltaje suministrado por el *host* esta entre 4.35V y 5.25V. La corriente suministrada máxima en funciones es de 100mA y en dispositivos suspendidos es de 0.5mA. La corriente máxima soportada por los *hubs* es de 500mA por puerto, después de estos 500mA se auto-apaga el puerto, por lo que si un usuario sobrecarga un puerto (recordemos que puede sostener máximo 127 dispositivos por puerto), se deberá desconectar alguno de estos elementos y conectarse a un puerto que lo soporte, o usar un puerto con fuente de poder independiente. (Nota. Esta especificación está siendo cambiada debido a las opciones de **On-The-Go** y **Battery Charging Specification**, que son dos nuevos modos de alimentación en USB).

Se utiliza una modulación *FSK (Frequency-shift keying)*⁶, tipo de modulación digital, donde el “1” es representado por una frecuencia, y el “0” es representado con otra) para el envío de datos (una portadora de 23.2Mhz y una desviación de 500Khz), y codificación llamada *NRZI (Non-Return-to-Zero Inverted)*, el cual cambia cada vez que hay un “1” lógico, y con un “0” lógico no hay transición.

⁶ Tipo de modulación digital en la cual se selecciona una frecuencia diferente para cada “símbolo”

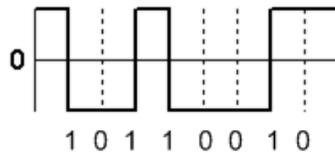


Figura 3.3 Gráfica de un ejemplo del NRZI

Velocidades

Las velocidades varían según la versión de *USB* que se use, pero entre mayor sea la versión, se pueden usar velocidades más bajas, dependiendo del cable y la función (se le llama función al dispositivo *USB*) usada.

Las velocidades de transferencia son:

1.5Mbps \pm 15000ppm (192KB/s) *Low-Speed (USB 1.0)*

12Mbps \pm 2,500ppm (1.5MB/s) *Full-Speed (USB 1.1)*

480Mbps \pm 500ppm (60MB/s) *High-Speed (USB 2.0)*

5.0Gbps (600MB/s) *Ultrahigh-Speed (USB 3.0)*

Hay que recordar que estas velocidades no toman en cuenta todo el protocolo que se tiene, por lo que no son velocidades de transferencia de datos función-host, host-función.

La forma en que se diferencian las velocidades (*Low-Speed* y *Full-Speed*) es por medio de una resistencia de 1.5K Ω , para *Full-Speed*, se conecta la resistencia entre D+ y 3.3V, y para *Low-Speed*, se conecta entre D- y 3.3V

Protocolos

La comunicación entre la función y el *software*, está dividido en capas, que se comunican lógicamente y van encapsulando el mensaje, generando segmentos, paquetes, tramas y bits, protocolizando la comunicación del *USB*, y el diseñador solo debe preocuparse por las capas altas, ya que las capas bajas son controladas por las configuraciones y por los circuitos integrados.

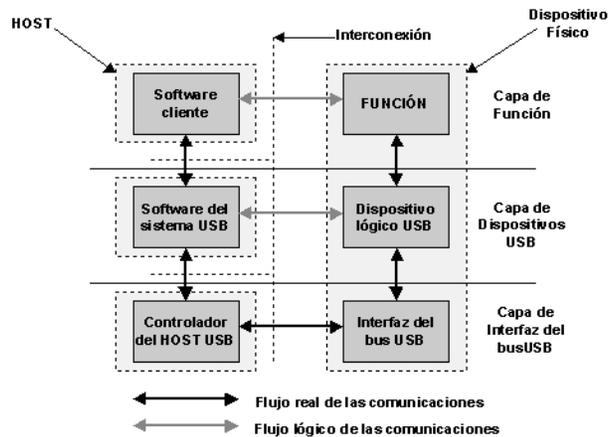


Figura 3.4 Explicación por capas de la protocolización del USB2.0

Como se conforman las tramas usadas en el USB2.0

A continuación se explicaran cada uno de los elementos de la trama del protocolo de comunicación que se está utilizando.

-Sync

Para la comunicación, la función y el host deben de hacer un Sync antes de empezar a realizar una comunicación, el cual consiste en usar 8 o 32 bits para hacer una sincronización de relojes, donde los dos últimos bits son usados para indicar en donde el *Packet ID* empieza. Se debe de tomar en cuenta que la información transmitida comienza a enviar con el bit menos significativo (LSB).

-Packet ID

El *Packet ID* se utiliza para identificar que se está enviando/recibiendo, que tipo de paquete, y dentro de cada paquete, que identificador se está usando. Se utilizan 8 bits para esto, 4 bits para el identificador y los siguientes 4 son una copia negada lógicamente, de los primeros 4, esto con el objetivo de verificar que llegaron correctamente.

Grupo	Valor del <i>Packet ID</i>
<i>Token</i>	XX01b
<i>Data</i>	XX11b
<i>Handshake</i>	XX10b
<i>Special</i>	XX00b

Tabla 3-6: Posibles valores del Packet ID según el tipo de paquete enviado

-ADDR

Son 7 bits, los cuales especifican cuál de los 127 dispositivos se va a usar, la dirección 0 no se le puede asignar a nadie.

-ENDP

Este campo está hecho para seleccionar el *Endpoint* y *Pipe* que se va a usar, consta de 4 bits (2 para dispositivos *Low-Speed*), con lo cual se tienen 16 posibilidades de mandar la información (4 para dispositivos *Low-Speed*).

-CRC

Cyclic Redundancy Checks (CRC)-Este campo es usado para el chequeo de errores, dependiendo del tipo de paquete, se usa un *CRC5* (5 bits), o un *CRC 16* (16 bits, solo usado en los paquetes de datos).

-EOP

End of Packet. Este campo, conformado por 2 bits es usado para asegurar que al terminar el mensaje, el nivel de voltaje esté en bajo (representa un cero lógico), esto para asegurar que no haya un *crossover* entre los canales de información.

Tipos de paquetes (*Packet ID*)

Existen 4 tipos de paquetes, *Token*, *Data*, *Handshake* y *Special*, los cuales se subdividen de la siguiente manera:

Token (xx01b):

- *Out* (0001b): Informan que enviará información.
- *In* (1001b): Informa que se quiere leer información.
- *Setup* (1101b): Empieza etapa de control.
- *SOF*(0101b): *Start-of-Frame*

<i>Sync</i>	<i>Packet ID</i>	<i>ADDR</i>	<i>ENDP</i>	<i>CRC5</i>	<i>EOP</i>
-------------	------------------	-------------	-------------	-------------	------------

Tabla 3-7 Como se compone la trama “token”

Data(xx11b):

La mayor cantidad de datos que se puede enviar, depende de la velocidad, en *low-speed* son 8 bytes, en *full-speed* son 1023bytes, y en *high-speed* son 1024bytes.

- *DATA0* (0011b)
- *DATA1*(1011b)
- *DATA2*(0111b): *High-bandwidth isochronous microframe*
- *MDATA*(1111b): *High-bandwidth isochronous transaction*

<i>Sync</i>	<i>Packet ID</i>	<i>Data</i>	<i>CRC16</i>	<i>EOP</i>
-------------	------------------	-------------	--------------	------------

Tabla 3-8 Como se compone la trama “Data”

Handshake (xx10b):

- *ACK* (0010b): Sirve para avisar que recibió mensaje.
- *NAK* (1010b): Sirve para avisar que en esos momentos no puede recibir o mandar información, o en una transacción de interrupción, informar que no hay información que mandar.
- *STALL* (1110b): El dispositivo informa al host que necesita de su intervención.
- *NYET* (0110b): Sin respuesta del receptor.

<i>Sync</i>	<i>Packet ID</i>	<i>EOP</i>
-------------	------------------	------------

Tabla 3-9 Como se compone la trama “Handshake”

Special(xx00b):

- *PREamble* (1100b): (usa la trama “token”) *Host-issued preamble*.
- *ERR* (1100b): (usa la trama “handshake”) *Split Transaction error handshake*.
- *Split* (1000b): (usa la trama “token”) *High-speed transaction token*.
- *Ping* (0100b): (usa la trama “token”) *High-speed flow control probe for bulk/control Endpoint*.

- *Reserved (0000b): Packet ID reserved.*

Endpoints y pipes

Los *endpoints* es la forma de selección por la cual va a enviar/recibir la función (dispositivo) la información al/del host, determinando la frecuencia del acceso al bus y su latencia, el ancho de banda que requiere, el número de *endpoint*, la forma que se va a manejar el *error handling*, el tipo de transferencia que se va a realizar y la dirección de la transferencia.

Los *pipes* son la asociación entre el *endpoint* (del dispositivo) y el *software* del host. Representan la transferencia de información del *software* y el *endpoint* por medio de un *buffer* de memoria. Existen 2 tipos de comunicación por medio de *pipes*.

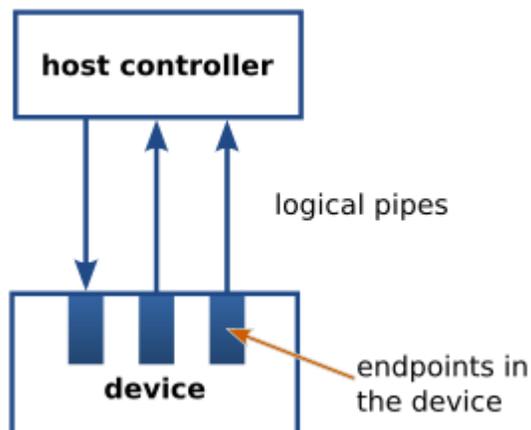


Figura 3.5 Diagrama funcional de los pipes y endpoints en un USB

Tipos de pipes

- *Stream pipe*: La información del *pipe* no tiene una estructura definida por el *USB*, soporta los *endpoints* del tipo *bulk*, *isochronous* e *interrupt*
- *Message pipe*: La información del *pipe* tiene una estructura definida por el *USB*, soporta el *endpoint control*.

Tipos de *endpoints*

- *Control*: Utilizado prioritariamente para operaciones de chequeo de estatus y comando. Transferencias en ráfaga, no periódicas, de tipo *polling*.
- *Isochronous*: Comunicación periódica y continua entre el host y el dispositivo, la cual lleva usualmente información que relevante con respecto al tiempo de llegada, y en su encapsulado conserva el concepto de encapsulación por tiempo. Pero no se hace un manejo de esta información como “crítica” o de alta importancia de transferencia. Se usa prioritariamente en audio y video.
- *Interrupt*: Es un tipo de comunicación de baja frecuencia, y una latencia específica, en el cual el *pipe* es unidireccional y se le da una prioridad alta en la transferencia.
- *Bulk*: Este *endpoint* está diseñado para que pueda ser mandada una alta cantidad de datos en tiempos variable, y puede usar cualquier cantidad de ancho de banda, y garantiza que el dato llegó (en caso de no haber llegado, lo vuelve a mandar).

Descriptores

Los descriptores es la forma en la cual la computadora puede distinguir una memoria USB de una cámara web, cuantos *endpoints* tiene, quien lo hizo, que versión de USB soporta, cuantos pueden entrar en la configuración, etc. Los descriptores más comúnmente usados son:

- *Device*: Describe la información general del dispositivo, incluyendo la información que sirve para la configuración global de este tipo de dispositivos. Como el *idVendor*, el *idProduct*, quien lo produjo y el producto, al igual que un número de serie, el número de versión del dispositivo, entre otros.
- *Configuration*: Este descriptor nos da información específica a de la configuración del dispositivo, como es si la fuente de poder es por medio del USB o es externa, el tipo de *wake-up* que se va a usar, corriente máxima permitida/suministrada, número de interfaces, entre otras cosas.
- *Interface*: Este habla del número de *endpoints* por interface que se van a usar, la clase que se usa (ejemplo HID, comunicación, almacenamiento, etc.), esto último, para evitar escribir drivers genéricos para cada uno de los dispositivos.
- *Endpoint*: Para este descriptor, sirve para configurar cada uno de los *endpoint* usados, dándole los atributos del tipo de *endpoint* que se va a

emplear, el tipo de sincronización que va a manejar, el tamaño máximo de información que puede suministrar el *endpoint*.

- String: Este último provee la información en un formato que cualquier persona puede leer, configurándola en cierto lenguaje.

Desarrollo de la interfaz y protocolo de comunicación *USB2.0* para Autocorrelador por Absorción de Dos Fotones (TPA) portátil

Una vez comprendidas las especificaciones para establecer la adquisición y transferencia de datos por medio de la interfaz *USB2.0*, se eligieron los protocolos y se procedió a implementarlos en el instrumento que se propone en el presente proyecto. La elección de protocolos se describe a continuación.

En una etapa inicial se generaron los descriptores para incluirlos en el programa de firmware, para que pueda ser reconocido y ejecute la función que nosotros queremos. Al igual de la generación de los drivers para que el sistema operativo reconozca al dispositivo.

Se utiliza un solo *endpoint* unidireccional tipo *Bulk* para la transferencia de la información, con el fin de garantizar que la información llegue correctamente. Se tiene la posibilidad, a futuro, aumentar el número de *endpoints* para incrementar las capacidades de la tarjeta DAQ, y en dado momento, poder unir las tarjetas DAQ y VCO en una sola, pudiendo controlar la frecuencia por medio de *software*.

A continuación se enumeran las ventajas que la *USB2.0* ofrece a nuestro instrumento:

- Compatibilidad con computadoras con puertos para este estándar
- Alta velocidad de comunicación entre la computadora y el dispositivo
- Facilidades de conexión, el uso de cualquier cable USB (que cumpla el estándar)
- Alimentación del dispositivo por medio del mismo puerto, quitando la necesidad de alimentación externa al puerto
- Facilidad de hacer un *upgrade* y ampliación tanto al *software* como al dispositivo, al igual que de la computadora (por si se requiere cambiar la

computadora por daño o por mejora, se tiene la facilidad de poder usar cualquiera que cumpla con los requerimientos)

- Posible utilización de este dispositivo en diferentes sistemas operativos solo utilizando diferentes drivers según el OS usado (esto debido a que cada sistema operativo tiene una manera diferente del reconocimiento, manejo y utilización de periféricos).

Capítulo 4. Sistema Opto-Electro-Mecánico

4.1.- Bloque I “Línea de Retardo”

La línea de retardo es un módulo esencial para este instrumento, ya que este, genera el retardo temporal/espacial de los pulsos y crea la autocorrelación. Para la línea de retardo se buscaron y evaluaron 3 diferentes alternativas.

1.-En un primer paso se intentó construir la línea de retardo a partir del mecanismo de desplazamiento de la lente de enfoque que utiliza un reproductor de CD (Figura 4.1). Se hicieron pruebas y se encontró que no era lo suficientemente robusto para soportar cargas de peso, en este caso un espejo.

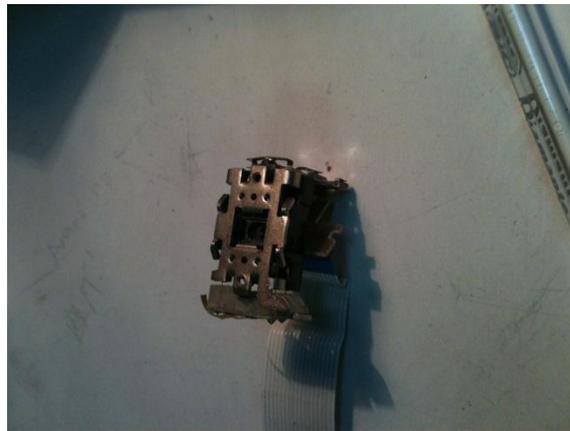


Figura 4.1 Sistema de enfoque de un CD

2.-Otra alternativa planteada fue utilizar una bocina (Figura 4.2), lo cual resultó ser la opción más convencional en este tipo de dispositivos. En este caso la limitante fue el tamaño. De acuerdo a las pequeñas dimensiones del instrumento que se planteó, la bocina era muy grande y además no lograba generar el desplazamiento o retardo suficiente para construir trazas de autocorrelación de pulsos mayores a 100 fs. En este caso en un ciclo de barrido completo se traslapaban las funciones de autocorrelación generando un error de *aliasing* (error de traslape o repliegue espectral).

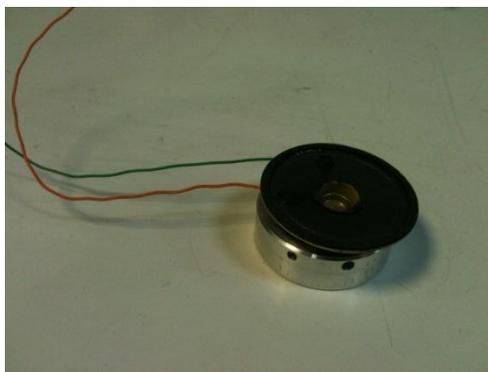


Figura 4.2. Sistema en base a la bocina

3.-En el tercer caso se utilizó el mecanismo electro-magnético (que posteriormente llamaremos actuador) de enfoque de una cámara web. Con este se logró obtener un retardo suficientemente amplio para la generación de las trazas de autocorrelación. Dicho actuador es el que se usa actualmente como línea de retardo. En este caso lo único que debe ser tomado en consideración es el hecho de que existen posiciones en las que no se mantiene una linealidad en cuanto al movimiento del actuador. Para resolver este problema solo se delimita el barrido en aquella región donde mantiene un comportamiento lineal, el cual se logra por medio de un ajuste directamente en el programa.

Para el VCO (*Voltage Control Oscillator*) se usó inicialmente un ICL8038 - *Precision Waveform Generator/Voltage* - unido a una etapa de potencia, diseño que se eligió cambiar (debido a la utilización de un actuador diferente para el cual fue diseñada esta etapa) por uno que entrega menos potencia (la etapa de potencia original llegaba a dar 48W máximo), y menor voltaje de operación (de una fuente dual de $\pm 12V$ se cambió a una fuente sencilla de 5V), esto por considerar el origen del actuador, que es una webcam conectada a USB, donde el voltaje máximo es de 5V y la potencia máxima que puede llegar a usar es de 500mW.

a) Descripción

El actuador que se aplica es el mecanismo autoenfoco de una cámara web *HD logitech C920*, por lo cual, para generar el VCO se utilizó una lógica de

0-5V, considerando el hecho de que la cámara *web* es activada por protocolo *USB* (ver *Capítulo 3: Estándares de voltajes y corrientes del USB 2.0*).

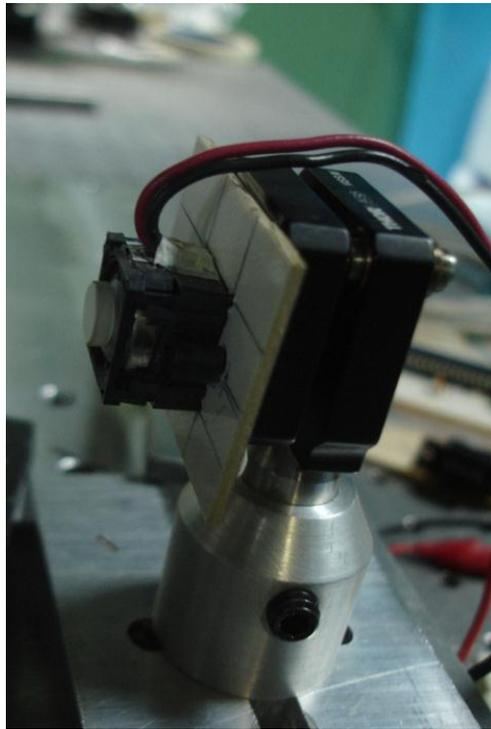


Figura 4.3 Actuator Electro-mecánico en uso

En cuanto al VCO, se utilizó un microprocesador *PIC 18F4550*, debido a su facilidad para generar la modulación *PWM* (*Apéndice C*), y que se pueden generar diferentes señales por medio de algoritmos que son programados en el microprocesador. También se ha considerado la posibilidad de que el VCO se convierta en un oscilador controlado por *software*, aprovechando la capacidad del microchip para usar la comunicación por *USB* y ser alimentarlo por medio de la *PC*.

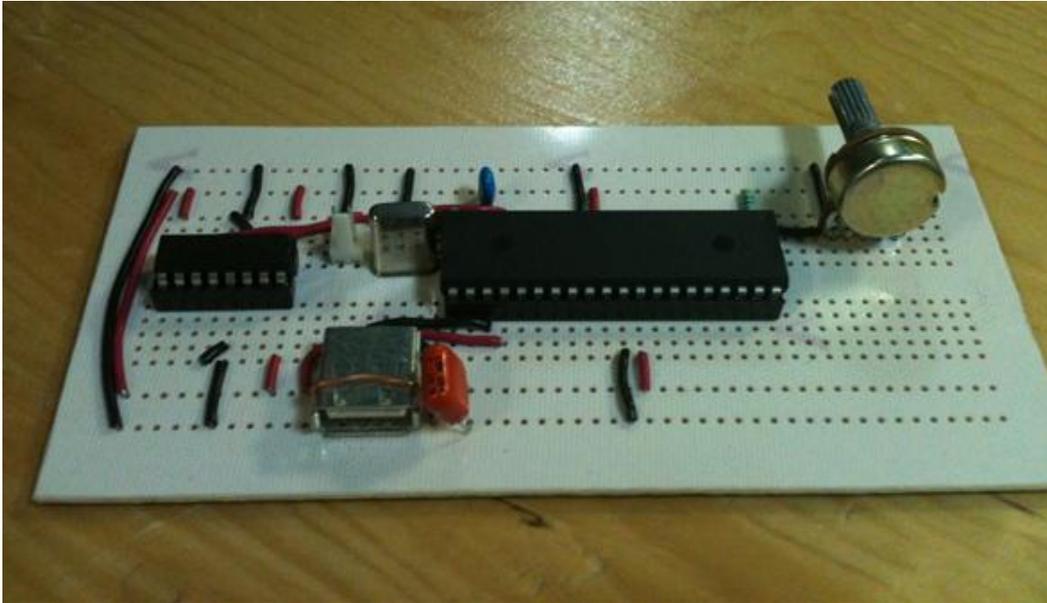


Figura 4.4 Prototipo del oscilador controlado por voltaje (VCO)

Para la etapa de potencia eléctrica tomada del *USB*, se usa un puente H, L293D, sin dispersor de calor, debido a que la potencia usada por el actuador es muy baja. En caso de hacer un cambio de actuador, por uno que utilice más potencia, se podría utilizar un pequeño ventilador y si supera la potencia máxima que puede entregar el *USB*, utilizar una fuente externa.

26

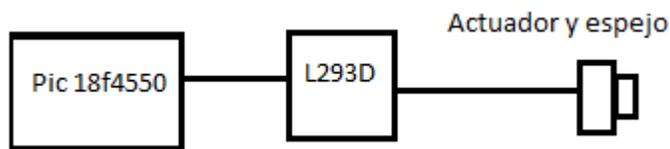


Figura 4.5 Diagrama lógico de la línea de retardo

b) Diagrama

A continuación mostramos el diagrama lógico que se diseñó para este bloque

cristal, filtro y sensor. En cuanto a la adquisición de la señal, se tuvieron que aplicar diversas técnicas, entre ellas el acoplamiento de voltaje, para no demandarle al sensor potencia o corriente (dificultando así la medición), una medición de la tierra para eliminar el ruido en el sensor (usando una tierra flotada, y haciendo una comparación entre la “tierra” y la señal leída). Se propuso utilizar filtros activos (filtro pasa-altas, apéndice B) para la eliminación del ruido, pero mejor se realizó el uso de promedios para la eliminación de ruidos no periódicos por medio de *software*, dejando así solo la posibilidad de problemas con ruidos periódicos (Ej. Ruido de línea el cual es inducido por la misma red eléctrica; eliminado generalmente por medio del uso de un filtro tipo notch o rechaza-bandas, apéndice B).

a) Descripción

En esta etapa, se llevó a cabo un acoplamiento de la señal, para que esta pudiera ser leída por medio de la DAQ. Este acoplamiento consistió en ajuste de un arreglo pasivo de resistencias, esto para realizar un acoplamiento de voltaje y no demandarle potencia ni darle una carga al sensor (Apéndice E).

Dado que el sensor (Hamamatsu G1115) tiene una impedancia alta, se usó una impedancia de entrada de aproximadamente $40\text{M}\Omega$ (la impedancia aproximada del sensor es de $1\text{M}\Omega$, una relación de resistencias (entrada-salida) de 10 a 1 se puede considerar aceptable, una condición óptima sería de 100 a 1).

En el caso de la entrada del amplificador operacional, se usa una tierra flotada y una entrada diferencial para eliminar lo más posible el ruido que se pueda presentar entre el sensor y la toma de datos (tomando uno de los extremos del sensor y restándosele al otro extremo).

Se decidió hacer por *hardware*, usando un *op-amp*, ya que si se hubiera usado una amplificación por *software*, el ruido de altas frecuencias, errores sistemáticos de medición entre otros también podrían ser amplificados, al igual que para tener una mayor velocidad de respuesta (teniendo en cuenta que la amplificación por *hardware* es analógica y prácticamente instantánea, a diferencia

de una amplificación por *software*, que consume recursos de la máquina y puede llegar a hacer más lento los procesos).

Como se desea evitar la utilización de un cristal de segundo armónico y usar un sensor de luz de bajo precio y de fácil obtención, esto se integra usando la técnica de absorción de dos fotones, en donde se utiliza un fotodiodo (o LED) debido a las características del semiconductor, en este caso se utiliza un LED de GaAsP (arseniuro fosfuro de galio), esto debido a la condición de la banda prohibida, en la cual, un solo fotón no es suficiente para llenar el salto que hay entre una banda de valencia y la banda de conducción (esto debido a la energía del mismo fotón), y se necesitan dos fotones (de ahí viene el nombre de la técnica, absorción de dos fotones) [5] (Figura 1.2).



Figura 4.7 Sensores usados.

Izquierda: fotosensor Hamamatsu G1115

Derecha: LED (GaAs/GaAsP)

b) Diagrama

A continuación se muestra el diagrama del diseño y construcción de la etapa de adquisición de la señal óptica.

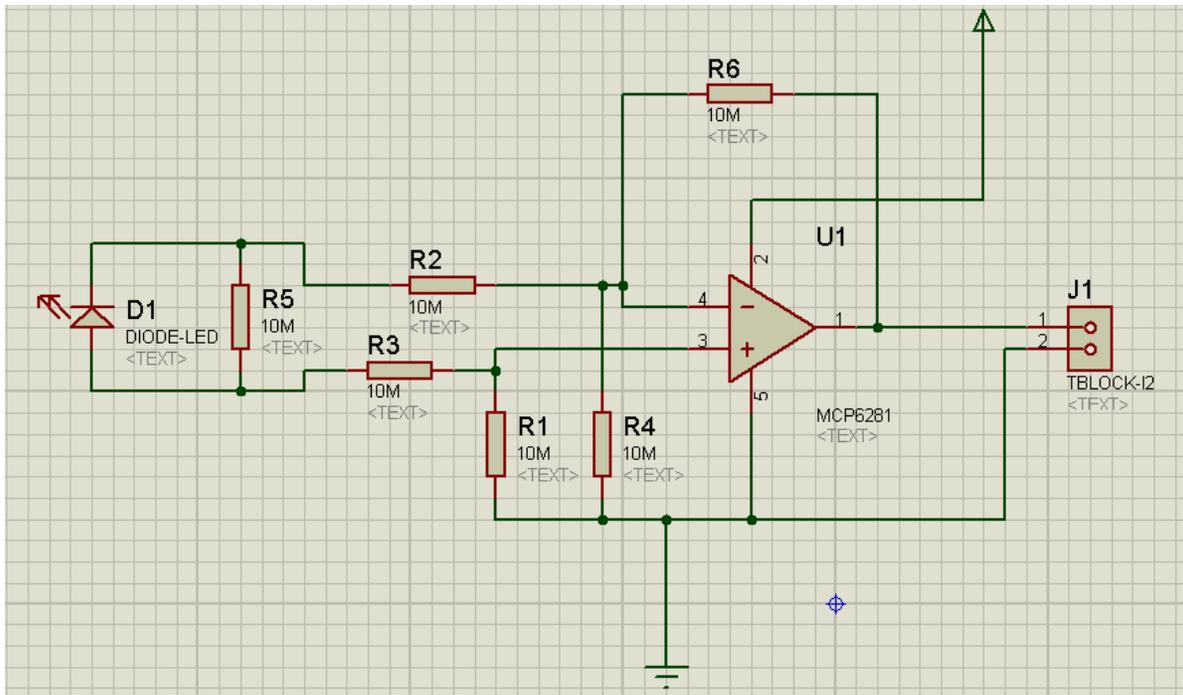


Figura 4.8 Esquema de conexiones del sensor y tratamiento de señal

c) Componentes Utilizados

1. LED(GaAs/GaAsP) o fotosensor Hamamatsu G1115 (D1)
2. MCP6281 (OP-AMP, U1)
3. Resistencias de diferentes valores

nota: la alimentación del amplificador operacional se toma por parte del USB, por lo cual no se necesita generar una fuente de potencia para esta parte.

4.3.- Bloque III “Interferómetro *Michelson*”

El interferómetro de *Michelson* es una parte primordial de este instrumento, ya que con él se genera la señal de autocorrelación de Intensidad. Este está conformado por un divisor de haz de película modelo BP145B1 (TorLabs 45R-55T) y dos brazos. En uno de los brazos se encuentra la línea de retardo y en el otro un espejo fijo montado en una platina traslacional controlada con un micrómetro (12XT40v de la marca GMP, Comar). En una etapa inicial de alineación se procura ajustar el retardo empatando la longitud de los dos brazos respecto al

divisor de haz. A la salida del divisor ambos haces se unen colinealmente, y son enviados a una lente que enfoca la señal en el sensor.

a) Descripción

La señal de autocorrelación de intensidad es generada por medio de la división del haz de luz en proporciones iguales, esto para no darle un peso diferente a cada brazo del interferómetro. Y la línea de retardo es usada para generar un desfase temporal/espacial entre los dos pulsos, generando así la función de autocorrelación en intensidad.

b) Prototipo

A continuación se muestra una imagen del prototipo de la base del interferómetro.

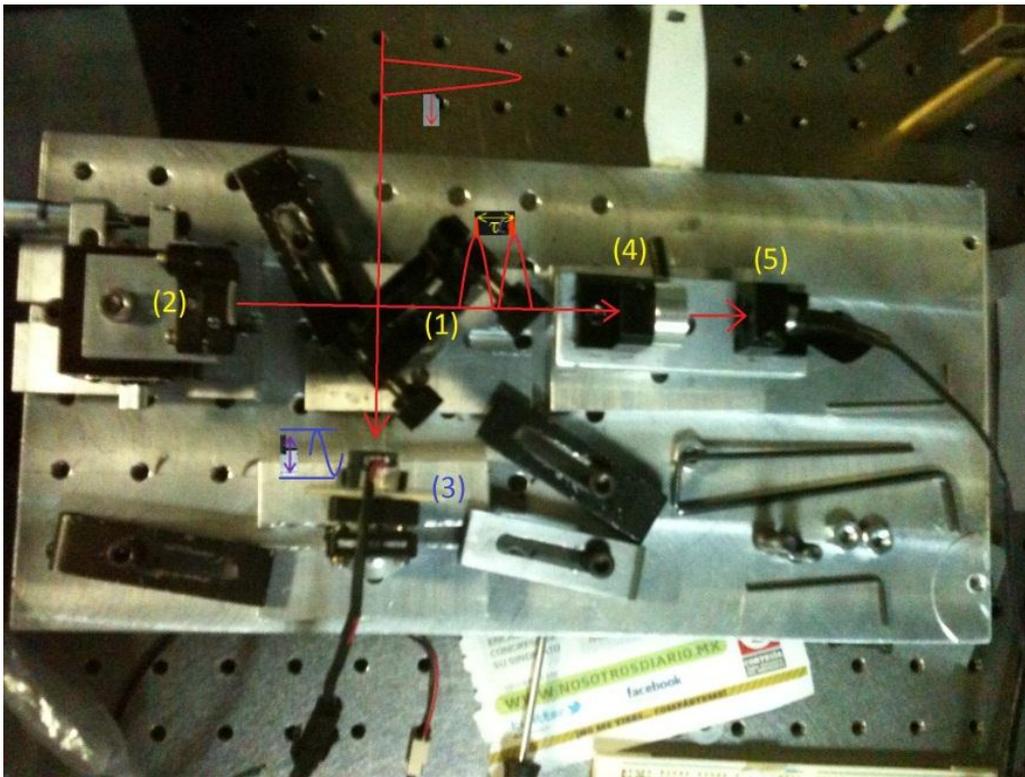


Figura 4.9 Base Prototipo.

- (1) divisor de haz. (2) Espejo fijo con ajuste de distancia (3) Línea de retardo (4) Lente de enfoque (5) Sensor

c) Componentes Utilizados

1. Bloque de aluminio
2. Bases ópticas ajustables para espejos y divisor de haz
3. Divisor de haz de película ThorLabs modelo BP145B1
4. Platina traslacional Comar modelo 12XT40v
5. Base para lente de enfoque y LED

4.4- Bloque IV “Muestreo de Señales” –DAQ–

Para este módulo, se diseñó una *DAQ – Data Acquisition* (siglas en inglés para tarjeta de adquisición de datos) que pudiera obtener una buena velocidad de adquisición (y buena respuesta frecuencial), de un costo bajo, y que fuera fácil de conectar a una PC, por ello se buscó una comunicación por medio del USB.

La forma en que se muestre una señal es muy importante (en especial en este tipo de aparatos), ya que si no se hace correctamente, se le puede introducir ruido a la señal medida, mala lectura del sensor, se pueden omitir frecuencias en la medición (teorema de *Nyquist*), con lo cual se pueden omitir información necesaria en el experimento.

a) Descripción

Esta etapa fue de las más difíciles, ya que el problema mayor que se tuvo fue la comunicación de la tarjeta de adquisición de datos con la computadora, debido a la propuesta de utilizar la conexión USB, ya que nos proporcionaba: una mayor velocidad de comunicación (evitando que fuera este el cuello de botella en la velocidad de muestreo); el poder usar esta misma conexión como fuente de alimentación del sistema; y la posibilidad de hacer *upgrades* con facilidad.

La comunicación por puerto serie hubiera limitado la velocidad, el tener que usar una fuente de potencia externa, y el uso de un adaptador o una computadora que tuviera este puerto.

En una primera versión para controlar la toma de datos, se intentó utilizar la interrupción por reloj del microchip, pero debido a problemas, posiblemente con el compilador, no fue posible controlarlo de esta manera, lo cual restringió mucha de la capacidad del microprocesador y dejándolo con una función única de toma de datos.

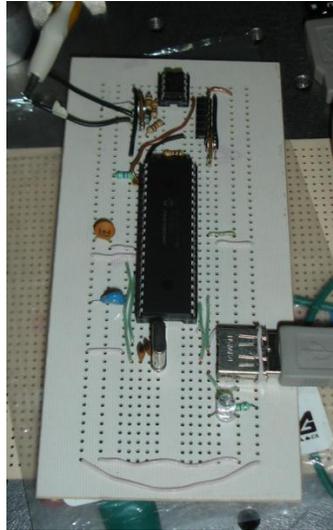


Figura 4.10 Prototipo de la tarjeta de adquisición de datos

b) Diagrama

A continuación se muestra el diagrama usado para la construcción del dispositivo DAQ.

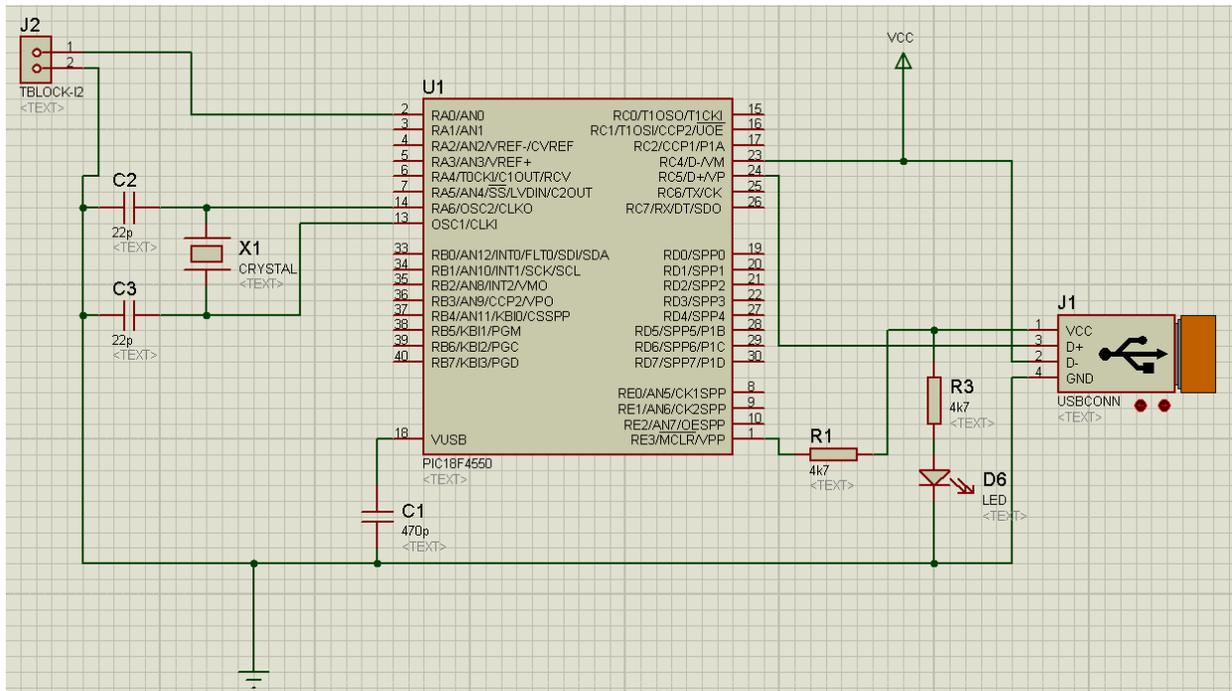


Figura 4.11 Diseño lógico de la tarjeta de adquisición de datos (DAQ)

c) Componentes Utilizados

1. PIC 18F4550 (U1) con todos los componentes requeridos para su operación (el programa usado se encuentra en el anexo A, (a)).
2. Conector USB (J1)

4.5.- Bloque V “Software Interfaz Hombre-Máquina”

En este bloque se presentaran algunos algoritmos que se utilizan en la implementación de los elementos del instrumento, tal como son los *firmwares* de la DAQ (a) y del VCO (b). Así como una descripción a grandes rasgos de la interfaz que fue creada para esta aplicación. (*Firmware en Anexo A, (a) y (b)*) (Nota. Para la generación de este software se utilizó la librería abierta libusbdotnet, para C#, obtenida en <http://sourceforge.net/projects/libusbdotnet/>)

a) Firmware DAQ (Pic18F4550)

Aquí se muestra el diagrama lógico que se siguió para el diseño del *firmware* de la DAQ.

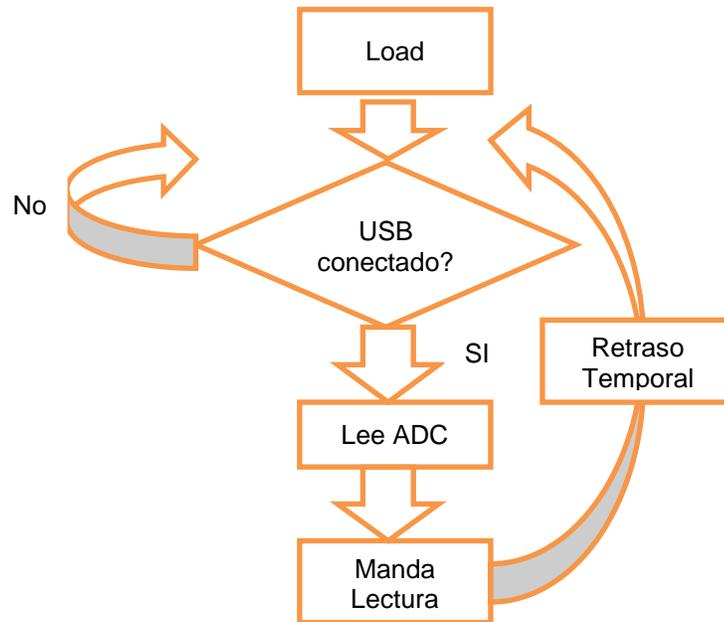


Figura 4.12 Esquema lógico de cómo se conforma el *firmware* de la DAQ

b) Firmware VCO (Pic18F4550)

Aquí se describe la lógica usada para el desarrollo del dispositivo VCO.

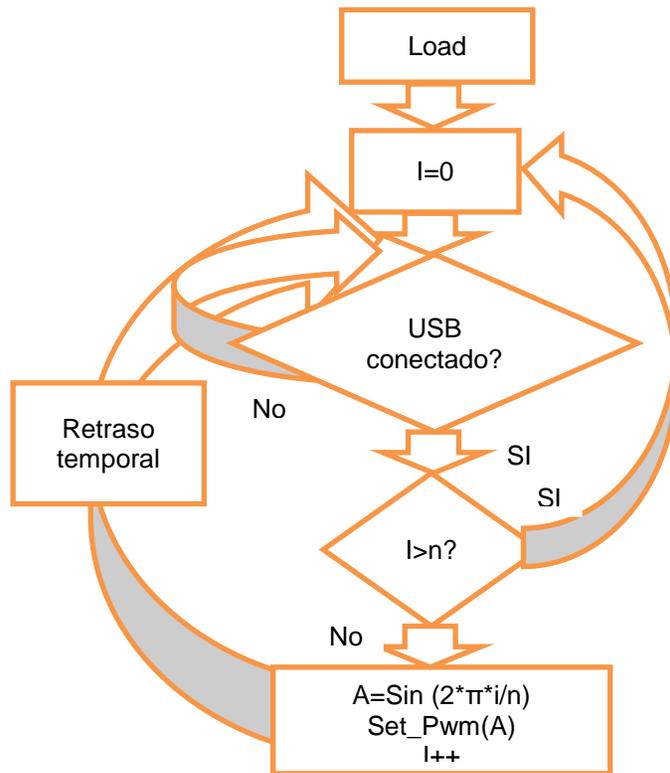


Figura 4.13 Esquema lógico de cómo está construido el Firmware del VCO

4.6.- Sistema para la caracterización de dispositivos Ópticos ultrarrápidos

El sistema de prueba o caracterización consiste en la base que sostiene al detector, y el lente de enfoque, en la cual, el lente es fácilmente cambiado por la base de sostén en la que está colocado. Facilitando así la caracterización de la aberración espacio temporal [6] que sufren los pulsos al ser enfocados, los parámetros variables son:

- Tipo de lente
- El tamaño de haz de entrada

- Iluminación homogénea o gaussiana
- Ancho de pulso de entrada
- Longitud de onda

Para que sea un buen sistema de caracterización, el Autocorrelador no debe de inducir aberraciones cromáticas, ni GVD (*group velocity dispersion*) en el pulso ultracorto. En caso del GVD, se reduce o evita mediante el uso de un divisor de película delgada.

Los dispositivos bajo prueba, son colocados en la base que se muestra en la Figura 4.14 (la base que está señalada con el círculo rojo).

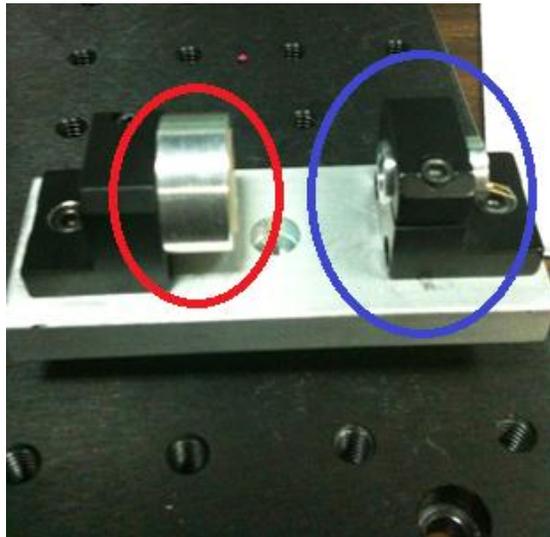


Figura 4.14 Dispositivo óptico ultrarrápido bajo prueba (Rojo) y base del sensor (Azul)

4.7.- Interfaz y comunicación del dispositivo con la computadora

Se propuso usar la comunicación por medio de USB debido a la velocidad de comunicación que ofrece, la facilidad de transporte, opción de alimentación por medio de este, versatilidad en cuanto a la opción de ser usado en múltiples sistemas operativos (el *driver* se diseñó para la familia de *Windows*®, y se pueden crear para *Mac OS*®, *Linux* y otros OS). Para ello se necesita los drivers según el sistema operativo, en este caso los drivers fueron hechos para el entorno *Windows XP*® y *Windows 7*® (*drivers* en Anexo B)

4.8.- Descripción del software

Este programa fue hecho en visual C#, debido a la facilidad del control de puertos, una sintaxis sencilla (evolución de C++), entornos de programación fáciles de utilizar, se pueden utilizar compilaciones a código intermedio (independiente de la máquina que se programó y de la máquina que se vaya a utilizar), entre otras ventajas sobre otros lenguajes. Se utilizó este lenguaje sobre *Labview®*, *Matlab®* (*Simulink®*), y otros lenguajes de este estilo, ya que estos requieren una licencia para ser usados, al igual que permisos para poder ser empleados en un instrumento que puede llegar a ser manufacturado, lo cual dificultaría o constituiría un precio extra.

Para el registro de la información, una vez pasado todo el proceso de adquisición, muestreo, cuantificación, codificación, y comunicación (entre la DAQ y la PC), la información es guardada en una lista dinámica, en el cual para lograr una mejor visualización se le sometió a un proceso selección de un nivel, y es disparada por un proceso de gatillo (*triggering*), y cuando es sobrepasado este nivel (definido por el usuario) se empieza a graficar desde ese punto, y se guarda la posición del valor en una segunda lista dinámica. Esta segunda lista, es llamada como referencia a la primera, con lo que podemos iniciar la gráfica de la señal iniciando en esta referencia, al igual se puede generar un promedio de los pulsos que sobrepasen el nivel de disparo, y al ser una señal periódica, se puede eliminar la mayor cantidad de ruido (no periódico) posible sin necesidad de filtros.

Este programa tiene la facilidad de poder guardar los datos medidos en un archivo .CSV para facilidad de manejo (comma-separated values, Apéndice D), ya sea los datos de la señal promediada, o los datos originales medidos.

Este software tiene una pantalla principal en donde se puede ver en un origen la señal leída por la tarjeta de adquisición, emulando un osciloscopio. Una vez que se ajusta el nivel de gatillo y se activa, se empieza a ver solo un pulso (el gatillo también reduce el tamaño del rango en tiempo), y ya con el gatillo activado,

es posible el poder usar la función de promedio, en el cual cambia la pantalla principal por la señal promediada, se le elimina la señal de corriente directa (se utiliza el primer valor del promedio) y es normalizada (se normaliza con respecto al mayor valor del promedio), y una pantalla secundaria para observar la señal original con gatillo.

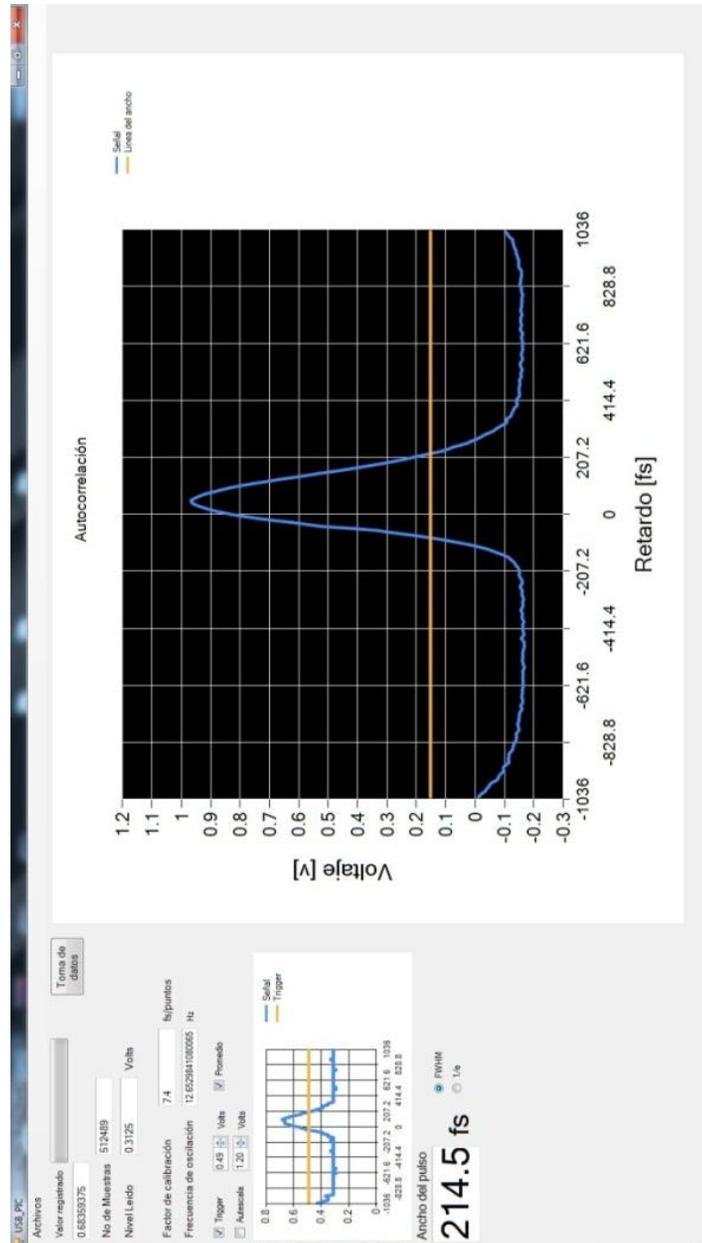


Figura 4.15 Snapshot de una versión de prueba del software en uso, tomando una traza de autocorrelación

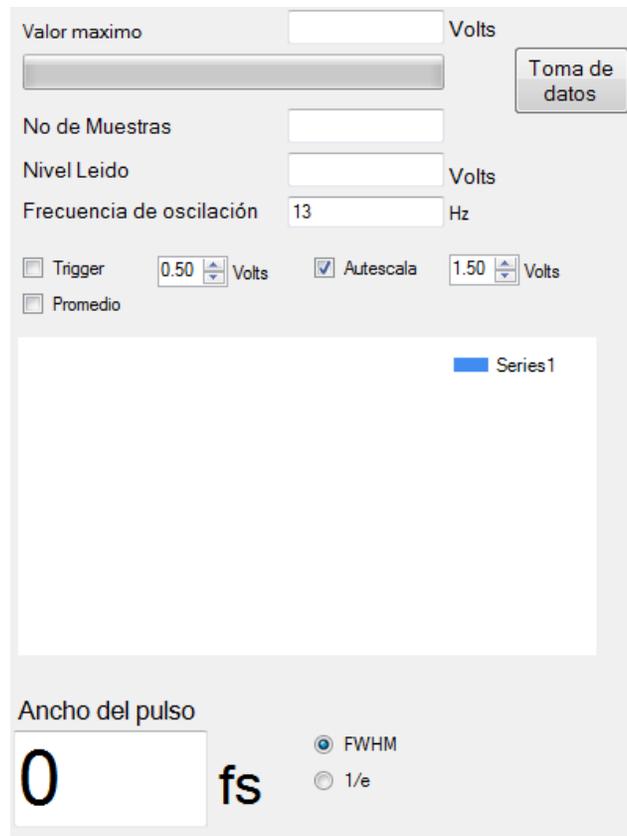


Figura 4.16 Parte de control del software

En la Figura 4.16 podemos observar el botón para iniciar la toma de datos (parte superior derecha, Anexo C, (b) y (f)), los *checkbox* para activar el *trigger* (junto con el nivel que se quiere para activar el gatillo, Anexo C, (e)), el *checkbox* para iniciar el promedio (no puede ser activado si el *trigger* no está encendido, Anexo C, (c)), y un *checkbox* (caja en la cual se puede seleccionar encendido o apagado, por medio de un tache) para acceder a la escala de voltaje en la gráfica (o dejarlo como auto escala). Al igual podemos ver una gráfica, esta gráfica se activa cuando el proceso de promedio se inicia, y lo que se puede ver en la gráfica es el valor real de los datos que se están tomando y el nivel del *trigger* (que originalmente se encuentra en la gráfica principal, Anexo C, (c)). Se puede ver al igual una barra que indica el valor obtenido de la DAQ (al igual que de valor numérico, de 0 a 5), valor máximo de la gráfica que se está leyendo, número de muestras que se han adquirido (desde que se inicia el proceso de adquisición, contador de los datos guardados), un cuadro que nos indica la frecuencia a la que

la línea de retardo se está moviendo y originalmente se tenía el factor de calibración en una ventana para tener acceso a ella y poderse cambiar, pero, debido a que se ideó la forma de que se autoajustara, se quitó del acceso al usuario. (Anexo C, (c)). Y un poco más abajo se encuentra la medición del ancho del pulso, con dos *radio button* (botones en forma redonda en los cuales solo puede estar uno encendido), los cuales sirven para seleccionar el umbral de medición del ancho del pulso, uno de ellos es FWHM y el otro es 1/e (Anexo C, (d)).

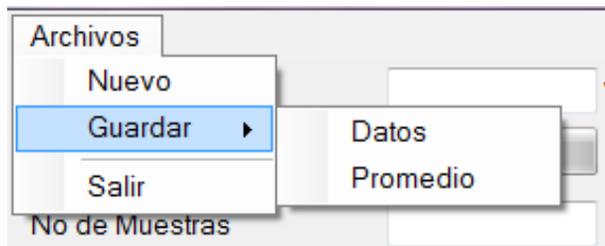


Figura 4.17 Menu strip

En el *menu strip* (Figura 4.17) tenemos las opciones de borrar todos los datos y empezar de nuevo (Nuevo), guardar los datos obtenidos (todos los datos obtenidos en la medición) o los datos que se tienen en el promedio, y la opción de salir del programa.

41

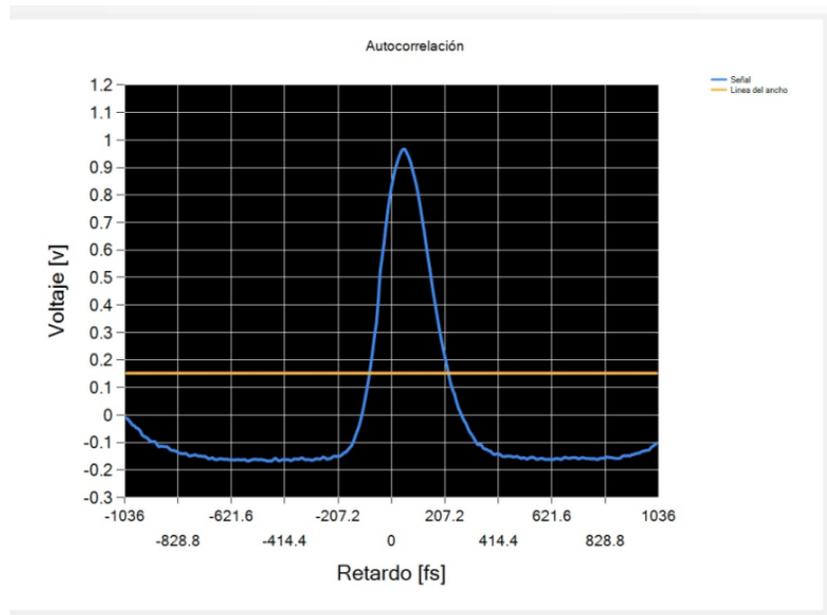


Figura 4.18 Gráfica principal

En la gráfica principal (Figura 4.18) se pueden leer una de dos mediciones:

- La medición que se está obteniendo en la DAQ, junto con el nivel del *trigger*, o
- Cuando esta activada la opción de promedio, (*checkbox* que se muestra en la Figura 4.16), se muestra el promedio de 500 trazas y el nivel de umbral para obtener el tamaño del pulso (Anexo C, (c), (d), (e)).

Una vez que se tiene la traza promedio, se procede a eliminar el DC de toda la señal, y se normaliza (Anexo C, (c)).

4.9.- Calibración

Para la calibración del instrumento, se tuvo que obtener una relación muestra/retardo, y retardo/tiempo. Para ello, se usó el tornillo micrométrico del espejo fijo (Figura 4.19), con ello se obtuvo una relación de movimiento espacial y el movimiento temporal (por medio de un osciloscopio y el software, Figura 4.20), teniendo esto se midió el tamaño del ancho del pulso y la distancia entre los pulsos y se hizo una comparación con los pulsos medidos por el aparato, y con ello se obtuvo el factor de calibración, también se tuvo que obtener la frecuencia en la que el retardo está funcionando, ya que este factor de calibración depende de la frecuencia en la cual este mueve la línea de retraso, por lo que para el factor de calibración se le multiplica por una relación entre la frecuencia que se está moviendo, y la frecuencia original en la que se calibró.

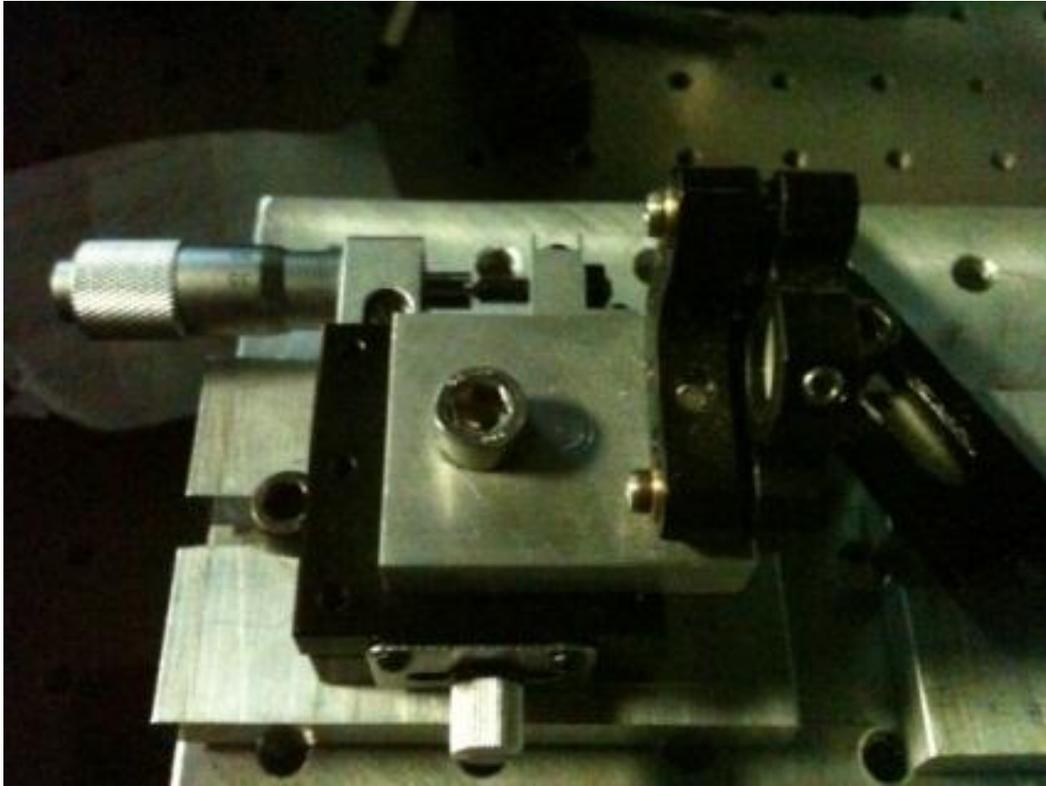


Figura 4.19 Tornillo micrométrico

Se utilizó la siguiente ecuación para obtener esta calibración:

$$\Delta t_p = \frac{2\Delta d_p}{c} \quad \text{Ecuación 4.1}$$

$$\text{f. c.} = \frac{\Delta t_p}{\Delta t_d} \left[\frac{fs}{ms} \right] \quad \text{Ecuación 4.2}$$

Donde Δt_p es el tiempo que representa el movimiento del tornillo micrométrico Δd_p , y Δt_d es el tiempo real del movimiento según el tornillo micrométrico.

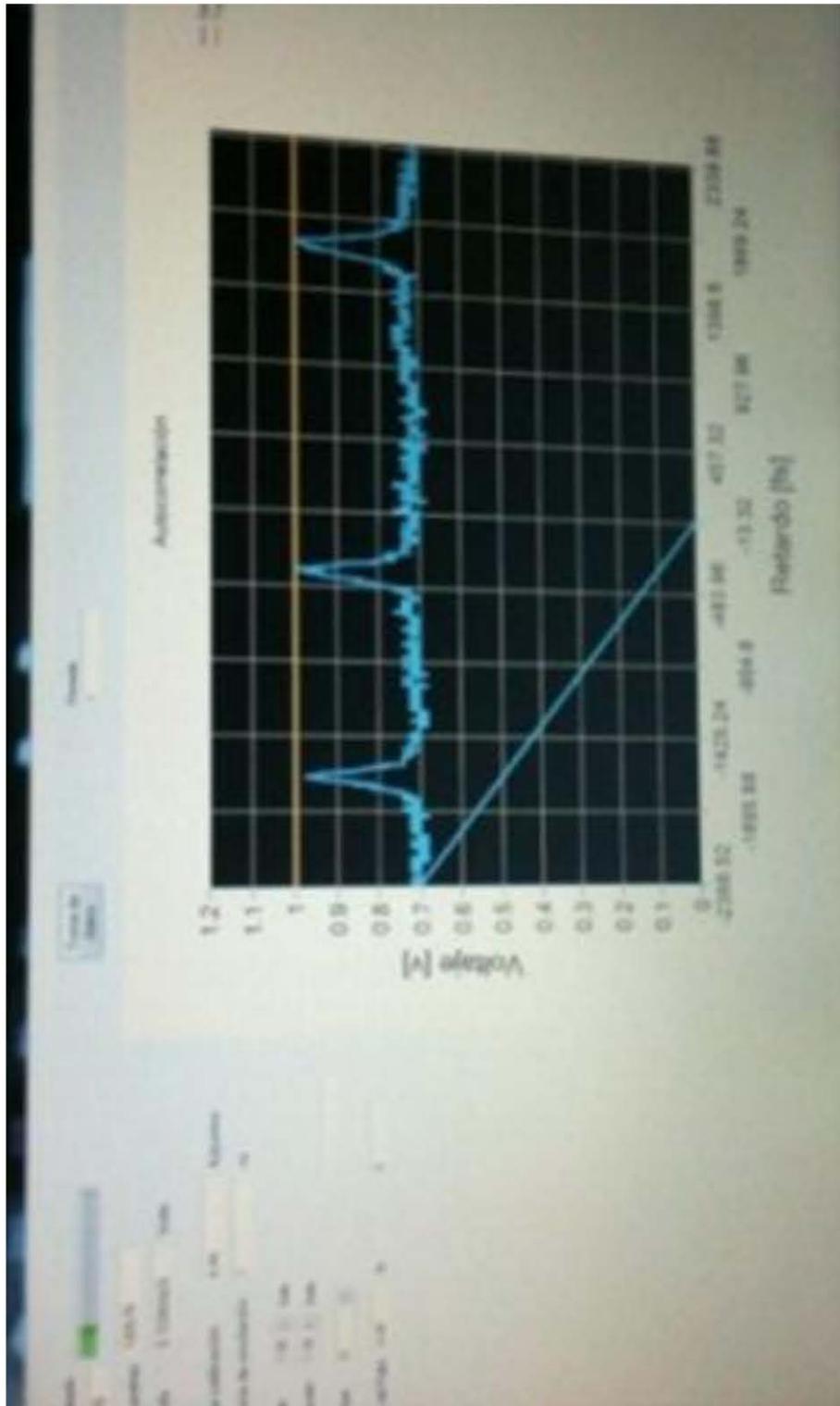


Figura 4.20 Obtención de calibración visto en el software

El factor de calibración que se obtuvo es de 7.4 fs/muestra, medido a una frecuencia de 13.38 Hz.

Capítulo 5. Resultados

Resultados experimentales

Después de alinear la parte óptica (interferómetro colineal de *Michelson*), se procura obtener el mayor voltaje posible leído en el diodo y empatar las longitudes del camino óptico en ambos brazos, tras esto, se obtuvieron las trazas de autocorrelación.

Se hicieron algunas consideraciones con respecto a las trazas:

- La traza tiene que ser simétrica (debido a que es una autocorrelación de segundo orden). Para obtener una lectura correcta de esta señal, se debe de tomar en cuenta que el diodo integra la señal adquirida, Ecuación 1.3 en Descripción del “Fenómeno *TPA (two photon absorption)*”, y por lo tanto se debe de utilizar un buen acoplamiento, y una resistencia de descarga lo suficientemente grande. Se debe de tener en cuenta que si la resistencia es muy pequeña, empieza a atenuar la señal, y si es muy grande, puede llegar a genera problemas de acoplamiento. Si no se utiliza esta resistencia de descarga, la integración puede generar una traza asimétrica (Figura 5.1),

45

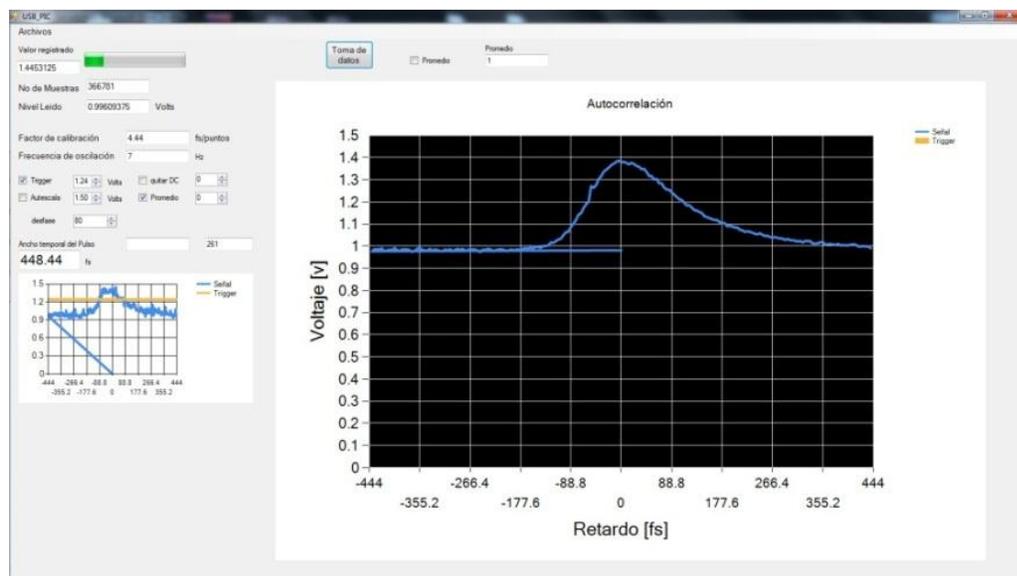


Figura 5.1 Traza asimétrica mal calibrada

- Cuidar que el recorrido de la línea de retraso sea lo suficientemente amplia para cubrir la traza completa, ya que puede hacer que se sobrepongan las trazas, o si es largo el recorrido, puede volver lento el proceso de adquisición (razón por la cual se cambió de actuador en la línea de retardo, de bocina a sistema de autoenfoco).
- Para obtener una buena cantidad de muestras en cada traza (según el teorema de Nyquist la frecuencia de muestreo debe ser mínimo 2 veces mayor a la máxima frecuencia que se desea leer, en este caso se busca tener una relación de 10 veces, para tener una buena visión de la señal sin tener que utilizar métodos de reconstrucción de la misma), las frecuencias de la señal medida depende de la frecuencia de la línea de retardo, ya que esta dicta la velocidad en la que se genera la señal, por lo que si se quiere ver mayor contenido frecuencial, se puede reducir la velocidad de barrido de la línea de retraso, esto debido a que entre mayor sea esta frecuencia, la traza de autocorrelación será más breve en cuanto a tiempo, y la tarjeta de adquisición podría omitir frecuencias necesarias en la medición.

A continuación se presentara un promedio de 500 muestras, tomado por medio del software (Figura 5.2 y Figura 5.3), y usando lo datos guardados, se utilizó el *autofit* de Matlab® (Figura 5.4) para procesar la información obtenida por el instrumento.

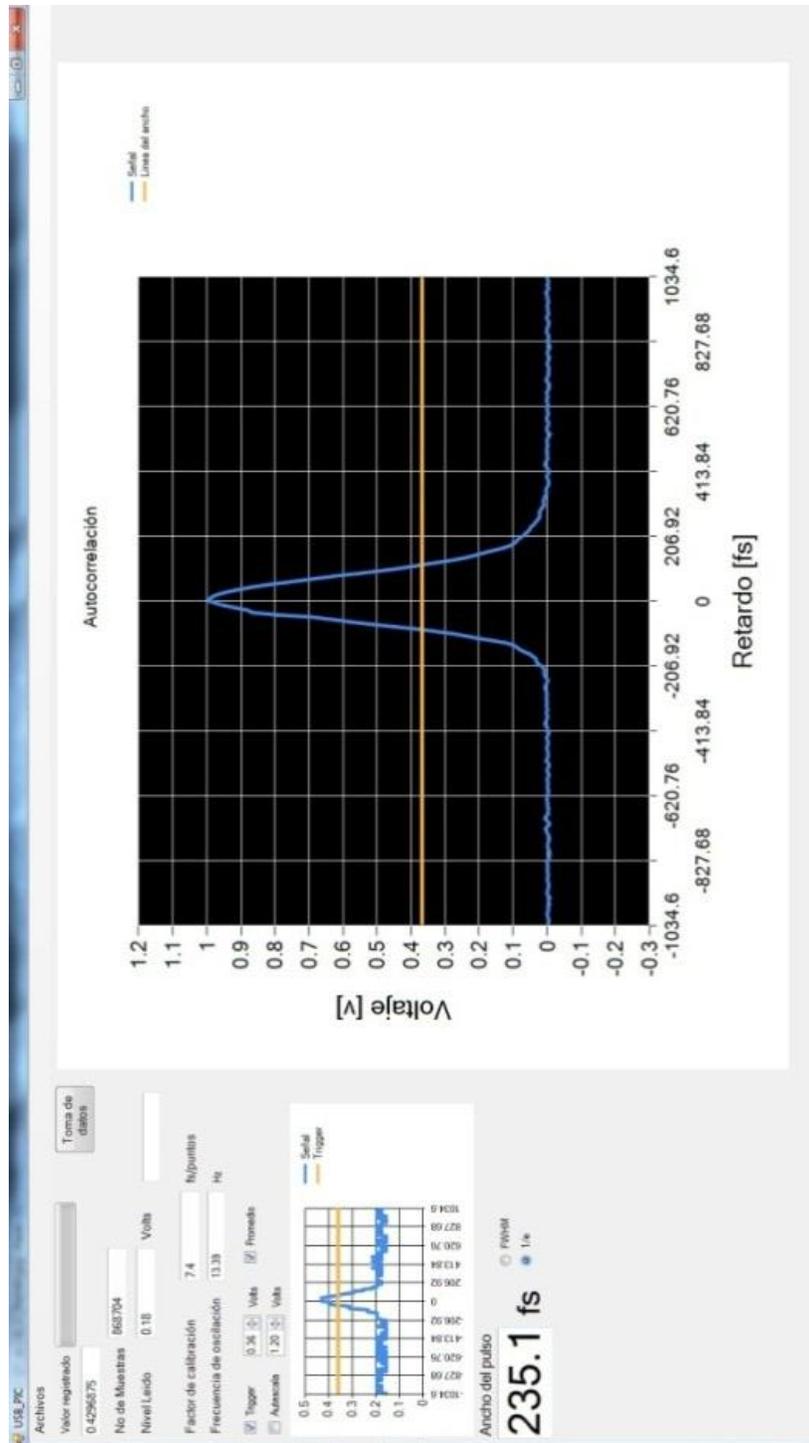


Figura 5.2 Pulso ultracorto usando promedio de 500 trazas, mostrando el ancho del pulso en 1/e

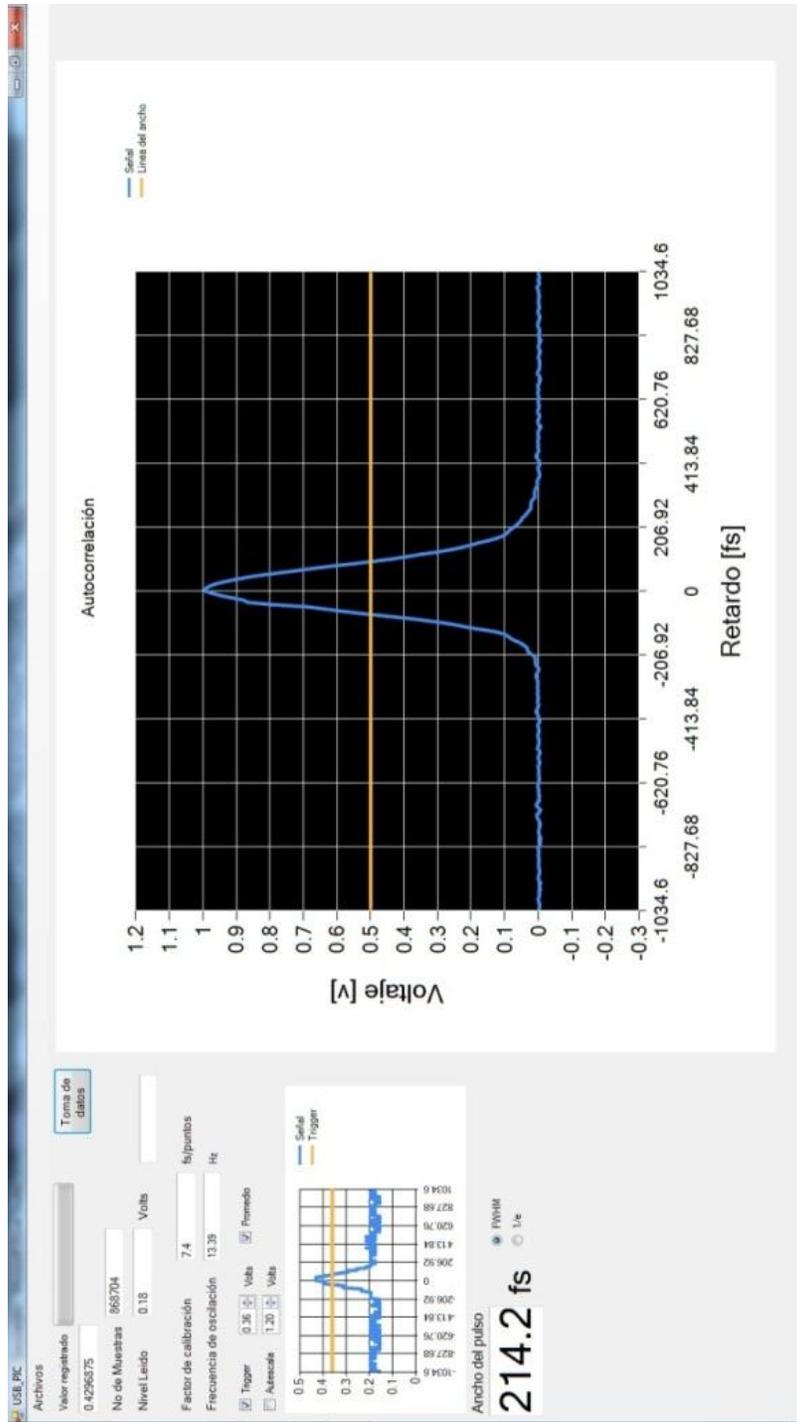


Figura 5.3 Pulso ultracorto usando promedio de 500 trazas, mostrando el ancho del pulso en FWHM

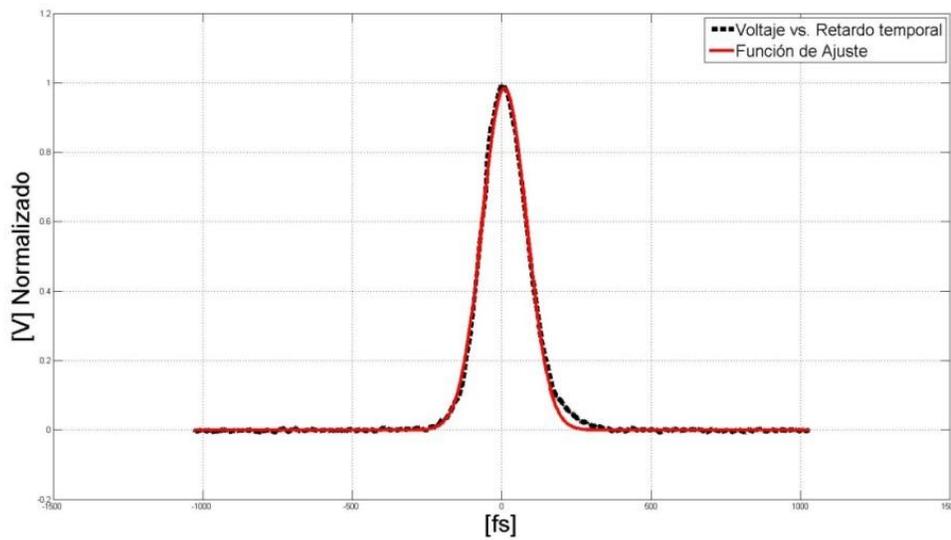


Figura 5.4 Datos procesados en Matlab ®

Con la utilidad cftool, de Matlab®, se utilizó la función $f(x) = a1 \cdot \exp(-((x-b1)/c1)^2)$, para aproximar y se obtuvo (Figura 5.4):

Coefficients (with 95% confidence bounds):

- a) $a1 = 0.9838$ (0.9769, 0.9906)
- b) $b1 = 9.94$ (9.34, 10.54)
- c) $c1 = 105.5$ (104.7, 106.4)

Goodness of fit:

- a) *SSE:* 0.03985
- b) *R-square:* 0.9974
- c) *Adjusted R-square:* 0.9973
- d) *RMSE:* 0.01202

Capitulo 6. Conclusiones

Conclusiones

- Se realizó un montaje experimental de un Autocorrelador TPA para la caracterización de pulsos ultracortos, utilizando materiales accesibles y de un bajo costo.
- Se logró que no se requirieran de instrumentos periféricos, a excepción de una PC para la adquisición de datos, como son un generador de funciones, un osciloscopio, un sistema (platina y control de la misma) para la línea de retardo.
- Se pudo obtener la alimentación de la tarjeta de adquisición (DAQ) y de la línea de retardo con el VCO, de la computadora que se está usando (*Power over USB*), haciendo que este instrumento no necesite algún tipo de alimentación extra.
- El uso de la DAQ y el VCO en distintos sistemas operativos, solo cambiando el driver usado para cada uno (actualmente solo se tiene el driver para Windows y se probó en las versiones de Windows 7 y Windows XP). Si no se cuenta con el driver para otros sistemas, sigue funcionando, pero algunas funciones son limitadas por el sistema operativo.
- La calibración temporal por punto es de $7.4\text{fs/punto}@13.38\text{Hz}$ (el tamaño temporal por muestra varía según la frecuencia utilizada en la línea de retardo).
- La tamaño de paso en voltaje mínimo leído es de 0.00489 Volts, esta resolución está limitada debido a el ADC que se encuentra en el microcontrolador que se está usando. Si se busca tener un ADC de mayor resolución (más número de pasos), podría llegar a afectar en la tasa de muestreo, debido a los tiempos de conversión de señal analógica a señal digital.

- La resolución espacial de la línea de retardo correspondiente a un paso de voltaje de 0.00489 Volts es de $\sim 80\text{nm}$, que su equivalente temporal es de $\sim 0.5\text{fs}$.

Trabajo futuro

- Mejorar la tasa de muestreo del ADC.
- Generar un programa capaz de manejar el VCO por medio de *software*, y tener acceso a más funciones, no solo la función sinusoidal (igualmente con una conexión USB a la PC).
- Mejorar la base ópto-mecánica, ya que es posible tener un mejor rendimiento del espacio que se dispone, y/o reducir el espacio del Autocorrelador.
- Buscar mejores opciones de microprocesadores (como por ejemplo un microprocesador de 32-bits), y de montaje superficial, por el cual, se obtendrá mejor rendimiento, una mayor velocidad de muestreo, entre otros detalles.
- Buscar el integrar el VCO y el ADC en un solo microprocesador, sin afectar la tasa de muestreo del ADC, ni la generación de señales del VCO.
- Buscar una forma de tener el instrumento en forma *stand-alone* y prescindir de la PC para realizar las mediciones. Teniendo una pantalla exclusiva para este sistema (conexión a la PC para guardado de datos y opciones para usar dispositivos de almacenaje como son tarjetas SD y memorias USB).
- Mejorar la resolución del tamaño de paso de la línea de retardo, ajustando el PWM al voltaje necesario para la corrida del espejo en la línea de retardo.
- Se busca la posibilidad de pasar al protocolo de comunicación *USB 3.0*
-

Apéndice

Apéndice A. Generación de pulsos ultracortos y mode-locking

A) Descripción de “Pulso Ultracorto”

Un pulso laser ultracorto es una breve ráfaga de energía electromagnética. Este pulso, al ser un pulso de luz, esta descrito por su campo eléctrico, en función del espacio-tiempo, $E(x, y, z, t)$. Haciendo una aproximación, con la suposición de que el pulso esta polarizado, podemos dejarlo en función puramente del tiempo [3]:

$$E(t) = \frac{1}{2} \sqrt{I(t)} e^{i[\omega_0 t - \varphi(t)]} + C. C. \quad \text{Ecuación A.1}$$

Para el análisis, haremos consideraciones para facilitar y simplificar los cálculos. Primero, el complejo conjugado (marcado con un “C.C.”) puede ser ignorado debido a la aproximación escalar. Segundo, la frecuencia angular portadora, ω_0 , cambia con mayor rapidez que con respecto a la fase del pulso $\varphi(t)$, considerando que los métodos usados serán promedios de la intensidad, podemos ignorar la frecuencia angular. Obteniendo la siguiente ecuación:

$$E(t) = \sqrt{I(t)} e^{-j\varphi(t)} \quad \text{Ecuación A.2}$$

Por lo que conociendo el campo eléctrico de la señal, podemos conocer la intensidad y la fase, ya sea en tiempo(Ecuación A.3) y Ecuación A.4)), o utilizando la transformada de Fourier, en el dominio frecuencial(Ecuación A.5 y Ecuación A.6).

$$I(t) = |E(t)|^2 \quad \text{Ecuación A.3}$$

$$\varphi(t) = -\arctan\left(\frac{\text{Im}[E(t)]}{\text{Re}[E(t)]}\right) \quad \text{Ecuación A.4}$$

A

$$S(\omega) = |\tilde{E}(\omega)|^2 \quad \text{Ecuación A.5}$$

$$\varphi(\omega) = -\arctan\left(\frac{\text{Im}[\tilde{E}(\omega)]}{\text{Re}[\tilde{E}(\omega)]}\right) \quad \text{Ecuación A.6}$$

$S(\omega)$ corresponde al espectro (aplicación de transformada de Fourier en Ecuación A.3 y Ecuación A.4)

B) Mode-locking

Mode-locking o amarre de modos es un método el cual consiste en seleccionar un modo fundamental, seleccionar los modos de oscilación y empatar las fases, para poder generar un pulso de cierta forma y cierta duración, en este caso, entre más modos haya (esto implica mayor ancho espectral) de menor duración será el pulso⁷.

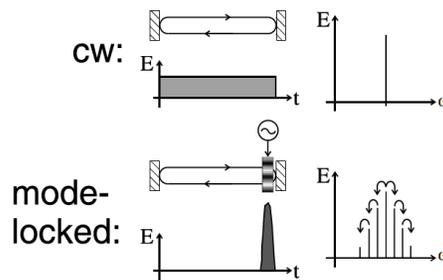


Figura APA.1 comparación entre un láser de modo continuo (CW) y un láser pulsado por medio de mode-locking

La técnica que se utiliza en este caso para el amarre de modos es un efecto no lineal de tercer orden, llamado efecto *Kerr* óptico, el cual consiste en un cambio del índice de refracción auto-inducido (el índice de refracción cambia debido a la cantidad de intensidad que llega).

$$n(t) = n_0 + n_{12}I(t) \quad \text{Ecuación B.1}$$

$$n_{12} = \frac{6\chi^3}{8\varepsilon_0cn_0} \quad \text{Ecuación B.2}$$

⁷ [9] Y [10]

Donde n_0 es el índice de refracción lineal, ϵ_0 es la constante de permisividad dieléctrica en el vacío, c es la velocidad de la luz, y χ^3 es la susceptibilidad dieléctrica de tercer orden del material. Con esto, se obtiene un ensanchamiento espectral del pulso laser (*self phase modulation* o automodulación de fase donde $n_{12} < 0$). Con esto, los pulsos de mayor intensidad empiezan a enfocarse más que los pulsos débiles, y genera una modulación de intensidad (*self-focusing* o autoenfocamiento, debido $n_{12} > 0$). Después, se puede utilizar un diafragma (modulación por pérdidas), o se cambia el acople entre el haz de bombeo y la cavidad donde se encuentra el medio no lineal (modulación por ganancia), todo esto para favorecer el amarre de modos.

Apéndice B. Filtros frecuenciales

Existen diferentes tipos de filtros frecuenciales, se pueden dividir de diferentes maneras, por su forma de hacerlos (*software* o *hardware*), por el espectro que delimitan (Pasa-bajas, pasa-altas, pasa-bandas, rechaza-bandas, pasa-todo).

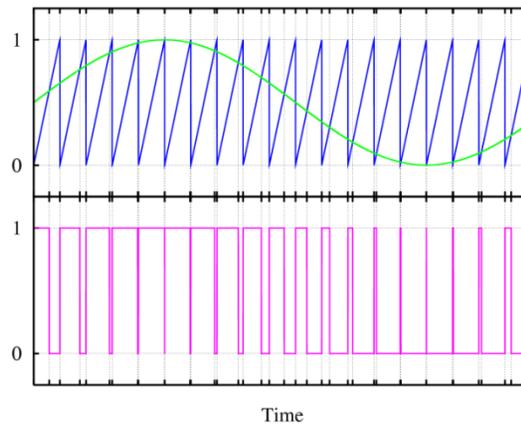
1. *Software* (llamados también filtros digitales) depende de si los términos generados a la salida del filtro (debido a un estímulo unitario) son finitos, infinitos o se truncaron a un número delimitado.
 - a. FIR (*finite impulse response*)
 - b. IIR (*infinite impulse response*)
 - c. TIIR (*Truncated Infinite Impulse Response*),
2. *Hardware* (filtros analógicos)
 - a. Filtros activos (usan generalmente un elemento que amplifique la señal o parte de ella)
 - b. Filtros pasivos (usa solamente elementos pasivos y atenúa la señal)

Cuando se escoge la respuesta a la frecuencia:

1. Pasa-bajas: Por su nombre, indica que dejara pasar las bajas frecuencias (desde 0 hasta la frecuencia de corte)
2. Pasa-altas: Este tipo de filtro deja pasar frecuencias altas y, atenúa o delimita las frecuencias bajas (obstruye las frecuencias de 0 a la frecuencia de corte)
3. Pasa-bandas: Este filtro es una combinación de un filtro pasa alta y un filtro pasa bajas, solo que la selección de las frecuencias de corte es importante, ya que para este filtro se debe de tener que la frecuencia de corte del filtro pasa altas debe ser menor a la del filtro pasa bajas, y poner los dos filtros en serie (la salida del primero se conecta a la entrada del segundo).
4. Rechaza-bandas (Notch): A diferencia del filtro pasa-bandas, en este, la frecuencia de corte del filtro pasa-bajas debe de ser menor a la frecuencia de corte del filtro pasa-altas, y deben de estar en paralelo (los dos filtros tienen la misma entrada, y las salidas se suman).
5. Pasa-todo: Este filtro no delimita ninguna frecuencia, lo que realiza este filtro es un cambio de fase. Por lo general este tipo de filtros solamente se implementan como filtro digital, o como filtro analógico activo, muy rara vez se diseñan como filtros pasivos.

Apéndice C. PWM (Pulse-width modulation) [7]

PWM o Modulación por ancho del pulso, es un tipo de modulación en la cual se hace la comparación de una señal “mensaje” con una señal portadora (por ello se ocupa normalmente un OPAMP en modo diferencial para generar esta señal), donde la frecuencia de la señal portadora es mucho mayor a frecuencia máxima del mensaje, y en relación de amplitud, la portadora tiene que ser mayor o igual a la máxima amplitud del mensaje.



*Fig APC.1. Ejemplo de la modulación PWM (normalizados)
Superior: Mensaje: señal senoidal. Portadora: diente de sierra
Inferior: PWM de la comparación de senoidal y diente de sierra*

Los parámetros de la salida de la comparación se delimitan con valores lógicos de “alto” y “bajo” (esto relacionando que señal es mayor, si el mensaje o la portadora), con ello tenemos una señal digital, donde varían los tiempos de alto y bajo en un periodo de la señal portadora.

Para la demodulación, se utiliza un filtro pasa-bajas (LPF), esto con la finalidad de hacer un promedio RMS, ya que si se quitan las frecuencias altas obtendríamos el mensaje original. Este tipo de modulación es comúnmente usada en el manejo de motores, servomotores y audio digital, ya que es más eficiente y tiene un bajo consumo de potencia (no se necesitan grandes disipadores de calor en la etapa de amplificación de potencia)

Apéndice D. CSV (comma-separated values)

Este formato de archivos está estructurado como un formato de texto, en el cual cada línea representa una fila, y cada celda esta diferenciada usando una coma ente cada grupo de información (puede ser texto ASCII y/o números). Las extensiones que comúnmente usa este formato son *.csv y *.txt.

Este formato es soportado por múltiples hojas de cálculo y su generación es relativamente sencilla, lo cual lo hace un formato versátil y útil para este tipo de proyectos.

Ej.

Aaa, bbb, ccc, ddd

Eee, fff, ggg, hhh

123, 456, 789, 000

Tabla vista en un editor de texto

Aaa	bbb	ccc	ddd
Eee	fff	ggg	hhh
123	456	789	000

Tabla vista en una hoja de cálculo

Apéndice E. Acoplamientos

Por el momento solo se explicaran 2 tipos de acoplamiento eléctrico:

- Acoplamiento de voltaje:

Este acoplamiento es mayormente usado para no cambiar la carga de la etapa anterior, por ello, se utiliza una resistencia de salida de la etapa 1 muy baja a comparación de la resistencia de entrada de la etapa 2. (Utilizado en el DAQ)

- Acoplamiento de potencia:

Este acoplamiento en particular, utiliza la Ley de Jacobi, donde se estipula que la resistencia interna de la fuente debe ser igual a la resistencia de la carga, para que exista una máxima transferencia de potencia entre la fuente y la carga, en este caso, la fuente puede ser tomada como una etapa anterior a la carga. (Utilizado en el VCO)

Apéndice F. Análisis monetario de los materiales usados

DAQ (con cable USB incluido)

Componente	Precio (mx\$)
Cable USB	\$23.00
Pic18f4550	\$100.00
MCP6281	\$10.00
Diodo sensor	\$12.00
Resistencias, capacitores, cristal de reloj, conector USB, PCB	\$60.00
Mano de obra	\$60.00
Total:	\$265.00

VCO (con cable USB incluido)

Componente	Precio (mx\$)
Cable USB	\$23.00
Pic18f4550	\$100.00
L293D	\$51.00
Resistencias, capacitores, cristal de reloj, conector USB, PCB y mano de obra.	\$60.00
Mano de obra	\$60.00
Total:	\$294.00

(en moneda nacional, se utilizó una tasa de cambio de us\$1 = mx\$13.05, diciembre 2012)

H

Interferómetro (óptica y base)

Componente	Precio (mx\$)
Pedazo de aluminio para la base e implementos	\$400.00
Bases ópticas	\$2,073.10
Divisor de haz de ThorLabs® modelo BP145B1	\$2,063.40
Espejos Comar® modelo 16MX16 (dos espejos)	\$ 1155.40
Actuador de línea de retardo	\$1200.00
Platina traslacional Comar® modelo 12XT40v	\$2,570.37
Mano de obra	\$1,500
Total	\$ 10,962.27

Suma de los totales

Concepto	Precio (mx\$)
DAQ	\$265.00
VCO	\$294.00
Interferómetro (óptica y base)	\$10,962.27
Total	\$11,521.27

Apéndice G. Desarrollo matemático de la correlación

A continuación se hará el desarrollo matemático para pasar de la Ecuación 1.5 a la Ecuación 1.7 [3].

$$S^{(2)} \approx A\beta \int_{-\infty}^{\infty} [I^2] dt \quad \text{Ecuación 1.4}$$

$$I = |E_0(t) + E_0(t + \tau)|^2 \quad \text{Ecuación 1.5}$$

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} |E_0^2(t) + E_0^2(t + \tau) + 2E_0(t)E_0(t + \tau)|^2 dt \quad \text{Ecuación 1.6}$$

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} |E_0^2(t) + E_0^2(t + \tau) + 2E_0^2(t)E_0^2(t + \tau)| |E_0^2(t) + E_0^2(t + \tau) + 2E_0^2(t)E_0^2(t + \tau)| dt$$

Haciendo la multiplicación se obtiene que:

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} |E_0^2(t)|^2 + E_0^2(t)E_0^{*2}(t + \tau) + 2E_0^2(t)E_0^*(t)E_0^*(t + \tau) + E_0^2(t + \tau)E_0^{*2}(t) + |E_0^2(t + \tau)|^2 + 2E_0^2(t + \tau)E_0^*(t)E_0^*(t + \tau) + 2E_0(t)E_0(t + \tau)E_0^{*2}(t) + 2E_0(t)E_0(t + \tau)E_0^{*2}(t + \tau) + 4|E_0(t)||E_0(t + \tau)| dt$$

Agrupando los términos obtenemos que:

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} \{(|E_0^2(t)|^2 + |E_0^2(t + \tau)|^2) + (4|E_0(t)||E_0(t + \tau)|) + (2E_0^2(t)E_0^*(t)E_0^*(t + \tau) + 2E_0^2(t + \tau)E_0^*(t)E_0^*(t + \tau) + 2E_0(t)E_0(t + \tau)E_0^{*2}(t) + 2E_0(t)E_0(t + \tau)E_0^{*2}(t + \tau)) + (E_0^2(t)E_0^{*2}(t + \tau) + E_0^2(t + \tau)E_0^{*2}(t))\} dt$$

$$S^{(2)}(\tau) \approx A\beta \int_{-\infty}^{\infty} \{(d) + (a) + (b) + (c)\} dt \quad \text{Ecuación 1.7}$$

Dónde:

$$4E_0^2(t)E_0^2(t + \tau) \quad \text{(a)}$$

$$4[E_0^3(t)E_0(t + \tau) + E_0(t)E_0^3(t + \tau)] \cos(\omega\tau) \quad \text{(b)}$$

$$2E_0^2(t)E_0^2(t + \tau) \cos(2\omega\tau) \quad \text{(c)}$$

$$E_0^4(t) + E_0^4(t + \tau) \quad \text{(d)}$$

J

Anexos

Anexo A. Firmware

(a) DAQ

```
#include <18F4550.h>
#DEVICE ADC=10
#fuses
HSPLL,MCLR,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN
#use delay(clock=48000000)

#define USB_HID_DEVICE FALSE
//deshabilitamos el uso de las directivas HID
#define USB_EP1_TX_ENABLE USB_ENABLE_BULK
//turn on EP1(EndPoint1) for IN bulk/interrupt transfers
#define USB_EP1_RX_ENABLE USB_ENABLE_BULK
//turn on EP1(EndPoint1) for OUT bulk/interrupt transfers
#define USB_EP1_TX_SIZE 8 //size to allocate for the tx endpoint 1 buffer
#define USB_EP1_RX_SIZE 2 //size to allocate for the rx endpoint 1 buffer

#include <pic18_usb.h> //Microchip PIC18Fxx5x Hardware layer for CCS's PIC USB driver
#include "USB_BULK.h" //Configuración del USB y los descriptores para este dispositivo
#include <usb.c> //handles usb setup tokens and get descriptor reports

int16 envia;
int c=0;

void main(void)
{
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    disable_interrupts(GLOBAL);

    usb_init();
    usb_task();
    usb_wait_for_enumeration();

    setup_port_a( ALL_ANALOG );
    #asm
    movlw 0b10001110; //justificacion_izquierda,4Tad,16Tosc
    iorwf 0xFC0,1 ; //direccion de ADCON2
    #endasm
    //setup_adc(ADC_CLOCK_DIV_32);
    set_adc_channel(0);
    while (usb_enumerated())
    {
        envia=read_adc();
        usb_put_packet(1,envia,8,USB_DTS_TOGGLE);
    }
}
```

(b) VCO

```
#include <18f4550.h>
#device adc=10;

#fuses
HSPLL,MCLR,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN
#use delay(clock=48000000)

#include <math.h>

int i=0;
long a=255;
long b=500;
void main()
{
    setup_adc_ports(AN0);
    setup_adc(ADC_CLOCK_DIV_2);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DIV_BY_4,128,4);
    setup_ccp1(CCP_PWM);
    set_pwm1_duty(a);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

    while(true)
    {
        i=0;
        for(i=0;i<255;i++)
        {
            a=191+63*sin((2*pi*i)/256);

            set_pwm1_duty(a);

            b=read_adc();
            delay_us(b);
        }
    }
}
```

L

Anexo B. Drivers

(a)DAQ

```
; VCO_Pablo.inf
; Copyright (c) 2010 libusb (GNU LGPL)
;
;-----
; libusb-win32 (Strings)
;-----
[Strings]
DeviceName = "VCO Pablo"
VendorName = ""
SourceName = "VCO Pablo Install Disk"
DeviceID = "VID_04D8&PID_000B"
DeviceGUID = "{43e5727e-a426-436b-b220-7d3224b3d6c5}"

;-----
; libusb-win32 (Version / ClassInstall32 / Manufacturer)
;-----
[Version]
Signature = "$Windows NT$"
Class = "libusb-win32 devices"
ClassGUID = {EB781AAF-9C70-4523-A5DF-642A87ECA567}
Provider = %VendorName%
CatalogFile = VCO_Pablo.cat
DriverVer = 10/05/2012, 1.0.0.0

[ClassInstall32]
Addreg = libusb_class_install_add_reg

[libusb_class_install_add_reg]
HKR,,0,"libusb-win32 devices"
HKR,,Icon,,-20

[Manufacturer]
%VendorName% = Devices, NT, NTAMD64, NTIA64

;-----
; libusb-win32 registry
;-----
[LIBUSB_WIN32_DEV.NT.HW]
DelReg = libusb_del_reg_hw
AddReg = libusb_add_reg_hw

[LIBUSB_WIN32_DEV.NTAMD64.HW]
DelReg = libusb_del_reg_hw
AddReg = libusb_add_reg_hw

[LIBUSB_WIN32_DEV.NTIA64.HW]
DelReg = libusb_del_reg_hw
AddReg = libusb_add_reg_hw

; Older versions of this .inf file installed filter drivers. They are not
; needed any more and must be removed
```

M

```

[libusb_del_reg_hw]
HKR,,LowerFilters
HKR,,UpperFilters

; libusb-win32 device properties
[libusb_add_reg_hw]
HKR,,SurpriseRemovalOK,0x00010001,1

; (Optional) the usb configuration value to select when this device
; is started. If this key does not exist the first config is selected.
;HKR,,InitialConfigValue,0x00010001,<your config value>

;-----
; libusb-win32 files
;-----
[SourceDisksNames]
1 = %SourceName%

[SourceDisksFiles.x86]
libusb0.sys = 1,x86
libusb0_x86.dll = 1,x86

[SourceDisksFiles.amd64]
libusb0.sys = 1,amd64
libusb0.dll = 1,amd64
libusb0_x86.dll = 1,x86

[SourceDisksFiles.ia64]
libusb0.sys = 1,ia64
libusb0.dll = 1,ia64
libusb0_x86.dll = 1,x86

[DestinationDirs]
libusb_files_sys = 10,system32\drivers
libusb_files_dll = 10,system32
libusb_files_dll_wow64 = 10,syswow64
libusb_files_dll_x86 = 10,system32

[libusb_files_sys]
libusb0.sys

[libusb_files_dll]
libusb0.dll

[libusb_files_dll_x86]
libusb0.dll, libusb0_x86.dll

[libusb_files_dll_wow64]
libusb0.dll, libusb0_x86.dll

[LIBUSB_WIN32_DEV.NT]
CopyFiles = libusb_files_sys, libusb_files_dll_x86

[LIBUSB_WIN32_DEV.NTAMD64]
CopyFiles = libusb_files_sys, libusb_files_dll, libusb_files_dll_wow64

```

N

```
[LIBUSB_WIN32_DEV.NTIA64]
CopyFiles = libusb_files_sys, libusb_files_dll, libusb_files_dll_wow64
```

```
;-----
; libusb-win32 service
;-----
```

```
[LIBUSB_WIN32_DEV.NT.Services]
AddService = libusb0, 0x00000002, libusb_add_service
```

```
[LIBUSB_WIN32_DEV.NTAMD64.Services]
AddService = libusb0, 0x00000002, libusb_add_service
```

```
[LIBUSB_WIN32_DEV.NTIA64.Services]
AddService = libusb0, 0x00000002, libusb_add_service
```

```
[libusb_add_service]
DisplayName = "libusb-win32 - Kernel Driver 10/05/2012 1.0.0.0"
ServiceType = 1
StartType = 3
ErrorControl = 0
ServiceBinary = %12%\libusb0.sys
```

```
;-----
; libusb-win32 devices
;-----
; Hardware IDs in a 'Devices' section can be installed by libusb-win32
; using usb_install_driver_np(), usb_install_driver_np_rundll() functions.
; Example:
; RUNDLL32 libusb0.dll,usb_install_driver_np_rundll <your_inf_file.inf>
;
```

```
[Devices]
%DeviceName% = LIBUSB_WIN32_DEV, USB\%DeviceID%
```

```
[Devices.NT]
%DeviceName% = LIBUSB_WIN32_DEV.NT, USB\%DeviceID%
```

```
[Devices.NTAMD64]
%DeviceName% = LIBUSB_WIN32_DEV.NTAMD64, USB\%DeviceID%
```

```
[Devices.NTIA64]
%DeviceName% = LIBUSB_WIN32_DEV.NTIA64, USB\%DeviceID%
```

Anexo C. Software

(a), (c) y (d) se encuentran en un *timer*

(b), (e) se encuentra en un *background worker*

la parte marcada de (f) se coloca en un botón

(a) Variables usadas

```
public static UsbDevice MyUsbDevice;
public static UsbDevice MyUsbDevice2;

#region Pid y Vid de la DAQ y VCO
public static UsbDeviceFinder MyUsbFinder = new UsbDeviceFinder(0x04D8, 0x000C); //DAQ
public static UsbDeviceFinder MyUsbFinder2 = new UsbDeviceFinder(0x04D8, 0x000B); //VCO
#endregion

#region Listas de datos
ArrayList Datos = new ArrayList(); //lista de datos RAW
ArrayList IdTrig = new ArrayList(); //lista de gatillo para el promedio
ArrayList listProm = new ArrayList(); // lista para grabar de
#endregion

#region Variables usadas
double lectura;
public string txtLoad1;
public string txtLoad2;

int a; //variable del tamaño del spam de la gráfica (aprox 7500=1 segundos) en el eje x
int a1;
int max;
int b = 0; //variable para llevar cuenta de cuantas matrices se han hecho
int n = 2; // variable de promedio
int ii = 0; //variable para "for" en promedio
double nC = 7.4; // variable de factor de calibracion
double frec = 13.38;
double vMax = 0;
double ancho = 2; //variable para medir el ancho del pulso
double tnPul = 0;
int prmmedio = 500; //número de trazas que va a promediar

#region Valores del trigger
bool primDat = true;
double trigg = 0;
int eMat = 0;
int eMat2 = 0;
#endregion

DateTime saveNow = DateTime.Now; //tiempo actual
string start;
string finish;
#endregion
```

(b) Obtención de datos (conexión con USB)

```
MyUsbDevice = UsbDevice.OpenUsbDevice(MyUsbFinder);
//
int[] lec = new int[n];
for (int i = 0; i < n; i++)
{
    lec[i] = 0;
}
ErrorCode ec = ErrorCode.None;

try
{
    if (MyUsbDevice == null) throw new Exception("DAQ Not Found.");
    IUsbDevice wholeUsbDevice = MyUsbDevice as IUsbDevice;

    if (!ReferenceEquals(wholeUsbDevice, null))
    {
        wholeUsbDevice.SetConfiguration(1);
        wholeUsbDevice.ClaimInterface(0);
    }
    UsbEndpointReader reader = MyUsbDevice.OpenEndpointReader(ReadEndpointID.Ep01);

    byte[] readBuffer = new byte[2];
    while (ec == ErrorCode.None)
    {
        lectura = 0;
        int bytesRead;
        ec = reader.Read(readBuffer, 5000, out bytesRead);
        if (ii == n)
        {
            ii = 0;
        }
        lec[ii] = Convert.ToInt16(readBuffer[0]);
        for (int i = 0; i < n; i++)
        {
            lectura += lec[i] / n;
        }
    }

    //región del gatillo (e)

    ii++;
    Datos.Add(lectura); //agrega a la lista un valor de promedio

    #region conteo de cuantas matrices lleva para tiempo real
    if ((Datos.Count % a1) == 0 && Datos.Count != 0)
    {
        int b1 = b;
        b = b1 + a1;
    }
    #endregion

    //para poder salir del ciclo y parar el background worker
    if (backgroundWorker1.CancellationPending)
```

Q

```

        {
            e.Cancel = true;
            return;
        }
        //
    }
}
catch (Exception ex)
{
    MessageBox.Show((ec != ErrorCode.None ? ec + ":" : String.Empty) + ex.Message);
    if (MyUsbDevice != null) { MyUsbDevice.Close(); }

    MyUsbDevice = null;
}
finally
{
}
}

```

(c) Llenado de las matrices

#region Llenado de matriz de tiempo real

for (int i = 1; i < a; i++) //metodo para llenar la matriz que se va a ver

```

{
//
    if (checkBox2.CheckState == CheckState.Unchecked) // sin trigger
    {
        if (Datos.Count > i + b)
        {
            aY[i] = Convert.ToDouble(Datos[i + b]) * 5 / 256;
        }
        else
        {
            if (Datos.Count > a)
            {
                aY[i] = Convert.ToDouble(Datos[i + b - a]) * 5 / 256;
            }
            else
            {
                aY[i] = 0;
            }
        }
    }
    if (checkBox2.CheckState == CheckState.Checked) //con trigger
    {
        if (Datos.Count > i + eMat & eMat + i - Convert.ToInt16(a * .478) > 0)
        {
            aY[i] = (Convert.ToDouble(Datos[eMat + i - Convert.ToInt16(a * .478)]) * 5 / 256);
        }
        else
        {
            if (Datos.Count > i + eMat2 & eMat2 != 0 & eMat + i - Convert.ToInt16(a * .478) > 0)
            {

```

R

```

        aY[i] = (Convert.ToDouble(Datos[eMat2 + i - Convert.ToInt16(a * .478)]) * 5 / 256);
    }
    else
    {
        aY[i] = 0;
    }
}

}
aX[i] = (i - a / 2) * nC; //eje x, que se expresa [-a/2, a/2]
}
#endregion

#region Llenado de matriz de promedio
if (checkBox2.CheckState == CheckState.Checked)
{
    if (checkBox3.CheckState == CheckState.Checked)
    {
        if (ldTrig.Count != 0)
        {
            int[] alfa = new int[prmedio];
            for (int i = 0; i < prmedio; i++) // genera una lista de pointers de donde estan los gatillos
            {
                if (ldTrig.Count > i + 1)
                {
                    alfa[i] = Convert.ToInt32(ldTrig[ldTrig.Count - 2 - i]);
                }
            }
            if (prmedio > 0) //proteccion para que no se divida entre 0 o numeros negativos
            {
                for (int i = 0; i < bY.Length; i++) //ciclo de recorrer matriz
                {
                    bY[i] = 0; //pone en ceros la matriz
                    for (int j = 0; j < prmedio; j++)//promedio
                    {
                        if (alfa[j] > 0 & Convert.ToDouble(Datos[alfa[j]]) > 0)
                        {
                            bY[i] += ((Convert.ToDouble(Datos[alfa[j]] - Convert.ToInt16(a * .478) + i)) * 5 / 256) / prmedio);
                        }
                    }
                }
            }
        }

        #region quitar DC
        for (int i = 1; i < bY.Length; i++)
        {
            bY[i] = bY[i] - bY[0];
        }
        #endregion
        //
    }
}
}
#endregion

```

S

(d) Tamaño del pulso

```
#region tamaño del pulso
tnPul = 0;
for (int i = 1; i < bY.Length; i++) //obtiene valor máximo de la matriz de promedio
{
    if (bY[i] > vMax)
    {
        vMax = bY[i];
    }
}
for (int i = 1; i < bY.Length; i++) //normalización
{
    bY[i] = bY[i] / vMax;
}
for (int i = 1; i < bY.Length; i++)
{
    if (bY[i] > vMax / ancho)
    {
        tnPul++;
    }
}
#endregion
```

(e) Trigger

```
#region Gatillo

if (lectura * 5 / 256 >= trigg & primDat == true & checkBox2.CheckState == CheckState.Checked)
//checa el nivel de disparo
{
    max = Datos.Count - eMat;
    eMat2 = eMat;
    eMat = Datos.Count;
    IdTrig.Add(eMat);
    primDat = false;
}
if (lectura * 5 / 256 < trigg & primDat == false & Datos.Count - eMat > a/2)
{
    primDat = true;
}
#endregion
```

T

(f) Encendido y apagado del Background worker

```
//se coloca en un botón
switch (backgroundWorker1.IsBusy)
{
    case false:
        StartKilling();
        break;
    case true:
        StopAllThatKilling();
        break;
    default:
        break;
}
//

private void StartKilling() // inicia background worker 1 y salva la hora en que se inició
{
    backgroundWorker1.RunWorkerAsync();
    start = Convert.ToString(saveNow);
}

private void StopAllThatKilling() // detiene background worker 1 y salva la hora de finalizado
{
    if (backgroundWorker1.WorkerSupportsCancellation == true)
    {
        backgroundWorker1.CancelAsync();
        finish = Convert.ToString(saveNow);
    }
}
```

U

Anexo D. Contribuciones

En el proceso de este trabajo se presentó el desarrollo en el siguiente evento:

SOMI XXVII, Octubre del 2012

SOMI XXVII
Congreso de Instrumentación
Culiacán, Sinaloa, del 29 al 31 de octubre de 2012

El Centro de Ciencias Aplicadas y Desarrollo Tecnológico
de la Universidad Nacional Autónoma de México y
el Instituto Tecnológico de Culiacán

Otorgan la presente

CONSTANCIA

a: Pablo Castro Marín, Jesús Garduño Mejía, Martha Rosete Aguilar,
Carlos Jesús Román Moreno, Roberto Ortega Martínez

Por haber presentado en el **SOMI XXVII** Congreso de Instrumentación su trabajo:

DISEÑO Y CONSTRUCCIÓN DE UN AUTOCORRELADOR
TPA DE FEMTOSEGUNDOS PORTÁTIL


Dr. José Manuel Saniger Blesa
PRESIDENTE DEL COMITÉ ORGANIZADOR



“Diseño y construcción de un Autocorrelador TPA de femtosegundos Portátil”



P. Castro-Marín, J. Garduño-Mejía, M. Rosete-Aguilar, C. J. Román-Moreno y R. Ortega-Martínez

Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), UNAM

Apdo. Postal 70-186, CP 04510. Coyoacán Cd. Universitaria, México D.F.

Corresponding authors email: jesus.garduño@ccadetunam.mx, pablo_mx@hotmail.com

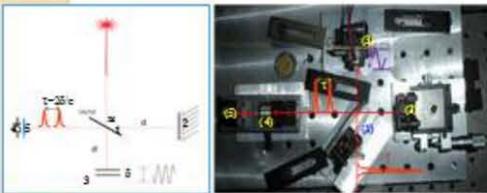
Resumen

En este trabajo se presenta el diseño y construcción de un autocorrelador de intensidad de segundo orden tipo TPA (*two photon absorption*), que sea de un tamaño portátil, de fácil alineación y calibración sencilla.

Diseño experimental

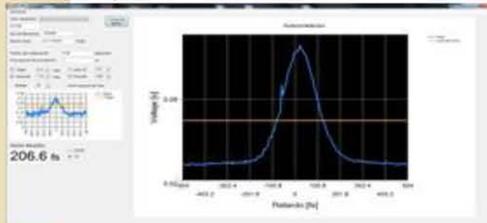
Se utilizó un interferómetro de Michelson para generar el autocorrelador de intensidad de segundo orden, el cual sigue el modelo:

$$A(\tau) = \int_{-\infty}^{\infty} I(t)I(t - \tau)dt$$



(1) Divisor de haz; (2) Brazo fijo; (3) Línea de Retardo variable; (4) Lente de Enfoque; (5) Fotodiodo

Se propuso un instrumento modular para un fácil manejo del instrumento, dividiéndolo en una línea de retardo (un generador de funciones con etapa de potencia y actuador), una tarjeta de adquisiciones, la base óptica y el software.



Software

El software se diseñó para poder obtener un promedio de señales (en este caso son periódicas debido a la etapa de retardo), esto por medio de un gatillo (generado por software) y con ello eliminar ruido no periódico (como el ruido blanco y el ruido térmico).



DAQ

Se hizo el sistema de adquisición de datos para que pudiera ser conectado a la computadora por medio de la comunicación USB, permitiendo así que solo dependa de controladores para su aplicación en diferentes sistemas operativos.

Comparación de sistemas (Nuevo vs. Anterior)

Línea de retardo

Interferómetro



Sensor y osciloscopio

Control de línea de retardo



Conclusiones

En este proyecto se desarrolló un instrumento autocontenido capaz de medir pulsos láser de femtosegundos. El sistema está basado en un Autocorrelador de Intensidad de segundo orden, colineal. El sistema incluye línea de retardo controlada con un microprocesador, tarjeta de adquisición de datos, interfaz de usuario para diferentes sistemas operativos, conectividad por USB, sistema de detección por absorción de dos fotones (TPA), despliegue en tiempo real de la traza de autocorrelación y la medida del pulso en la misma interfaz.

Referencias

- [1] R. Trebino and E. Zeek, "The autocorrelation, the spectrum, and phase retrieval". In: R. Trebino, "Frequency-Resolved Optical Gating". USA: Kluwer Academic Publishers, 2000. pp. 61-99.
- [2] D.T. Reid, M. Padgett, C. McGowan, W.E. Sleaz, and W. Sibbet, "Light-emitting diodes as measurement devices for femtosecond laser pulses". OPTICS LETTERS February 15, 1997/ vol.22, No.4, pp.233-235

Agradecimientos

Se agradece el financiamiento y apoyo en becas a la DGAPA-UNAM. PAPIIT proyectos: IN113809, IB101212, IACOD-11100611, IN104112, BID-UNAM 98-B3-C-DGP-L0034-1077, CONACYT proyecto CB 131746.

Índice de Figuras

Figura 1.1 Autocorrelador de segundo orden (TPA).....	4
Figura 1.2 Absorción de dos fotones, S0 banda de valencia y Sn banda de conducción	5
<i>Figura 1.3</i> Autocorrelación de segundo orden, franjas interferométricas y su comparación con la autocorrelación de intensidad.....	7
Figura 1.4 Autocorrelación de intensidad y su comparación con el pulso generado por el cual se obtiene su autocorrelación.	8
<i>Figura 2.1</i> Diagrama por bloques del instrumento.....	9
Figura 3.1 USB 1.0, 1.1 y 2.0	11
Figura 3.2 USB 3.0.....	12
Figura 3.3 Gráfica de un ejemplo del NRZI	15
Figura 3.4 Explicación por capas de la protocolización del USB2.0.....	16
Figura 3.5 Diagrama funcional de los pipes y endpoints en un USB.....	19
<i>Figura 4.1</i> Sistema de enfoque de un CD	23
<i>Figura 4.2.</i> Sistema en base a la bocina	24
Figura 4.3 Actuador Electro-mecánico en uso.....	25
Figura 4.4 Prototipo del oscilador controlado por voltaje (VCO).....	26
Figura 4.5 Diagrama lógico de la línea de retardo.....	26
Figura 4.6 Esquema de conexiones del oscilador controlado por voltaje (VCO)...	27
Figura 4.7 Sensores usados.....	29
Figura 4.8 Esquema de conexiones del sensor y tratamiento de señal	30
Figura 4.9 Base Prototipo.....	31
Figura 4.10 Prototipo de la tarjeta de adquisición de datos.....	33
Figura 4.11 Diseño lógico de la tarjeta de adquisición de datos (DAQ)	34

Figura 4.12 Esquema lógico de cómo se conforma el firmware de la DAQ	35
Figura 4.13 Esquema lógico de cómo está construido el Firmware del VCO.....	36
Figura 4.14 Dispositivo óptico ultrarrápido bajo prueba (Rojo) y base del sensor (Azul).....	37
Figura 4.15 Snapshot de una versión de prueba del software en uso, tomando una traza de autocorrelación.....	39
Figura 4.16 Parte de control del software.....	40
Figura 4.17 Menu strip	41
Figura 4.18 Gráfica principal	41
Figura 4.19 Tornillo micrométrico.....	43
Figura 4.20 Obtención de calibración visto en el software	44
Figura 5.1 Traza asimétrica mal calibrada	45
Figura 5.2 Pulso ultracorto usando promedio de 500 trazas, mostrando el ancho del pulso en $1/e$	47
Figura 5.3 Pulso ultracorto usando promedio de 500 trazas, mostrando el ancho del pulso en FWHM.....	48
Figura 5.4 Datos procesados en Matlab ®.....	49

Índice de Tablas

Tabla 3-1: <i>USB A y USB B</i>	11
Tabla 3-2: <i>Mini-A, Mini-B, Micro-AB y Micro-B</i>	12
Tabla 3-3: <i>USB3.0 A</i>	12
Tabla 3-4: <i>USB3.0 B y Bpowered</i>	13
Tabla 3-5: <i>Características según el American wire gauge</i>	14
Tabla 3-6: <i>Posibles valores del Packet ID según el tipo de paquete enviado</i>	17
Tabla 3-7 Como se compone la trama “ <i>token</i> ”	18
Tabla 3-8 Como se compone la trama “ <i>Data</i> ”	18
Tabla 3-9 Como se compone la trama “ <i>Handshake</i> ”	18

Bibliografía

- [1] F. G. Stremler, Introduction to communication systems 3rd ed, Addison-Wesley Pub., 1992.
- [2] J.-C. Diels y W. Rudolph, Ultrashort Laser Pulses Phenomena 2nd Ed., N.Y.: Academic Press, 2006.
- [3] R. Trebino, Frequency-Resolved Optical Gating: the measurement of ultrashort laser pulses, Massachusetts: Kluwer Academic Publishers, 2000.
- [4] R. W. Boyd, Nonlinear Optics, Third Edition, Orlando: Academic Press, 2008.
- [5] D. T. Reid, W. Sibbett, J. M. Dudley, L. P. Barry, B. Thomsen y J. D. Harvey, «Commercial Semiconductor devices for two photon absorption auto-correlation of ultrashort light pulses,» *Engineering & laboratory notes*, pp. 8142-8144, 1994.
- [6] F. C. S. Estrada , Propagación de pulsos ultracortos en sistemas refractivos, Posgrado de ingeniería, 2012.
- [7] MIT, «The Center for bits and atoms,» [En línea]. Available: <http://fab.cba.mit.edu/classes/MIT/961.04/topics/pwm.pdf>.
- [8] S. A. Akhmanov, V. A. Vysloukh y A. S. Chirkin, Optics of Femtosecond Laser Pulses, American Institute of Physics, 1992.
- [9] D. T. Reid, C. McGowan, W. W. Sleat y W. Sibbett, «Light-Emitting Diodes as measurement devices for femtosecond laser pulses,» *OPTICS LETTERS*, pp.

AA

233-235, 1997.

[10] A. E. Siegman, Lasers, University Science Books, 1986.

[11] G. Steinmeyer, «A review of ultra fast optics and optoelectronics,» *Journal of optics A: pure and applied optics*, pp. R1-R15, 2003.

[12] USB Implementers Forum, Inc, «Universal Serial Bus,» [En línea]. Available: <http://www.usb.org/developers/docs/>.

[13] «Handbook of hardware schemes, cables layouts and connectors,» [En línea]. Available: <http://pinoutsguide.com/>.

[14] «USBlyer,» [En línea]. Available: <http://www.usblyzer.com/brief-usb-overview-and-history.htm>.

[15] USB G-Jack, «G-Jack,» [En línea]. Available: <http://www.g-jack.com/apdf/USB3.0-cable.pdf>.

BB

[16] D. C. O'Shea, W. R. Calen y W. T. Rhode, Introduction to lasers and their applications, Addison-Wesley, 1977.

Hojas de especificaciones

- Fotodiodo

http://sales.hamamatsu.com/assets/pdf/parts_G/g11115_etc_kgpd1002e02.pdf

- Microcontrolador Pic 18f4550

<http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>

- Puente H L293D

<http://www.ti.com/lit/ds/symlink/l293d.pdf>

- Op-Amp MCP6281

<http://ww1.microchip.com/downloads/en/devicedoc/21811d.pdf>

- Divisor de haz de película ThorLabs modelo BP145B1

<http://www.thorlabs.com/thorproduct.cfm?partnumber=BP145B1>

- Platina traslacional Comar modelo 12XT40v

http://www.comaroptics.com/pdfs/Comar_Euro_price_list.pdf