

## **INFORME FINAL DE PROYECTO**

Anteproyecto:	<b>Controlador de un actuador electrodinámico aplicado a interferometría dinámica</b>	
Autores:	Roman, Santiago	93947
	Verstraeten, Federico	92892
	Zarza, Ezequiel M.	93883
Turno de TP:	1 (Martes 19 a 22hs)	
Docente Guía:	-	

Observaciones generales	

Aceptación del docente:	
Fecha	

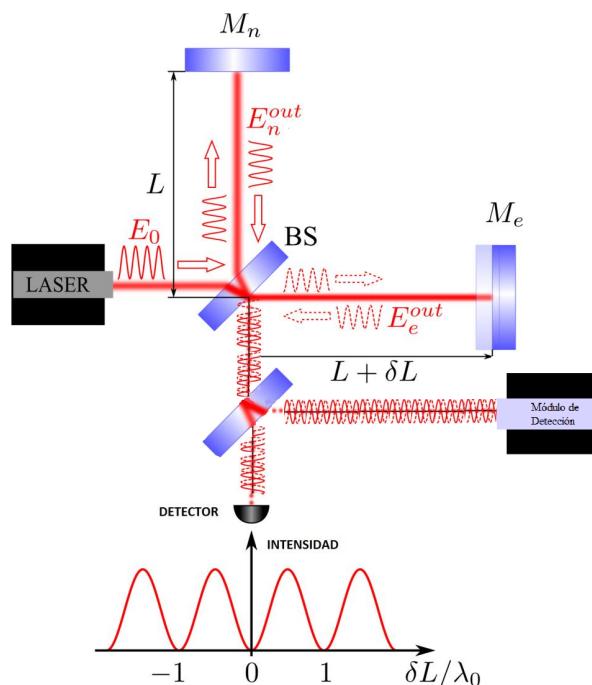
# Índice

1. Objetivos	1
2. Descripción	2
3. Especificaciones	3
4. Periféricos Principales	4
5. Diagrama de bloques (hardware)	7
6. Diagrama de flujo (firmware)	8
7. Esquemático	9
8. Listado de componentes y costos estimados	10
9. Resultados	10
10. Conclusiones	13

## 1. Objetivos

Un actuador es un dispositivo capaz de transformar energía (eléctrica para nuestro caso de interés) en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Al mismo tiempo, existe un controlador que se encarga de enviar órdenes al actuador con las cuales se activa un elemento final de control. Un ejemplo de actuador utilizado frecuentemente son los actuadores piezoelectricos, los cuales producen una deformación mecánica sobre el material piezoelectrico a partir de la aplicación de un campo eléctrico externo y viceversa. Estos tipos de actuadores pueden funcionar como por ejemplo, microposicionadores.

En este proyecto se busca diseñar un actuador que se utilizará en un esquema interferométrico, más precisamente en un interferómetro de Michelson. Este tipo de esquema se describe en la figura 1.1.



**Figura 1.1** – Interferómetro de Michelson

Mediante un divisor de haz (*BS*), se superponen dos campos proveniente de un fuente monocromática y coherente (*laser*). El campo que se desvía hacia la rama superior se refleja mediante un espejo fijo ( $M_n$ ) mientras que el campo que se desvía hacia la derecha se refleja en un espejo móvil ( $M_e$ ). De esta manera, los campos reflejados se dirigen hacia la rama inferior, donde se superponen en un fotodetector por medio del cual se puede registrar la intensidad de estos dos campos superpuestos. El modelo de esta intensidad es el siguiente:

$$I(t) = A + B \cos \left[ \frac{4\pi}{\lambda_0} (L_1 - (L_2 + \delta L(t))) \right] \quad (1.1)$$

donde  $L_1$ ,  $L_2$  y  $\delta L$  son las que se muestran en la figura 1.1, y  $\lambda_0$  es la longitud de onda del haz incidente.

De esta manera, con la ayuda del fotodetector se puede medir la intensidad producida por la diferencia de camino óptico ( $\delta L$ ). Si las ramas del interferómetro tienen la misma distancia (es decir,  $L_1 = L_2 = L$ ), la intensidad registrada sólo depende del movimiento del espejo móvil  $\delta L$ .

Por lo tanto: Existe una variación de la intensidad detectada producto del desplazamiento del espejo móvil. Si conocemos  $A$ ,  $B$  y la longitud de onda de la fuente es posible recuperar la información del desplazamiento producido por  $M_e$  midiendo la intensidad en el detector. Claramente este proceso también puede aplicarse de manera inversa, esto es: si podemos controlar con una resolución elevada la posición del espejo podemos determinar la longitud de onda de la fuente lumínica y los parámetros  $A$  y  $B$ .

## 2. Descripción

Retomando la ecuación 1.1, si en lugar de colocar un espejo fijo ( $M_n$ ) colocamos un espejo móvil, la expresión 1.1 ahora se puede escribir como:

$$I(t) = A + B \cos[\delta L(t) + \Delta\phi] \quad (2.1)$$

donde  $\Delta\phi$  corresponde a la variación de camino óptico introducido por el nuevo espejo móvil. Si logramos controlar el movimiento del espejo móvil, es posible recuperar la información de fase  $\delta L(t)$ . Acorde a [1],  $\Delta\phi$  puede expresarse como:

$$\Delta\phi(t) = \frac{4\pi}{\lambda_0} d_M(t) \quad (2.2)$$

Dada una determinada posición inicial del espejo supongamos que introducimos "saltos" de desplazamiento los cuales conllevan obtener "saltos" de fase  $\Delta\phi_1 = 0$ ;  $\Delta\phi_2 = \pi/2$ ,  $\Delta\phi_3 = \pi$ ;  $\Delta\phi_4 = 3\pi/4$ . Las intensidades obtenidas para estos saltos son:

$$I_1 = A + B \cos[\delta L(t) + 0] \quad (2.3)$$

$$I_2 = A + B \cos[\delta L(t) + \frac{\pi}{2}] \quad (2.4)$$

$$I_3 = A + B \cos[\delta L(t) + \pi] \quad (2.5)$$

$$I_4 = A + B \cos[\delta L(t) + \frac{3\pi}{2}] \quad (2.6)$$

Es posible recuperar  $\delta L(t)$  como:

$$\delta L(t) = \arctan \left[ \frac{I_2(t) - I_4(t)}{I_1(t) - I_3(t)} \right] \quad (2.7)$$

Esto es posible gracias al hecho de poder controlar la fase introducida por el espejo móvil  $M_n$ . De esta forma es posible obtener  $\delta L(t)$  introduciendo por otro tipo de desfasaje (ya sean discretos o continuos). Este tipo de técnicas se denominan *PSI* (*Phase Shifting Interferometry*).

Por otro lado, como ya hemos mencionado, es posible determinar los parámetros del interferograma ( $A$  y  $B$ ) para una figura de interferencia arbitraria, utilizando el movimiento del espejo móvil. De esta forma es posible compensar el interferómetro

cuando el mismo se vé afectado por fluctuaciones aleatorias de intensidad debido a distintos fenómenos (turbulencias, vibraciones, etc.).

En general se utilizan transductores piezoeléctricos, donde además es necesario abrir su sistema de control y tienen un costo aproximado de cientos de dólares.

En este caso particular, dado que se utiliza un esquema interferométrico como el de Michelson, puede no ser necesario contar con transductores tan precisos como los piezoeléctricos, pudiéndose optar por alternativas que resulten más económicas. De esta manera, en el presente trabajo nos planteamos obtener un método de transducción eléctrica-mecánica a partir de dispositivos de uso común cuyo sistema de control incluya tanto la posibilidad de definir el tipo de desplazamiento que se desea utilizar por medio de Matlab, como también poder definirlo desde el microcontrolador, mediante un display y pulsadores.

Particularmente, se buscará generar el desplazamiento de un espejo conectándolo a un parlante o altavoz, el cual podría ser controlado por Matlab mediante la presencia de un microcontrolador como intermediario entre estos, como también desde el microcontrolador directamente. Esta dualidad del microcontrolador para trabajar desde él mismo o como intermediario le brinda al sistema mayor "portabilidad".

### 3. Especificaciones

En cuanto a lo que se refiere al hardware propio del proyecto, es decir el parlante que se encarga de transformar energía eléctrica y producir un desplazamiento físico; tras realizar diferentes pruebas con varios modelos de parlantes elegimos aquel que mostró una mayor respuesta lineal en conjunto con un desplazamiento en una sola dirección. Esto último es de especial importancia ya que existe la posibilidad de que algunos parlantes generen un desplazamiento oblicuo.

El elemento central de control del actuador electrodinámico es el microcontrolador ATMega32. La elección de este tuvo como premisa fundamental de que disponga de la suficiente memoria volátil (RAM) y la necesaria para alojar el programa que ejecutará finalmente (ROM).

Estas consideraciones se deben a que en este trabajo utilizamos memoria ROM para almacenar las Lookup Tables de la función seno pre cargada (es decir, se encuentran guardadas una determinada cantidad de muestras de cada función que en total forman un período de las mismas). Por otro lado, la memoria RAM se utiliza principalmente para almacenar las muestras correspondientes a la señal generada en Matlab, la cual se exporta al microcontrolador por medio del protocolo de comunicación serie USART. Desde ya, dado que solamente vamos a utilizar señales periódicas, es posible generar una señal continua a lo largo de un período de tiempo contando únicamente con las muestras correspondientes a un período.

Otro punto importante de la elección del microcontrolador en cuestión es que debe contar con la cantidad suficiente pines I/O para poder relacionarse con los distintos periféricos externos que conforman a este proyecto. A continuación mostramos una tabla con la cantidad de pines que requiere cada parte del proyecto:

Elemento	# pines
Parlante (DAC)	8
Display	7
Pulsadores	4
Matlab/USART ( $T_X$ y $R_X$ )	2
Módulo de detección (ADC)	1
Cristal	2
Total	24

**Tabla 3.1** – Análisis de la cantidad de pines I/O requeridos para cada parte del programa.

La versión final del proyecto se presenta ensamblada sobre una placa experimental. La fuente de alimentación es provista por una placa conversora USB/Serie, la cual cuenta con salidas de 5 V y 3,3 V. Esto permite evitar el uso de una fuente separada para poder alimentar al microcontrolador.

Al iniciar el programa se muestra el menú principal en el display LCD, por el cual puede desplazarse utilizando los botones *Arriba* y *Abajo*. Para seleccionar alguna de las opciones se utiliza el botón *Enter* y para regresar al menú principal el restante, el cual ejecuta una interrupción externa y regresa al menú principal.

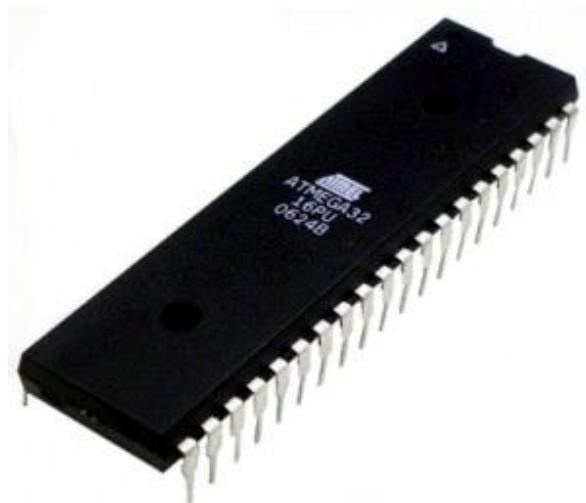
Por último, se utiliza un módulo de detección de intensidad, el cual permite medir este parámetro y mostrar un promedio del mismo en el display. El haz de luz al cual se calcula la intensidad sale de colocar un divisor de haz extra en la última rama del interferómetro (ver imagen 1.1), a partir de la cual queda colocado en una sección el detector que va conectado al osciloscopio que ya se mencionó anteriormente, y en la otra el módulo de detección que esta conectado al microcontrolador.

## 4. Periféricos Principales

- Microcontrolador ATMega32: Es el periférico central de este trabajo. Toda la información entra o sale por el mismo y es el encargado de interconectar al resto de los periféricos entre sí. Algunos de los motivos por los cuales se eligió este microcontrolador entre la diferente variedad que ofrece Atmel son:
  - Posee la cantidad de pines suficiente para poder conectar el resto de los periféricos.
  - La cantidad de memoria RAM y ROM que posee resulta más que suficiente para poder almacenar una tabla de las funciones pre cargadas y también para guardar en memoria volátil las muestras de una función periódica que se envía desde Matlab.
  - Cuenta con el protocolo USART.
  - Tiene un conversor analógico-digital integrado, el cual se utiliza para poder medir la intensidad haciendo uso del módulo de detección. Esto se debe a

que dicho módulo transforma la intensidad del haz que incide sobre el mismo en un valor de tensión (señal analógica).

- Al ser frecuentemente utilizado en sistemas embebidos, es posible encontrar en internet información y documentación sobre cómo utilizar cada una de las funciones que posee, en contraposición a otros tipos de microcontroladores.



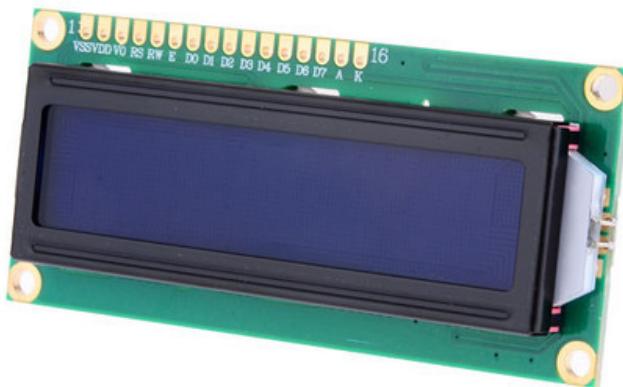
**Figura 4.1 – Microcontrolador ATmega32**

- Placa conversora USB a Serie, necesaria para poder conectar a una computadora va puerto USB y que la computadora sea capaz de reconocer al dispositivo como un puerto serie virtual. De esta manera es posible utilizar las funciones de Matlab que permiten enviar datos a través de un puerto serie. Además estas placas incluyen conexiones para alimentar al microcontrolador con lo cual se lo podría energizar solamente con la conexión a una computadora sin necesidad de utilizar un cable adicional. En el mercado argentino se encuentran este tipo de placas basadas en dos integrados distintos: Cp2102 y CH340. De estos dos se utilizó una placa basada en Cp2102 ya que era la de menor costo.



**Figura 4.2** – Ejemplo de placa conversora usb a serie

- Red R2R: Se trata de una implementación sencilla de un conversor digital-analógico (DAC). Tal cual indica su nombre, R2R (también conocido como escalaera de resistencias), se trata de un circuito electrónico formado por resistencias alternando dos valores posibles, donde un valor debe ser el doble del otro. Tiene como ventajas su fácil implementación, su conversión eficiente y la independencia que presenta en comparación con DACs integrados, donde estos últimos necesitan varias fuentes de alimentación, donde en algunos casos dichos valores de alimentación son atípicos.
- Display LCD: Es necesario para poder visualizar las opciones a elegir, para definir el tipo de desplazamiento desde el microcontrolador.

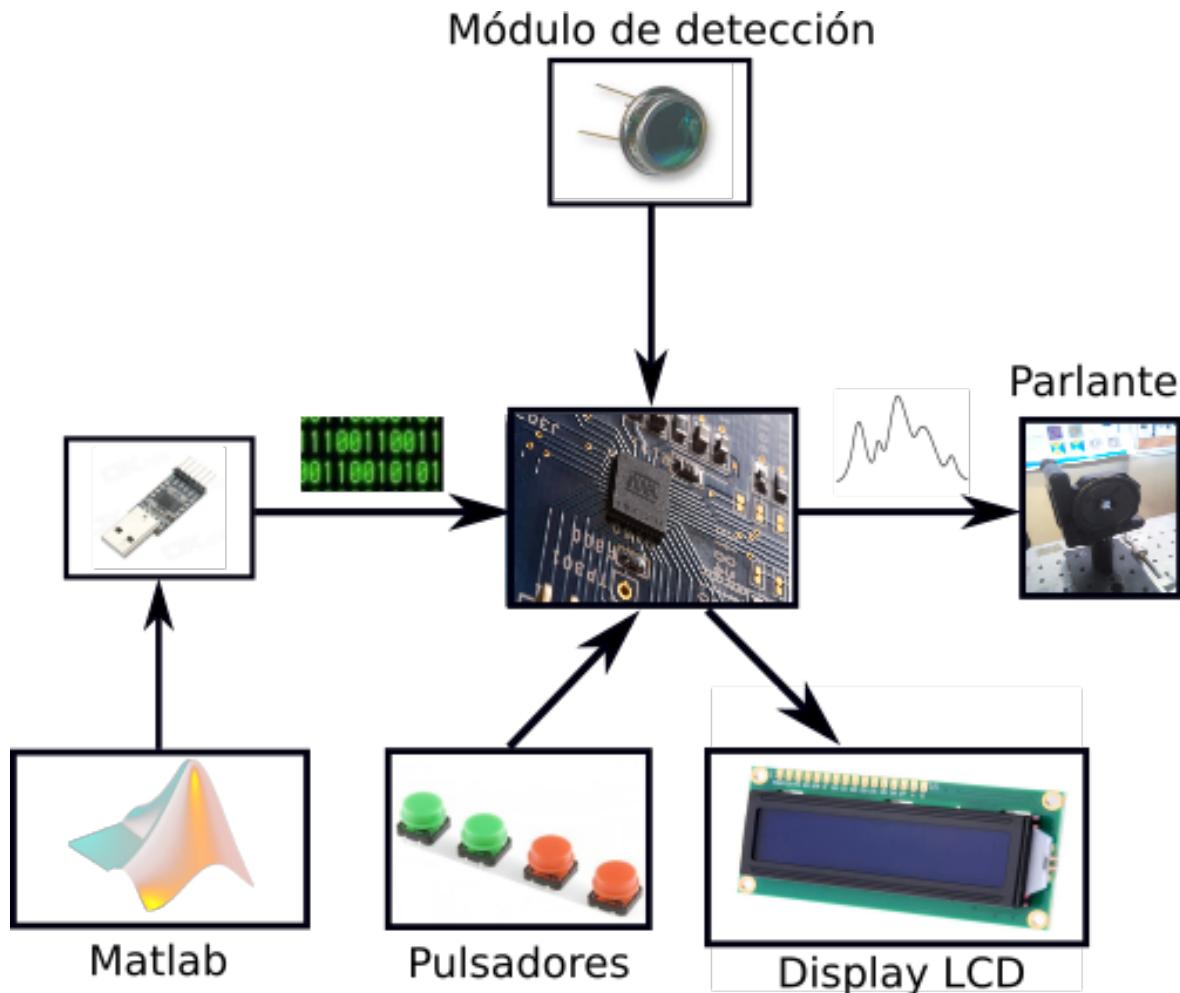


**Figura 4.3 – Ejemplo Display LCD 16x2**

- Pulsadores: Son necesarios para seleccionar el tipo de desplazamiento a efectuar, donde las opciones serán visualizadas en el display.
- Parlante o Altavoz: Necesario para poder generar el desplazamiento en el espejo. Será necesario evaluar distintos tipos de parlantes para entender cuál resulta útil para cumplir nuestros objetivos.

## 5. Diagrama de bloques (hardware)

A continuación se muestra el diagrama de bloques del sistema utilizado. El microcontrolador puede recibir o establecer el tipo de desplazamiento que se desea aplicar sobre el parlante. Dicho modo de desplazamiento puede ser recibido mediante la interfaz USB/Serie que vincula al microcontrolador con la PC, como también ser seleccionado a través de un teclado (pulsadores) conectado al microcontrolador, donde el usuario elegirá el tipo de desplazamiento a realizar mediante las opciones vistas en el display, y será el microcontrolador el encargado de efectuarlas, a diferencia de la otra forma, donde solo actuará como intermediario.



**Figura 5.1** – Diagrama de bloques que muestra cómo interactúan los diferentes dispositivos que conforman el proyecto.

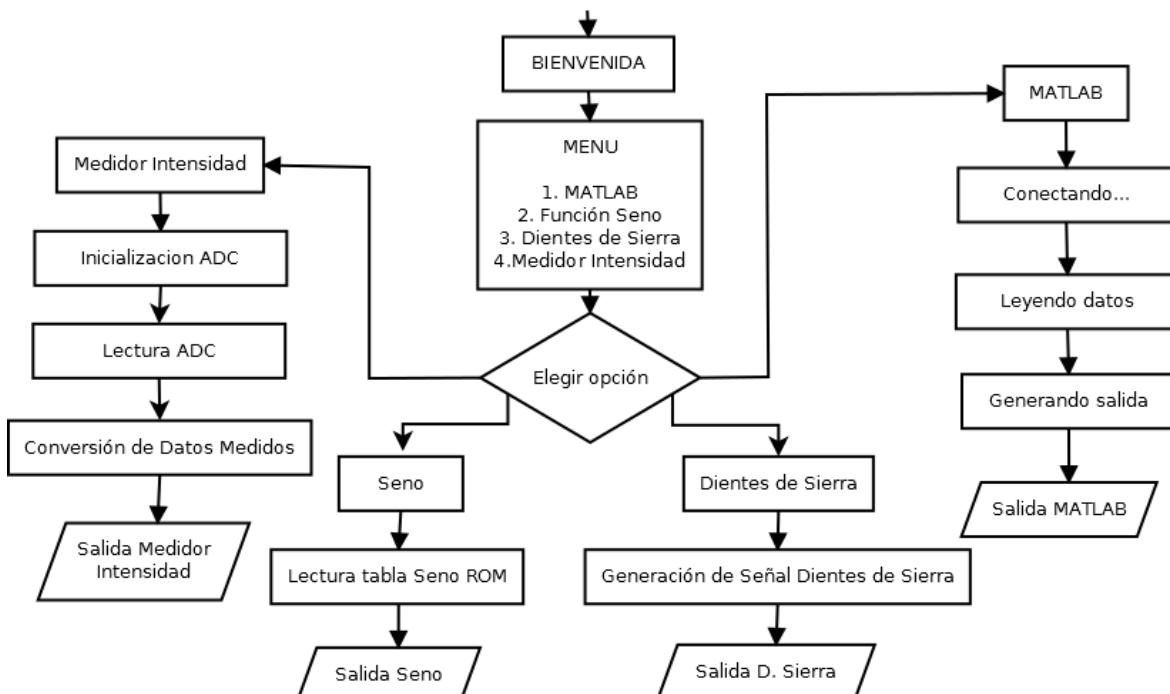
## 6. Diagrama de flujo (firmware)

A continuación se presenta el Diagrama de flujo correspondiente al Menú principal del proyecto. A través de este, que funciona como interfaz con el Usuario, se puede seleccionar los distintos modos de operación que puede ejecutar el microcontrolador.

- Seno: generación de una señal senoidal precargada en memoria del programa (ROM), con una Frecuencia y Amplitud prefijadas.
- Dientes de Sierra: generación de una señal Dientes de Sierra, a partir de una función implementada en Assembly, con una Frecuencia y Valor Pico prefijados.
- Matlab: el programa abre un canal de comunicación de transmisión Serie entre Matlab y el Microcontrolador, donde este último recibe datos, los almacena en RAM, procesa y genera la salida con los datos acondicionados entre 0 y 255 valores.
- Intensidad: Este modo de operación permite al Microcontrolador medir la intensidad lumínica de las ramas del Interferómetro. Esto se hace mediante un fotodiodo

en serie con una resistencia, la cual esta conectada al conversor Analogico-Digital interno del Microcontrolador.

Las señales se generaran de forma continuada hasta que el usuario detiene el proceso mediante una Interrupción Externa activada al presionar un pulsador, la cual salta el programa al Inicio del Menú. Dichas señales se generan en un Puerto establecido como salida, el cual tiene conectado un conversor Digital-Analógico (DAC).



**Figura 6.1** – Diagrama de flujo del menú principal.

## 7. Esquemático

A continuación se muestra el esquemático realizado en Proteus.

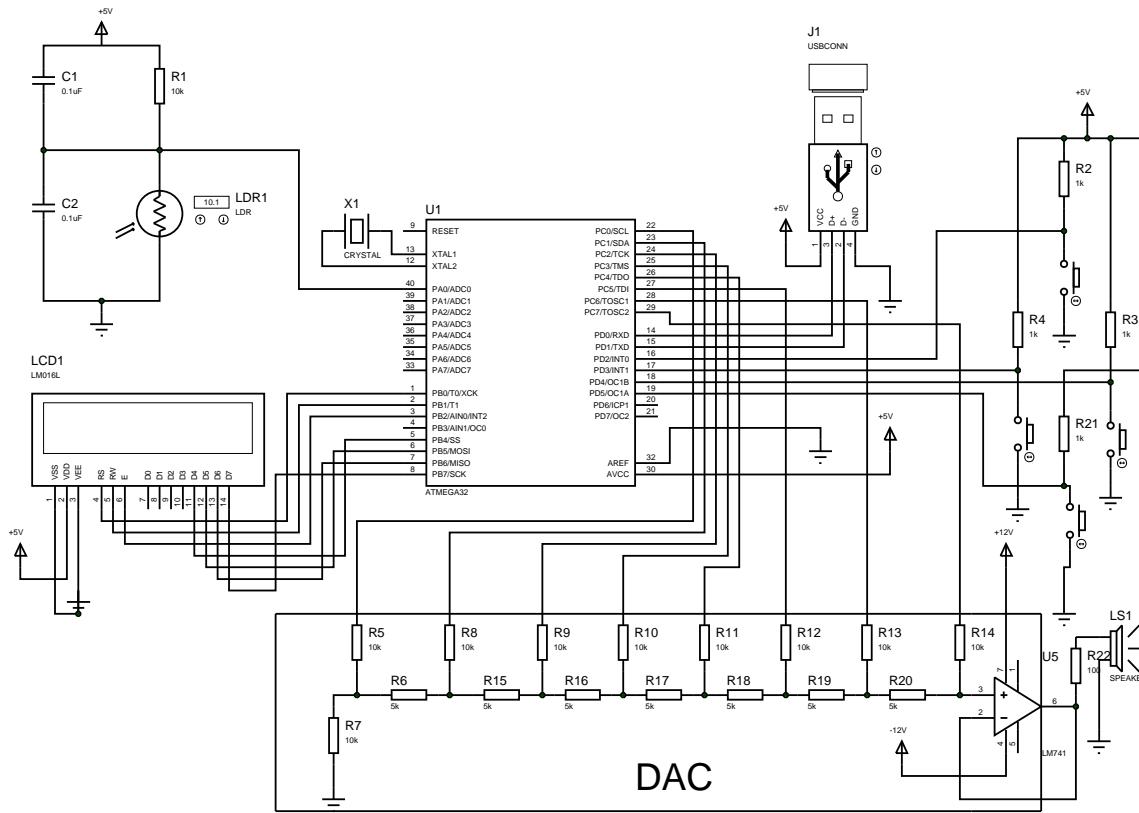


Figura 7.1 – Esquemático del proyecto en Proteus.

## 8. Listado de componentes y costos estimados

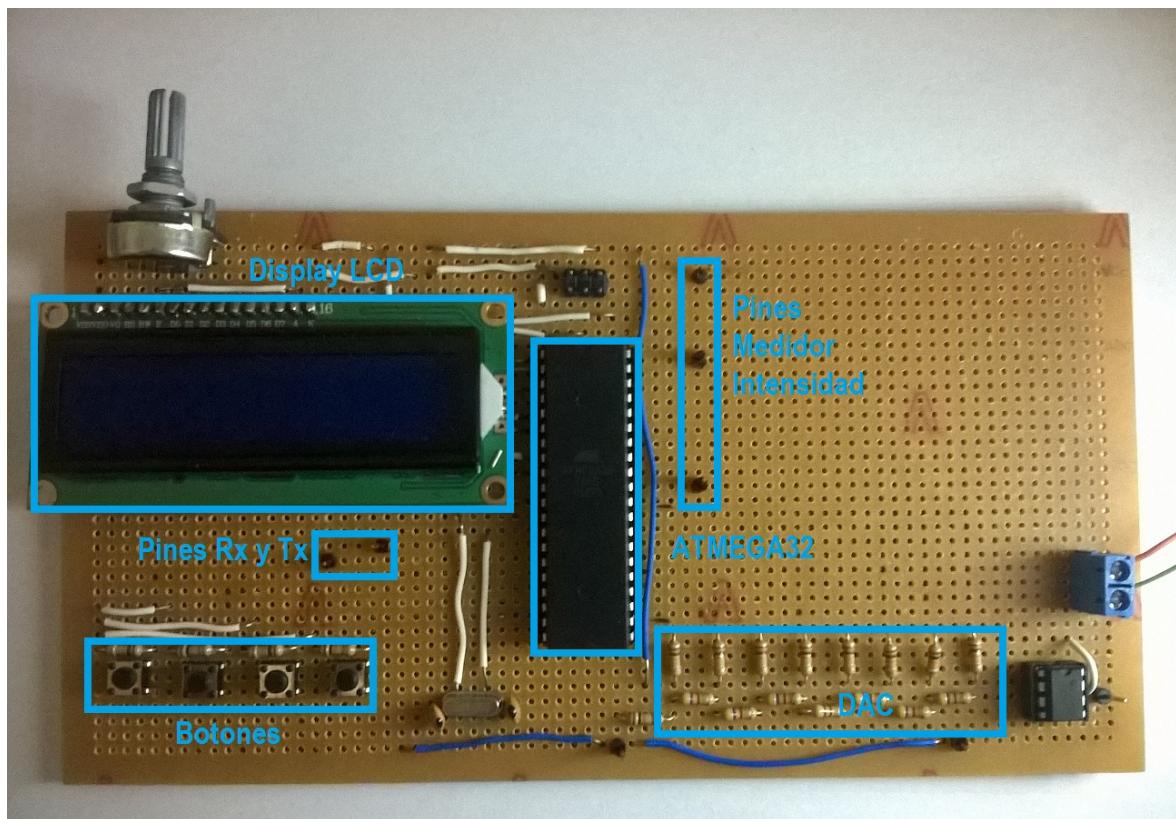
Componente	Costo estimado
Microcontrolador ATMega32	\$ 225
Placa experimental	\$ 150
Soporte	\$ -
Placa USB a Serie	\$ 120
Display LCD	\$ 120
Parlante	\$ 50
4x Botones	\$ 20
Fotodiodo	\$ 5
16x Resistencias <sup>1</sup>	\$ 10
Cristal 16 MHz	\$ 6
Total	≈ \$ 706

Tabla 8.1 – Costos estimados para la realización del proyecto.

## 9. Resultados

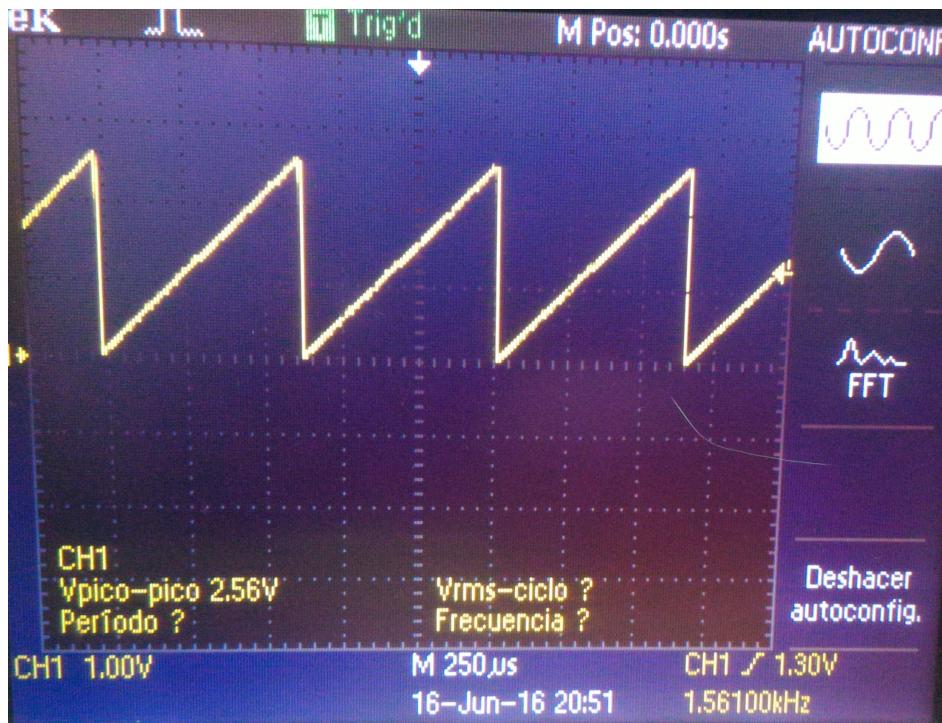
Tras haber verificado el correcto funcionamiento de cada parte que conforma el proyecto se realizó una implementación del mismo en una placa experimental. En la

figura 9.1 se puede ver el resultado final de la misma.

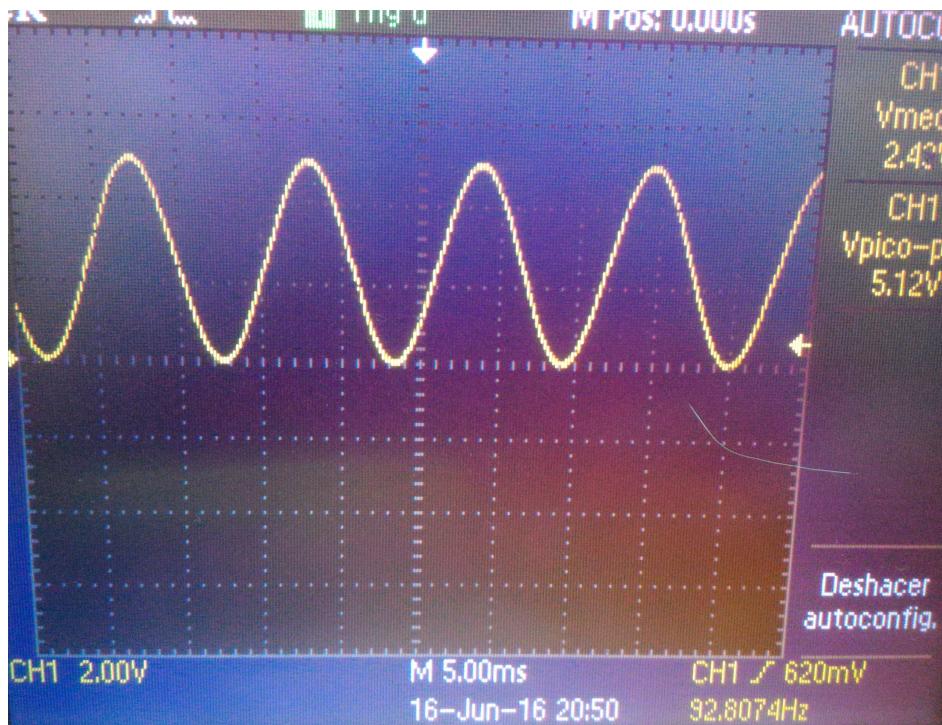


**Figura 9.1** – Versión final del proyecto implementado en una placa experimental.

Se realizaron mediciones en el osciloscopio de las salidas que se obtienen para las funciones seno y una rampa y en las figuras 9.2 y 9.3 se muestran capturas de pantalla del osciloscopio para cada señal.



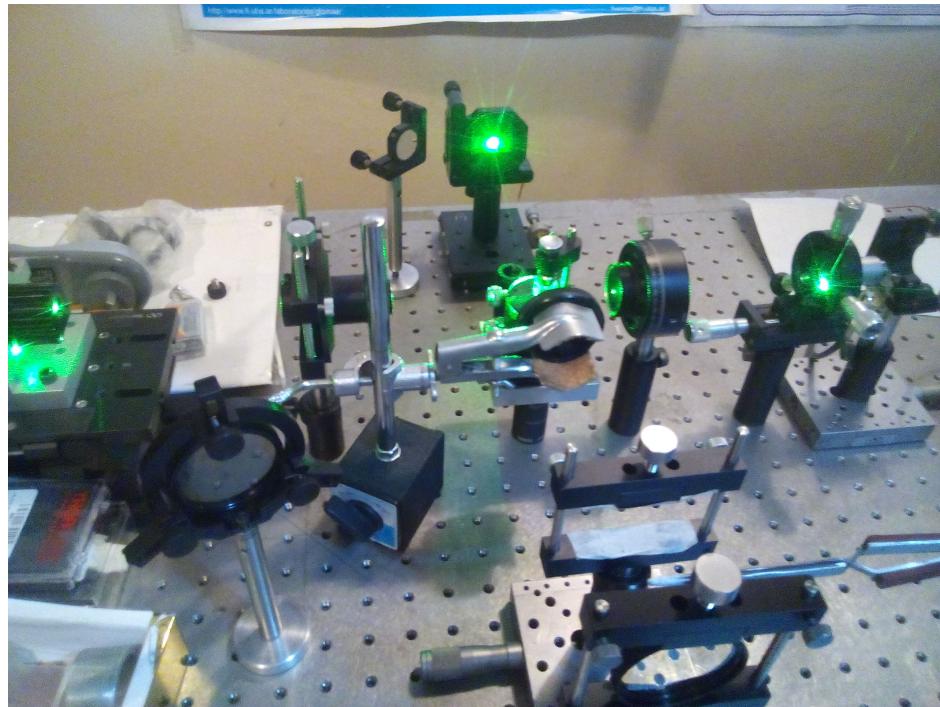
**Figura 9.2** – Salida obtenida para la función dientes de sierra/rampa.



**Figura 9.3** – Salida obtenida para la función seno.

En cuanto a los riesgos que se podían presentar que analizamos en el informe de anteproyecto es posible aclarar que el parlante utilizado para poder generar un desplazamiento físico a partir de una señal eléctrica no tuvo problemas de alinealidad y pudo ser incluido en el esquema interferométrico sin mayores problemas. De todas formas, es necesario calibrar la señal de salida para poder obtener resultados óptimos cuando

se lo usa en el esquema interferométrico.



**Figura 9.4** – Interferómetro encendido con el parlante en una de sus ramas

Por otro lado, otro posible problema era el desarrollo del middleware para pasar las muestras de una señal de Matlab hacia el microcontrolador. Gracias a que Matlab cuenta de antemano con funciones diseñadas para poder comunicarse con un dispositivo serie fue muy sencillo crear un programa que tome muestras de una señal periódica y las envíe al microcontrolador.

Otra de las funciones del proyecto es la medición de intensidad lumínica, se realizaron diversas pruebas sencillas y se verificó el correcto funcionamiento ya que al dejar el módulo de detección completamente a oscuras el valor medido de potencia se vuelve nulo mientras que al utilizar un puntero laser el valor de potencia que se muestra en el display aumenta exponencialmente.

Finalmente, es importante aclarar que si bien es posible verificar el correcto funcionamiento de las diferentes funciones del dispositivo es necesario realizar diferentes experimentos a partir de las cuales va a ser necesario modificar el código del proyecto de manera tal que se logre una precisión mayor que permita que se pueda utilizar en una medición de laboratorio.

## 10. Conclusiones

Primeramente se debe concluir que fue posible implementar correctamente en el proyecto todas las funciones que se plantearon en el informe de anteproyecto. Esto es, crear un menu para poder desplazarse a través de las diferentes opciones que modifican el modo de funcionamiento del dispositivo. Se puede seleccionar entre generar funciones seno o rampa precargadas en el microcontrolador, tomar una función enviada por medio

del protocolo USART desde Matlab, y mostrar en el display la intensidad medida en el módulo de detección.

Entre una de las cosas más importantes que se aprendieron durante la realización de este proyecto es la correcta utilización de las hojas de datos que provee Atmel para sus microcontroladores AVR. Esto es de vital importancia ya que muchas veces existen errores que se pueden solucionar entendiendo cómo fue diseñado el microcontrolador. Un ejemplo de esto es que el microcontrolador ATMega32 cuenta activado por defecto la funcionalidad de debuggeo JTAG, la cual impide el correcto uso del puerto C y en caso de ser necesario hay que desactivar JTAG por medio del seteo de fusibles.

Este proyecto nos permitió aprender a utilizar diversas utilidades de los microcontroladores, entre las que se destacan el protocolo de comunicación serie USART, el conversor ADC, la configuración de interrupciones y el modo de comunicación con un display LCD externo.

## Referencias

- [1] E. Hecht. *Optics*. 4.<sup>a</sup> ed. Addison-Wesley, 1998.