

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Licenciatura en Ciencia de la Computación

TRABAJO DE DIPLOMA

Prácticas de laboratorio de PDS aplicadas a señales de audio en Arduino

Autor: Abraham Pérez Hernández

Tutor: MsC. Roberto Díaz Amador

Santa Clara, junio, 2018
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-1419



Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

Microprocessors are getting into everything. We won't be able to pick up a single piece of equipment in the near future, except maybe a broom, that hasn't got a microprocessor in it.
Arthur C. Clarke (1979).

A mi familia:

A mi papá y mi mamá por estos veinticuatro años que no me ha faltado nada.

A mi abuela por estos últimos cuatro años que me ha atendido.

A mi hermana, por las peleas de cuando éramos niños.

A mi sobrino para que se motive con esta carrera.

A mi familia de Cabaiguán por su cariño.

A mis amigos:

Por apoyarme durante estos cinco años de carrera.

Por ayudarme a salir adelante en la Universidad.

A todos aquellos que contribuyeron de una manera u otra:

Muchas Gracias.

RESUMEN

En la asignatura de PDS actualmente se cuenta con un conjunto de prácticas de laboratorio que permiten al estudiante desarrollar el contenido aprendido en clase, pero a pesar de esto se busca una forma de motivar la enseñanza de esta asignatura mediante la presentación de herramientas de hardware. Por lo tanto el objetivo general de este trabajo de diploma implementar un conjunto de algoritmos que conformen un sistema de prácticas de laboratorio de Procesamiento Digital de Señales utilizando Arduino como plataforma, con esto se espera lograr un mayor interés por parte de los alumnos hacia la asignatura. Los principales resultados alcanzados en son: (1) la implementación de los cuatro tipos de filtros que conformaran cuatro prácticas independientes y (2) la creación de un manual que permita la realización de dichas prácticas.

Palabras Clave: *Arduino, filtro, sonido, digital, frecuencia, prácticas*

ABSTRACT

In the subject of PDS currently has a set of laboratory practices that allow the student to develop the content learned in class, but despite this is looking for a way to motivate the teaching of this subject through the presentation of hardware tools. Therefore, the general objective of this research is to implement a set of algorithms that make up a system of laboratory practices of Digital Signal Processing using Arduino as a platform, with this it is expected to achieve greater interest on the part of students towards the subject. The main results achieved are: (1) the implementation of the four types of filtering that will make up four independent practices and (2) the creation of a manual that allows these practices to be carried out.

Keywords: Arduino, filtering, sound, digital, frequency, practices

Tabla de Contenidos

| | |
|--|-----------|
| INTRODUCCIÓN..... | 10 |
| CAPÍTULO 1: INTRODUCCIÓN A LAS PRÁCTICAS DE LABORATORIO EN LA ASIGNATURA APLICACIONES DEL PROCESAMIENTO DIGITAL DE SEÑALES..... | 14 |
| 1.1 RELACIÓN TEORÍA-PRÁCTICA EN EL PROCESO DOCENTE. | 14 |
| 1.1.1 <i>La práctica de laboratorio en el sistema de educación superior.</i> | 14 |
| 1.1.2 <i>Clasificación de los laboratorios en la ciencia.</i> | 15 |
| 1.2 SOFTWARE PARA EL DESARROLLO DE LAS PRÁCTICAS DE LABORATORIO..... | 17 |
| 1.2.1 <i>Software para creación de filtros y obtención de coeficientes.</i> | 17 |
| 1.2.2 <i>Herramientas para la creación de sketches.</i> | 17 |
| 1.3 CONCLUSIONES DEL CAPÍTULO. | 18 |
| CAPÍTULO 2. IMPLEMENTACIÓN DE LAS PRÁCTICAS REALES | 19 |
| 2.1 GENERALIDADES DE LA PLACA ARDUINO UNO. | 19 |
| 2.2 FLUJO DE TRABAJO PARA UN PROYECTO EN ARDUINO UNO. | 20 |
| 2.3 MATERIALES Y COMPONENTES PARA LAS PRÁCTICAS DE LABORATORIO. | 21 |
| 2.3.1 <i>Práctica 1: Filtrado digital en tiempo real de ruido.</i> | 21 |
| 2.3.2 <i>Práctica 2: Karaoke</i> | 21 |
| 2.3.3 <i>Práctica 3: Detector de frecuencias.</i> | 21 |
| 2.3.4 <i>Práctica 4: Filtrado adaptativo.</i> | 22 |
| 2.4 IMPLEMENTACIÓN DE LAS PRÁCTICAS..... | 22 |
| 2.4.1 <i>Práctica 1: Filtrado digital en tiempo real de ruido.</i> | 22 |
| 2.4.3 <i>Práctica 2: Karaoke</i> | 25 |
| 2.4.4 <i>Práctica 3: Detector de frecuencias.</i> | 27 |
| 2.4.4 <i>Práctica 4 Filtrado adaptativo.</i> | 29 |
| 2.5 CONCLUSIONES DEL CAPÍTULO. | 30 |
| CAPÍTULO 3. RESULTADOS..... | 31 |
| 3.1 DISTRIBUCIÓN DE LAS PRÁCTICAS..... | 31 |

| | |
|--|-----------|
| 3.1.1 Práctica 1: Filtrado digital en tiempo real de ruido. | 32 |
| 3.1.2 Práctica 2 Karaoke..... | 33 |
| 3.1.3 Práctica 3 Detector de frecuencias..... | 35 |
| 3.1.4 Práctica 4 Filtrado adaptativo. | 36 |
| 3.2 CONCLUSIONES DEL CAPÍTULO. | 38 |
| CONCLUSIONES..... | 39 |
| RECOMENDACIONES:..... | 40 |
| REFERENCIAS BIBLIOGRÁFICAS: | 41 |
| ANEXOS:..... | 42 |
| CÓDIGO FILTRADO DIGITAL EN TIEMPO REAL DE RUIDO..... | 42 |
| CÓDIGO PRÁCTICA KARAOKE:..... | 45 |
| CÓDIGO PRÁCTICA DETECTOR DE FRECUENCIAS: | 48 |
| CÓDIGO PRÁCTICA FILTRO ADAPTATIVO:..... | 51 |

INTRODUCCIÓN

En la actualidad el movimiento DIY (Do It Yourself) el cual es la práctica de la fabricación o reparación de cosas por uno mismo, ha tomado puesto entre el desarrollo tecnológico, propiciando la distribución del conocimiento de las tecnologías a una amplia gama de usuarios, no solo científicos, sino también a artistas, emprendedores, etc. Este movimiento ha propiciado el surgimiento de tecnologías de “free hardware”. Se llama “free hardware”, hardware de código abierto, electrónica libre o máquinas libres a aquellos dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago, o de forma gratuita. La filosofía del software libre es aplicable a la del hardware libre, y por eso forma parte de la cultura libre.

El “free hardware” ha traído consigo que el “hardware” sea flexible y más próximo al “software”, con lo cual no es necesario tener amplios conocimientos de electrónica para el desarrollar sistemas funcionales. Una gran aplicación de esto es en el uso de sistemas de tiempo real, el cual interacciona con su entorno físico y responde a los estímulos del entorno dentro de un plazo determinado.

El hardware libre (también llamado “open-source” o “de fuente abierta”) comparte muchos de los principios y metodologías del software libre. En particular, el hardware libre permite que a gente pueda estudiarlo para entender su funcionamiento, modificarlo, reutilizarlo, mejorarlo y compartir dichos cambios. Para conseguir esto, la comunidad ha de poder tener acceso a los ficheros esquemáticos del diseño del hardware en cuestión (que son ficheros de tipo CAD). Estos ficheros detallan toda la información necesaria para que cualquier persona con los materiales, herramientas y conocimientos adecuados pueda reconstruir dicho hardware por su cuenta sin problemas. La placa Arduino es hardware libre y uno de los más difundidos y popularizados (Artero, 2013).

Existen muchas otras placas de diferentes fabricantes que, aunque incorporan diferentes modelos de microcontroladores, son comparables y ofrecen una funcionalidad más o menos similar a la de las placas Arduino. Todas ellas también vienen acompañadas de un entorno de desarrollo agradable y cómodo y de un lenguaje de programación sencillo y completo. No

obstante, la plataforma Arduino (hardware + software) ofrece una serie de ventajas: Arduino es libre y extensible: esto quiere decir que cualquiera que desee ampliar y mejorar tanto el diseño hardware de las placas como el entorno de desarrollo software y el propio lenguaje de programación, puede hacerlo sin problemas. Esto permite que exista un rico “ecosistema” de extensiones, tanto de variantes de placas no oficiales como de librerías software de terceros, que pueden adaptarse mejor a nuestras necesidades concretas. Además, la organización del lenguaje en librerías permite que los programas resulten más pequeños y sean escritos de una forma más flexible y modular, ya que solo están disponibles en cada momento las instrucciones ofrecidas por las librerías utilizadas en ese instante. Arduino tiene una gran comunidad: muchas personas lo utilizan, enriquecen la documentación y comparten continuamente sus ideas. Su entorno de programación es multiplataforma: se puede instalar y ejecutar en sistemas Windows, Mac OS X y Linux. Esto no ocurre con el software de muchas otras placas. Su entorno y el lenguaje de programación son simples y claros: son muy fáciles de aprender y de utilizar, a la vez que flexibles y completos para que los usuarios avanzados puedan aprovechar y expresar todas las posibilidades del hardware. Además, están bien documentados, con ejemplos detallados y gran cantidad de proyectos publicados en diferentes formatos. Las placas Arduino son baratas: la placa Arduino estándar (llamada Arduino UNO) ya preensamblada y lista para funcionar cuesta alrededor de 20 euros. Incluso, uno mismo se la podría construir adquiriendo los componentes por separado, con lo que el precio total de la placa resultante sería incluso menor. Las placas Arduino son reutilizables y versátiles: reutilizables porque se puede aprovechar la misma placa para varios proyectos (ya que es muy fácil de desconectarla, reconectarla y reprogramarla), y versátiles porque las placas Arduino proveen varios tipos diferentes de entradas y salidas de datos, los cuales permiten capturar información de sensores y enviar señales a actuadores de múltiples formas (Artero, 2013).

En la carrera de Ciencia de la Computación de la UCLV no se ha mostrado el gran potencial de esta herramienta y no existe en su programa docente ninguna aproximación a este. Por lo cual se les está negando a los estudiantes una forma poderosa de resolución de problemas. En la asignatura de Procesamiento Digital de Señales I y II impartidas en el primer y segundo semestre del cuarto año de la carrera es posible fomentar el conocimiento y el uso de estas

tecnologías lo cual tendría impacto directo y positivo en el desarrollo profesional de los estudiantes de la carrera.

La implementación de un conjunto de prácticas que permitan al estudiante familiarizarse con la herramienta Arduino tendría impacto en el estudiante dentro de las ramas:

Vocacional:

Al ser una herramienta de uso práctico el estudiante se vería motivado a la experimentación y al desarrollo.

Metodológico:

El estudiante tendría un conocimiento más amplio de las tecnologías que actualmente se usan en el desarrollo profesional.

Práctico:

El profesional que haya recibido la formación en estas herramientas tendrá una nueva forma de aproximarse a la solución de un problema.

Objetivo General:

Implementar un conjunto de algoritmos que conformen un sistema de prácticas de laboratorio de Procesamiento Digital de Señales utilizando Arduino como plataforma.

Para alcanzar este objetivo general, se hace necesario trazar los siguientes objetivos específicos.

Objetivos Específicos:

Escoger las herramientas que faciliten el desarrollo de las prácticas de laboratorio.

Seleccionar las prácticas de laboratorio de PDS adecuadas para ser implementadas utilizando Arduino.

Implementar las prácticas de laboratorio sobre Arduino.

Escribir un manual que permita que profesores y estudiantes puedan realizar las prácticas

Para dar solución a este problema se hacen las siguientes preguntas de investigación:

Preguntas de Investigación:

¿Qué prácticas de laboratorio son adecuadas para ser montadas en el “free hardware” Arduino?

¿Qué herramientas faciliten el desarrollo de la práctica de laboratorio?

Este documento está dividido en tres capítulos. El CAPÍTULO 1 hace un análisis del estado actual de las prácticas de laboratorios existentes para la asignatura de PDS para la carrera de Ciencia de la Computación, así como también de las bases metodológicas para la creación de las prácticas propuestas. En el CAPÍTULO 2 se expone la implementación de los ejercicios propuestos para las prácticas. En el CAPÍTULO 3 se muestran los resultados obtenidos mediante la exposición del manual creado para la realización de las prácticas. Finalmente se exponen las conclusiones, las recomendaciones para una continuación futura y las referencias bibliográficas.

Capítulo 1: Introducción a las prácticas de laboratorio en la asignatura Aplicaciones del Procesamiento Digital de Señales

1.1 Relación teoría-práctica en el proceso docente.

En el ámbito educativo la teoría y la práctica constituyen dos realidades autónomas que gestionan conocimientos de diferente envergadura y se desenvuelven en contextos también distintos. La teoría definida como un conjunto de leyes, enunciados e hipótesis que configuran un corpus de conocimiento científico, sistematizado y organizado, que permite derivar a partir de estos fundamentos reglas de actuación. Mientras la práctica la podemos entender como una praxis que implica conocimiento para conseguir determinados fines. Por lo tanto si la teoría representa el conocimiento y la práctica la aplicación del mismo, es necesario la práctica como elemento sintetizador del conocimiento. Esta relación entre teoría y práctica desarrolla habilidades y sistematiza conocimientos. La práctica sirve al conocimiento, por lo cual se centra en actividades derivativas, en la manipulación de herramientas y experimentación a base de prueba y error (Álvarez, 2012).

1.1.1 La práctica de laboratorio en el sistema de educación superior.

La práctica de laboratorio es el tipo de clase que tiene como objetivos instructivos fundamentales que los estudiantes adquieran las habilidades propias de los métodos de la investigación científica, amplíen, profundicen, consoliden, realicen, y comprueben los fundamentos teóricos de la asignatura mediante la experimentación empleando los medios de enseñanza necesarios, garantizando el trabajo individual en la ejecución de la práctica. Esta persigue objetivos similares a los de las clases prácticas. En las prácticas de laboratorio los objetivos se cumplen a través de la realización de experiencias programadas con el apoyo de un manual (Iglesias, Manuel, & Mesa, 2008).

Por su esencia el proceso de realización de las prácticas de laboratorio constituye parte integrante del trabajo independiente de los estudiantes, el cual está constituido por tres etapas (Iglesias et al., 2008):

- Preparación previa a la práctica

- Realización de la práctica
- Conclusiones de la práctica

La preparación previa a la práctica es desarrollada sobre la base del estudio teórico orientado por el profesor como fundamento de la práctica, así como el estudio de las técnicas de los experimentos correspondientes. El desarrollo se caracteriza por el trabajo de los estudiantes con el material de laboratorio, la reproducción de los fenómenos deseados, el reconocimiento de los índices característicos de su desarrollo, la anotación de las observaciones, entre otras tareas docentes. Durante las conclusiones el estudiante deberá analizar los datos de la observación y arribar a las conclusiones y generalizaciones que se derivan de la práctica en cuestión. El profesor deberá tener en cuenta que el trabajo independiente en el laboratorio es muy complejo si se realiza conscientemente, por cuanto debe combinar las acciones físicas y mentales de forma paralela. Muchas veces los estudiantes se limitan a la reproducción mecánica de los pasos. Esto en gran medida se puede evitar si el conjunto de experimentos propuestos en la técnica presupone un enfoque investigativo de los estudiantes para su realización. Este enfoque investigativo requiere de la existencia de una técnica de laboratorio tal, que en la misma no se de toda la información detallada, sino que una buena parte de dicha información debe ser extraída por el estudiante a partir del conocimiento de los objetivos. Este enfoque resume una de las posibles formas que pueda adoptar el experimento con carácter investigativo (Iglesias et al., 2008).

Al igual que en otras tipos de clases, es necesario durante su preparación tener en cuenta las etapas del proceso de enseñanza-aprendizaje: motivación, orientación y evaluación (Iglesias et al., 2008).

1.1.2 Clasificación de los laboratorios en la ciencia.

La Ciencia es una actividad eminentemente práctica, además de teórica, lo cual hace que en su enseñanza el laboratorio sea un elemento indispensable. El objetivo fundamental de los trabajos prácticos es fomentar una enseñanza más activa, participativa e individualizada, donde se impulse el método científico y el espíritu crítico. De este modo se favorece que el

alumno: desarrolle habilidades, aprenda técnicas elementales y se familiarice con el manejo de las herramientas ("Las prácticas de laboratorio importancia, diseño y elaboración," 2010).

Los laboratorios se clasifican en función de dos criterios: (1) La forma de acceder a los recursos (local o remota) para propósitos de experimentación y (2) la naturaleza del sistema físico (real o virtual), con lo que los entornos de experimentación quedarían clasificados de acuerdo a (Tabla 1) (Calvo et al., 2009):

| | Real | Virtual |
|--------|--|--|
| Local | Laboratorio en el cual todas las herramientas y sistemas que se utilizan existen físicamente. | Laboratorio en el cual todas algunas de las herramientas o sistemas que se utilizan son virtualizados. Ej.: Máquinas virtuales, simuladores. |
| Remoto | Laboratorio en el cual todas las herramientas y sistemas que se utilizan existen físicamente pero se acceden a ellas a través de la red. Ej.: Clúster. | Laboratorio en el cual todas algunas de las herramientas o sistemas que se utilizan son virtualizados pero en servidores remotos. |

Tabla 1: Clasificación de los laboratorios.

Disponer de un laboratorio local y real, con las características apropiadas para el procesamiento digital de señales, planteando ejercicios que utilicen directamente equipamiento e instalaciones reales. Este tipo de laboratorio resulta más motivador, ya que permite hacer uso de hardware, en contraparte de emplear tan solo software. Esto no quiere decir que dichos entornos virtuales sean completamente descartados, ya que pueden representar una ayuda muy importante en el momento de realizar labores de diseño. Sin embargo, se considera indispensable el contacto directo con el equipamiento. La importancia de los laboratorios tanto en la enseñanza de las ciencias como en la investigación y en la

industria, es indispensable. No se puede negar que el trabajo práctico en el laboratorio proporciona la experimentación y el descubrimiento. El uso de laboratorio, requiere de tiempo adicional al de una clase convencional, para descubrir y aprender de los propios errores. El laboratorio es el elemento y complemento más distintivo de la educación científica, tiene gran relevancia en el proceso de formación y en este se puede conocer al estudiante con las habilidades que posee para desarrollar las actividades de montaje y la experiencias con la finalidad de afianzar los conocimientos y actitudes adquiridas en el aula de clase (Frías, 2016).

1.2 Software para el desarrollo de las prácticas de laboratorio.

Para el desarrollo de las prácticas de laboratorio es necesario interactuar con software, por lo tanto es necesario escoger el mismo de forma que facilite a los estudiantes y a los profesores el flujo de la clase.

1.2.1 Software para creación de filtros y obtención de coeficientes.

En orden de realizar un filtrado es necesario obtener los coeficientes del filtro que se desea construir. El Matlab es el software escogido por varios motivos. Esta herramienta permite la obtención de los coeficientes de los filtros de manera muy cómoda, así como también trae un plugin que permite crear los filtros desde un GUI donde se especifican las principales características del mismo, lo cual es muy cómodo pues no es objetivo enseñar a usar el código de Matlab tan solo para obtener coeficientes. Además que en este tema Matlab cuenta con mucha bibliografía, en internet la mayoría de artículos y libros que hablan sobre el filtrado digital usan Matlab para sus ejemplos. Por otro lado las prácticas de laboratorio existentes están basadas en Matlab y eso crea una base para los profesores que ya están familiarizados con el tema además que pueden dar los cursos antiguos como bibliografía.

1.2.2 Herramientas para la creación de sketches.

Sin duda la herramienta más popular para la creación de sketches para Arduino es Fritzing, otra muy popular es Proteus al cual se le puede instalar un plugin que simula el Arduino UNO. Proteus permite no solo la creación de sketches sino también la simulación del circuito

con el código. Fritzing tan solo nos permite la creación del sketch y el guardado del código conjunto con el sketch, pero producto de que el objetivo de las prácticas es el filtrado y no el diseño de circuitos se va a usar Fritzing que permite más claridad en el diseño.

1.3 Conclusiones del capítulo.

Las prácticas de laboratorio son fundamentales para la consolidación de conocimiento y es importante la motivación de los estudiantes ante las mismas, lo cual les permite desarrollarse mejor durante esta. Para motivar el interés de los estudiantes se va a hacer uso de un tipo de práctica de laboratorio que es el Real-Local el cual permite que los estudiantes interactúen con todas las herramientas de forma directa, motivando así el interés en la clase. Los sketches estarán diseñados en Fritzing dado que este permite una mayor claridad en el diseño de los circuitos.

CAPÍTULO 2. Implementación de las Prácticas Reales

2.1 Generalidades de la placa Arduino UNO.

La placa Arduino UNO usa un microcontrolador ATmega328P, dicha placa opera con 5V que pueden ser suministrados a través del puerto USB Tipo B o a través de un puerto Jack por el cual se le deben suministrar de 7 a 12V recomendados y como límite de 6 a 20V. Posee 14 pines de entrada/salida digital de los cuales 6 admiten PWM y posee 6 de entrada Analógica. Otras características del Arduino UNO se presentan en (Tabla 2) (ARDUINO).

| | |
|-----------------------------|---|
| Microcontroller | ATmega328P |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |

| | |
|-------------|--------|
| Clock Speed | 16 MHz |
| LED_BUILTIN | 13 |

Tabla 2: Especificaciones Arduino UNO

2.2 Flujo de trabajo para un proyecto en Arduino UNO.

Para el desarrollo de un proyecto en Arduino UNO debemos conocer las características básicas mencionadas con anterioridad y tenerlas en cuenta durante el diseño de nuestro circuito y posteriormente durante la implementación del código. En el desarrollo de un proyecto de este tipo, lo ideal es diseñar nuestro circuito sobre algún software antes de montarlo sobre el Arduino físicamente, para una mayor seguridad se podría probar el circuito en algún software de simulación. En este caso se ha escogido Fritzing (Imagen 1), que aunque no permite simulación, es una herramienta de “free software” y además da facilidades para guardar el código junto con el diseño del circuito.

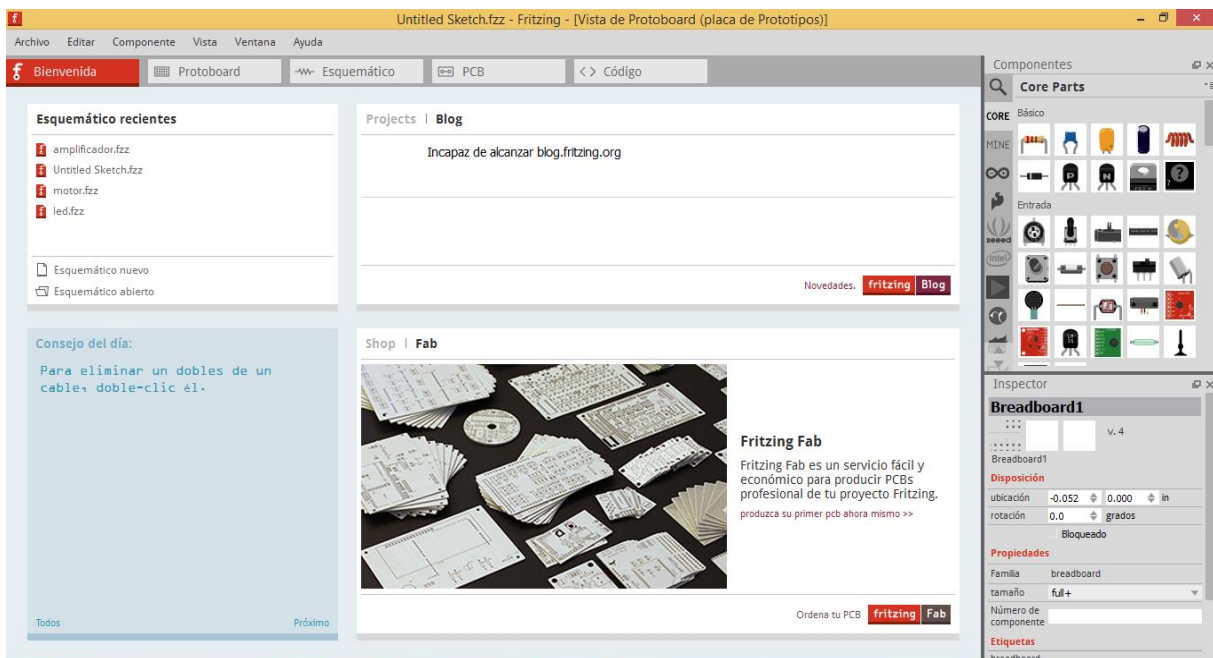


Imagen 1: Software de diseño Fritzing.

El próximo paso sería montar el diseño del circuito sobre la placa física y comenzar a escribir el código. Durante la implementación del código debemos tener en cuenta, la capacidad de memoria del Arduino UNO y no hacer uso indiscriminado de ella.

2.3 Materiales y Componentes para las prácticas de laboratorio.

Para la implementación de todas las prácticas es necesario contar con la placa física Arduino UNO, cuyas características han sido mencionadas con anterioridad, se debe contar con una Protoboard o de las herramientas necesarias para soldar y además será necesario el uso de transistores, resistencias y otros componentes pero el uso de estos es descrito para cada práctica individualmente más adelante, así como sus cantidades y principales características.

2.3.1 Práctica 1: Filtrado digital en tiempo real de ruido.

Para la implementación de esta práctica es necesario la construcción de un amplificador para un micrófono. Para el cual se usará un potenciómetro de hasta $10k\Omega$, una resistencia de $1k\Omega$, una de $10k\Omega$ y una de $100k\Omega$ (aunque en la práctica la hemos sustituido por diez resistencias de $10k\Omega$ conectadas en serie), todas las resistencias usadas tienen una tolerancia de un 5%. Además será usado un transistor BC547B NPN de silicio, un micrófono de escritorio, un puerto hembra Jack 3.5mm y conectores de cobre.

2.3.2 Práctica 2: Karaoke.

Para la implementación de esta práctica además de los componentes mencionados con anterioridad en el inicio del subcapítulo solamente necesitaremos contar con un puerto hembra Jack 3.5mm y conectores de cobre.

2.3.3 Práctica 3: Detector de frecuencias.

Para la implementación de esta práctica son necesario cuatro LEDs los cuales estarán conectados cada uno a un pin de salida del Arduino, además son necesarias cuatro resistencias de 220Ω y conectores de cobre. Esta práctica es hace uso también del circuito del amplificador expuesto con anterioridad.

2.3.4 Práctica 4: Filtrado adaptativo.

Para la construcción de esta práctica necesitaremos construir un dispositivo capaz de introducir ruido a una señal, para el cual necesitaremos un potenciómetro de hasta $10k\Omega$, un capacitor de $100\mu F$ y conectores de cobre.

2.4 Implementación de las prácticas.

Teniendo todos los componentes es necesario pasar a realizar el diseño del circuito y luego a la implementación del código.

2.4.1 Práctica 1: Filtrado digital en tiempo real de ruido.

Esta práctica concierte en la implementación del filtrado digital en tiempo real de ruido a una señal de audio con voz. Es necesario eliminar el ruido, el ruido en este caso será ruido blanco, lo que significa que la frecuencia del ruido es diferente a la frecuencia de la señal y por lo tanto son separables implementando un filtro paso banda, donde la banda de paso está en el rango de la frecuencia de la voz humana, por lo tanto lo que este fuera de ese rango de frecuencia es considerado ruido y debe ser eliminado. Para ello es necesario diseñar el circuito que amplifique la señal proveniente del micrófono, diseñar el filtro, obtener los coeficientes de dicho filtro e implementar el filtrado en el Arduino UNO.

El amplificador (Diagrama 1) necesita de resistencias de $1k\Omega$, $10k\Omega$, $100k\Omega$, un potenciómetro un transistor NPN. En el circuito la señal proveniente del micrófono conectado al puerto Jack 3.5mm es amplificada y finalmente se lee por el pin de lectura analógica A0 del Arduino.

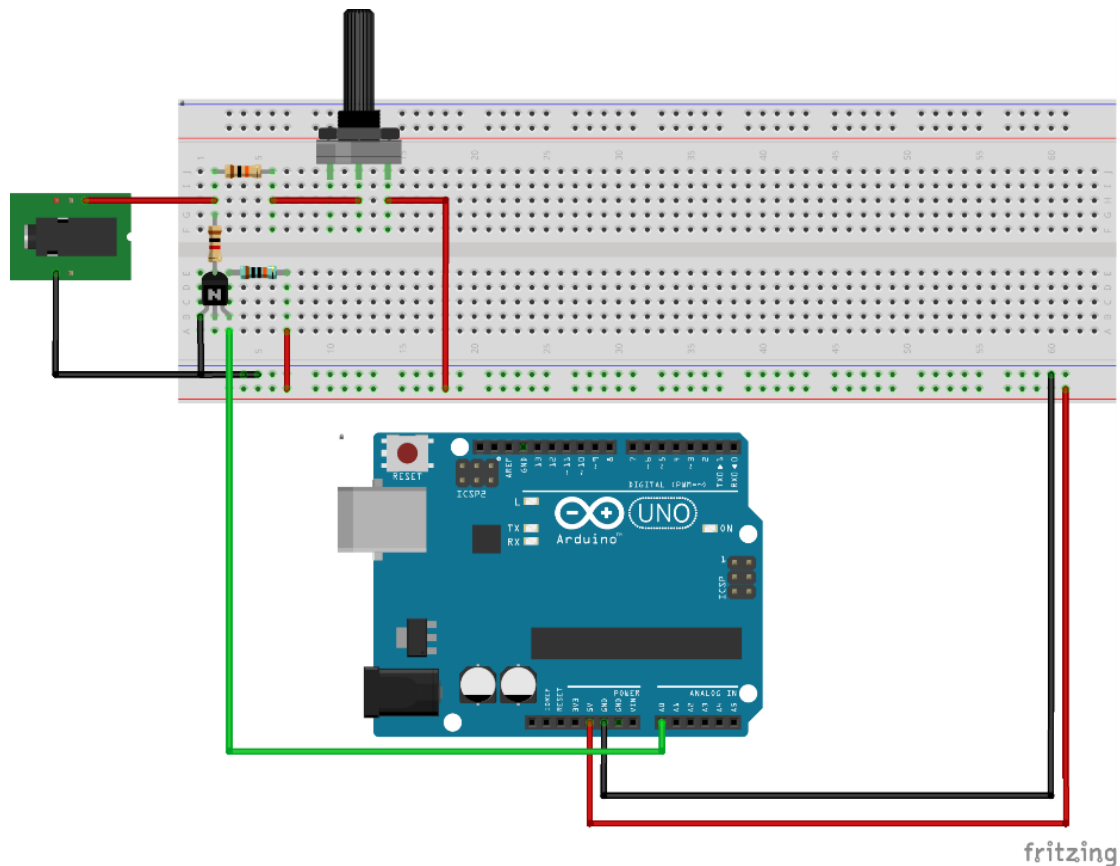


Diagrama 1:Amplificador2.4.2

Para el diseño del filtro y la obtención de sus coeficientes se utilizó la herramienta FDATool de Matlab. En la Imagen 2 se muestra como el uso de dicha herramienta y como quedó el diseño del filtro. El filtro es un pasa banda donde la banda de paso esta entre los 100Hz a 250Hz a una frecuencia de muestro de 1kHz. Aclarar que la frecuencia de la voz puede alcanzar los 4kHz pero estamos enfocando la realización de este ejercicio que es metodológico en un rango menor que en la frecuencia fundamental de la voz.

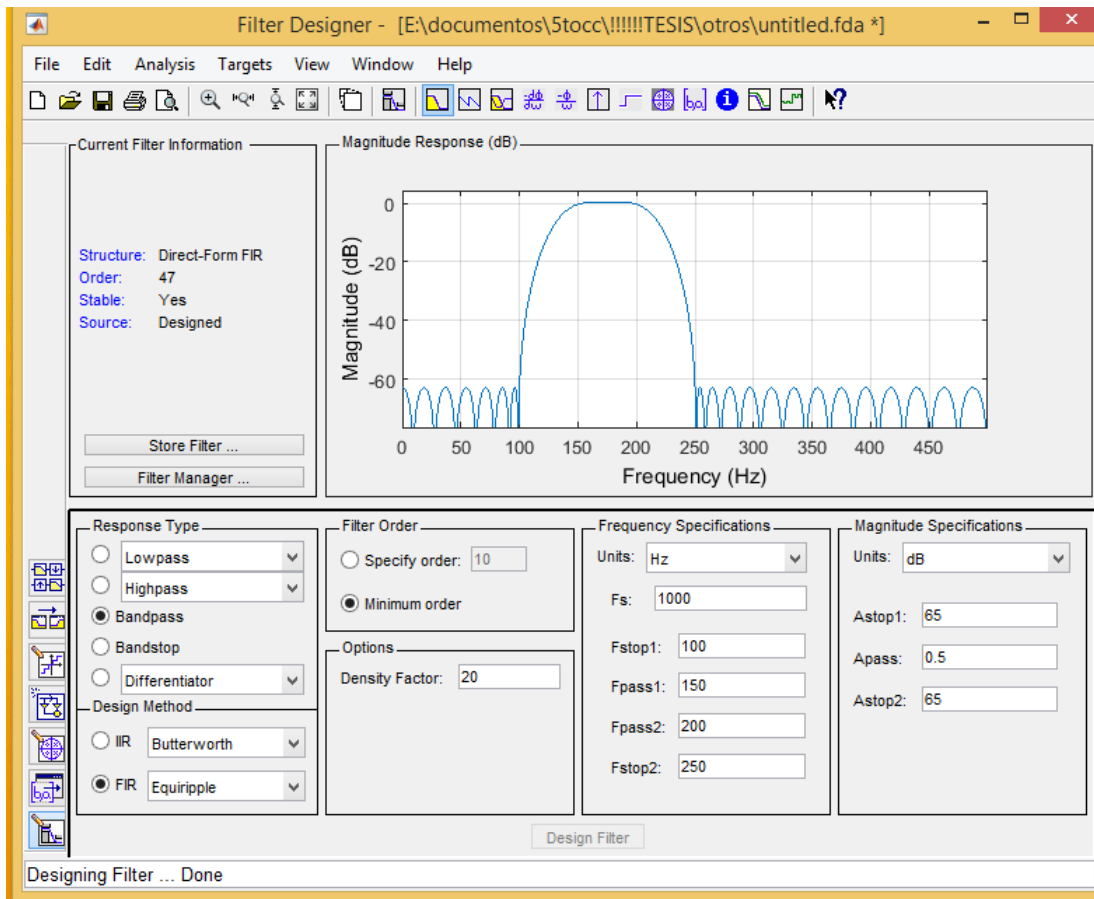


Imagen 2: FDATOOL.

De la herramienta FDATOOL también fueron extraídos los coeficientes del filtro diseñado los cuales los mostramos como parte del código de Arduino (ver Código 1).

```
const float coeff[]={-0.00124524577937771,-0.00280198539369773,0.0000006045039216,
    0.00630328657649506,0.00823661763838471,-0.00151431405782056,
    -0.014151805764884,-0.0125056431623591,0.00461013098918164,
    0.0159865225517666,0.00807653016950389,-0.00290328185391139,
    0.00224594284891645,0.00759719823859967,-0.0155188779102616,
    -0.0471417531356446,-0.0253465673428238,
    0.0572600967874697, 0.105432948687573, 0.0295343280364442,
    -0.111073177809333, -0.147660220290832, -0.0131440745330448,
    0.149359471577213, 0.149359471577213, -0.0131440745330448,
    -0.147660220290832, -0.111073177809333, 0.0295343280364442,
    0.105432948687573, 0.0572600967874697, -0.0253465673428238,
    -0.0471417531356446, -0.0155188779102616, 0.00759719823859967,
    0.00224594284891645, -0.00290328185391139, 0.00807653016950389,
    0.0159865225517666, 0.00461013098918164, -0.0125056431623591,
    -0.014151805764884,-0.00151431405782056, 0.00823661763838471,
    0.00630328657649506, 0.0000006045039216, -0.00280198539369773,
    -0.00124524577937771};
```

Código 1: Coeficientes del filtro en el código Arduino.

En el código la variable `signlx` es un arreglo de enteros que guarda los cuarenta y ocho valores previos de la señal. La función `add` (Código 2) adiciona de forma cíclica valores al arreglo `signlx` y la función `eval` (Código 2) aplica el filtro sobre la señal.

```
//función que adiciona de forma ciclica los
// valores de la señal a la matriz
void add(int val){
  if(cont<48){
    signlx[47-cont]=val;
    cont++;
  }else{
    for(int i=0; i<48; i++){
      int v1 = signlx[i];
      signlx[i]=val;
      val=v1;
    }
  }
}

//funcion que aplica el filtro
float eval(){
  val = 0;
  for(int i = 0; i < 48; i++){
    val = val+ (signlx[47-i]*coeff[i]);
  }
}
```

Código 2: Función `add` y función `eval`.

La función `loop` es natural de Arduino y se encarga de ejecutar cíclicamente el código dentro de ella. El código dentro de la función `loop` es el encargado de leer la señal del micrófono y evaluar la función `eval` para obtener la señal filtrada. (Ver Código 3).

```
void loop() {
  int sensor = analogRead(sign);
  add(sensor);
  Serial.println(eval());
}
```

Código 3: Función `loop`

2.4.3 Práctica 2: Karaoke.

Esta práctica consiste en separar de una canción la voz de los instrumentos, dado que la voz y los instrumentos comparten frecuencias, no es posible separarlos mediante un filtrado convencional PB, PA o PB y para esto es necesario tener un audio que proceda de una fuente con dos canales derecho e izquierdo. Para ello se necesita solamente del Arduino y un puerto hembra Jack 3.5mm (Diagrama 2).

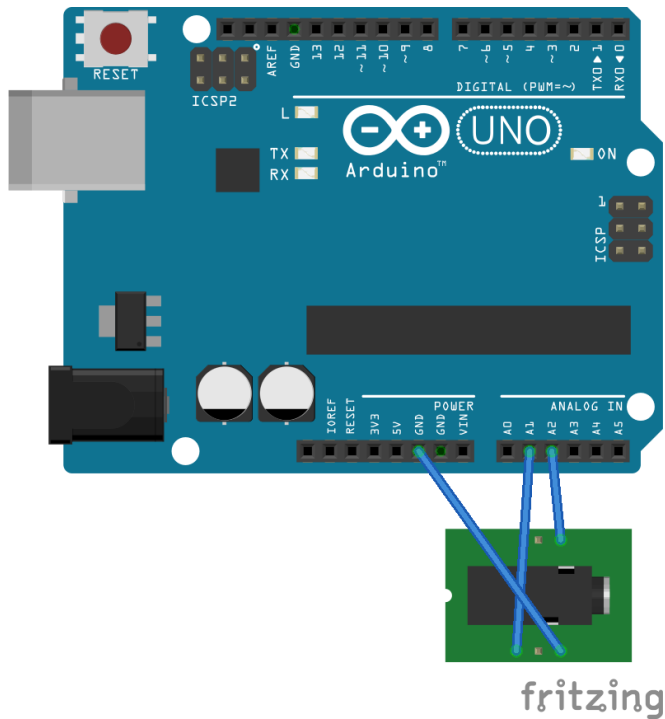


Diagrama 2: Karaoke

El código (Código 4) de esta práctica consiste en ir adicionando de forma cíclica las entradas de los canales derecho e izquierdo, restarlos y luego aplicar un filtrado de tipo Butterworth. El código de ambas funciones está recogido en los anexos para simplificar espacio.

```
void loop() {
    // put your main code here, to run repeatedly:
    add();
    filter();
    cont++;
    for(int i=0;i<5;i++ ){
        Serial.println(out[i]);
    }
}
```

Código 4: loop

2.4.4 Práctica 3: Detector de frecuencias.

Esta práctica concierte en la detección de las frecuencias existentes en una señal de audio. Para ello se van a utilizar cuatro LEDs los cuales van a representar cada uno un rango de frecuencias, de forma que si un LED está encendido significa que la señal presenta una frecuencia comprendida en el rango que él representa. Para ello se va a utilizar un micrófono como entrada de audio, por lo cual será necesario utilizar el amplificador antes diseñado y diseñar además el circuito de los LEDs, hay que decidir que rango de frecuencias convendrá ser representados e implementar el código que reconozca las frecuencias que se van activando.

El diseño del circuito necesario para la práctica, es mostrado en el (Diagrama 2), el circuito del amplificador envía la señal al Arduino el cual la lee por el pin A0, la procesa da la salida por los pines 2, 3, 4 y 5 para los LEDs verde, rojo, azul y amarillo respectivamente. Los LEDs están conectados directamente a la GND (ground o tierra) del Arduino y a los pines 2, 3, 4 y 5 a través de resistencias de 220Ω .

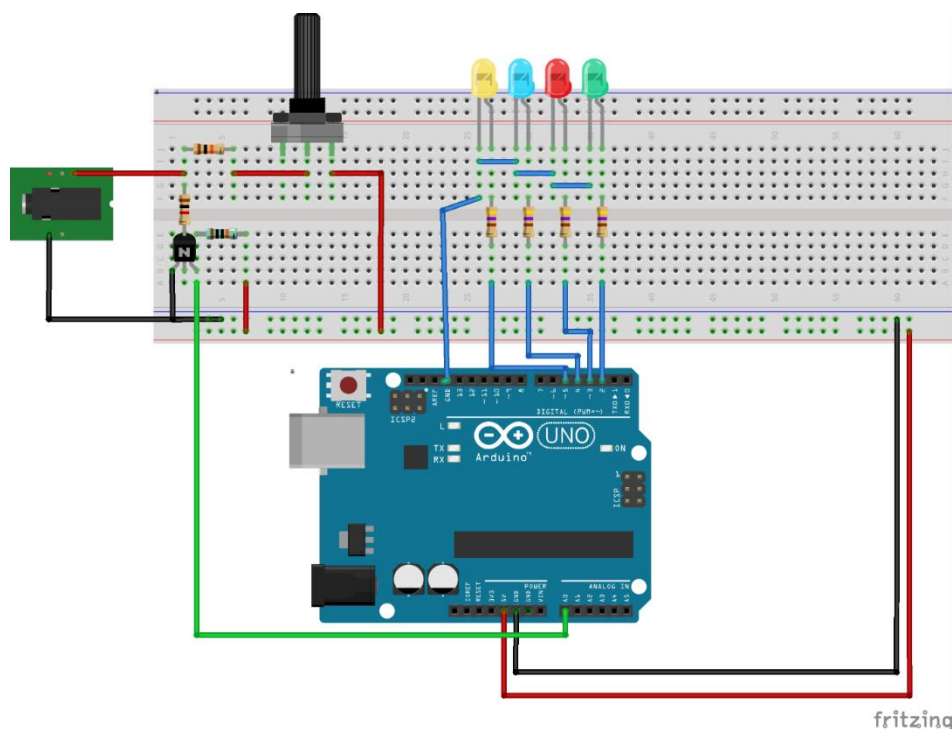


Diagrama 3: Circuito Práctica 2.

Lo función setup es propia de Arduino y si inicia antes de loop y se ejecuta solamente una vez el iniciar. En se le dice al Arduino que prepare los pines correspondientes a los LEDs para escribir en ellos. (Código 5).

```
void setup() {  
    pinMode(amarillo, OUTPUT);  
    pinMode(azul, OUTPUT);  
    pinMode(rojo, OUTPUT);  
    pinMode(verde, OUTPUT);  
}
```

Código 5: Función setup

En el loop se implementa el proceso de analizar qué frecuencia está presente y encender el LED en el rango correspondiente. En el segmento de código (Código 6) se muestra como se implementa esto para las frecuencias menores de 100Hz, cuyo rango corresponde al LED verde y se muestra la implementación para el rango entre 100Hz y 200Hz, correspondiente al LED rojo. Los demás rangos se implementan de forma similar y corresponden a: LED azul de 200Hz a 400Hz y LED amarillo de 400Hz a 800Hz.

```
//frecuencias menores a 100Hz  
if(time2 - timeV >=10000 and sensor>=100){  
    digitalWrite(verde,HIGH);  
    timeV=time2;  
}  
else{  
    digitalWrite(verde,LOW);  
}  
  
//frecuencias entre 100Hz y 200Hz  
if(time2 - timeR <=10000 and time2 - timeR >=5000 and sensor>=100){  
    digitalWrite(rojo,HIGH);  
    timeR=time2;  
}  
else{  
    if(time2 - timeR >=10000){  
        timeR=time2;  
    }  
    digitalWrite(rojo,LOW);  
}
```

Código 6: Analisis de frecuencias activas.

2.4.4 Práctica 4 Filtrado adaptativo.

Esta práctica consiste en la implementación de un filtro adaptativo. Los filtros funcionan de manera similar a las redes neuronales, mientras que un filtro convencional mantiene sus coeficientes invariables en el tiempo, un filtro adaptativo itera sobre ellos para modificarlos. Los filtros adaptativos están hechos para adaptarse a un determinado tipo de ruido y luego de concluidas sus iteraciones puede ser utilizado para eliminar ese ruido, para ello necesitan de una señal con ruido y de la señal limpia.

```
void loop() {
  // put your main code here, to run repeat:
  //Serial.println(D[0]);
  if(n < 15){
    digitalWrite(out,D[n]);
    delay(1000);
    add(n);
    Serial.print(n);
    Serial.print(": ");
    Serial.println(X[n]);
    for(int i=0;i<15;i++){
      Y[n]+=W[i]*X[i];
      E[n] = D[n]-Y[n];
    }
    for(int i=1;i<15;i++){
      W[i]= W[i-1] -2 * E[n] * X[n-1];
    }
    n++;
  }else{
    for(int i=0;i<15;i++){
      Serial.println(W[i]);
    }
  }
}
```

Código 7: loop filtro adaptativo

En el código (Código 7) podemos ver como se crean los coeficientes del filtro. En cada iteración es necesario evaluar el filtro, sus coeficientes en la primera iteración deben ser llenados aleatoriamente. Luego es evaluado y se obtiene el error de la resta de la señal deseada (señal limpia) y la señal con ruido, entonces son recalculados los coeficientes, esto para en una cantidad determinada de iteraciones o puede ser implementado también para que para cuando llegue a un margen de error mínimo. Aclarar que estos filtros tienden a converger a mínimos locales.

2.5 Conclusiones del capítulo.

Las prácticas de laboratorio son variadas y se hace uso tanto del “software” como del “hardware”. Estas son sencillas pero cubren rasgos abarcadores tanto de las herramientas usadas como de los contenidos impartidos en clase.

CAPÍTULO 3. Resultados.

La creación de una guía que permita el desarrollo de la práctica de laboratorio es necesaria tanto para estudiante como para profesores.

3.1 Distribución de las prácticas.

Para el desarrollo de las prácticas de laboratorio se tienen en cuenta los siguientes aspectos (Frías, 2016):

- **Tema:** Es una frase relativamente pequeña que da a conocer el tema a tratar, este se puede desarrollar en diferentes actividades.
- **Título:** Debe expresar el nombre de la práctica. El título deberá ser sugerente, atractivo y relacionado con el tema.
- **Objetivo:** Se plantean las metas a alcanzar con la realización de la práctica. Identifican la finalidad a la cual deben dirigirse los recursos y esfuerzos en aras de dar cumplimiento a los propósitos.
- **Materiales y equipos:** Se mencionan y describen los materiales y equipos necesarios para el desarrollo de la práctica de laboratorio. Especifica todo lo requerido en cuanto al tipo de equipos, materiales, tecnologías, herramientas y *software*, tanto para la etapa de experimentación como para la reproducción futura del problema en estudio
- **Conocimientos previos:** Se refiere a los conocimientos teóricos que el estudiante debe poseer, para contribuir al cumplimiento de los objetivos trazados en la práctica.
- **Habilidades:** Se define como la destreza para ejecutar alguna acción con el fin de alcanzar un objetivo.
- **Técnica operatoria:** En este paso se describe de forma ordenada, los pasos a seguir durante el laboratorio. Este procedimiento debe conducir a que el estudiante adquiera la habilidad de seguir una secuencia de pasos a la hora de realizar la práctica.

- **Conclusiones:** Se desarrollarán de manera participativa donde el profesor guiará, mediante preguntas, a los estudiantes para que concluyan los principales aspectos tratados durante la práctica de laboratorio. Contiene las acciones que el estudiante debe seguir durante el desarrollo de las prácticas en aras de cumplir los objetivos propuestos.
- **Estudio independiente:** Se debe orientar al estudiante, la realización de un ejercicio de laboratorio que permita sistematizar el conocimiento obtenido durante la práctica.
- **Bibliografía:** Se indica la bibliografía básica y complementaria con la que fueron elaborados los contenidos de la práctica.

3.1.1 Práctica 1: Filtrado digital en tiempo real de ruido.

Tema: Eliminación de ruido blanco de una señal de voz.

Título: Filtrado digital en tiempo real de ruido.

Objetivo:

Familiarizarse con el IDE de Arduino, así como con su lenguaje.

Hacer uso de componentes básicos en la electrónica.

Consolidación de los contenidos teóricos con respecto al filtrado digital.

Materiales y equipos:

Arduino UNO

Computadora con el IDE Arduino instalado.

Componentes electrónicos tales como capacitores, resistencias y potenciómetros.

Conocimientos previos:

Conocer la herramienta Matlab y su extensión fdatool.

Procesamiento digital de señales.

Habilidades:

Montado de circuito sobre la protoboard y el uso de herramientas.

Técnica operatoria:

1. Montar sobre la protoboard y el Arduino el circuito mientras se tiene este desconectado de la corriente y del pc.
2. Verificar con los esquemáticos las conexiones hechas en el Arduino.
3. Conectar el Arduino a la pc mediante el cable USB.
4. Implementar el código para la actividad
5. Subir al Arduino el código.
6. Esperar a que se subido el código.
7. Realizar pruebas e interactuar con los sensores del Arduino.

Conclusiones:

¿Qué es Arduino?

¿Qué es un microcontrolador?

¿Qué utilidades tiene Arduino?

¿Qué es un filtro digital?

¿Qué tipo de filtro digital se usó en la clase, qué características tiene?

Estudio independiente:

Familiarizarse con el IDE y el lenguaje.

Modificar el código para que permita obtener el ruido y no solamente la señal filtrada.

Bibliografía:

"Getting Started with Arduino" de Massimo Banzi.

Conferencia Filtrado digital.

3.1.2 Práctica 2 Karaoke.

Tema: Separación de la voz y los instrumentos en una canción.

Título: Karaoke.

Objetivo:

Consolidación en el lenguaje de Arduino.

Hacer uso de componentes básicos en la electrónica.

Consolidación de los contenidos teóricos con respecto al filtrado digital.

Materiales y equipos:

Arduino UNO

Computadora con el IDE Arduino instalado.

Un puerto Jack 3.5mm, la protoboard y el Arduino UNO.

Conocimientos previos:

Procesamiento digital de señales.

Familiarización con el lenguaje e IDE de Arduino.

Habilidades:

Montado de circuito sobre la protoboard y el uso de herramientas.

Técnica operatoria:

1. Montar sobre la protoboard y el Arduino el circuito mientras se tiene este desconectado de la corriente y del pc.
2. Verificar con los esquemáticos las conexiones hechas en el Arduino.
3. Conectar el Arduino a la pc mediante el cable USB.
4. Implementar el código para la actividad
5. Subir al Arduino el código.
6. Esperar a que se subido el código.
7. Realizar pruebas e interactuar con los sensores del Arduino.

Conclusiones:

¿Qué utilidades tiene Arduino?

¿Qué es un filtro digital?

¿Qué tipo de filtro digital se usó en la clase, qué características tiene?

Estudio independiente:

Familiarizarse con las librerías de Arduino existentes.

Modificar el código para que permita obtener los instrumento y no solamente la voz.

• **Bibliografía:**

"Getting Started with Arduino" de Massimo Banzi.

Conferencia filtrado digital.

3.1.3 Práctica 3 Detector de frecuencias.

Tema: Detectar las frecuencias presentes en una señal de audio.

Título: Detector de frecuencias.

Objetivo:

Consolidación en el lenguaje de Arduino.

Hacer uso de componentes básicos en la electrónica.

Consolidación de los conocimientos acerca de frecuencia y su definición.

Materiales y equipos:

Arduino UNO

Computadora con el IDE Arduino instalado.

Un puerto Jack 3.5mm, la protoboard, el Arduino UNO y otros componentes electrónicos.

Conocimientos previos:

Procesamiento digital de señales.

Familiarización con el lenguaje e IDE de Arduino.

Conceptos de frecuencia y periodo.

Habilidades:

Montado de circuito sobre la protoboard y el uso de herramientas.

Técnica operatoria:

1. Montar sobre la protoboard y el Arduino el circuito mientras se tiene este desconectado de la corriente y del pc.
2. Verificar con los esquemáticos las conexiones hechas en el Arduino.
3. Conectar el Arduino a la pc mediante el cable USB.
4. Implementar el código para la actividad
5. Subir al Arduino el código.
6. Esperar a que se subido el código.
7. Realizar pruebas e interactuar con los sensores del Arduino.

Conclusiones:

¿Qué utilidades tiene Arduino?

¿Qué es la frecuencia?

¿Qué es el periodo?

¿Cómo se relacionan frecuencia y periodo?

Estudio independiente:

Familiarizarse con las librerías de Arduino existentes.

Modificar el código para que permita analizar la existencia de otros rangos de frecuencias.

• Bibliografía:

"Getting Started with Arduino" de Massimo Banzi.

Conferencia filtrado digital.

Conferencia introducción a PDS.

3.1.4 Práctica 4 Filtrado adaptativo.

Tema: Implementar un filtrado adaptativo.

Título: Filtrado adaptativo.

Objetivo:

Consolidación en el lenguaje de Arduino.

Hacer uso de componentes básicos en la electrónica.

Explicar el uso de los filtro adaptativos y una de sus implementaciones.

Materiales y equipos:

Arduino UNO

Computadora con el IDE Arduino instalado.

Un capacitor un potenciómetro, la protoboard y el Arduino UNO.

Conocimientos previos:

Procesamiento digital de señales.

Familiarización con el lenguaje e IDE de Arduino.

Conocimientos teóricos acerca de los filtro adaptativos.

Habilidades:

Montado de circuito sobre la protoboard y el uso de herramientas.

Técnica operatoria:

1. Montar sobre la protoboard y el Arduino el circuito mientras se tiene este desconectado de la corriente y del pc.
2. Verificar con los esquemáticos las conexiones hechas en el Arduino.
3. Conectar el Arduino a la pc mediante el cable USB.
4. Implementar el código para la actividad
5. Subir al Arduino el código.
6. Esperar a que se subido el código.
7. Realizar pruebas e interactuar con los sensores del Arduino.

Conclusiones:

¿Qué utilidades tiene Arduino?

¿Qué es un filtro adaptativo?

¿En qué se diferencia de otros filtros?

¿Cuál es su utilidad?

Estudio independiente:

Familiarizarse con las librerías de Arduino existentes.

Modificar el programa para que los coeficientes iniciales sean iniciados aleatoriamente por el Arduino.

Bibliografía:

"Getting Started with Arduino" de Massimo Banzi.

Conferencia filtrado digital.

Conferencia filtrado adaptativo.

3.2 Conclusiones del capítulo.

Se creó la guía que permite el desarrollo de la práctica, está dividida en secciones que de manera que cada sección permite el desarrollo de una actividad específica relacionada a la práctica de laboratorio.

Conclusiones:

Como resultado de este trabajo se escogieron las cuatro prácticas de manera que si integra el contenido propuesto en las conferencias de la asignatura de PDS, sirviendo las prácticas de continuacion a la teoría. Se escogió el software que permite un mejor flujo de trabajo durante las prácticas y resulta más ameno para alcanzar los objetivos propuestos en cada una de las prácticas. Además se implementaron las prácticas y se crearon las bases que permite que los profesores las impartan y que sirve de guía a los estudiantes.

Recomendaciones:

1. La implementación de otras prácticas adicionales para la asignatura PDS II.
2. Montar las prácticas sobre algún software que permita la simulación.

Referencias Bibliográficas:

- Álvarez, C. Á. (2012). La relación teoría-práctica en los procesos de enseñanza-aprendizaje. *Educatio Siglo XXI*, 30(2), 383-402.
- ARDUINO, U. REV3 [https://www.arduino.cc/en/Main: ArduinoBoardUno](https://www.arduino.cc/en/Main:ArduinoBoardUno).
- Artero, Ó. T. (2013). *Arduino: curso práctico de formación*: RC Libros.
- Calvo, I., Zulueta, E., Gangoiti, U., López, J. M., Cartwright, H., & Valentine, K. (2009). *Laboratorios remotos y virtuales en enseñanzas técnicas y científicas* (Vol. 3): Ikastorratza.
- Frías, Y. C. (2016). Propuesta de prácticas de laboratorio para la asignatura Seguridad de redes inalámbricas.
- Iglesias, C., Manuel, C., & Mesa, M. C. (2008). Fundamentos teóricos para la implementación de la didáctica en el proceso enseñanza-aprendizaje. *Consultado en* [http://www. eumed. net/libros-gratis/2008b/395/CARACTERIZACION% 20DE% 20LA% 20CONFERENCIA. htm](http://www.eumed.net/libros-gratis/2008b/395/CARACTERIZACION%20DE%20LA%20CONFERENCIA.htm).
- Las prácticas de laboratorio importancia, diseño y elaboración. (2010).

Anexos:

Código filtrado digital en tiempo real de ruido

```
//A0 pin de lectura de la señal del mic
```

```
const int sign = A0;
```

```
int cont =0;
```

```
//coeficientes del filtro
```

```
const float coeff[]={-0.00124524577937771,-0.00280198539369773,0.0000006045039216,  
0.00630328657649506,0.00823661763838471,-0.00151431405782056,  
-0.014151805764884,-0.0125056431623591,0.00461013098918164,  
0.0159865225517666,0.00807653016950389,-0.00290328185391139,  
0.00224594284891645,0.00759719823859967,-0.0155188779102616,  
-0.0471417531356446,-0.0253465673428238,  
0.0572600967874697, 0.105432948687573, 0.0295343280364442,  
-0.111073177809333, -0.147660220290832, -0.0131440745330448,  
0.149359471577213, 0.149359471577213, -0.0131440745330448,  
-0.147660220290832, -0.111073177809333, 0.0295343280364442,  
0.105432948687573, 0.0572600967874697, -0.0253465673428238,  
-0.0471417531356446, -0.0155188779102616, 0.00759719823859967,  
0.00224594284891645, -0.00290328185391139, 0.00807653016950389,  
0.0159865225517666, 0.00461013098918164, -0.0125056431623591,  
-0.014151805764884, -0.00151431405782056, 0.00823661763838471,
```

```
0.00630328657649506, 0.0000006045039216, -0.00280198539369773,  
-0.00124524577937771};
```

```
//valores de la señal
```

```
int signlx[]={    0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0,  
                 0,0,0    };
```

```
//función que adiciona de forma ciclica los
```

```
// valores de la señal a la matriz
```

```
void add(int val){
```

```

if(cont<48){
    signlx[47-cont]=val;
    cont++;
}else{
    for(int i=0; i<48; i++){
        int v1 = signlx[i];
        signlx[i]=val;
        val=v1;
    }
}

float val = 0;

//funcion que aplica el filtro
float eval(){
    val = 0;
    for(int i = 0; i < 48; i++){
        val = val+ (signlx[47-i]*coeff[i]);
    }
    return val;
}

int[] a(){

```

```
}
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int sensor = analogRead(sign);  
  add(sensor);  
  Serial.println(eval());  
}
```

Código práctica karaoke:

```
int left[] = {0,0,0,0,0};
```

```
int righth[] = {0,0,0,0,0};
```

```
float out[] = {0,0,0,0,0};
```

```
const float a[] = { 0.000001552450732844424, 0.000006209802931377695,  
0.000009314704397066543, 0.000006209802931377695, 0.000001552450732844424};
```

```
const float b[] = { 1.0000000000000000,-3.811217313542288,5.451267531368796,-  
3.467935474171039,0.8279100955562573};
```

```

const int lR = A1;

const int rR = A2;


int cont =0;


void add(){

    int lV = analogRead(lR);

    int rV = analogRead(rR);

    if(cont<=4){

        left[cont]=lV;

        righth[cont]=rV;

    }else{

        for(int i=0; i<4; i++){

            left[i] = left[i+1];

            righth[i] = righth[i+1];

        }

        left[4]=lV;

        righth[4]=rV;

    }

}

```

```

void filter(){

    float o=0;

```

```

for(int i= 0; i<5 ; i++){

    int temp = righth[i] - left[i];

    o+=(a[i]*temp-b[i]*out[i]);

}

```

```

if(cont<=4){

    out[cont]=o;

}else{

    for(int i=0; i<4; i++){

        out[i] = out[i+1];

    }

    out[4]=o;

}

}

```

```

void setup() {

    // put your setup code here, to run once:

    Serial.begin(9600);

}

```

```

void loop() {

    // put your main code here, to run repeatedly:

    add();

    filter();

}

```

```

cont++;

for(int i=0;i<5;i++){

    Serial.println(out[i]);

}

}

```

Código práctica detector de frecuencias:

```

const int sign = A0;

const int amarillo = 5;

const int azul = 4;

const int rojo = 3;

const int verde = 2;


bool first = true;


void setup() {

    pinMode(amarillo, OUTPUT);

    pinMode(azul, OUTPUT);

    pinMode(rojo, OUTPUT);

    pinMode(verde, OUTPUT);

}


unsigned long timeV=0;

unsigned long timeR=0;

```



```

unsigned long timeA=0;

unsigned long timeAm=0;

unsigned long time2=0;


unsigned long timez=0;


int sensor = 0;


void loop() {

    sensor= analogRead(sign);


    if(first){

        timeV=micros();

        timeR=timeV;

        timeA=timeV;

        timeAm=timeV;

        first=false;

    }else{


        time2 = micros();


        //frecuencias menores a 100Hz

        if(time2 - timeV >=10000 and sensor>=100){

            digitalWrite(verde,HIGH);

```

```

    timeV=time2;

}else{

    digitalWrite(verde,LOW);

}

//frecuencias entre 100Hz y 200Hz

if(time2 - timeR < 10000 and time2 - timeR >=5000 and sensor>=100){

    digitalWrite(rojo,HIGH);

    timeR=time2;

}else{

    if(time2 - timeR >=10000){

        timeR=time2;

    }

    digitalWrite(rojo,LOW);

}

//frecuencias entre 200Hz y 400Hz

if(time2 - timeA < 5000 and time2 - timeA >=2500 and sensor>=100){

    digitalWrite(azul,HIGH);

    timeA=time2;

}else{

    if(time2 - timeA > 10000){

        timeA=time2;

    }

}

```

```

    digitalWrite(azul,LOW);
}

//frecuencias entre 400Hz y 800Hz
if(time2 - timeAm < 2500 and time2 - timeAm >=1250 and sensor>=100){
    digitalWrite(amarillo,HIGH);
    timeAm=time2;
}else{
    digitalWrite(amarillo,LOW);
    if(time2 - timeAm >=10000){
        timeAm=time2;
    }
}

delayMicroseconds(10);
}
}

```

Código práctica filtro adaptativo:

```

int D[] = {HIGH,LOW,LOW,HIGH,HIGH,HIGH,LOW,HIGH,LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH};
int X[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int Y[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int W[] = {0.3,0.9,0.5,0.1,0.8,0.4,0.7,0.11,-0.99,0.6,-0.33,0.9,-0.1,0.3,-0.4};
int const in = A3;

```

```
int n =0;
```

```
const int out = 6;
```

```
int E[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```
void add(int n){  
    int xn = analogRead(in);
```

```
    xn/=1024;
```

```
    if(n<15){
```

```
        X[n]=xn;
```

```
    }else{
```

```
        for(int i=0; i<14; i++){
```

```
            X[i] = X[i+1];
```

```
        }
```

```
        X[14]=xn;
```

```
    }
```

```
}
```

```
void setup() {
```

```
    // put your setup code here, to run once:
```

```
    Serial.begin(9600);
```

```
    pinMode(out, OUTPUT);
```

```
}
```

```

void loop() {

    // put your main code here, to run repeatedly:

    //Serial.println(D[0]);

    if(n <15){

        digitalWrite(out,D[n]);

        delay(1000);

        add(n);

        Serial.print(n);

        Serial.print(": ");

        Serial.println(X[n]);

        for(int i=0;i<15;i++){

            Y[n]+=W[i]*X[i];

            E[n] = D[n]-Y[n];

        }

        for(int i=1;i<15;i++){

            W[i]= W[i-1] -2 * E[n] * X[n-1];

        }

        n++;

    }else{

        for(int i=0;i<15;i++){

            Serial.println(W[i]);

        }

    }
}

```

}