

Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Cuautitlán

Nombre:

Edgar Palomino Alfonso

Carrera:

Ingeniería en Telecomunicaciones, Sistemas y Electrónica

No.Cuenta:

420075710

Correo electronico:

erreapesrk@gmail.com

Dependencia donde se relizo el servicio social:

Facultad de Estudios Superiores Cuautitlán Campo 4

Nombre del programa:

Apoyo a las prácticas y apuntes del laboratorio de control digital y
del laboratorio de electrónica

Responsable:

I.M.E. Héctor Miguel Santoyo Morales

Fecha de inicio y término del servicio social:

01 de marzo del 2023 al 01 de septiembre del 2023



Índice

1. Introducción.	2
2. Objetivo general del programa.	2
3. Objetivo específico del alumno en el programa de servicio social.	2
4. Descripción del programa en que se prestó el servicio social.	2
5. Resultados obtenidos.	7
6. Resultados en beneficio a la sociedad.	16
7. Resultados en cuanto a la formación profesional.	16
8. Conclusiones.	17



1. Introducción.

En la Facultad de Estudios Superiores Cuautitlán específicamente en el departamento de Ingeniería, se desglosan varias áreas. El área donde se realizó el servicio social fue en el área de Control, específicamente en la asignatura de Control Digital en donde se ven temas relacionados como las transformadas Z, controladores PI, ecuaciones en diferencia entre otros temas.

En este servicio social nos enfocamos en el tema de identificación de sistemas donde su principal función es obtener el modelo de un sistema dinámico a partir de datos experimentales, con la ayuda de la herramienta de Matlab veremos cómo se comportan dichos sistemas.

2. Objetivo general del programa.

Realización, revisión y corrección de apuntes y prácticas de los laboratorios de control digital y electrónica.

3. Objetivo específico del alumno en el programa de servicio social.

Mejorar las prácticas del laboratorio de control digital y electrónica aplicando los conocimientos aprendidos a lo largo de la carrera.

4. Descripción del programa en que se prestó el servicio social.

La identificación del sistema es un método experimental en la cual se realizan algunas pruebas sobre el sistema que permite obtener los datos para estimar el valor de los parámetros del modelo representativo del sistema. El procedimiento para la identificación de un sistema son los siguientes puntos:

- Diseño del experimento de identificación. En esta primera etapa es necesario decidir, entre otros aspectos; el tipo de señales de excitación, el mejor periodo para la adquisición de datos, la cantidad de datos necesarios.
- Observación y mejora de la calidad de los datos capturados. Antes de utilizar los métodos de estimación de parámetros es necesario observar y reparar los datos erróneos, filtrar las altas frecuencias, eliminar offsets y tendencias.
- Determinación de la estructura del modelo. En esta etapa es necesario definir el tipos de modelos a utilizar, continuos o discretos, tipos de ruido, lineales o no lineales, regresiones, redes neuronales, ... Y es necesario adoptar un procedimiento para determinar el orden del modelo.
- Estimación de los parámetros. Etapa la mayoría de las veces muy relacionada con la anterior, en ella se presenta el problema de decidir el método o métodos de estimación de parámetros que se va a utilizar para calcular el valor de estos.
En general se puede escoger entre dos técnicas distintas: en el dominio temporal y en el dominio frecuencial.



- Validación del modelo. Es la etapa en la que debe preguntarse si el modelo identificado es suficientemente representativo del proceso estudiado. En él se debe definir un criterio para evaluar la calidad. Generalmente se dispone de varios modelos candidatos y debe escogerse uno de ellos basándose en algún criterio.

A través de la metodología descrita se procedió de la siguiente manera en los pasos del proyecto realizado. Se procedió a realizar el armado de circuitos electrónicos para corroborar los resultados obtenidos de manera teórica en el aula de clases, brindando a los alumnos la oportunidad de experimentar y validar los conceptos aprendidos en un entorno práctico.

Al armar los circuitos electrónicos, los alumnos tienen la posibilidad de aplicar los conocimientos teóricos adquiridos durante la clase control digital así como del resto de materias del área electrónica que ayudan a comprender y aplicar los diferentes circuitos implicados en los resultados finales, esto ayuda a poner en práctica los principios junto a las técnicas discutidas durante estos. Lo que nos permite visualizar de manera concreta cómo funcionan los componentes electrónicos, cómo interactúan entre sí, así como cuáles son los efectos sobre el comportamiento global del circuito. Al experimentar con diferentes configuraciones y parámetros, se puede comprobar de manera directa si los resultados que se esperan según la teoría son reales o como sucede en diferentes casos esto difiere de la realidad, con lo que se debe de corregir y parametrizar los valores correctos con los cuales se obtenga el resultado deseado. El armado de circuitos electrónicos también conlleva la oportunidad de desarrollar habilidades prácticas, como el manejo de herramientas, la soldadura y el montaje de componentes. Reforzar el conocimiento para interpretar esquemas o diagramas, a seleccionar los componentes adecuados para proseguir con el procedimientos de montaje y conexión. Estas habilidades son fundamentales en el área de electrónica, ya que nos proporciona una base sólida para trabajar en el diseño, así como la implementación de sistemas electrónicos. Durante el armado de circuitos electrónicos, se tuvo la oportunidad de identificar posibles problemas y errores en los diseños. Esto nos ayuda a desarrollar habilidades de solución de problemas, debido a el análisis para el diagnóstico de las fallas, para posteriormente realizar las correcciones necesarias lo que permite obtener los resultados esperados. Esta experiencia nos permitió comprender la importancia de la precisión, la atención en el diseño además de la implementación de circuitos electrónicos. Para este punto se puede destacar ejemplo prácticos como el armado de un generador de señal triangular con un CI 555, que al pasar por un filtro de primer orden se puede obtener una señal cuadrada que sirve como entrada a un comparador esto con el fin de generar un nivel de offset de 0 volts, el cual anula un voltaje de offset diferente de 0 volts que se genera al salir de una pin de propósito general debido a que los valores que se toman como salida digital comprenden un valor ligeramente mayor a 0 para no tener problemas al momento de utilizar la salida, pero que para el caso de control esto puede representar un problema ya que el análisis de la señal se hace referenciado a un nivel de 0 volts. El circuito armado se puede apreciar en la figura 1.

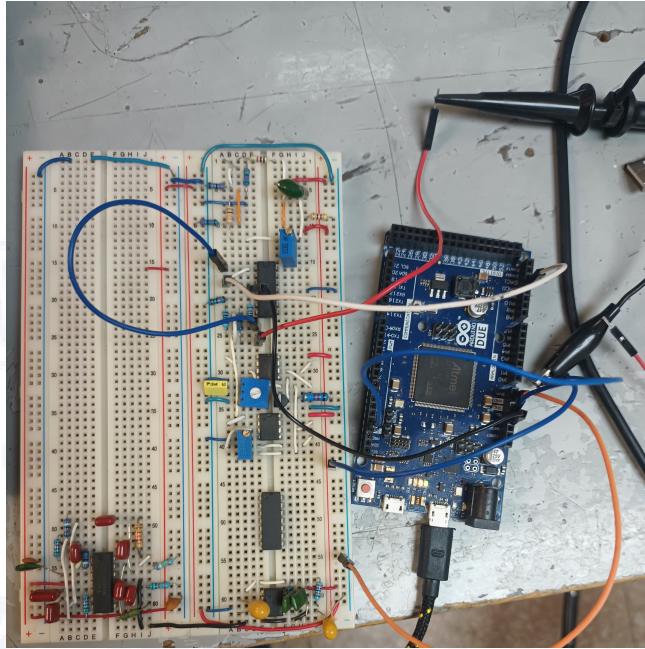


Figura 1: Circuito generador de onda triangular.

Otro proyecto a destacar es el armado de filtros analógicos, donde se usaron los modelos Butterworth que es un tipo de filtro utilizado en el procesamiento de señales para lograr una respuesta de frecuencia plana en la banda de paso y una rápida atenuación de las frecuencias fuera de esta banda. Es conocido por su característica de respuesta plana.^{en} la región de paso, lo que significa que no introduce distorsiones significativas en la señal en esa banda.

El filtro de Butterworth se basa en la función de transferencia del polinomio de Butterworth, que se caracteriza por tener una caída gradual en la respuesta de frecuencia en lugar de una caída abrupta. Esto significa que el filtro de Butterworth no tiene una capacidad de rechazo de frecuencias tan pronunciada como otros tipos de filtros, como los filtros de Chebyshev de los cuales también se realizaron armados de circuitos. Sin embargo, su ventaja radica en su respuesta plana en la banda de paso, lo que lo hace adecuado para aplicaciones en las que se requiere una atenuación suave fuera de la banda de interés.

Por su parte el filtro Chebyshev es otro tipo de filtro ampliamente utilizado en el procesamiento de señales para lograr una respuesta de frecuencia específica. A diferencia del filtro de Butterworth, el filtro de Chebyshev permite un control más preciso de la respuesta de frecuencia, lo que significa que puede proporcionar una atenuación más pronunciada de las frecuencias fuera de la banda de paso.

El filtro de Chebyshev se basa en los polinomios de Chebyshev, que tienen la propiedad de minimizar el error máximo entre la respuesta de frecuencia ideal y la respuesta real del filtro. Esto permite un diseño más eficiente para obtener una atenuación específica de las frecuencias no deseadas.

La principal ventaja del filtro de Chebyshev es su capacidad para lograr una alta tasa de atenuación de las frecuencias fuera de la banda de paso, lo que lo hace adecuado para aplicaciones que requieren una fuerte supresión de las frecuencias no deseadas. Sin embargo, esto puede estar acompañado de ondulaciones en la respuesta de frecuencia en la banda de paso o corte.

Para una comprensión rápida entre las diferencias se tiene la siguiente figura 2, donde se aprecia la diferencia de las respuesta en los rangos de frecuencia de cada uno de los filtros.

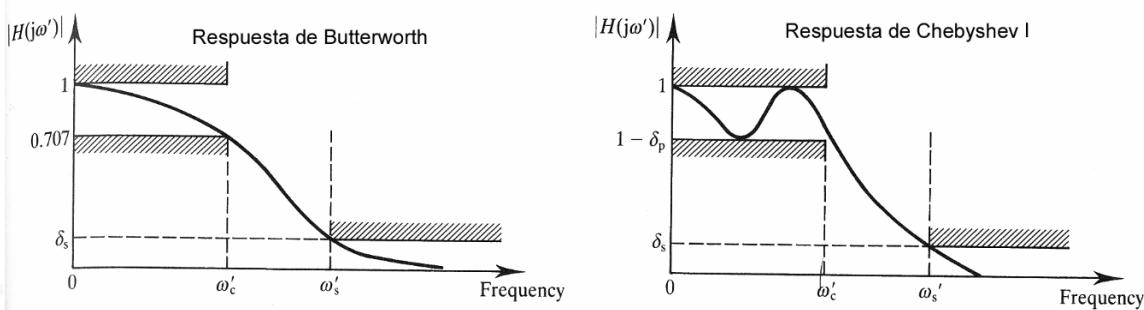


Figura 2: Respuesta de los filtros.

Donde se puede apreciar como la caída del Filtro Butterworth es más suave pero sin ondulaciones, mientras que el filtro chebyshev tiene una caída más brusca pero con unas ondulaciones, su uso dependerá de las necesidades del proyecto, cosa que se hizo notorio mientras hacíamos la pruebas de estos filtros.

Para el armado del circuito se hizo uso de modelos chebyshev el cual conviene más a nuestro interés de generar una banda de paso más exacta, es decir donde la banda de paso debía de ser más tajante con las señales que requerimos que pasen así como las que no requerimos o no queremos dejar pasar ya que deseamos usarlo para filtrar señales de audio cuyo espectro electromagnético es corto, la imagen siguiente es de un filtro de segundo orden chebyshev con frecuencia de paso de 10 a 100 hz, la mejor manera de identificar que es de segundo orden es debido a que tiene 2 pares de resistencias y capacitores, es decir por cada capacitor con resistencia podemos asegurar el orden del filtro, de esta manera el circuito presentado en la figura 3, se puede apreciar un integrador correspondiente a un amplificador operacional acompañado de resistencia y capacitadores para configurarlo como un filtro de segundo orden.

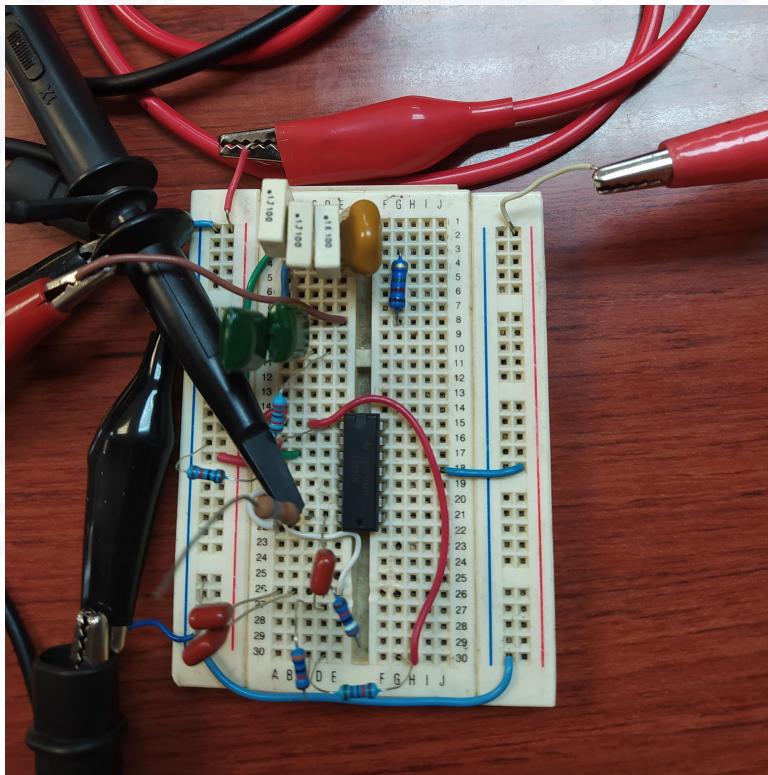


Figura 3: Circuito del filtro pasa banda de segundo orden.



Para este caso particular a armar este filtro no se tenían los valores exactos de capacitores, con lo cual se tuvo que armar de tal manera que la suma y configuración de los capacitores resultaron en un valor cercano a la cual se requería para de esta manera obtener los resultados deseados.

Una vez que se realizaba el funcionamiento así como la obtención de los circuitos analogicos se utilizaban la transformada de laplace, la transformada z para después hacer el despeje de las ecuaciones resultantes consiguiendo una ecuación en diferencias que representará al sistema, esta se modelaba para poder hacer uso de un arduino, el cual nos permite pasar de una manare sencilla de lo analogico a digital, esto debido a que la plataforma ya integra un ambiente de desarrollo tanto en hardware como software que por algunos es criticado mientras que por otros es amado debido a la facilidad con la que se puede hacer implementaciones de proyectos digitales, pero más allá de eso está el entendimiento del porqué es que estos system on a chip (SoC), funcionan así, también como es que podemos implementarlos en proyectos más avanzados, tomando en cuenta el concepto de utilizar tecnología de vanguardia para los retos del campo laboral moderno o simplemente tener la facilidad y práctica de las nuevas tecnologías e implementarlas en proyectos personales que siendo ambiciosos pueden genera una solución necesaria a problemas actuales que se tienen en la sociedad.

Como último punto durante el desarrollo del servicio desafortunadamente o afortunadamente se tuvo la necesidad de realizar el ensamblaje de circuitos para proyectos externos lo cual nos dio pie a hacer uso de tarjetas perforadas donde se soldaron terminales conocidos como headers para la integración de diferentes módulos a la tarjeta de arduino, dando a la tarea de aprender a utilizar de manera correctas las diferentes opciones para realizar los puntos de soldadura sobre las placas, dando como resultado los circuitos de la figura 4 y figura 5.

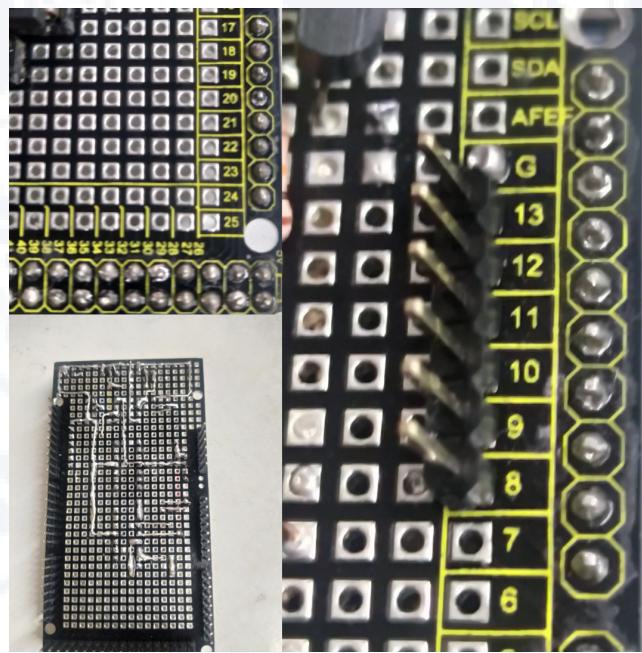


Figura 4: Tarjeta con componentes soldados.

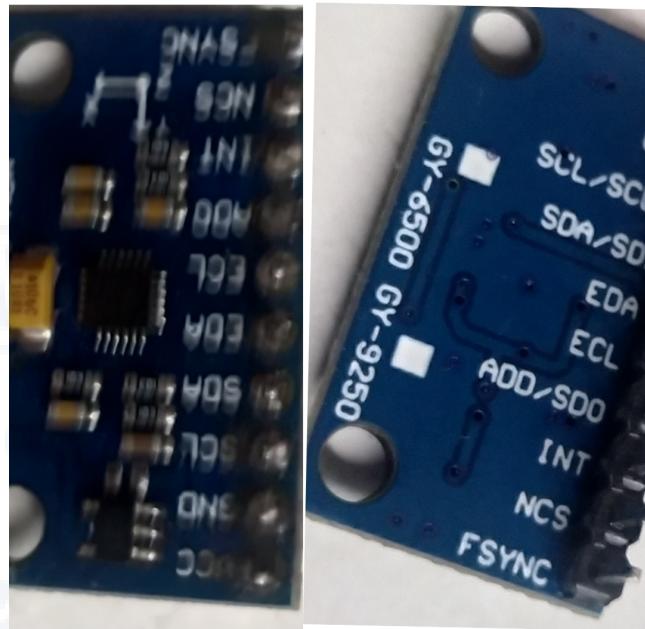


Figura 5: MPU soldado.

5. Resultados obtenidos.

Se hace implementación de un panel de instrumentación electrónica básica, para poder obtener las lecturas desde una tarjeta Arduino, la cual sirve como tarjeta de adquisición de datos, la interfaz recibe dos señales analogicas a través de los pines A0 y A1, pero modificando el código podemos recibir las señales que se requieran, aunque se consideró que serían suficientes para nuestro propósito, donde con una sola señal podemos realizar el control de temperatura de un sistema, además se pueden enviar dos señales de PWM para el control de la intensidad lumínica de un led. así que podemos escribir y recibir instrucciones desde dicha interfaz. En conjunto se diseñó una tarjeta perforada para poder tratar la señal, a través de dos procesos que son los siguientes:

- Limpiar la señal.

Al tener componentes externos, generamos ruido dentro de la adquisición de la señal con lo que debemos implementar un filtro que ayude a tener una lectura más precisa, esto se hace mediante un filtro Butterworth pasa bajas de segundo orden.

- Eliminar el offset.

un caso particular ocurre al querer obtener la señal desde la tarjeta o pasando por la tarjeta y es que este le genera un offset de 0.5 volts el cual mueve el eje real de donde se obtuvo la señal, con lo cual es necesario antes de representarla para obtener la tabulación de los datos, quitar dicho voltaje de offset y mantener una lectura real de la entrada adquirida por la tarjeta, esto se hace con un generador diente de sierra mediante una configuración de un IC 555 que puede contrarrestar este efecto.

Después de tratar las señales de manera analógica podemos usar la interfaz gráfica sin errores. La cual hace uso de Python para generar una imagen animada de las señales obtenidas mediante los puertos indicados además de un estado de los leds que se pueden controlar, adicional a eso se denota que originalmente se tenía pensado hacer uso de visual studio para a través de la interface



grafica poder generar dicho panel, pero esto limita su uso a solo windows, ya que se genera un archivo .net, debido a esto y haciendo uso de tecnologías Open Source, se optó por a través de Python y librerías como PyQt5, matplotlib y Pyserial hacer una interfaz multiplataforma.

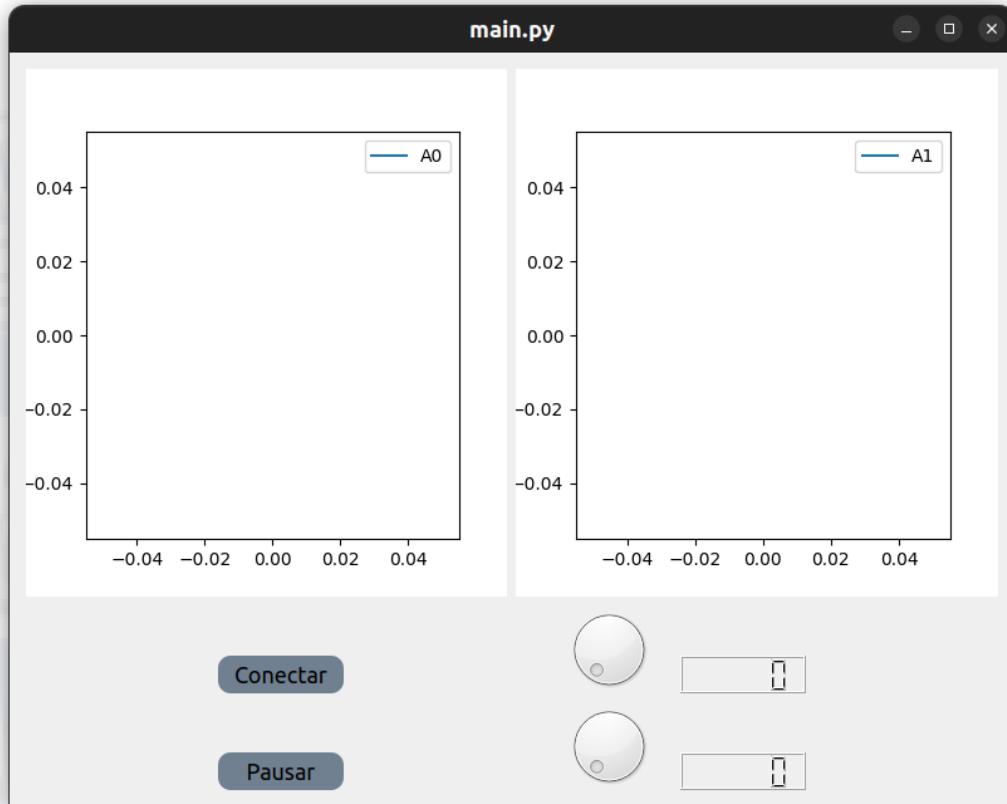


Figura 6: Interfaz.

En la figura 6 podemos apreciar la interfaz limpia, donde se aprecia los botones para conectar, desconectar, pausar y reanudar la gráfica, esto se hace manejando los estados de la conexión serial y actualizando los datos sobre la interface. a continuación explicare que es cada estado.

- **Conectado.**

Mediante la librería Pyserial podemos conectar y desconectar la tarjeta por completo del interfaz, de esta manera no se envían ni reciben datos, esto es lo que maneja este botón, pasando entre los estados, ademas de indicarlo con un cambio de color sobre al interfaz.

- **Pausar.**

Este botón aisla las gráficas, si queremos hacer una captura de algún pedazo de la gráfica en un tiempo específico podemos hacer uso de este botón ya que este dejará de reportar el estado del puertos analogicos a la grafica, pero sin desconectar la tarjeta, con lo cual podemos seguir manejando el estado del led.

- **Diales.** Las ruedas que se aprecian sobre las gráficas sirven para de manera digital pasar un estado a los leds o en otras palabras cambiar la intensidad de la iluminación, cosa que se aprecia en la figura 7 gracias al display, además podemos verificarlo en la figura 8 con las intensidades del led.



Figura 7: Elemento display de la interfaz

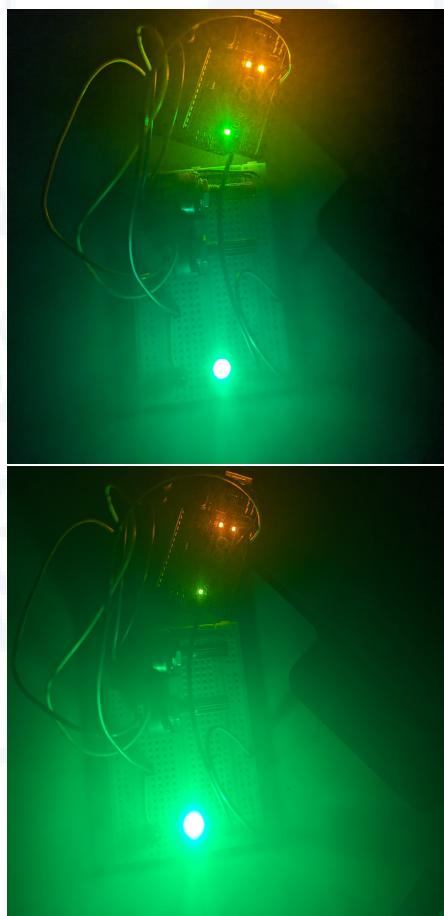


Figura 8: Intensidad del led



Figura 9: Graficas del potenciómetro

Como se ha mencionado la intención de este panel es poder visualizar las señales que se obtiene mediante un puerto analógico de la tarjeta, pero haciendo uso de todo lo desarrollado durante el curso de control digital para adecuar la señal y controlar la temperatura de un sistema o cualquier uso que se le quiera dar al poder obtener la señal de la tarjeta sin necesidad de un osciloscopio o no de manera indispensable.

Por último el código utilizado para crear la interfaz. se divide por módulos, tomando como la convención dicta, el módulo main como el punto de entrada de toda la aplicación, tenemos el módulo comunicación serial, que hace el enlace entre Python y Arduino, el módulo Gráfica genera toda la animación de la interfaz.[1]



```
1 # main.py
2 import sys
3 from PyQt5.QtWidgets import QApplication
4 from Grafica import Grafica
5
6 if __name__ == '__main__':
7     app = QApplication(sys.argv)
8     ventana = Grafica()
9     ventana.show()
10    sys.exit(app.exec_())
```

Código 1 : Main

```
1 # ComunicacionSerial.py
2 import serial
3
4 class ComunicacionSerial:
5     def __init__(self, puerto, baudios):
6         self.puerto = puerto
7         self.baudios = baudios
8         self.conexion = None
9
10    def conectar(self):
11        try:
12            self.conexion = serial.Serial(self.puerto, self.baudios, timeout=1)
13            print(f"Conectado al puerto {self.puerto}")
14        except serial.SerialException as e:
15            print(f"Error de conexión: {e}")
16
17    def desconectar(self):
18        if self.conexion and self.conexion.isOpen():
19            self.conexion.close()
20            print("Desconectado")
21
22    def enviar_datos(self, datos):
23        if self.conexion and self.conexion.isOpen():
24            self.conexion.write(datos.encode())
25            print(f"Datos enviados: {datos}")
26
27    def recibir_datos(self):
28        if self.conexion and self.conexion.isOpen():
29            return self.conexion.readline().decode().strip()
30
31    def cerrar_puerto(self):
32        if self.conexion and self.conexion.isOpen():
33            self.conexion.close()
```

Código 2: Comunicación serial entre Arduino y Python



```
1 # Grafica.py
2 import matplotlib.pyplot as plt
3 from PyQt5.QtWidgets import QWidget, QVBoxLayout, QPushButton, QHBoxLayout,
4     QSlider, QLabel,QDial,QLCDNumber
5 from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
6 from PyQt5.QtCore import QTimer, Qt
7 from ComunicacionSerial import ComunicacionSerial
8
9 crimson = '#DC143C'
10 seaGreen = '#2E8B57'
11
12 steelBlue = '#4682B4'
13 slateGray = '#708090'
14 violetRed = '#DB7093'
15
16 des = '#7FFFDD'
17 con = '#CD5C5C'
18
19 class Grafica(QWidget):
20     def __init__(self, parent=None):
21         super().__init__(parent)
22         self.comunicacion = ComunicacionSerial("/dev/ttyACM0", 9600)
23         self.comunicacion.conectar()
24
25         self.layout_principal = QVBoxLayout(self)
26         self.inicializar_ui()
27
28         self.conectado = False
29         self.timer = QTimer(self)
30         self.timer.timeout.connect(self.actualizar_graficas)
31         self.timer_envio_datosA = QTimer(self)
32         self.timer_envio_datosA.timeout.connect(self.enviar_datos_timerA)
33         self.timer_envio_datosB = QTimer(self)
34         self.timer_envio_datosB.timeout.connect(self.enviar_datos_timerB)
35         self.setGeometry(300, 300, 800, 600)
36
37     def inicializar_ui(self):
38         #LAYOUT
39         self.layout_graficas = QHBoxLayout()
40         self.layout_slider_boton = QHBoxLayout()
41         self.layout_slider = QVBoxLayout()
42         self.layout_boton = QVBoxLayout()
43         self.layout_lcd = QVBoxLayout()
44
45         #GrAFICAS
46         self.inicializar_grafica("AO")
47         self.inicializar_grafica("A1")
48
49         #BOTONES
50         self.btn_conectar = QPushButton('Conectar', self)
51         self.btn_conectar.clicked.connect(self.conectar_desconectar)
52         self.btn_conectar.setStyleSheet(f"background-color: {slateGray};
```



```
        border-radius: 10px; text-align: center;")  
    self.btn_conectar.setFixedSize(100, 30)  
  
    51  
    52    self.btn_pausar = QPushButton('Pausar', self)  
    self.btn_pausar.clicked.connect(self.pausar_graficas)  
    self.btn_pausar.setStyleSheet(f"background-color: {slateGray};  
        border-radius: 10px; text-align: center;")  
    self.btn_pausar.setFixedSize(100, 30)  
  
    55  
    56  
#DIAL  
    57    self.dial_ledA = QDial()  
    58    self.dial_ledA.setRange(0, 255)  
    59    self.dial_ledA.valueChanged.connect(self.control_ledA)  
  
    61  
    62    self.valor_dialA = QLCDNumber(self)  
    63    self.valor_dialA.setFixedSize(100, 30)  
  
    64  
    65    self.dial_ledB = QDial()  
    66    self.dial_ledB.setRange(0, 255)  
    67    self.dial_ledB.valueChanged.connect(self.control_ledB)  
#self.dial_ledB.setFixedSize(100, 30)  
  
    69  
    70    self.valor_dialB = QLCDNumber(self)  
    71    self.valor_dialB.setFixedSize(100, 30)  
#self.valor_dialB.setStyleSheet(f"background-color: {slateGray};  
        border-radius: 10px; text-align: center; color = {violetRed};")  
  
    73  
    74  
    75  
#AGREGANDO  
    76    self.layout_principal.addLayout(self.layout_graficas)  
    77    self.layout_boton.addWidget(self.btn_conectar, alignment=Qt.AlignBottom |  
        Qt.AlignRight )  
    78    self.layout_boton.addWidget(self.btn_pausar, alignment=Qt.AlignBottom |  
        Qt.AlignRight )  
    79    self.layout_slider.addWidget(self.dial_ledA, alignment=Qt.AlignBottom |  
        Qt.AlignRight )  
    80    self.layout_lcd.addWidget(self.valor_dialA, alignment=Qt.AlignBottom |  
        Qt.AlignLeft )  
    81    self.layout_slider.addWidget(self.dial_ledB, alignment=Qt.AlignBottom |  
        Qt.AlignRight )  
    82    self.layout_lcd.addWidget(self.valor_dialB, alignment=Qt.AlignBottom |  
        Qt.AlignLeft )  
  
    84  
    85    self.layout_slider_boton.setLayout(self.layout_boton)  
    86    self.layout_slider_boton.setLayout(self.layout_slider)  
    87    self.layout_slider_boton.setLayout(self.layout_lcd)  
    88    self.layout_principal.setLayout(self.layout_slider_boton)  
  
    89  
    90  
def inicializar_grafica(self, nombre):
```



```
92     canvas = FigureCanvas(plt.Figure())
93     self.layout_graficas.addWidget(canvas)
94     ax = canvas.figure.add_subplot(111)
95     line, = ax.plot([], [], label=nombre)
96     ax.legend()
97
98     setattr(self, f"canvas_{nombre}", canvas)
99     setattr(self, f"ax_{nombre}", ax)
100    setattr(self, f"line_{nombre}", line)
101
102   def conectar_desconectar(self):
103       if not self.conectado:
104           self.comunicacion.conectar()
105           self.btn_conectar.setText('Desconectar')
106           self.btn_conectar.setStyleSheet(f"background-color: {des};"
107                                           "border-radius: 10px; text-align: center;")
108           self.timer.start(100)
109       else:
110           self.comunicacion.desconectar()
111           self.btn_conectar.setText('Conectar')
112           self.btn_conectar.setStyleSheet(f"background-color: {con};"
113                                           "border-radius: 10px; text-align: center;")
114           self.timer.stop()
115
116   self.conectado = not self.conectado
117
118   def actualizar_graficas(self):
119       if not self.timer.isActive():
120           return
121
122       datos_A0 = self.comunicacion.recibir_datos()
123       datos_A1 = self.comunicacion.recibir_datos()
124
125       if datos_A0 and datos_A1:
126           valor_A0 = float(datos_A0)
127           valor_A1 = float(datos_A1)
128
129           self.actualizar_grafica("A0", valor_A0)
130           self.actualizar_grafica("A1", valor_A1)
131
132   def actualizar_grafica(self, nombre, valor):
133       line = getattr(self, f"line_{nombre}")
134       ax = getattr(self, f"ax_{nombre}")
135       max_puntos = 100
136
137       x = list(line.get_xdata())
138       y = list(line.get_ydata())
139
140       x.append(len(x))
141       y.append(valor)
```



```
141     if len(x) > max_puntos:
142         x = list(range(max_puntos))
143         y = y[-max_puntos:]
144
145     line.set_data(x, y)
146     ax.relim()
147     ax.autoscale_view()
148
149     canvas = getattr(self, f"canvas_{nombre}")
150     canvas.draw()
151
152
153
154     def pausar_graficas(self):
155         if self.timer.isActive():
156             self.timer.stop()
157             self.btn_pausar.setText('Reanudar')
158             self.btn_pausar.setStyleSheet(f"background-color: {steelBlue};"
159                                         "color:white; border-radius: 10px; text-align: center;")
160         else:
161             self.timer.start(100)
162             self.btn_pausar.setText('Pausar')
163             self.btn_pausar.setStyleSheet(f"background-color: {seaGreen};"
164                                         "color:white; border-radius: 10px; text-align: center;")
165
166
167     def control_ledA(self, valor):
168         if isinstance(valor, (int, float)):
169             valor = int(valor)
170             self.valor_dialA.display(valor)
171             self.timer_envio_datosA.start(100)
172
173     def enviar_datos_timerA(self):
174         valor = self.dial_ledA.value()
175         self.comunicacion.enviar_datos(f'LED1:{valor}\n')
176         self.timer_envio_datosA.stop()
177
178     def control_ledB(self, valor):
179         if isinstance(valor, (int, float)):
180             valor = int(valor)
181             self.valor_dialB.display(valor)
182             self.timer_envio_datosB.start(100)
183
184     #enviar datos desde el timer para el led
185     def enviar_datos_timerB(self):
186         valor = self.dial_ledB.value()
187         self.comunicacion.enviar_datos(f'LED2:{valor}\n')
188         self.timer_envio_datosB.stop()
```

Código 3: Animación de la aplicación



6. Resultados en beneficio a la sociedad.

Incorporar una evaluación de los beneficios directos generados por mi labor implica destacar los resultados positivos y cambios tangibles que aportaron a la sociedad. En primer lugar, se logró una actualización significativa en la formación profesional de la comunidad, dotándola de habilidades más avanzadas para abordar problemáticas mediante el uso de la tecnología actual. Esta iniciativa puede crear un impacto directamente en el aumento de profesionales capacitados y preparados.

Además, se proporcionó a la comunidad un panorama amplio de las nuevas tecnologías disponibles, permitiéndoles aprovechar al máximo las herramientas más innovadoras. La difusión de este conocimiento contribuyó a mejorar la eficiencia y competitividad en diversas áreas de conocimiento.

La implementación de proyectos prácticos y tangibles en los laboratorios resultará en beneficios de los grupos posteriores de alumnos. Estos proyectos no sólo ofrecen soluciones innovadoras a problemas locales, sino que también servirán como ejemplos inspiradores para futuras iniciativas similares.

Una de las metas alcanzadas fue la reducción progresiva del gasto en componentes individuales, que históricamente se convertían en solo material de práctica. Este enfoque estratégico no solo puede optimizar los recursos financieros, sino que también fomenta una práctica más sostenible y responsable.

En resumen, los beneficios directos derivados de mi labor abarcan desde la formación profesional avanzada y la difusión de conocimientos tecnológicos hasta la implementación de proyectos concretos y la gestión eficiente de recursos, impactando positivamente en la sociedad y sentando las bases para un futuro prometedor.

7. Resultados en cuanto a la formación profesional.

Dentro de mi formación escolar, he reconocido que la experiencia que puedo presentar a un futuro empleador puede parecer limitada. Sin embargo, el proyecto de desarrollo de una interfaz gráfica para un panel de control asistido por Arduino y Python ha sido una oportunidad invaluable para demostrar habilidades fundamentales que trascienden la teoría académica.

Este proyecto no solo implicó la aplicación de conocimientos técnicos, sino que también resaltó mi capacidad para llevar a cabo un proyecto desde sus fases iniciales hasta su implementación. A lo largo de este proceso, tuve que enfrentarme a los desafíos de definir requerimientos, planificar adecuadamente, llevar a cabo investigaciones detalladas y colaborar efectivamente en un equipo multidisciplinario.

En particular, la fase de desarrollo de código en lenguajes de programación, como Python, me permitió consolidar y ampliar mis habilidades en esta área. Durante los últimos semestres, he trabajado intensamente con diversos lenguajes. Esta diversidad me ha brindado la capacidad de evaluar recursos disponibles, comprender los objetivos de un sistema y analizar las ventajas y desventajas de distintas soluciones.

El tránsito entre sistemas de lenguajes de programación me ha enseñado a realizar un análisis completo de un problema, considerando las particularidades de cada lenguaje. Esto no solo implica la capacidad de escribir código, sino también de comprender la estructura y la sintaxis de manera fluida. Podemos decir que, este enfoque me ha permitido desarrollar la habilidad de proponer soluciones exhaustivas a problemas, esencial en el ámbito de la ingeniería.



El servicio social, en este contexto, ha actuado como un catalizador de aprendizaje, proporcionándome herramientas, metodologías, conocimientos y práctica para abordar problemáticas del mundo actual. La resolución de problemas se ha vuelto una segunda naturaleza, respaldada por la comprensión de que existen diversos caminos para alcanzar una meta, y cada uno de ellos puede derivarse del entendimiento y las limitantes de una solución frente a un problema.

8. Conclusiones.

En este trabajo se ha desarrollado un panel de instrumentación electrónica básica utilizando una tarjeta Arduino como tarjeta de adquisición de datos. Se implementó un filtro Butterworth pasa bajas de segundo orden para limpiar la señal adquirida y se utilizó un generador diente de sierra con un IC 555 para eliminar el offset de la señal.

La interfaz gráfica fue desarrollada utilizando Python y las librerías PyQt5, matplotlib y Pyserial, lo que permitió generar una imagen animada de las señales obtenidas y controlar la intensidad lumínica de un LED. Además, se implementaron botones para conectar, desconectar, pausar y reanudar la gráfica, así como ruedas para cambiar la intensidad de iluminación del LED.

Los resultados obtenidos demuestran que este panel de instrumentación es una alternativa viable para visualizar y controlar señales adquiridas a través de una tarjeta Arduino, sin necesidad de utilizar un osciloscopio. Esto puede ser de gran utilidad en aplicaciones de control de temperatura u otras aplicaciones que requieran el análisis de señales analógicas.

En cuanto a los beneficios generados por este trabajo, se destaca la actualización de la formación profesional de la comunidad, dotándola de habilidades más avanzadas en el uso de tecnología actual. Además, se proporcionó a la comunidad un panorama amplio de las nuevas tecnologías disponibles, mejorando la eficiencia y competitividad en diversas áreas de conocimiento. Los proyectos prácticos implementados en los laboratorios servirán como ejemplos inspiradores para futuras iniciativas similares.

En conclusión, este trabajo ha logrado desarrollar un panel de instrumentación electrónica básica que permite visualizar y controlar señales adquiridas a través de una tarjeta Arduino. Los beneficios generados incluyen la actualización de la formación profesional, la difusión de conocimientos tecnológicos, la implementación de proyectos concretos y la gestión eficiente de recursos. Este trabajo sienta las bases para un futuro prometedor en el campo de la instrumentación electrónica.

Referencias

- [1] B. Meier, *Python GUI Programming Cookbook: Develop functional and responsive user interfaces with tkinter and PyQt5*. Packt Publishing Ltd, 2019.