

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**CONSTRUCCIÓN DE 5 MÓDULOS DIDÁCTICOS PARA
CONECTAR Y CONTROLAR LAS ENTRADAS Y SALIDAS DE
ARDUINO DESDE MATLAB**

DISEÑO DE LOS MÓDULOS DIDÁCTICOS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO
SUPERIOR EN ELECTROMECAÁNICA**

JARAMILLO GONZÁLEZ CARLA PRISCILA

DIRECTOR: ARMAS FREIRE ELIZABETH CATALINA

QUITO, OCTUBRE 2023

CERTIFICACIONES

Yo, CARLA PRISCILA JARAMILLO GONZÁLEZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

CARLA PRISCILA JARAMILLO GONZÁLEZ

carla.jaramillo@epn.edu.ec

litalow_j7@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por CARLA PRISCILA JARAMILLO GONZÁLEZ, bajo mi supervisión.

Elizabeth Armas Freire

DIRECTOR

elizabeth.armas@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CARLA PRISCILA JARAMILLO GONZÁLEZ

DEDICATORIA

Dedico este logro a mis padres, cuyo amor y comprensión incondicional me han guiado a lo largo de este viaje. A mi padre, cuyo ejemplo a seguir me inspira a ser perseverante, y a mi madre, cuyo amor ha sido mi fuerza motriz. A mi querido hermano, cuyo constante apoyo me ha impulsado a alcanzar nuevas alturas. A mi novio, quien no solo me brindó su inquebrantable apoyo, sino que también se convirtió en mi fuente de fuerza en los momentos en que la tesis parecía abrumadora.

Esta tesis es el resultado del amor, la comprensión y el respaldo de estas personas excepcionales en mi vida. Gracias por ser mi inspiración y mi impulso constante.

Carla Jaramillo

AGRADECIMIENTO

En primer lugar, deseo expresar mi gratitud a mi asesor de tesis, Ingeniera Elizabeth Armas, por su orientación experta y su valioso conocimiento. Su compromiso y dedicación hacia mi trabajo han sido fundamentales para el éxito de esta investigación.

También quiero agradecer a todos los docentes de la Escuela de Formación de Tecnólogos que han aportado con información valiosa a mi formación como profesional.

Finalmente, a mi familia y amigos por su apoyo incondicional, su paciencia y generosidad al compartir su tiempo conmigo para poder crecer como persona.

Carla Jaramillo

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE TABLAS	VII
ÍNDICE DE FIGURAS	VIII
RESUMEN	IX
ABSTRACT	X
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo General	3
1.2 Objetivos Específicos.....	3
1.3 Alcance	3
1.4 Marco teórico	4
2 METODOLOGÍA.....	10
2.1 Dispositivos periféricos (digitales/analógicos).....	10
2.2 Código de Matlab	15
2.3 IDE de Arduino.....	20
2.4 Diseño de los módulos didácticos en Inventor	23
3 RESULTADOS	26
3.1 Interfaz gráfica de Matlab.....	26
3.2 Comand Window	34
3.3 IDE Arduino.....	35
3.4 Diseño y construcción del módulo.....	37
4 CONCLUSIONES	38

5	RECOMENDACIONES.....	40
6	REFERENCIAS BIBLIOGRÁFICAS	41

ÍNDICE DE TABLAS

Tabla 1.1 Características y requerimientos de los módulos.	7
Tabla 2.1 Características técnicas de sensor de temperatura LM35, [1].....	10
Tabla 2.2 Características técnicas de sensor fototransistor, [2].	10
Tabla 2.3 Características técnicas del potenciómetro, [1].	10
Tabla 2.4 Características técnicas de teclado matricial, [3].....	11
Tabla 2.5 Características técnicas de integrado 74C922N, [4].....	11
Tabla 2.6 Características técnicas del LED, [1].....	12
Tabla 2.7 Características técnicas del módulo de dos relés, [5].....	12
Tabla 2.8 Características técnicas del servomotor, [6].....	12
Tabla 2.9 Características técnicas de la bocina, [7].	13
Tabla 2.10 Características técnicas del LCD, [8].....	13
Tabla 2.11 Características técnicas del módulo I2C, [9].	13
Tabla 2.12 Características técnicas de bombilla incandescente, [10].	14
Tabla 2.13 Características técnicas del ventilador, [11].	14
Tabla 2.14 Características técnicas de teclado matricial.....	21
Tabla 3.1 Funcionamiento sensor temperatura.	27
Tabla 3.2 Funcionamiento sensor fototransistor.....	29
Tabla 3.3 Encendido y apagado del LED rojo.	30
Tabla 3.4 Funcionamiento de la bocina.....	31
Tabla 3.5 Variación de intensidad del LED amarillo.	32
Tabla 3.6 Control de la posición del servomotor.	33
Tabla 3.7 Funcionamiento con comandos.....	34
Tabla 3.8 Funcionamiento teclado matricial.	35

ÍNDICE DE FIGURAS

Figura 1.1 Esquema general del proyecto.....	6
Figura 2.1 Apartado de la interfaz para el potenciómetro.....	16
Figura 2.2 Apartado de la interfaz para el sensor de temperatura	16
Figura 2.3 Apartado de la interfaz para el sensor fototransistor	17
Figura 2.4 Apartado de la interfaz para el LED rojo	17
Figura 2.5 Apartado de la interfaz para el LED amarillo.....	18
Figura 2.6 Apartado de la interfaz para el foco incandescente y el ventilador .	18
Figura 2.7 Apartado de la interfaz para la bocina.....	19
Figura 2.8 Apartado de la interfaz para el servomotor	19
Figura 2.9 Diagrama de funcionamiento del Comand Window.....	20
Figura 2.11 Diagrama de funcionamiento del IDE de Arduino.....	21
Figura 2.10 Diagrama de funcionamiento del IDE de Arduino.....	22
Figura 2.12 Diseño del módulo físico en Inventor	23
Figura 2.13 Conexión de teclado matricial y LCD con reducción de pines.....	24
Figura 2.14 Diseño del módulo en Proteus para la PCB	25
Figura 2.15 Diseño del módulo en Proteus para la PCB 3D	25
Figura 3.1 Interfaz en App Designer.....	26
Figura 3.2 Muestra de datos en el LCD.....	37
Figura 3.3 Diseño en dos y tres dimensiones el módulo	37

RESUMEN

El proyecto desarrolló una solución educativa para integrar los entornos de desarrollo Arduino y Matlab. Los módulos permitirán a los estudiantes conectar y controlar los dispositivos de entrada y salida de Arduino utilizando Matlab como interfaz de programación.

En la primera sección, se revisó los fundamentos teóricos relacionados con Arduino, Matlab y la comunicación entre ambos. Se investigaron las características y funcionalidades de ambas plataformas, así como las herramientas y librerías disponibles para su integración.

Posteriormente, se especificaron los dispositivos periféricos digitales y analógicos que se utilizaron en cada módulo. Se identificaron los sensores y actuadores necesarios, como sensores de temperatura y luz, servomotores, bocina y luces.

En la segunda sección se escribió el código en Matlab para controlar las entradas y salidas del Arduino. Se desarrollaron algoritmos y programas que permitieron la lectura de los sensores y la activación de los actuadores en función de los valores leídos. Se establecieron las condiciones de comunicación y conexión (Arduino - periféricos o Matlab – Arduino -periféricos) y los comandos necesarios para lograr una interacción adecuada entre el Arduino y Matlab. Se creó una interfaz intuitiva mediante App Designer, este entorno de desarrollo interactivo genera un código orientado a objetos para una conexión sencilla entre la distribución de los dispositivos y el diseño de la interfaz.

Finalmente, la tercera sección consistió en el diseño de los módulos didácticos. Se tomaron en cuenta consideraciones ergonómicas y de facilidad de uso para los estudiantes. Se realizó el diseño de los módulos didácticos utilizando el software Inventor, donde se crearon representaciones en dos y tres dimensiones de los módulos donde se indicaron las medidas de los mismos, la ubicación de los periféricos, los puertos de conexión y la distancia entre los elementos.

PALABRAS CLAVE: Matlab, Arduino, interfaz, programación, sensores, actuadores, App Designer

ABSTRACT

The project developed an educational solution to integrate the development environments of Arduino and Matlab. The modules will allow students to connect and control Arduino's input and output devices using Matlab as the programming interface.

In the first section, theoretical fundamentals related to Arduino, Matlab, and their communication were reviewed. The characteristics and functionalities of both platforms were investigated, as well as the available tools and libraries for integration.

Subsequently, the digital and analog peripheral devices used in each module were specified. Necessary sensors and actuators such as temperature and light sensors, servomotors, speakers, and lights were identified.

In the second section, Matlab code was written to control Arduino's inputs and outputs. Algorithms and programs were developed to read sensor data and activate actuators based on the readings. Communication and connection conditions (Arduino - peripherals or Matlab - Arduino - peripherals) were established, along with the necessary commands to achieve proper interaction between Arduino and Matlab. An intuitive interface was created using App Designer, an interactive development environment that generates object-oriented code for easy connection between device distribution and interface design.

Finally, the third section consisted of designing didactic modules. Ergonomic considerations and ease of use for students were taken into account. The didactic modules were designed using Inventor software, creating two and three-dimensional representations that included measurements, peripheral locations, connection ports, and distances between elements.

KEYWORDS: Matlab, Arduino, interface, programming, sensors, actual

1 DESCRIPCIÓN DEL COMPONENTE

DESARROLLADO

Este proyecto se plantea con el objetivo de construir 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab, proporcionando una experiencia práctica para que los estudiantes contemplen la utilidad de enlazar el lenguaje de programación, la Electrónica Digital y Microprocesadores (TEMD333). Permite a los usuarios explorar de manera interactiva conceptos clave de la electrónica, como entradas y salidas digitales y analógicas (potenciómetro, servomotor, luces, bocina, entre otros.) en el hardware Arduino. Además, el uso de Matlab como interfaz de programación permite aprovechar las capacidades de análisis y visualización de datos.

El diseño de los módulos didácticos incluyó componentes tanto hardware como software. En el lado del hardware, se utilizaron diferentes dispositivos para establecer la conexión física entre Arduino y Matlab. Esto incluyó cables, conectores y adaptadores para garantizar una conexión adecuada y confiable entre los dos sistemas.

En el lado del software, se utilizaron bibliotecas y drivers específicos para establecer la comunicación entre Arduino y Matlab. Estos softwares establecieron una conexión serial entre ambos sistemas, lo que permitió enviar y recibir datos de forma bidireccional. Además, se utilizaron scripts o programas específicos en Matlab para controlar y monitorear las entradas y salidas de Arduino.

El diseño de los módulos didácticos incluyó una interfaz gráfica de usuario mediante App designer para facilitar la interacción con el usuario. Esta interfaz controla y configura los pines de entrada y salida de Arduino, visualiza los datos en tiempo real y procesa los datos adquiridos desde Matlab.

La elaboración de las hojas guías tienen como objetivo familiarizar a los estudiantes con los módulos y proporcionarles instrucciones paso a paso sobre cómo utilizarlos de manera efectiva. En ellas, se describieron detalladamente los

conceptos clave relacionados con cada módulo y se proporcionaron ejemplos prácticos de cómo utilizarlos en combinación con Arduino y Matlab. También se incluirán diagramas de conexión, explicando cómo conectar correctamente los componentes y cables, y se ofrecerán instrucciones de programación para realizar diferentes tareas y experimentos.

1.1 OBJETIVO GENERAL

Construir 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab

1.2 OBJETIVOS ESPECÍFICOS

1. Especificar los dispositivos periféricos (digitales/analógicos) de los módulos.
2. Escribir el código en Matlab para controlar las entradas y salidas del Arduino.
3. Diseñar los módulos didácticos

1.3 ALCANCE

El proyecto se enfocará en la construcción de 5 módulos didácticos para conectar y controlar las entradas y salidas de Arduino desde Matlab. El objetivo es diseñar y programar un sistema capaz de adquirir señales desde dispositivos de entrada (sensor de temperatura LM35, potenciómetro, sensor fototransistor y teclado matricial) para procesarlas y enviar señales a los dispositivos de salida (pantalla de cristal líquido (LCD), un servomotor, bocina, diodo emisor de luz (LED); y, dos relés que encenderán una bombilla incandescente y un ventilador, respectivamente.)

Para enlazar Arduino con Matlab se utilizarán los paquetes disponibles (arduinoio.mlpkginstall). Para crear la interfaz gráfica de Matlab que permita controlar los dispositivos se usará la opción App Designer. Se utilizarán códigos de programación tanto en el programa Arduino como en Matlab para realizar las siguientes acciones: recibir las señales de un pulsador, potenciómetro y teclado, realizar mediciones de temperatura y cantidad de iluminación, controlar el encendido y apagado de un LED, posicionar un servomotor, variar el nivel de frecuencia de una bocina, accionar los relés; y, enviar datos por el puerto serial I2C para mostrar información en un LCD.

Se diseñará la placa de circuito impreso (PCB) en el programa Proteus que permitirá la conexión de los periféricos con la placa del Arduino Uno.

Los módulos se diseñarán en dos y tres dimensiones en el software Inventor para distribuir el espacio de los periféricos, la placa Arduino Uno, protecciones eléctricas y el cableado.

1.4 MARCO TEÓRICO

En la Figura 1.1 se presenta un esquema general del proyecto en donde los periféricos conectados al Arduino se controlan mediante dos formas: control remoto y control físico.

El control remoto se lleva a cabo desde el ordenador mediante dos enfoques distintos: el Comand Window y App Designer en Matab, así como el entorno de desarrollo (IDE) de Arduino. La comunicación bidireccional entre Matlab y placas Arduino posibilita el envío y recepción de datos en tiempo real. En la Command Window de Matlab, se ejecutan comandos y secuencias de código para interactuar con dispositivos físicos, mientras que App Designer brinda una interfaz gráfica que crea aplicaciones interactivas con componentes como botones, deslizadores y gráficos.

La conexión de dispositivos de entrada y salida en el control físico del proyecto se realiza a través de la placa de Arduino, que actúa como el cerebro central del sistema. Para conectar los dispositivos de entrada, como el sensor de temperatura LM35, pulsadores, potenciómetro, sensor fototransistor y teclado matricial, se utilizan los pines de entrada/salida digitales y analógicos del Arduino. Estos datos son procesados en el IDE de Arduino mediante el código en lenguaje C/C++ y mostrados en el LCD.

Por otro lado, para controlar los dispositivos de salida, como el LCD, el servomotor, la bocina, el LED, y los dos relés para la bombilla incandescente y el ventilador, respectivamente, se utilizan los pines de salida digitales del Arduino.

Los controles remoto y físico permiten una interacción efectiva entre los dispositivos de entrada y salida, permitiendo que el sistema adquiera señales desde los dispositivos de entrada, las procese a través de Matlab o Arduino y envíe señales adecuadas para controlar los dispositivos de salida, logrando así el funcionamiento y el control deseado del sistema en su conjunto.

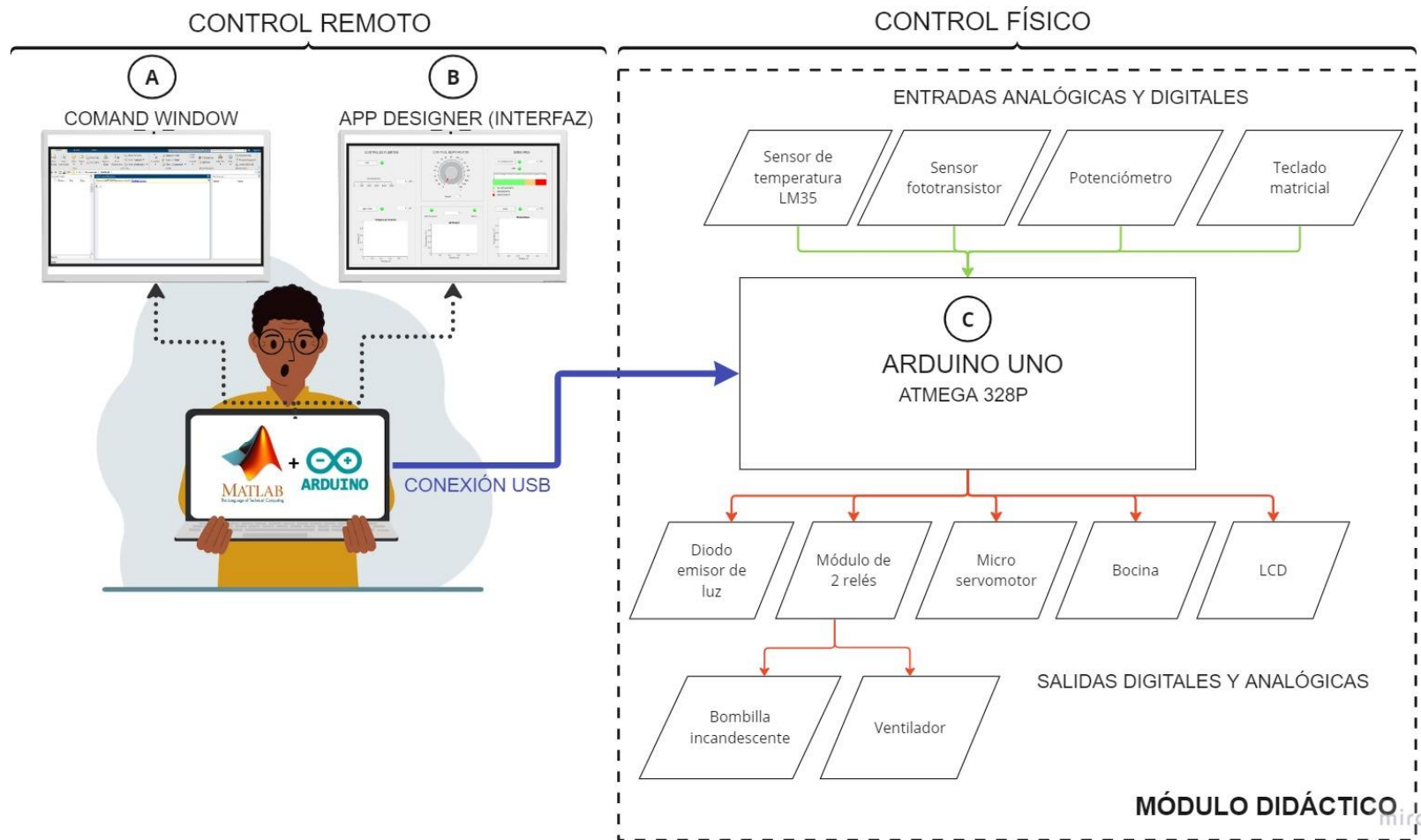


Figura 1.1 Esquema general del proyecto

Tabla 1.1 Características y requerimientos de los módulos.

Características	Descripción
Comunicación Arduino - Matlab	Soporte para la comunicación entre Matlab y Arduino, a través del "Matlab Support Package for Arduino" para establecer y mantener la conexión con el Arduino.
Comunicación bidireccional	Envío y recepción de datos y comandos en tiempo real entre Matlab y Arduino.
Control remoto versátil	Implementación de funciones de Matlab para enviar comandos al Arduino y recibir datos desde periféricos conectados al Arduino.
Flexibilidad en la Interfaz de usuario	Una interfaz gráfica creada con App Designer que permite a los usuarios interactuar con el Arduino y controlar periféricos de manera intuitiva mediante botones, sliders, campos de entrada, gráficos, etc.
Fácil conexión física	Utilizando un cable USB para conectar el Arduino a la computadora. Los controladores USB del Arduino deben estar correctamente instalados en la computadora para que Matlab reconozca el dispositivo.
Análisis en tiempo real	Monitoreo y análisis en tiempo real de datos provenientes de sensores y periféricos, lo que facilita la toma de decisiones y ajustes en función de la información actualizada.
Enfoque en educación y aprendizaje	Ideal para la enseñanza y el aprendizaje de conceptos de programación y control

CONTROL DE PERIFÉRICOS DESDE MATLAB A TRAVÉS DEL ARDUINO

El control de periféricos desde Matlab a través del Arduino es una metodología que permite la interacción bidireccional entre software y hardware, utilizando Matlab como interfaz de control y análisis, y el Arduino como plataforma para gestionar dispositivos periféricos. Esta integración se puede dividir en dos

enfoques principales dentro del entorno de Matlab: el Comand Window y App Designer (interfaz).

- **Comand Window**

En el Comand Window (Figura 1.1, sección A) los comandos y scripts de Matlab se utilizan para establecer comunicación serial con el Arduino, enviar comandos específicos y recibir datos en tiempo real. Esto facilita la interacción directa y rápida con los periféricos conectados al Arduino, permitiendo el control y la monitorización desde una sola ventana de comandos. Esta opción es adecuada para aplicaciones más simples y pruebas rápidas de concepto.

- **Interfaz de usuario**

App Designer en Matlab (Figura 1.1, sección B) es una herramienta de desarrollo gráfico que permite a los usuarios crear interfaces de usuario personalizadas y aplicaciones interactivas sin necesidad de escribir código manualmente. Con esta herramienta, se diseñan interfaces gráficas mediante la función de arrastrar y soltar elementos y componentes gráficos como: botones, deslizadores y visualizaciones en tiempo real, conectados a las operaciones del Arduino.

CONTROL DE PERIFÉRICOS DESDE EL ARDUINO

El control de periféricos desde Arduino (Figura 1.1, sección C) se refiere a la capacidad de utilizar un Arduino para gestionar y controlar dispositivos y componentes electrónicos externos. Arduino es una plataforma de hardware programable (IDE) que ofrece una variedad de pines digitales y analógicos, lo que permite interactuar con periféricos como motores, luces, sensores, pantallas, relés, etc.

- **IDE**

El IDE de Arduino ofrece una plataforma de software que permite a los usuarios crear y cargar programas en sus placas Arduino. Con esta herramienta, se escribe el código en el lenguaje de programación Arduino, se

verifica su sintaxis, se compila para generar el código binario específico de la placa y finalmente, cargar el código en el Arduino para que se ejecute.

2 METODOLOGÍA

El diseño de los módulos didácticos se realizó con un enfoque práctico y experimental. Este enfoque se centra en proporcionar a los estudiantes conceptos teóricos programables a través de la interacción con dispositivos físicos y el análisis de resultados en tiempo real.

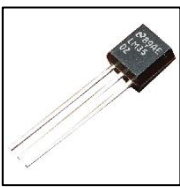
2.1 DISPOSITIVOS PERIFÉRICOS (DIGITALES/ANALÓGICOS)

De acuerdo con las características y especificaciones técnicas y físicas se seleccionaron los siguientes dispositivos:

PERIFÉRICOS DE ENTRADA:


- Sensor de temperatura LM35.

Tabla 2.1 Características técnicas de sensor de temperatura LM35, [1].

Rango de temperatura	-55 (°C) a +150 (°C)	
Precisión típica	±0,5 (°C) a +25 (°C)	
Voltaje de operación	4 (V _{DC}) a 30 (V _{DC})	
Corriente típica	60 (μA _{DC})	
Salida	10 ($\frac{mV_{DC}}{^{\circ}C}$)	


- Sensor fototransistor.

Tabla 2.2 Características técnicas de sensor fototransistor, [2].

Voltaje	3,3 (V _{DC}) o 5 (V _{DC})	
Salida de señal	La resistencia varía y se convierte en un valor de voltaje	


- Potenciómetro.

Tabla 2.3 Características técnicas del potenciómetro, [1].

Valor de resistencia	10 (k Ω)	
Potencia nominal	0,25 (W) o 0,5 (W)	
Tolerancia	10 %	
Voltaje	200 (V _{DC}) o más	
Diámetro del eje	6 (mm) o 1/4 (in)	

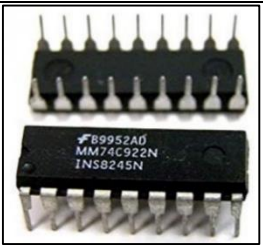
- Teclado matricial.

Tabla 2.4 Características técnicas de teclado matricial, [3].

Tamaño de la matriz	4 filas x 4 columnas (4x4)	
Tipo de teclado	Membrana	
Alimentación	5 (V _{DC})	
Máxima corriente operativa	30 (mA _{DC})	

- Integrado 74C922N.


Tabla 2.5 Características técnicas de integrado 74C922N, [4].

Tipo de integrado	Codificador/decodificador de teclado matricial	
Número de pines	18 pines	
Voltaje de operación	4,5 (V _{DC}) y 15 (V _{DC})	
Corriente de operación	Baja	

PERIFÉRICOS DE SALIDA:

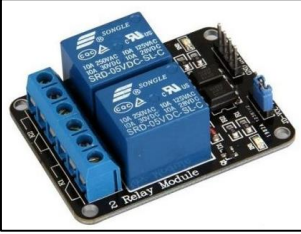
- LED

Tabla 2.6 Características técnicas del LED, [1].

Voltaje	2.2 (V_{DC})	
Corriente	20 (mA_{DC})	
Diámetro	5 (mm)	
Luminosidad	200 (mcd)	


- Módulo de dos relés.

Tabla 2.7 Características técnicas del módulo de dos relés, [5].

Número de Relés	2 relés	
Alimentación	5 (V_{DC}) desde Arduino o fuente de alimentación	
Capacidad de Conmutación	250 (V_{AC}) a 10 (A_{AC}); 30 (V_{DC}) a 10 (A_{DC})	
Tamaño	44,4 x 32,4 (mm)	


- Servomotor.

Tabla 2.8 Características técnicas del servomotor, [6].

Tipo de servomotor	servomotor de 180 °	
Torque máximo	1,5 (kg/cm)	
Velocidad de rotación	0,1 y 0,2 (s) por 60°	
Conexiones	Cable rojo: 5 (V_{DC}) Cable marrón o negro: (GND) Cable naranja: Señal de control (PWM)	

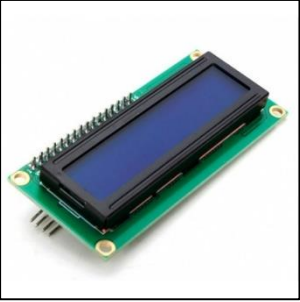
- Bocina.

Tabla 2.9 Características técnicas de la bocina, [7].

Voltaje	5 (V _{DC})	
Corriente	30 (mA _{DC})	
Dimensiones	12 (mm) x 9,5 (mm)	
Frecuencia	23 (KHz)	


- LCD.

Tabla 2.10 Características técnicas del LCD, [8].

Tamaño de pantalla	16 caracteres por 2 líneas (16x2)	
Tecnología de visualización	Pantalla de cristal líquido	
Tamaño físico	-Longitud: Aprox. 80 (mm) -Ancho: Aprox. 36 (mm) -Espesor: Aprox. de 12 (mm)	
Tamaño de los caracteres	5 x 8 (px)	
Voltaje	5 (V _{DC})	

- Módulo I2C.

Tabla 2.11 Características técnicas del módulo I2C, [9].


Protocolo I2C	(SDA y SCL)	
Voltaje	3,3 (V _{DC}) y/o 5 (V _{DC})	
[9]Pines	A0	Bit menos significativo
	A1	Segundo bit menos significativo
	A2	Bit más significativo
	GND	Tierra
	SDA	Transmitir los datos

	SCL	Sincronizar la comunicación
	WP	Protección contra escritura
	VCC	Alimentación

Cargas de los relés:


- Bombilla incandescente.

Tabla 2.12 Características técnicas de bombilla incandescente, [10].

Color	Ámbar	
Material	Metal, Vidrio	
Voltaje	12 (V _{DC})	
Consumo de energía	21 (W)	

- Ventilador

Tabla 2.13 Características técnicas del ventilador, [11].

Voltaje	12 (V _{DC})	
Dimensiones	40 (mm) x 40 (mm) x 10 (mm)	
Corriente	100 (mA _{DC})	
Consumo de energía	1.25 (W)	

2.2 CÓDIGO DE MATLAB

Se usaron librerías para establecer la conexión entre Arduino y Matlab. Se crearon los códigos correspondientes que permitieron las siguientes acciones:

- Usar los pines de entrada y salida del Arduino Uno.
- Ingresar valores numéricos desde el teclado matricial.
- Accionar los relés para el uso de la bombilla y el ventilador.
- Girar el servomotor en un determinado ángulo.
- Medir la temperatura con el LM35.
- Ingresar datos binarios desde los pulsadores.
- Ingresar valores analógicos desde el potenciómetro.
- Obtener datos de temperatura.
- Establecer la conexión I2C para el uso del LCD
- Encender y apagar el LED
- Generar sonido en la bocina.

Además, utilizando la App Designer se diseñó la interfaz gráfica para añadir componentes y elementos interactivos (botones de acción, botones de conmutación, cuadros de texto, gráficos de datos e imágenes ilustrativas); cada elemento cumplió las acciones específicas de acuerdo con los códigos correspondientes.

APP DESIGNER

Utilizando la App Designer se diseñó la interfaz gráfica para añadir componentes y elementos interactivos (botones de acción, botones de conmutación, cuadros de texto, gráficos de datos e imágenes ilustrativas); cada elemento cumplirá las acciones específicas de acuerdo con los códigos correspondientes.

El código establecido se muestra en el Anexo V.

Periféricos de entrada:

- A0: Potenciómetro

Se mide la tensión mediante la función "ledamarillovaluechanged" en la interfaz, como se muestra en la Figura 2.1.

El potenciómetro está conectada al pin analógico 0 del Arduino, y se muestra el valor medido en la interfaz.

La gráfica con ejes muestra la variación del voltaje al transcurrir el tiempo en segundos, este cambia según la manipulación del potenciómetro físico.

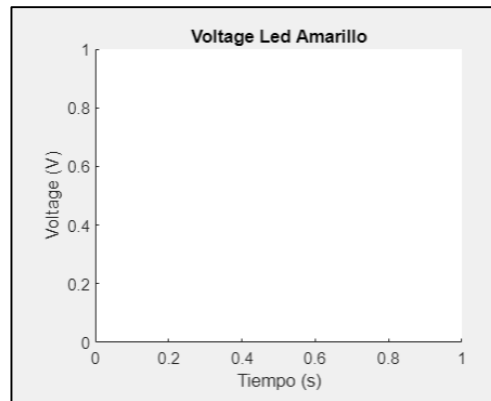


Figura 2.1 Apartado de la interfaz para el potenciómetro

- A1: Sensor de temperatura LM35

Se mide la tensión mediante la función "lm35valuechanged" en la interfaz, como se muestra en la Figura 2.2.

El sensor de temperatura LM35 está conectado al pin analógico 1 del Arduino, y se muestra el valor de temperatura en la interfaz. La gráfica con ejes muestra la temperatura en tiempo real que mide el sensor.

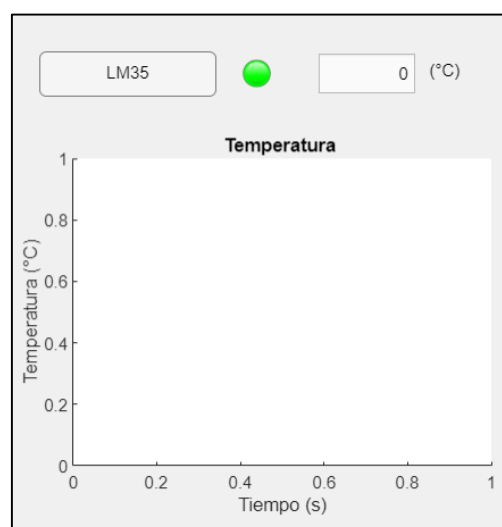


Figura 2.2 Apartado de la interfaz para el sensor de temperatura

- A5: Sensor de Fototransistor

Se mide la tensión mediante la función "fotoresisvaluechanged" en la interfaz, como se muestra en la Figura 2.3.

El fototransistor está conectado al pin analógico 5 del Arduino, y se muestra el valor medido de voltaje. Mientras menos iluminación detecte el sensor la perrilla se moverá con dirección a la derecha, por otro lado, si la iluminación detectada es mayor la perilla se dirige a la izquierda.

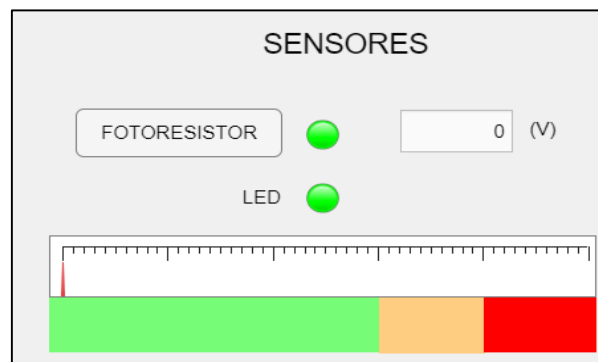


Figura 2.3 Apartado de la interfaz para el sensor fototransistor

Periféricos de salida:

- D13: LED rojo

Se controla mediante el botón "ledrojo" (uibutton) en la interfaz, como se muestra en la Figura 2.4.

Cuando se presione el botón está activado (valor 1), se enciende el LED rojo conectado al pin digital 13 del Arduino. Si está desactivado (valor 0), el LED se apaga.



Figura 2.4 Apartado de la interfaz para el LED rojo

- D3: LED amarillo

Se controla mediante el botón "ledamarillo" (uibutton) en la interfaz, como se muestra en la Figura 2.5.

Cuando el botón está activado (valor 1), el LED amarillo conectado al pin digital 3 del Arduino se enciende y se controla su intensidad a través del potenciómetro "volt_2".

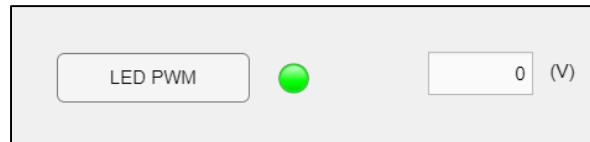


Figura 2.5 Apartado de la interfaz para el LED amarillo

- D4: Foco – D7: Ventilador

Si la temperatura que se desea obtener en el espacio en el que se encuentra el sensor debe ser alta el relé permitirá el encendido del foco caso contrario si se desea una temperatura menor el relé correspondiente al ventilador se accionará, como se muestra en la Figura 2.6.

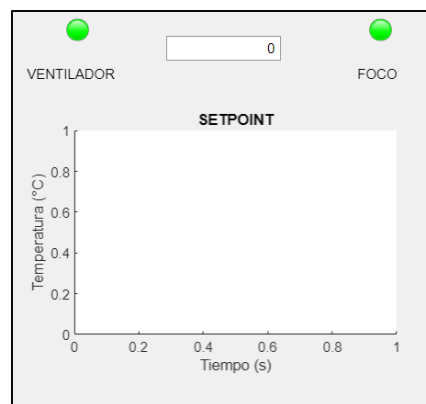


Figura 2.6 Apartado de la interfaz para el foco incandescente y el ventilador

- D12: Bocina

Se controla mediante el control deslizante "frecuenciabocina" (uislider) en la interfaz.

Cuando se mueve el control deslizante, se genera un tono de la frecuencia seleccionada a través del altavoz conectado al pin digital 12 del Arduino, como se muestra en la Figura 2.7.

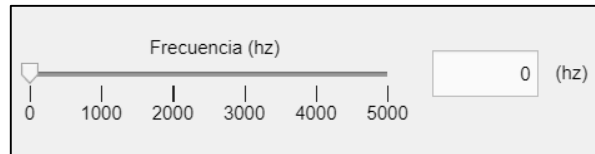


Figura 2.7 Apartado de la interfaz para la bocina

- D5: Servomotor

Se controla mediante una perilla "servo" (uiknob) en la interfaz.

El servomotor está conectado al pin digital 5 del Arduino, y la perilla permite ajustar el ángulo al cual se posiciona el servomotor, como se muestra en la Figura 2.8.

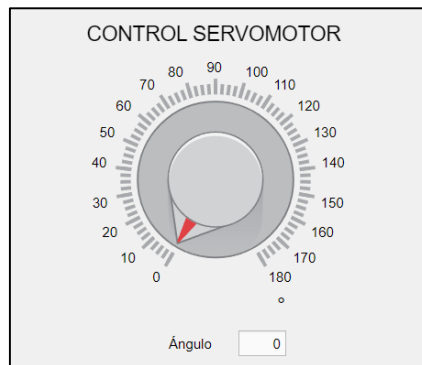


Figura 2.8 Apartado de la interfaz para el servomotor

COMAND WINDOW

En la Figura 2.9 se muestra el funcionamiento del entorno de Matlab que permite ingresar y ejecutar comandos directamente.

La lista de comandos establecidos se muestra en el Anexo IV.

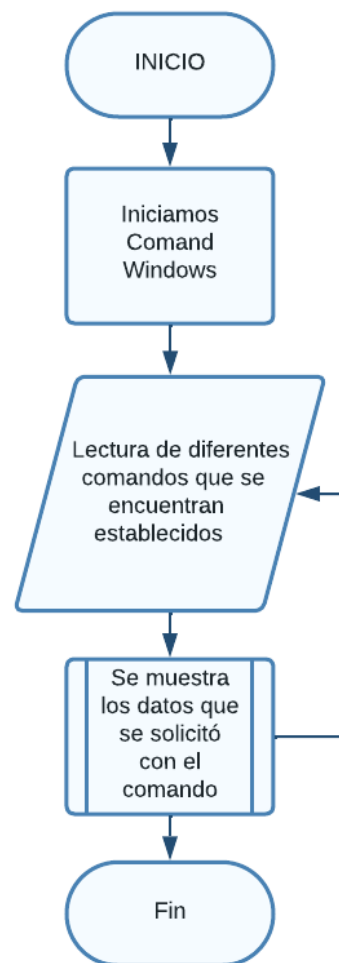


Figura 2.9 Diagrama de funcionamiento del Comand Window

Los dispositivos periféricos de entrada y salida que se controlan en el Comand Window son los mismos que se controlan mediante la interfaz.

2.3 IDE DE ARDUINO

En la Figura 2.10 se muestra el control físico donde la programación del IDE de Arduino se encarga de mostrar datos de temperatura del LM35 y grados de posición del servomotor, digitalizando previamente la contraseña establecida.

En la Tabla 2.14 se muestra el código binario para el uso del teclado en Proteus Figura 2.11.

Los códigos establecidos se muestran en el Anexo VI.

Tabla 2.14 Características técnicas de teclado matricial

Teclado	Binario			
7	0	0	0	0
9	0	0	0	1
8	0	0	1	0
A	0	0	1	1
1	0	1	0	0
3	0	1	0	1
2	0	1	1	0
B	0	1	1	1
4	1	0	0	0
6	1	0	0	1
5	1	0	1	0
C	1	0	1	1
*	1	1	0	0
#	1	1	0	1
"="	1	1	1	0
D	1	1	1	1

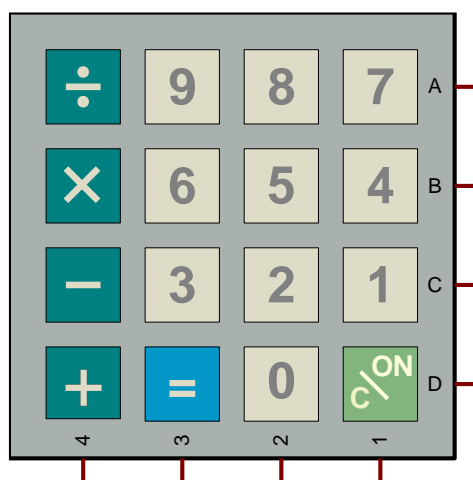


Figura 2.11 Diagrama de funcionamiento del IDE de Arduino

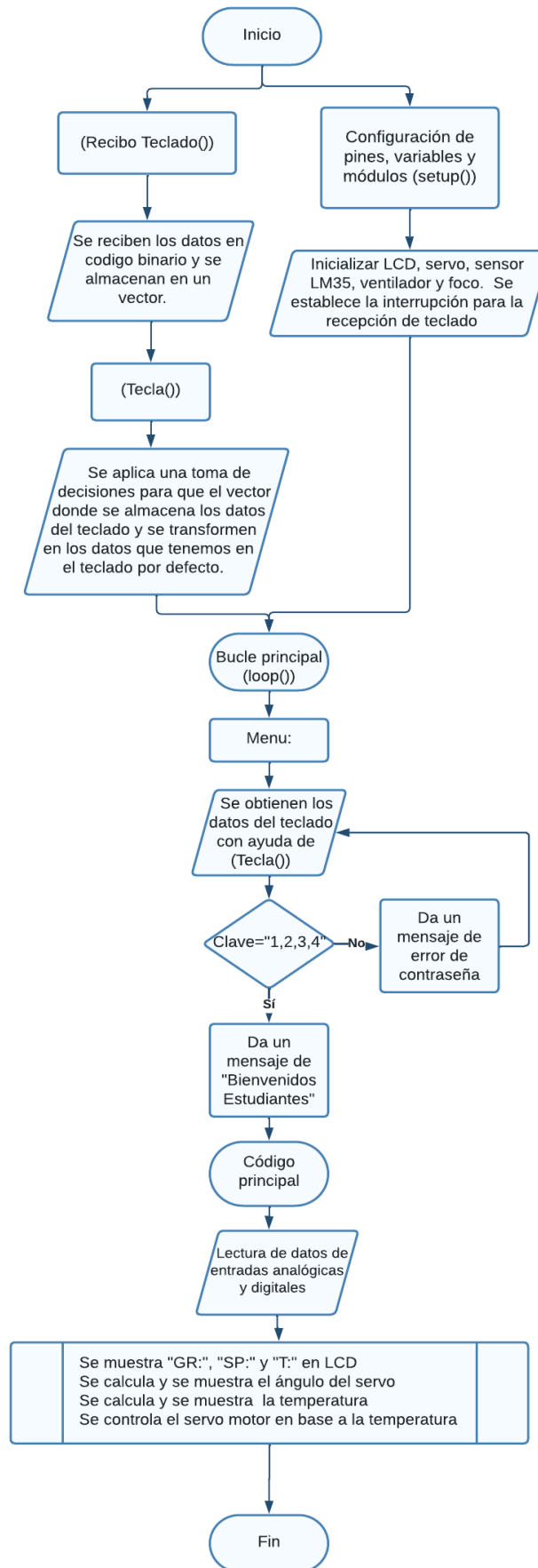


Figura 2.10 Diagrama de funcionamiento del IDE de Arduino

2.4 DISEÑO DE LOS MÓDULOS DIDÁCTICOS EN INVENTOR

En la Figura 2.12 se muestra el diseño de los módulos realizados en Inventor donde se proporciona información de: medidas de los módulos, ubicación de los periféricos y PCB, puertos de conexión y distancia entre elementos; el objetivo es garantizar que los módulos sean didácticos, modulares y robustos; para que los estudiantes puedan realizar las conexiones de forma sencilla.

En el lado izquierdo del módulo se ubicaron los dispositivos de entrada (sensor de temperatura LM35, pulsadores, potenciómetro, sensor fototransistor y teclado matricial), mientras que en el lado derecho se ubicaron los dispositivos de salida (LCD, un servomotor, bocina, LED; y, dos relés que encenderán una bombilla incandescente y un ventilador, respectivamente.). Esta disposición se determinó con el fin de que los estudiantes puedan seguir el flujo de información desde la entrada hasta la salida de una manera más intuitiva y también porque su fácil gestión del cableado y las conexiones, lo que contribuye a un diseño más limpio y ordenado.

El plano diseñado en Inventor se muestra en el Anexo III.

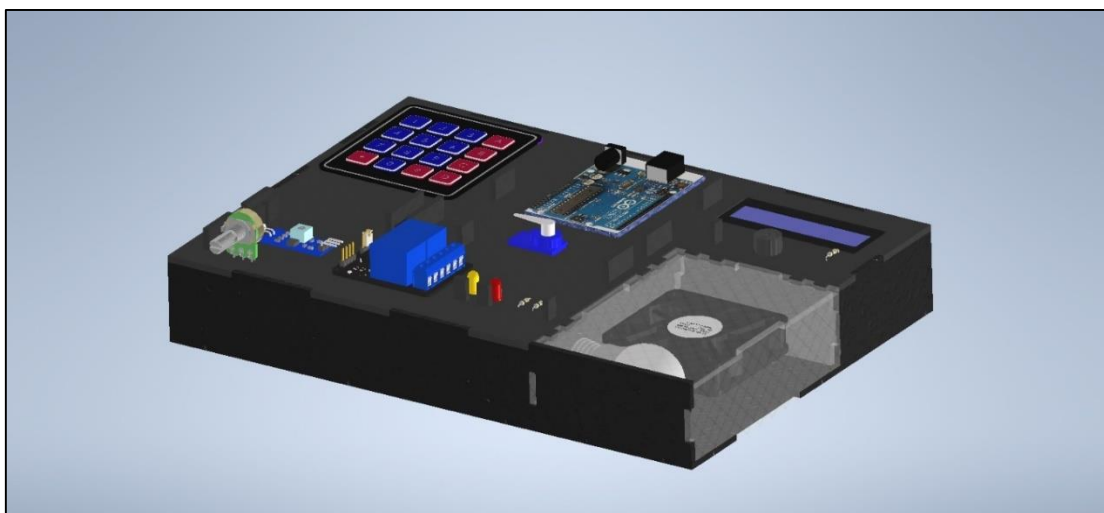


Figura 2.12 Diseño del módulo físico en Inventor

Funcionalidades implementadas:

En la Figura 2.13 se muestra la combinación del módulo I2C y LCD que ayuda a simplificar la conexión y el control de la pantalla, ya que requiere menos pines en comparación con una interfaz paralela tradicional. Por otro lado, el uso del teclado y el integrado MM74C922 ayuda a disminuir la cantidad de pines necesarios para su conexión.

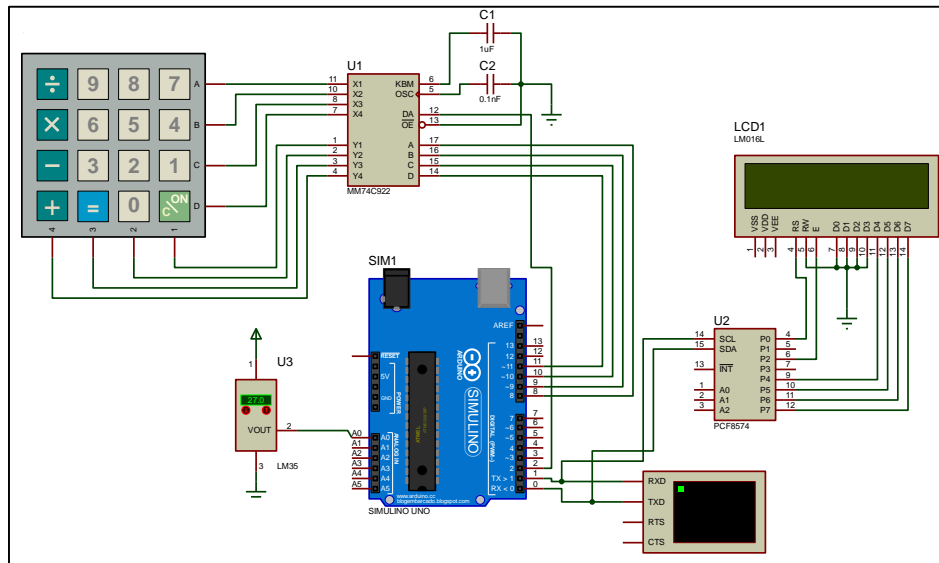


Figura 2.13 Conexión de teclado matricial y LCD con reducción de pines

En las Figuras 2.14 y Figura 2.15 se muestra el diseño de la PCB que se usó para la tapa superior del módulo, dando como resultado una integración más limpia y compacta de los componentes electrónicos, reduciendo el espacio y mejorando la organización. La disposición de los pines esta realizada para que se encuentre junto a cada dispositivo de tal forma que los estudiantes realicen la conexión por su cuenta.

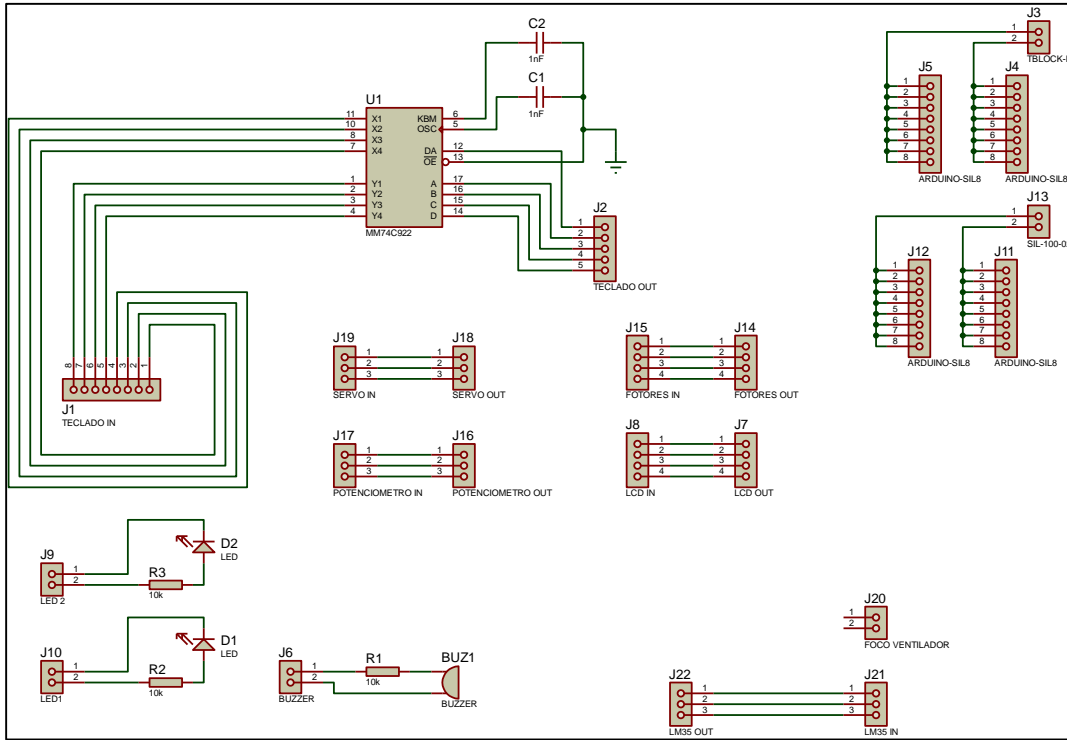


Figura 2.14 Diseño del módulo en Proteus para la PCB

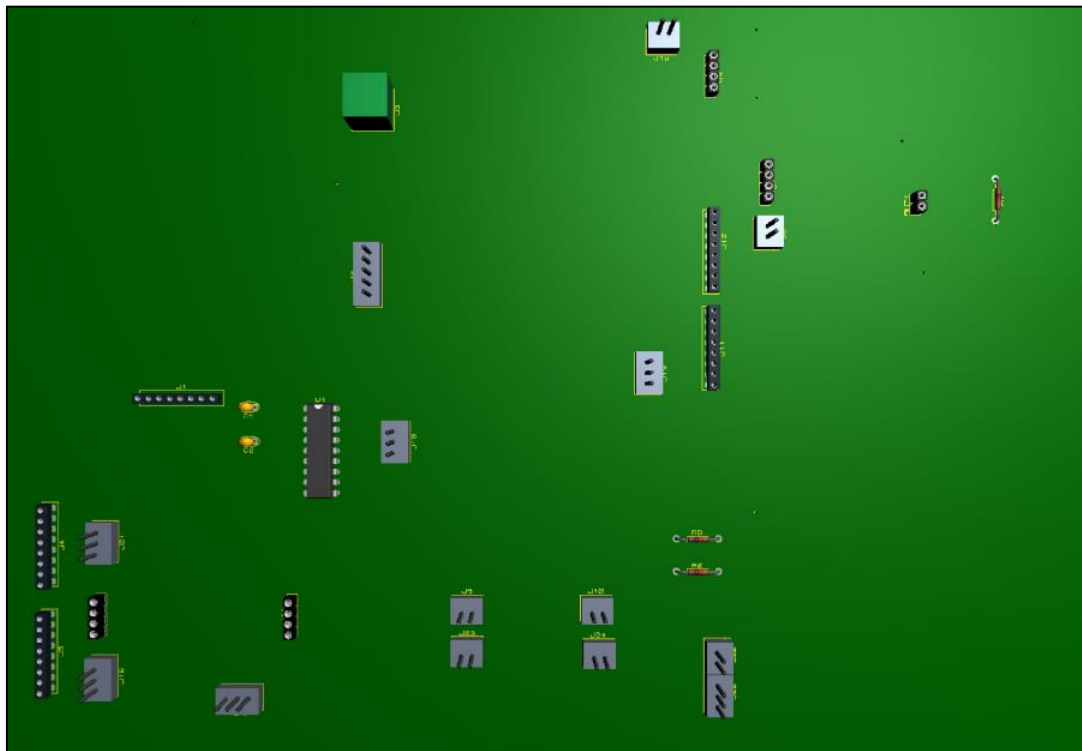


Figura 2.15 Diseño del módulo en Proteus para la PCB 3D

3 RESULTADOS

Se llevaron a cabo pruebas exhaustivas a través de las interfaces para verificar el correcto funcionamiento del módulo desarrollado en el componente de construcción. Estas pruebas permitieron evaluar la eficacia y precisión de las interfaces en la comunicación y control del sistema, asegurando que el módulo de construcción respondiera de manera adecuada a las señales y comandos enviados a través de dichas interfaces.

3.1 INTERFAZ GRÁFICA DE MATLAB

En la Figura 3.1 se muestra una integración de todos los elementos en una sola ventana de interfaz para mejor visualización y que sea amigable con el usuario. Los datos de los sensores de entrada, como el sensor de temperatura LM35 y el sensor fototransistor, se pueden visualizar en tiempo real en la interfaz gráfica de usuario, y controlar los periféricos de salida como el LED, bocina y servomotor

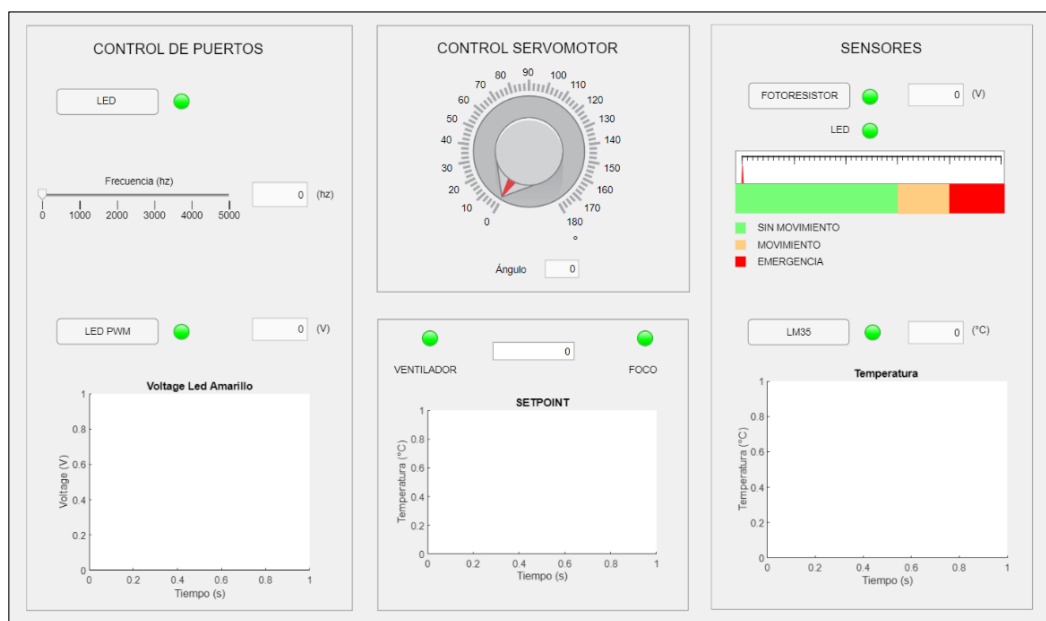


Figura 3.1 Interfaz en App Designer

OBTENCIÓN DE DATOS DEL SENSOR DE TEMPERATURA

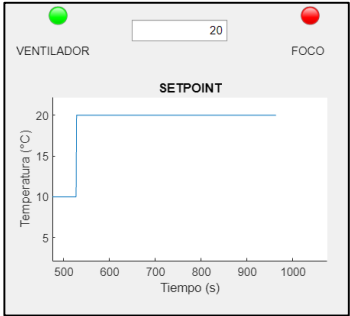
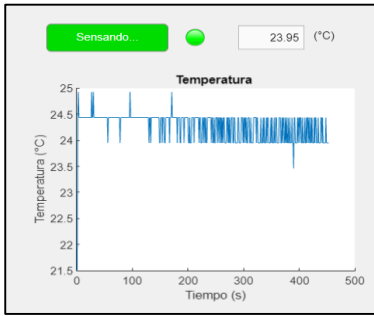
El objetivo central del funcionamiento del sensor de temperatura fue demostrar la capacidad de la interfaz para controlar y mantener la temperatura deseada a través de la activación de un ventilador y un foco, según corresponda. Los resultados obtenidos revelaron una respuesta eficaz y controlada del sistema, con un error promedio de 0.069 en las pruebas, como se muestra en la Tabla 3.1.

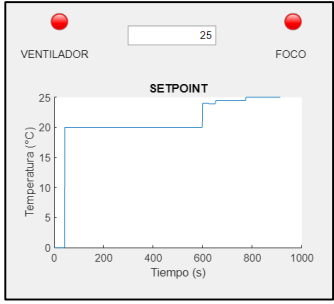
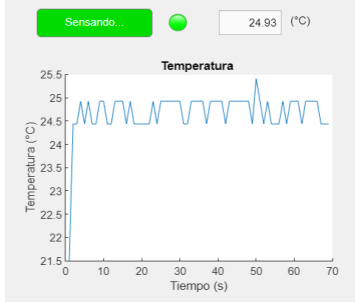
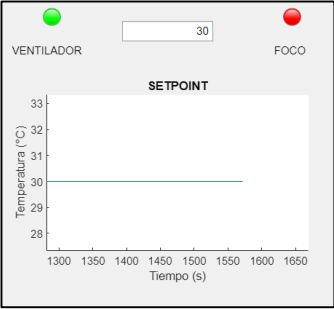
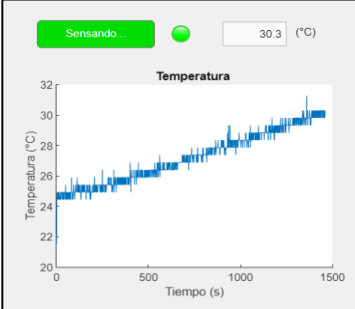
La interfaz de usuario se diseñó con el propósito de permitir a los usuarios establecer la temperatura deseada a través de la digitalización. Los valores de temperatura seleccionados se representaron en un eje de coordenadas, lo que facilitó la visualización de cómo la temperatura variaba en función del tiempo.

Para mantener la temperatura en los valores deseados, se implementaron dos estrategias de control. Cuando la temperatura ingresada era inferior a la deseada, se activaba un ventilador para enfriar el sistema y reducir la temperatura.

Por otro lado, si la temperatura ingresada superaba el valor objetivo, se encendía un foco para aumentar la temperatura ambiente. Estas acciones se llevaron a cabo para acercar la respuesta del sensor LM35 a la temperatura deseada.

Tabla 3.1 Funcionamiento sensor temperatura.

No. Prueba	Temperatura deseada (C°)	Temperatura medida (C°)	Error %	Resultado
1			19,70	✓

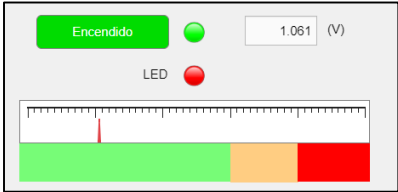
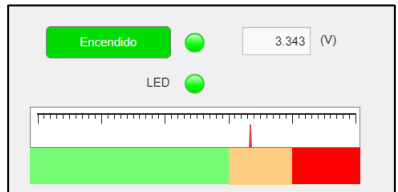
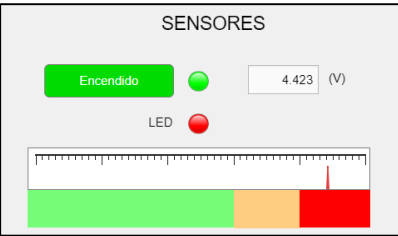
2			0,20	✓
3			1,00	✓

OBTENCIÓN DE DATOS DEL SENSOR FOTOTRANSISTOR

El objetivo principal consistió en demostrar cómo el sensor respondía a diferentes niveles de luz, y cómo estos cambios de intensidad luminosa se reflejaban en la interfaz, donde se representaba el voltaje medido. Además, se compararon los valores de voltaje obtenidos en la interfaz con mediciones reales realizadas utilizando un multímetro, como se muestra en la Tabla 3.2.

A través de la interfaz, se observó cómo la perilla, representada gráficamente en la pantalla, se inclinó hacia la izquierda cuando el voltaje aumentaba debido a una mayor intensidad luminosa. De manera similar, cuando se tapaba el sensor y el voltaje disminuye, la perilla se inclinó hacia la derecha.

Tabla 3.2 Funcionamiento sensor fototransistor.

No. Prueba	Voltaje Ideal (V _{DC})	Voltaje Medido (V _{DC})	Error %	Interfaz	Resultado
1	1,061	1,207	13,76		✓
2	3,343	2,740	18,04		✓
3	4,423	4,396	0,61		✓

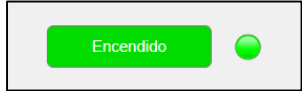
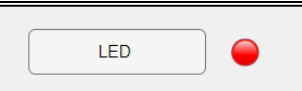
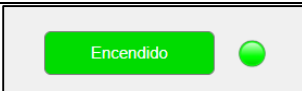
Es relevante destacar que, en todas las pruebas realizadas, tanto en condiciones de alta intensidad luminosa como en condiciones de baja luz, se observó una respuesta coherente y predecible del sensor de fototransistor en la interfaz. La inclinación de la perilla correspondía consistentemente a los cambios en el voltaje, lo que confirmaba la confiabilidad del sistema

CONTROL DEL ENCENDIDO Y APAGADO DEL LED ROJO

Durante el proceso de pruebas, se llevaron a cabo múltiples ciclos de encendido y apagado del LED rojo utilizando la interfaz de usuario. Se registraron los valores ideales de encendido y apagado, y se compararon con los valores medidos obtenidos de la interfaz. En todos los casos, se observó una coincidencia exacta entre los valores ideales y medidos, lo que indica una

sincronización perfecta entre la interfaz de usuario y el control del LED, como se muestra en la Tabla 3.3.

Tabla 3.3 Encendido y apagado del LED rojo.

No. Prueba	Voltaje Ideal (V_{DC})	Voltaje Medido (V_{DC})	Error %	Interfaz	Resultado
1	5	5,063	1,26		✓
2	0	0,0	0,00		✓
3	5	5,072	1,44		✓

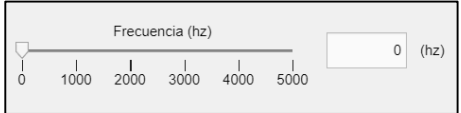
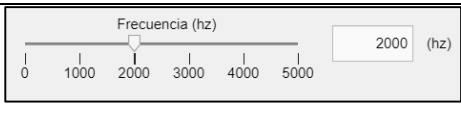

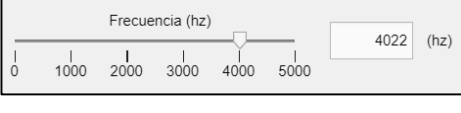
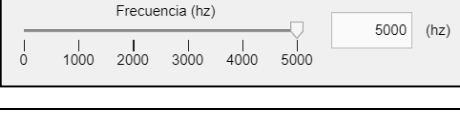
CONTROL DE FRECUENCIA DE LA BOCINA

Durante las pruebas de frecuencia, se evaluó la respuesta de la bocina ante una serie de frecuencias específicas. La interfaz permitió seleccionar diferentes frecuencias y reproducirlas a través de la bocina. Se registraron las respuestas de la bocina en términos de la claridad y la consistencia de la reproducción para cada frecuencia probada. Los resultados demostraron que la bocina fue capaz de reproducir una amplia gama de frecuencias con un nivel uniforme de calidad sonora.

Para evaluar la potencia acústica de la bocina, se realizaron mediciones de nivel de sonido utilizando la aplicación de celular "dB Meter" la cual lanzó resultados en unidades de decibelios (dB).

La interfaz se ajustó para variar el nivel de sonido emitido por la bocina y se midieron los valores correspondientes en (dB). Los resultados indicaron que la bocina mantuvo una respuesta lineal en términos de la relación entre la señal de entrada y el nivel de sonido en [dB], lo que sugiere una consistencia en la reproducción de sonido a diferentes niveles de intensidad, como se muestra en la Tabla 3.4.

Tabla 3.4 Funcionamiento de la bocina.

No. Prueba	Frecuencia (Hz)	Nivel de Sonido (dB)	Interfaz	Resultado
1	0	30		✓
2	2000	60		✓
3	3000	96		✓
4	4000	82		✓
5	5000	91		✓

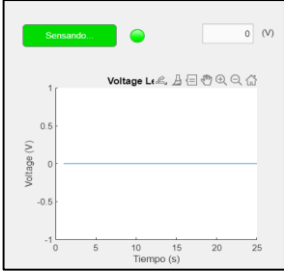
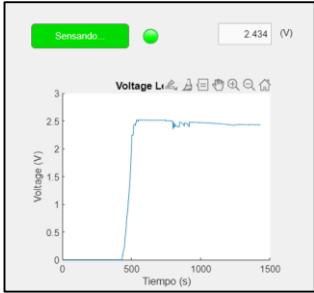
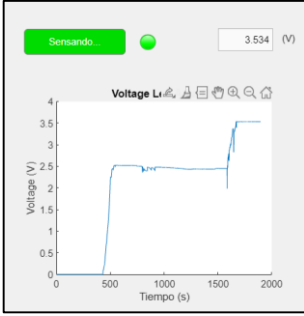
CONTROL INTENSIDAD DE LUZ DEL LED AMARRILLO

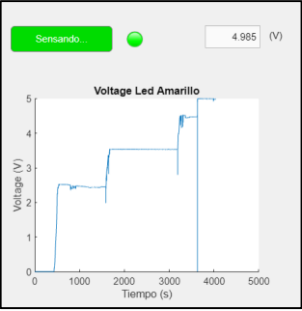
Se llevaron a cabo diversas iteraciones de variación de la intensidad del LED amarillo a través del potenciómetro físico, evidenciando los valores de voltaje en la interfaz. Se registraron los niveles de voltaje ideales y se compararon con las mediciones obtenidas de la configuración física, obtenidas mediante un multímetro. Las oscilaciones de voltaje evidencian la variación de intensidad del LED. Mientras más voltaje deje pasar nuestro potenciómetro, el LED aumentará su iluminación. Por otro lado, si el voltaje es menor, el LED disminuirá su intensidad.

En todas las instancias, se observó una coincidencia perfecta entre los niveles de voltaje deseados y los niveles medidos, evidenciando una correspondencia

precisa entre la interfaz de usuario y la modulación del LED, como se muestra en la Tabla 3.5.

Tabla 3.5 Variación de intensidad del LED amarillo.

No. Prueba	Voltaje Ideal (V_{DC})	Voltaje Medido (V_{DC})	Error %	Interfaz	Resultado
1	0	0,0	0,0		✓
2	2,5	2,426	2,96		✓
3	3,5	3,462	1,08		✓
4	4,5	4,385	2,50		✓

5	5	4,904	0,019		✓
---	---	-------	-------	--	---

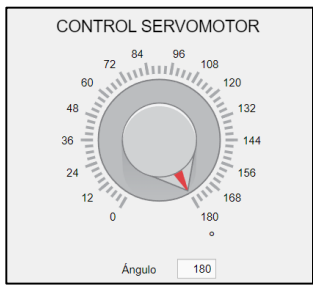
CONTROL DE LA POSICIÓN DEL SERVOMOTOR

Se llevaron a cabo diversos escenarios de control del servomotor mediante la interfaz.

Se evaluó la capacidad de la interfaz para posicionar el servomotor en ángulos específicos y controlar la velocidad de rotación de manera consistente. Los resultados fueron altamente satisfactorios, ya que el servomotor respondió con precisión a los comandos proporcionados a través de la interfaz, como se muestra en la Tabla 3.6.

Tabla 3.6 Control de la posición del servomotor.

No. Prueba	Ángulo Medido (°)	Interfaz	Resultado
1	0°		✓
2	90°		✓

3	180°		✓
---	------	---	---

3.2 COMAND WINDOW

Durante este estudio, se logró una integración efectiva del "Command Window" como una herramienta versátil para interactuar con diferentes componentes del sistema. Los resultados obtenidos demuestran la viabilidad de este enfoque para controlar y adquirir datos de manera interactiva. A continuación, se presentan en la Tabla 3.7.

Tabla 3.7 Funcionamiento con comandos

Comandos	Resultado	Característica
<code>>> a=arduino</code>	<pre>a = arduino with properties: Port: 'COM6' Board: 'Uno' AvailablePins: {'D2-D13', 'A0-A5'} AvailableDigitalPins: {'D2-D13', 'A0-A5'} AvailablePWM Pins: {'D3', 'D5-D6', 'D9-D11'} AvailableAnalogPins: {'A0-A5'} AvailableI2CBusIDs: [0] Libraries: {'I2C', 'SPI', 'Servo'} Show all properties</pre>	Determinar la variable "a" como Arduino
<code>>> writeDigitalPin(a, 'D13', 1)</code>	<pre>ans = 1</pre>	Encendido del LED rojo
<code>>> writeDigitalPin(a, 'D13', 0)</code>	<pre>ans = 0</pre>	Apagado del LED rojo
<code>>> readVoltage(a, 'A0')</code>	<pre>ans = 4.9951</pre>	Lectura del voltaje del potenciómetro
<pre>>> servoPin = 'D5'; >> writePosition(a, servoPin, 0.9)</pre>	<pre>ans = 0.9000</pre>	Se determina el ángulo del servomotor

<code>>> playTone(a, 'd12', 200, 5)</code>	<pre>ans = ans = 200 5</pre>	Se determina la frecuencia de la bocina
<code>>> readVoltage(a, 'a5')</code>	<pre>ans = 0.1222</pre>	Se muestra el valor de voltaje del sensor de luz
<code>>> writeDigitalPin(a, 'd4', 1)</code>	<pre>ans = 1</pre>	Encendido del foco
<code>>> readVoltage(a, 'a1')</code>	<pre>ans = 0.1222</pre>	Se muestra el valor de voltaje del sensor de temperatura

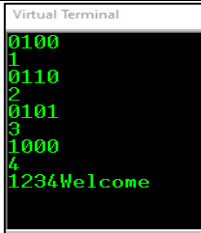
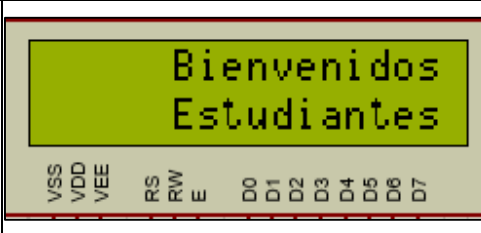
3.3 IDE ARDUINO

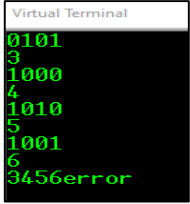
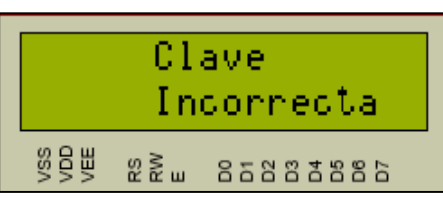
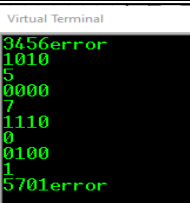
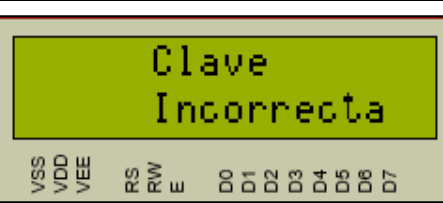
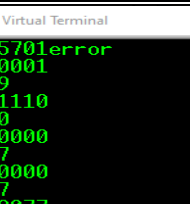
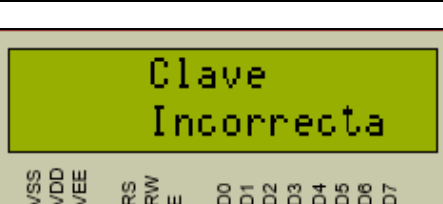
TECLADO MATRICIAL Y LCD

El objetivo principal fue establecer un sistema que permitiera a los usuarios introducir una contraseña a través del teclado matricial y, a continuación, mostrar en un LCD si la contraseña ingresada era correcta o incorrecta.

Se presentan los resultados obtenidos de esta implementación en la Tabla 3.8.

Tabla 3.8 Funcionamiento teclado matricial.

Contraseñas	Virtual terminal	LCD
1-2-3-4	 <pre>Virtual Terminal 0100 1 0110 2 0101 3 1000 4 1234Welcome</pre>	 <p>Bienvenidos Estudiantes</p> <p>VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7</p>

3-4-5-6		
5-7-0-1		
9-0-7-7		

Una vez que se ingresa una secuencia de teclas que representa una posible contraseña, el programa Arduino compara esta secuencia con la contraseña predefinida. Si la coincidencia es exitosa, se generaba un mensaje indicando que la contraseña era correcta y pasaba al menú principal. En caso contrario, se mostraba un mensaje que informaba que la contraseña era incorrecta.

Durante las pruebas realizadas, se introdujeron diferentes contraseñas a través del teclado matricial, y se observó que el sistema respondió correctamente en todos los casos. Los mensajes en el LCD fueron coherentes con la validez de las contraseñas ingresadas.

Una vez que se ha ingresado la contraseña exitosamente, el LCD despliega el mensaje: "Bienvenidos Estudiantes". Este saludo da la bienvenida y reconoce su acceso autorizado. Además, presenta datos en tiempo real de los grados del servomotor, los cuales varían en respuesta a las fluctuaciones de temperatura detectadas por el sensor LM35. La relación entre la temperatura y la posición del servomotor se refleja en los valores mostrados en el LCD, proporcionando una presentación visual inmediata y significativa del entorno térmico actual, como se muestra en la Figura 3.2.

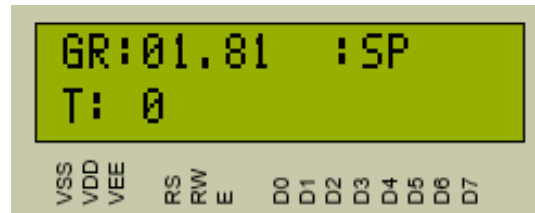


Figura 3.2 Muestra de datos en el LCD

3.4 DISEÑO Y CONSTRUCCIÓN DEL MÓDULO

Se logró diseñar los módulos en dos y tres dimensiones utilizando el software Inventor como se muestra en el Anexo III. Estos diseños muestran la distribución del espacio de los periféricos, la placa Arduino Uno, lo que facilita la comprensión y el manejo de los módulos.

Finalmente, en la Figura 3.3 se muestra un esquema en conjunto de todo el módulo con los dispositivos periféricos determinados.

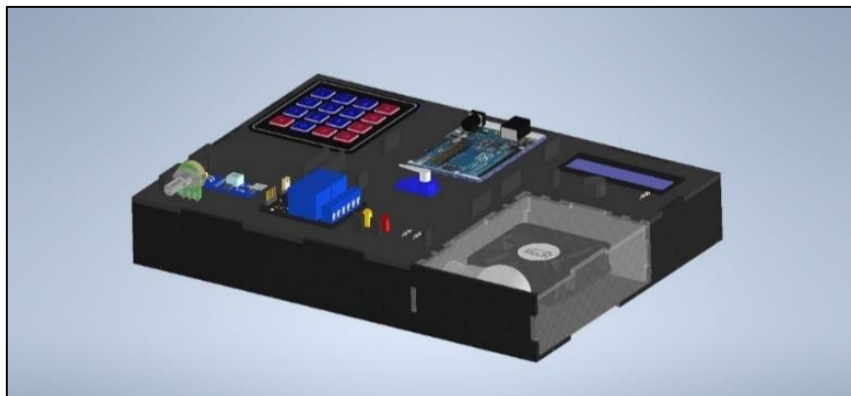


Figura 3.3 Diseño en dos y tres dimensiones el módulo

4 CONCLUSIONES

- El sistema en conjunto está diseñado para trabajar con dos tipos de controles:
 - El control remoto se basa en dos entornos de Matlab. El primero mediante la interfaz creada en App Designer, mediante gráficas intuitivas que permite visualizar y controlar componentes a través de elementos visuales. El segundo mediante el Comand Window utilizando comandos establecidos.
 - El control físico se basa en el Arduino IDE, permitiendo la programación y control de dispositivos físicos mediante la placa Arduino.
- Para el control remoto a través de la interfaz de App Designer se muestra la integración de dispositivos periféricos de entrada y salida.
 - Para los periféricos de entrada como: sensor de temperatura LM35, potenciómetro y sensor fototransistor, se muestran una apreciación clara de cómo estos valores fluctúan en función del tiempo mediante un eje de coordenadas con una fiabilidad del 100% que se verificó en las pruebas realizadas.
 - Para los periféricos de salida como: bocina, servomotor, ventilador y foco incandescente, se controla la orientación de grados del servomotor, la frecuencia de sonido emitido por la bocina y la determinación de temperatura para que el foco o el ventilador respondan de manera automática para cumplir con las condiciones de temperatura establecidas en la interfaz, con una fiabilidad del 100% que se verificó en las pruebas realizadas.
- La utilización del Comand Window como control remoto se muestra la integración de dispositivos periféricos de entrada y salida.
 - Para los periféricos de entradas, como: sensor de temperatura LM35, potenciómetro y el sensor fototransistor, se obtienen valores de voltaje.

- Para los periféricos de salida como: LED, bocina, servomotor, ventilador y foco incandescente proporcionaron una respuesta mecánica.

Para los dos casos los resultados fueron altamente satisfactorios, ya que los dispositivos respondieron con precisión a los comandos establecidos.

- Para el control físico a través del IDE de Arduino permite el uso del teclado matricial para activación de una contraseña la cual se visualiza en el LCD y al ser ingresada correctamente se visualizan los grados del servomotor y las lecturas de temperatura del sensor LM35.
- La utilización de Autodesk Inventor para el diseño del módulo didáctico permitió la mejor visualización y distribución, donde los dispositivos de entrada se colocaron a la izquierda y los dispositivos de salida a la derecha, esta disposición se determinó con el fin de que los estudiantes puedan seguir el flujo de información desde la entrada hasta la salida de una manera más intuitiva y también por su fácil gestión del cableado y las conexiones.

5 RECOMENDACIONES

- Utilizar una versión actualizada tanto para Matlab como para Arduino, esto puede solucionar errores y problemas de compatibilidad. Instalar los paquetes y soportes necesarios y para la comunicación utilizar controladores USB para reconocimiento del dispositivo
- Para que la comunicación sea eficiente primero conectar el Arduino con el Matlab y después ejecutar el IDE de Arduino en caso de que se haya realizado primero el IDE de Arduino cambiar de puerto para que Matlab pueda reconocer el dispositivo adecuadamente
- En el apartado del sensor de temperatura se debe esperar un tiempo suficiente para que el sistema alcanzara tanto temperaturas altas como bajas, lo que garantiza la estabilidad y la precisión de las mediciones en ambos extremos del espectro térmico.
- En la ejecución de la interfaz se recomienda desactivar después de haber utilizado cada apartado y no activarlos todos al mismo tiempo para evitar posibles daños en los dispositivos periféricos

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] T. instrument, «Alldatasheet.com,» Noviembre 2000. [En línea]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/517588/T11/LM35.html>.
- [2] T. i. Incorporated, «Texas instrument Incorporated,» 23 Octubre 1999–2018. [En línea]. Available: <https://www.ti.com/lit/ds/symlink/lm2903-n.pdf>. [Último acceso: 24 Julio].
- [3] «Electrónicos Caldas,» [En línea]. Available: <http://www.electronicoscaldas.com/datasheet/Teclado-membrana-matricial-4x4.pdf>.
- [4] «Fairchild Semiconductor,» Octubre 1999. [En línea]. Available: <https://pdf1.alldatasheet.es/datasheet-pdf/view/53731/FAIRCHILD/MM74C922.html>.
- [5] «Naylamp Mechatronics,» [En línea]. Available: <https://naylampmechatronics.com/drivers/31-modulo-relay-2-canales-5vdc.html>.
- [6] M. PC, «Servomotor SG90 datasheet,» 2014. [En línea]. Available: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf.
- [7] «Components 101,» 25 Septiembre 2017. [En línea]. Available: <https://components101.com/misc/buzzer-pinout-working-datasheet>.
- [8] «Digest, Curcuit,» 24 Octubre 2015. [En línea]. Available: <https://circuitdigest.com/article/16x2-lcd-display-module-pinout-datasheet>.

- [9] M. ELECTRÓNICA, 2004. [En línea]. Available:
<https://mvelectronica.com/producto/interfaz-serial-i2c-para-pantallas-lcd-1602-2004>.
- [10 «Wurth Peru,» 2022. [En línea]. Available:
] <https://eshop.wurth.pe/instalaciones-electricas/foco-1-contacto-12v-21w-rojo-2826.html>.
- [11 T. instrument, «Alldatasheet.com,» Noviembre 2000. [En línea]. Available:
] <https://pdf1.alldatasheet.com/datasheet-pdf/view/517588/T11/LM35.html>.
- [12 L. Carvajal, Metodología de la Investigación Científica. Curso general y
] aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.