



Proyecto Plotter Serial

NOMBRE	CARNÉ
Marvin Geobani Pretzantzìn Rosalío	202011394
Teresa Celeste Rodas Castillo	202010979
Andrea Rebeca Rubin Salazar	202000356
Irving Fernando Alvarado Asensio	202200349
Angel Samuel González Velásquez	202200263
Henri Eduardo Martinez Duarte	201704312
Andres Alejandro Agosto Mendez	202113580
Adler Alejandro Pérez Asensio	202200329
Ivan Alessandro Hilario Chacon	201902888

Índice

Índice	2
Introducción.....	3
Descripción del problema	4
Lógica del sistema.....	5
Funciones booleanas, mapas de Karnaugh y diagramas de estado	7
Diagramas del diseño del circuito.....	11
.....	12
Código comentado	13
Equipo utilizado	17
Presupuesto	19
Conclusiones.....	21
Recomendaciones.....	22
Anexos	23

Introducción

El presente documento detalla el desarrollo de un proyecto de impresión, un Plotter Cuadri-Color controlado por software. Este proyecto surge como respuesta a la necesidad de innovación en el campo de la impresión digital, buscando ofrecer una alternativa única y versátil a las impresoras tradicionales.

En las siguientes secciones, se describirán los requisitos, el diseño y la implementación de este Plotter Cuadri-Color, destacando cada aspecto relevante desde su concepción hasta su realización. El documento tiene como objetivo proporcionar una visión completa del proyecto, detallando su funcionamiento y capacidades.

A través de esta documentación, se invita a los lectores a explorar el proceso de desarrollo de este Plotter, con la esperanza de que inspire la creatividad y la innovación en el ámbito de la impresión digital.

Descripción del problema

El proyecto consiste en el desarrollo de un Plotter Cuadri-Color controlado por software, destinado a ofrecer una alternativa innovadora en el campo de la impresión digital. A diferencia de las impresoras convencionales, este Plotter se enfoca en la reproducción precisa de diseños utilizando un conjunto de cuatro colores distintos.

El objetivo principal del proyecto es proporcionar una solución que permita a los usuarios imprimir diseño con alta fidelidad y detalle, a la vez que ofrece una experiencia de usuario intuitiva y versátil. La necesidad de este proyecto surge de la limitación de las impresoras tradicionales para reproducir diseños con colores precisos y detallados, así como de la falta de opciones de personalización en el proceso de impresión.

Al desarrollar este Plotter Cuadri-Color, buscamos abordar estas limitaciones al ofrecer una solución que combine la precisión de la impresión por pluma con la versatilidad del control por software. Esto nos permitirá imprimir diseños con una calidad y detalle superiores, al tiempo que brindamos a los usuarios la capacidad de personalizar sus impresiones según sus necesidades y preferencias.

Lógica del sistema

El Plotter Cuadri-Color se compone de varios componentes clave que trabajan en conjunto para lograr su funcionamiento. A continuación, se describe la lógica general del sistema:

1. Aplicación de Escritorio:

- La aplicación de escritorio proporciona la interfaz de usuario para interactuar con el Plotter.
- Permite al usuario crear, editar y guardar imágenes en un lienzo virtual.
- Utiliza una matriz de 3x3 píxeles para representar el diseño de la imagen.
- Proporciona opciones para abrir, editar, guardar y guardar como archivos de extensión ".orga".
- Incluye un conjunto de figuras predefinidas para facilitar la creación de imágenes.

2. Interfaz PC-Plotter:

- La interfaz PC-Plotter establece la comunicación entre la aplicación de escritorio y el hardware del Plotter.
- Utiliza una conexión serial a través del Puerto paralelo LPT1 o Puerto Serial para enviar y recibir datos entre la PC y el Plotter.

3. Controlador del Plotter:

- El controlador del Plotter, implementado en un Arduino, recibe los datos enviados desde la PC a través de la interfaz serial.
- Procesa los comandos recibidos y controla los motores del Plotter para mover el cabezal de impresión a las coordenadas especificadas.
- Utiliza una matriz de Flip-Flops como memoria RAM para almacenar las coordenadas de impresión.

4. Cabezal de Impresión:

- El cabezal de impresión está equipado con un lápiz o herramienta de escritura que se desplaza sobre el área de impresión.
- Utiliza los datos proporcionados por el controlador del Plotter para determinar las coordenadas de impresión y realizar los trazos necesarios

5. Indicadores de Coordenadas:

- Se utilizan dos Displays para mostrar las coordenadas actuales en los ejes X y Y del cabezal de impresión.
- Estos indicadores son actualizados por el controlador del Plotter en tiempo real para reflejar la posición actual del cabezal.

6. Sensores de Alineación:

- Cuatro sensores de color están integrados en las esquinas del área de impresión para detectar la alineación correcta del papel.
- Emiten señales visuales indicando si la alineación es correcta o no, utilizando LEDs de color verde y rojo.

La lógica del sistema del Plotter Cuadri-Color implica la interacción entre la aplicación de escritorio, la interfaz PC-Plotter, el controlador del Plotter, el cabezal de impresión, los indicadores de coordenadas y los sensores de alineación para lograr la impresión precisa y personalizable de imágenes.

Funciones booleanas, mapas de Karnaugh y diagramas de estado

Contador de 10-0

Se eligieron flip flops de Tipo T- (4 Flip-Flops)

Tabla de Transiciones para el contador

Numero	Estados	Q0	Q1	Q2	Q3
0	S0	0	0	0	0
1	S1	0	0	0	1
2	S2	0	0	1	0
3	S3	0	0	1	1
4	S4	0	1	0	0
5	S5	0	1	0	1
6	S6	0	1	1	0
7	S7	0	1	1	1
8	S8	1	0	0	0
9	S9	1	0	0	1
10	S10	1	0	1	0

Transiciones	
S0	S10
S1	S0
S2	S1
S3	S2
S4	S3
S5	S4
S6	S5
S7	S6
S8	S7
S9	S8
S10	S9

Tabla de Estados para el contador

Estado Presente				Estado Futuro				T0	T1	T2	T3
Q0	Q1	Q2	Q3	Q0+	Q1+	Q2+	Q3+				
0	0	0	0	1	0	1	0	1	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	1	0	0	1	1
0	0	1	1	0	0	1	0	0	0	0	1
0	1	0	0	0	0	1	1	0	1	1	1
0	1	0	1	0	1	0	0	0	0	0	1
0	1	1	0	0	1	0	1	0	0	1	1
0	1	1	1	0	1	1	0	0	0	0	1
1	0	0	0	0	1	1	1	0	0	1	1
1	0	0	1	1	0	0	0	1	1	0	1
1	0	1	0	1	0	0	1	0	0	1	1

Contador de 20-0

Se eligieron flip flops de Tipo T- (5 Flip-Flops)

Numero	Estados	Q0	Q1	Q2	Q3	Q4	Transiciones	
0	S0	0	0	0	0	0	S0	S20
1	S1	0	0	0	0	1	S1	S0
2	S2	0	0	0	1	0	S2	S1
3	S3	0	0	0	1	1	S3	S2
4	S4	0	0	1	0	0	S4	S3
5	S5	0	0	1	0	1	S5	S4
6	S6	0	0	1	1	0	S6	S5
7	S7	0	0	1	1	1	S7	S6
8	S8	0	1	0	0	0	S8	S7
9	S9	0	1	0	0	1	S9	S8
10	S10	0	1	0	1	0	S10	S9
11	S11	0	1	0	1	1	S11	S10
12	S12	0	1	1	0	0	S12	S11
13	S13	0	1	1	0	1	S13	S12
14	S14	0	1	1	1	0	S14	S13
15	S15	0	1	1	1	1	S15	S14
16	S16	1	0	0	0	0	S16	S15
17	S17	1	0	0	0	1	S17	S16
18	S18	1	0	0	1	0	S18	S17
19	S19	1	0	0	1	1	S19	S18
20	S20	1	0	1	0	0	S20	S19

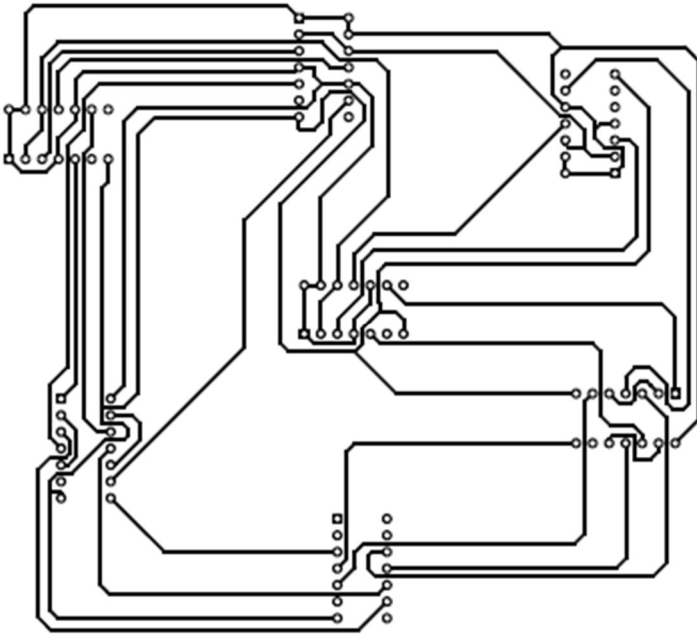
Tabla de Estados para el contador

1	Estado Presente					Estado Futuro					D0	D1	D2	D3	D4
	Q0	Q1	Q2	Q3	Q4	Q0+	Q1+	Q2+	Q3+	Q4+					
0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1
3	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0
4	0	0	1	0	0	0	0	0	1	1	0	0	0	1	1
5	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0
6	0	0	1	1	0	0	0	1	0	1	0	0	1	0	1
7	0	0	1	1	1	0	0	1	1	0	0	0	1	1	0
8	0	1	0	0	0	0	0	1	1	1	0	0	1	1	1
9	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0
10	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1
11	0	1	0	1	1	0	1	0	1	0	0	1	0	1	0
12	0	1	1	0	0	0	1	0	1	1	0	1	0	1	1
13	0	1	1	0	1	0	1	1	0	0	0	1	1	0	0
14	0	1	1	1	0	0	1	1	0	1	0	1	1	0	1
15	0	1	1	1	1	0	1	1	1	0	0	1	1	1	0
16	1	0	0	0	0	0	1	1	1	1	0	1	1	1	1
17	1	0	0	0	1	1	0	0	0	0	1	0	0	0	0
18	1	0	0	1	0	1	0	0	0	1	1	0	0	0	1
19	1	0	0	1	1	1	0	0	1	0	1	0	0	1	0
20	1	0	1	0	0	1	0	0	1	1	1	0	0	1	1

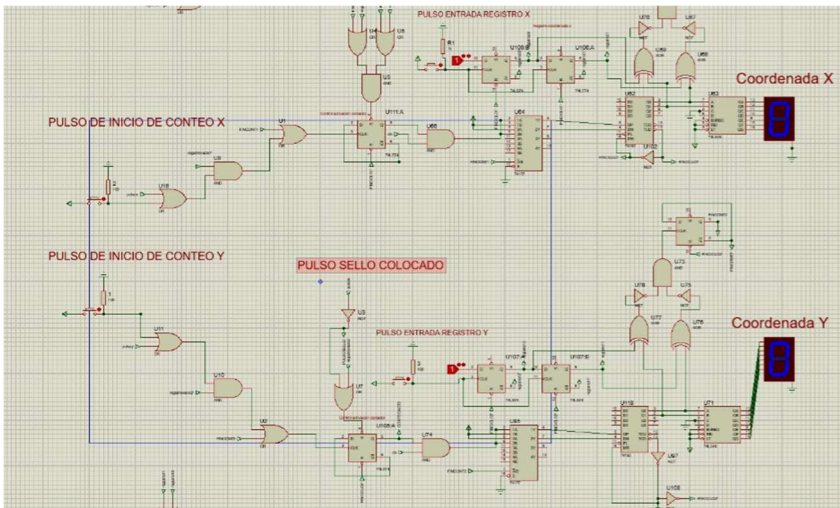
Serialización



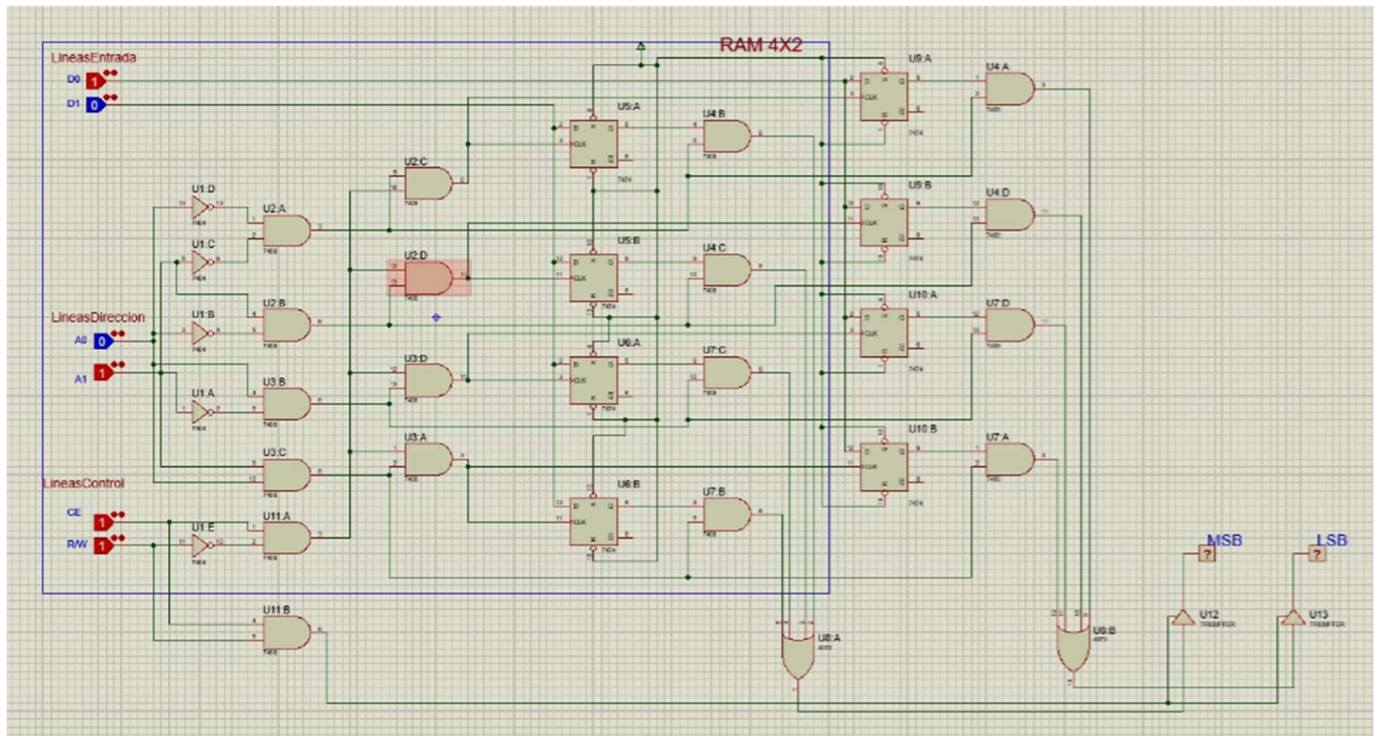
Diagramas del diseño del circuito



Diseño de la placa utilizada para implementar la memoria RAM



Circuito para mostrar la posición actual en X y en Y del cabezal de impresión



Diseño de la memoria ram de 4x2
implementada en el proyecto

Código comentado

Arduino

Librerías y Variables Globales:

- **Librerías:** La primera línea incluye la librería Servo.h, que proporciona las funciones necesarias para controlar un servo motor.
- **Variables Globales:** Se declaran varias variables globales para almacenar los pines de entrada/salida y otros parámetros necesarios para el funcionamiento del programa.

Configuración (setup()):

- **Inicialización de Pines:** Se configuran los pines de Arduino como entradas o salidas según sea necesario para conectar y controlar los componentes.
- **Inicialización de Servo:** Se inicializa un objeto de tipo Servo para controlar un servo motor conectado al pin especificado (pinServo). Se establecen los valores mínimo y máximo del pulso del servo (PULSOMIN y PULSOMAX).

Bucle Principal (loop()):

El bucle principal del programa realiza las siguientes acciones repetidamente:

- **Movimiento en Posición X:**
 - Se lee el estado de dos señales de entrada (posx1 y posx2).
 - Se determina el movimiento en la posición X basado en los estados de estas señales.
 - Se utiliza un bucle para controlar los motores paso a paso conectados a los pines INA1-INA4 o INB1-INB4 según sea necesario.
- **Movimiento en Posición Y:**
 - Similar al movimiento en posición X, pero para el eje Y y utilizando los pines INB1-INB4.

- **Dibujo de Formas Geométricas:**
 - Se leen las señales de entrada forma1 y forma2 para determinar qué forma dibujar.
 - Se ejecuta el código correspondiente para dibujar la forma seleccionada.
 - Cada forma tiene su propio bloque de código que controla los movimientos de los motores paso a paso para trazar la forma deseada.
- **Movimiento de Retorno a la Posición Inicial:**
 - Después de dibujar cada forma, los motores se mueven de regreso a la posición inicial en ambos ejes.
- **Movimiento de Retorno a la Posición X e Y:**
 - Después de dibujar cada forma, los motores se mueven de regreso a la posición inicial en ambos ejes.
- **Actualización del Ciclo de Coordenadas:**
 - Se activa y desactiva un pin (ciclosCoordenadas) para marcar el final de un ciclo de coordenadas.

Funciones de Movimiento (moverDer(), moverIzq(), moverAbajo(), moverArriba()):

- Estas funciones controlan los movimientos de los motores paso a paso en diferentes direcciones (derecha, izquierda, arriba, abajo).

Consideraciones Finales:

- El programa utiliza una secuencia de pasos para controlar los motores paso a paso, lo que permite un movimiento preciso en múltiples direcciones.
- El servo motor se utiliza para controlar el movimiento de un mecanismo de dibujo o cualquier otro dispositivo similar.

- El programa es modular y permite agregar fácilmente nuevas formas o funcionalidades al sistema de dibujo mecánico.

Python

Clase Movimiento:

Esta clase representa un movimiento en el plotter. Cada instancia de esta clase tiene atributos como la coordenada x, la coordenada y, el color y la figura que se va a dibujar.

Clase Analizador:

Esta clase es el analizador principal del código. Su objetivo es analizar el código fuente del lenguaje del plotter y extraer información relevante como los nombres de las impresiones, los conjuntos de impresión y los comentarios.

- Método `__init__`: Este método se llama cuando se crea una nueva instancia de la clase. En él se inicializan algunas variables y se definen expresiones regulares para buscar tokens específicos en el código.
- Métodos `obtener_contador_errores` y `obtener_movimientos`: Estos métodos se utilizan para obtener el recuento de errores y los movimientos extraídos del código, respectivamente.

Funciones para la interfaz gráfica:

Funciones como `cargar_archivo`, `imprimir`, `nuevo_archivo`, `guardar_como`, `guardar`, `salir` y `documentacion` son funciones asociadas a eventos en la interfaz gráfica. Por ejemplo, `cargar_archivo` abre un archivo de texto seleccionado por el usuario y carga su contenido en un área de texto.

Clase PixelArtEditor:

Esta clase define un editor de arte pixelado en una ventana Tkinter. Permite al usuario dibujar figuras simples (triángulos, estrellas, cruces y círculos) en un lienzo mediante clics y arrastres del mouse. También proporciona botones para seleccionar colores y formas.

Gramática de análisis sintáctico:

La gramática define las reglas de la estructura del código del lenguaje del plotter. Utiliza un analizador sintáctico LR para reconocer estas reglas y construir un árbol de análisis sintáctico.

En resumen, este código combina un analizador sintáctico para un lenguaje de plotter con una interfaz gráfica de usuario para permitir a los usuarios editar y visualizar el resultado de su código de forma interactiva.

Comunicación serial

- Importación de módulos: Al principio del script, se importan los módulos necesarios para la comunicación serial y para manejar el tiempo (serial y time, respectivamente).
- Establecimiento de la conexión serial: Se crea un objeto ser utilizando la función serial.Serial(). Se especifica el puerto serial al que está conectado el Arduino (por ejemplo, 'COM3' en sistemas Windows) y el baudrate de la comunicación (9600 baudios en este caso).
- Pausa de ejecución: Se utiliza time.sleep(2) para esperar dos segundos, lo que permite que se establezca correctamente la conexión serial antes de enviar datos.
- Envío de datos al Arduino: Se utiliza el método write() del objeto ser para enviar datos al Arduino.
- Espera de procesamiento: Se utiliza time.sleep(1) para esperar un segundo, permitiendo que el Arduino procese la información y realice alguna acción, como encender o apagar un LED.
- Envío de más datos.
- Cierre de la conexión serial: Finalmente, se utiliza ser.close() para cerrar la conexión serial una vez finalizada la comunicación.

Equipo utilizado

El desarrollo del Plotter Cuadri-Color requirió el uso de una variedad de equipos y herramientas, tanto de hardware como de herramientas manuales, que desempeñaron roles fundamentales en el proceso de diseño, fabricación y montaje. A continuación se detalla el equipo utilizado:

1. Computadora Personal:

Se utilizó una computadora personal para el desarrollo del software de control del Plotter, la creación de la interfaz de usuario y la generación de diseños gráficos.

2. Placa Arduino Uno:

Se empleó una placa Arduino Uno como el controlador principal del Plotter. Esta placa proporciona una interfaz de hardware programable para controlar los motores y gestionar las operaciones del Plotter.

3. Circuitos Integrados y Componentes Electrónicos:

Se utilizaron diversos circuitos integrados y componentes electrónicos, como compuertas lógicas, flip-flops y transistores, para construir la lógica de control del Plotter y la memoria RAM.

4. Equipo de Soldadura:

Se utilizó un soldador y estaño para soldar los componentes electrónicos en la placa de circuito impreso, incluyendo la memoria RAM y otros circuitos integrados.

5. Multímetro:

Se empleó un multímetro para realizar mediciones eléctricas, verificar la continuidad de los circuitos y solucionar problemas de conexión.

6. Herramientas de Corte y Perforación:

Se utilizaron herramientas como una Dremel para cortar y perforar la placa de circuito impreso, así como para ajustar y preparar los componentes electrónicos para el montaje.

7. Herramientas de Ensamblaje:

Se necesitaron herramientas como destornilladores y pinzas para ensamblar los componentes electrónicos en la placa de circuito impreso y montar el Plotter.

8. Materiales de Construcción:

Se utilizaron materiales como placas de circuito impreso, cables, conectores, madera, sensores y motores para construir y ensamblar el Plotter.

Estos son los principales equipos y herramientas utilizados en el desarrollo del Plotter Cuadri-Color. Cada uno de ellos desempeñó un papel importante en el proceso de diseño, fabricación y montaje del proyecto.

Presupuesto

Maqueta

Descripción	Cantidad	Precio unitario	Total
Motor Stepper	2	Q35.00	Q70.00
Servomotor	1	Q30.00	Q30.00
Sensor de color	2	Q75.00	Q150.00
Faja dentada (metro)	2.5	Q12.00	Q30.00
Pieza de cartón	2	Q24.00	Q48.00
Tubo de metal	1	Q10.00	Q10.00

Posición en X y en Y

Descripción	Cantidad	Precio unitario	Total
Circuito integrado AND	3	Q8.00	Q24.00
Circuito integrado OR	3	Q10.00	Q30.00
Circuito integrado XOR	1	Q12.00	Q12.00
Circuito integrado NOT	2	Q7.00	Q14.00
Multiplexor 2 a 1	2	Q9.00	Q18.00
Display de 7 segmentos	2	Q5.00	Q10.00
Decoder BCD a 7 segmentos	2	Q16.00	Q32.00
Circuito integrado de 8 flip flops D	2	Q12.00	Q24.00

Memoria RAM 4x2

Descripción	Cantidad	Precio unitario	Total
Circuito integrado AND	5	Q8.00	Q40.00
Circuito integrado OR	1	Q10.00	Q10.00
Circuito integrado NOT	2	Q7.00	Q14.00
Buffer	1	Q12.00	Q12.00
Circuito integrado de 8 flip flops D	1	Q12.00	Q12.00
Placa de cobre	1	Q18.00	Q18.00

Conclusiones

El desarrollo del Plotter Cuadri-Color ha sido un proceso que ha alcanzado los objetivos establecidos con éxito. A continuación, se presentan las principales conclusiones obtenidas durante el desarrollo del proyecto:

1. Logro de Objetivos:

Se ha alcanzado el desarrollo de un Plotter Cuadri-Color funcional que cumple con los requisitos establecidos inicialmente. El dispositivo es capaz de imprimir diseños.

2. Integración de Componentes:

Se logró una integración efectiva de los componentes electrónicos, incluyendo circuitos integrados, memoria RAM y sensores de color, para controlar con precisión el movimiento del cabezal de impresión y garantizar la calidad en la reproducción de imágenes.

3. Optimización de Recursos:

Se emplearon de manera eficiente los recursos disponibles, tanto en términos de hardware como de herramientas, para maximizar la eficiencia y la calidad del proyecto. La fabricación de la memoria RAM en placa ejemplifica la utilización de recursos disponibles para alcanzar los objetivos del proyecto.

4. Aprendizaje y Desarrollo:

El proyecto proporcionó una valiosa oportunidad de aprendizaje y desarrollo profesional para todos los miembros del equipo. Se adquirieron habilidades en diseño de circuitos electrónicos, programación de microcontroladores y colaboración en equipo.

En resumen, este proyecto ha demostrado el potencial de la creatividad, el trabajo en equipo y la aplicación de conocimientos técnicos para crear soluciones tecnológicas sólidas y eficaces.

Recomendaciones

Basándonos en la experiencia adquirida durante el desarrollo del proyecto del Plotter Cuadri-Color, se proponen las siguientes recomendaciones para futuros proyectos similares:

1. Planificación Detallada:

Es crucial realizar una planificación detallada antes de comenzar cualquier proyecto, identificando claramente los objetivos, los requisitos y el alcance del proyecto. Esto ayuda a evitar problemas y retrasos durante el desarrollo.

2. Pruebas y Validación Continuas:

Se recomienda realizar pruebas y validaciones continuas a lo largo del proceso de desarrollo, tanto a nivel de hardware como de software. Esto ayuda a detectar y corregir posibles problemas de manera temprana, minimizando los riesgos y mejorando la calidad del producto final.

3. Colaboración Efectiva en Equipo:

Fomentar una comunicación abierta y una colaboración efectiva entre los miembros del equipo es fundamental para el éxito del proyecto. Establecer reuniones regulares, asignar tareas claras y compartir información de manera transparente contribuye a mantener el proyecto en curso y a superar los desafíos que puedan surgir.

4. Documentación Exhaustiva:

Es importante llevar un registro detallado de todos los aspectos del proyecto, incluyendo el diseño, las decisiones tomadas, los problemas encontrados y las soluciones implementadas.

Estas recomendaciones están diseñadas para ayudar a garantizar el éxito y la efectividad en futuros proyectos, proporcionando una guía útil para abordar los desafíos y aprovechar las oportunidades que puedan surgir.

Anexos

