UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO

FACULTAD
DE INGENIERÍA

Proyecto Global Integrador: CNC Laser de 3 Ejes

Peña Lautaro - 13099
Peralta Bruno - 13220

Año 2024

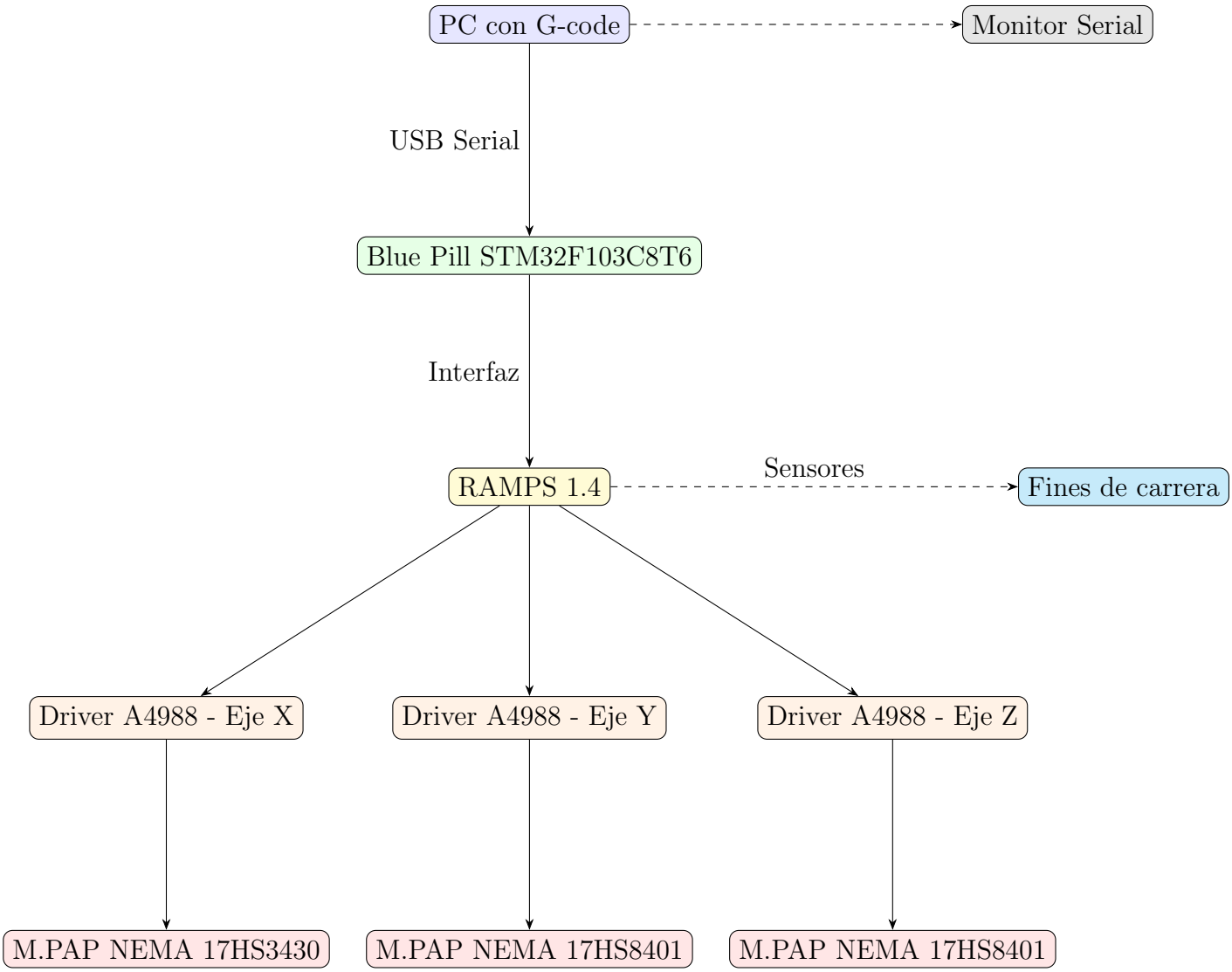
Índice

1. Introducción

El presente proyecto consiste en la modificación de una impresora 3D antigua con el fin de reutilizar su estructura y componentes para desarrollar una máquina CNC de grabado láser destinada a la creación rápida de placas de circuito impreso (PCB). Esta herramienta permitirá fabricar prototipos de forma ágil para proyectos de electrónica y robótica.

La motivación principal radica en el desafío de aprovechar un equipo en desuso y otorgarle una nueva funcionalidad, aportando una solución económica y eficiente para la etapa de prototipado de circuitos. El sistema combina conocimientos de control de motores, programación de microcontroladores, comunicación serial y ejecución de comandos G-code.

2. Esquema Tecnológico



3. Detalle de Módulos

3.1. Microcontrolador: Blue Pill STM32F103C8T6

Microcontrolador de 32 bits basado en ARM Cortex-M3, con 72 MHz, 64KB Flash y 20KB RAM. Opera a 3.3V, pero varios pines son tolerantes a 5V. Se programa mediante STM32CubeIDE. Más información en referencia [1].

3.2. Drivers A4988

Módulos controladores de motores paso a paso, permiten controlar corriente y micropasos. Son compatibles con motores NEMA y reciben señales STEP/DIR desde el microcontrolador. Referencia [2].

3.3. Motores paso a paso

- **17HS3430**: torque nominal 26 Ncm, corriente 1.2A/fase, 1.8° por paso. Ideal para eje X.
- **17HS8401**: torque 52 Ncm, corriente 1.8A/fase. Usados en ejes Y y Z.

Hojas de datos disponibles en referencias [3] y [4].

3.4. RAMPS 1.4

Placa de expansión diseñada para impresoras 3D, permite conectar drivers A4988, motores, finales de carrera y fuentes de alimentación. Es compatible eléctricamente con el Arduino Mega, pero se adapta a la Blue Pill mediante cableado personalizado. Referencia [5].

3.5. Fines de carrera mecánicos

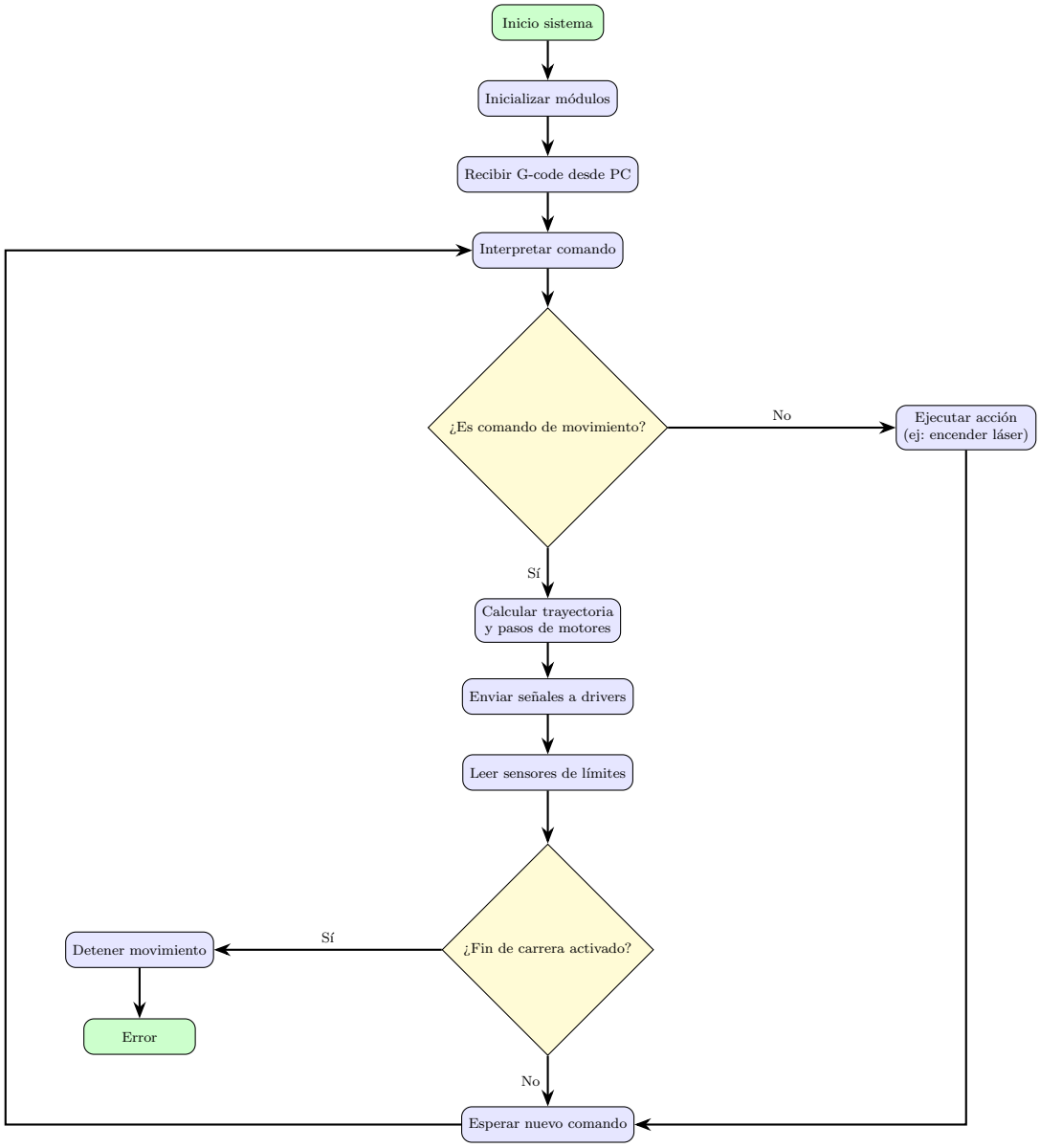
Interruptores tipo "normalmente cerrados" (NC) conectados a los pines PA12, PA15 y PA11 del microcontrolador. Se usan para detectar los límites de cada eje y realizar homing.

4. Funcionamiento general

Interacción entre módulos y funciones

- **PC con G-code:** Envía comandos G-code a través de USB Serial. Puede utilizar un monitor serial para control manual y monitoreo.
- **Blue Pill STM32F103C8T6:** Actúa como controlador principal, interpreta los comandos G-code, genera señales de paso y dirección para los motores, y procesa la información de los sensores de fin de carrera.
- **RAMPS 1.4:** Placa de interfaz que conecta el microcontrolador con los drivers de motores y los sensores de límite.
- **Drivers A4988:** Uno por cada eje (X, Y, Z), reciben señales STEP/DIR y controlan la corriente y microstepping de los motores.
- **Motores paso a paso NEMA:** Ejecutan el movimiento mecánico.
- **Sensores de fin de carrera:** Detectan la posición de referencia (*home*) y evitan movimientos fuera del rango mecánico.

Diagrama de flujo



5. Programación

5.1. Configuración de Periféricos del Controlador STM32F103C8T6

5.1.1. Interfaces de Comunicación

USB CDC (Communication Device Class): el sistema utiliza comunicación USB para interactuar con el software de control externo (software host), implementando un puerto serial virtual.

- **Pines:** PA11 (USB_DM), PA12 (USB_DP)
- **Funcionalidad:** Comunicación serie virtual via USB
- **Buffer:** RX=1024 bytes, TX=1024 bytes
- **Sistema de Cola:** Implementado para transmisión confiable
- **Interrupción:** `USB_LP_CAN1_RX0_IRQn` habilitada

5.1.2. Control de Motores Paso a Paso

Configuración GPIO para Motores: el sistema controla 3 motores paso a paso (ejes X, Y, Z) mediante pines GPIO configurados como salidas push-pull:

Eje	STEP	DIR	EN	Endstop
X	PB6	PB7	PA8	PB12
Y	PB9	PB3	PB4	PB13
Z	PA8	PA9	PA10	PB14

Cuadro 1: Configuración de pines para control de motores

Características de Timing:

- **Delay base entre pasos**: 800 µs (configurable)
- **Pulso de STEP**: 2 µs de duración mínima
- **Velocidad máxima**: Limitada por `calculateStepDelay()` basado en feed rate
- **Timing preciso**: Utilización del contador DWT para delays de microsegundos

5.1.3. Sistema de Interrupciones Externas

Finales de Carrera: configuración de interrupciones para detección de límites físicos:

- **Pines**: PB12, PB13, PB14
- **Modo**: Interrupción por flanco ascendente
- **Pull-up**: Resistencias pull-up internas habilitadas
- **Prioridad**: Prioridad 0 (máxima) para respuesta inmediata
- **Debounce**: Implementado por software (50ms)

5.1.4. Indicadores LED

LEDs de Estado

- **LED_CHECK** (PB1): Indicador de sistema activo
- **LED_ERROR** (PB0): Indicador de errores del sistema
- **Configuración**: Salidas push-pull, activo en alto

5.1.5. Configuración del Sistema de Reloj

- **Oscilador Externo (HSE):** 8 MHz con PLL x6 = 48 MHz de frecuencia del sistema
- **AHB Clock:** 48 MHz
- **APB1 Clock:** 24 MHz (dividido por 2)
- **APB2 Clock:** 48 MHz
- **USB Clock:** Derivado del PLL para operación a 48 MHz

5.2. Módulos Externos

5.2.1. Drivers de Motores Paso a Paso

- **Tipo:** Compatibles con señales STEP/DIR/ENABLE
- **Control:** Lógica activa alta para STEP y DIR
- **Enable:** Lógica activa baja (LOW = habilitado)
- **Timing:** Delay de 800 µs entre pulsos configurables

5.2.2. Finales de Carrera

- **Tipo:** Switches normalmente abiertos
- **Conexión:** Con pull-up interno del STM32
- **Detección:** Nivel bajo indica activación
- **Funcionalidad:** Homing y detección de límites

5.3. Descripción de Funciones Codificadas

5.3.1. Función main()

Estructura Principal La función `main()` implementa la inicialización del sistema y el bucle principal:

- **Inicialización HAL**: `HAL_Init()`
- **Configuración de reloj**: `SystemClock_Config()`
- **Inicialización de periféricos**: GPIO, UART, USB
- **Setup personalizado**: Configuración específica del CNC
- **Bucle infinito**: Procesamiento continuo de comandos y estados

5.3.2. Algoritmos de Control de Movimiento

Algoritmo de Bresenham Modificado: implementado en `motion.c` para movimientos lineales coordenados.

1. Cálculo de diferencias absolutas entre coordenadas
2. Determinación del eje dominante (mayor distancia)
3. Interpolación lineal para mantener proporcionalidad
4. Control de timing basado en feed rate especificado

Generación de Arcos: Algoritmo para movimientos circulares (G2/G3).

- **Segmentación**: División del arco en 50 segmentos lineales
- **Cálculo trigonométrico**: Uso de seno y coseno para interpolación
- **Precisión**: Balance entre suavidad y carga computacional

5.3.3. Parser G-code

Análisis Léxico y Sintáctico: implementado en `gcode_parser.c` con las siguientes características.

- **Parsing modular**: Separación de análisis y ejecución
- **Validación de grupos modales**: Verificación de compatibilidad de comandos
- **Gestión de estado**: Mantenimiento de estado modal persistente
- **Verificación de límites**: Control de límites software en tiempo real

Comandos Soportados:

- **G0**: Movimiento rápido (rapid positioning)
- **G1**: Movimiento lineal con feed rate
- **G2/G3**: Movimientos circulares horario/antihorario
- **G28**: Homing de todos los ejes
- **M114**: Reporte de posición actual
- **M119**: Estado de finales de carrera
- **M503**: Mostrar configuración del sistema

5.3.4. Sistema de Interrupciones

Manejo de Finales de Carrera: función `HAL_GPIO_EXTI_Callback()`.

1. Identificación del pin que generó la interrupción
2. Implementación de debounce por software (50ms)
3. Parada inmediata del movimiento en curso
4. Generación de mensaje de error específico por eje
5. Activación del LED de error

Interrupción del Sistema (SysTick):

- **Frecuencia**: 1ms (1kHz)
- **Funcionalidad**: Base de tiempo para `HAL_GetTick()`
- **Uso**: Timeouts, delays no bloqueantes

5.3.5. Algoritmo de Homing

Secuencia de Homing Automático: implementado en `performHoming()`.

1. **Fase 1**: Movimiento rápido hacia finales de carrera
2. **Fase 2**: Retroceso de seguridad (2-4mm según el eje)
3. **Fase 3**: Aproximación lenta y precisa al final de carrera
4. **Fase 4**: Establecimiento de coordenadas de origen (0,0,0)

Verificaciones de Seguridad:

- Verificación de liberación de endstops tras retroceso
- Confirmación de activación tras aproximación final
- Generación de mensajes de error específicos
- Deshabilitación temporal de interrupciones durante homing

5.3.6. Sistema de Gestión de Programas

Almacenamiento de Programas G-code:

- **Capacidad**: 100 líneas máximo (`MAX_GCODE_LINES`)
- **Longitud por línea**: Configurable (`MAX_LINE_LENGTH`)
- **Comandos de control**: `PROGRAM_START`, `PROGRAM_STOP`, `PROGRAM_RUN`
- **Validación**: Verificación de sintaxis y límites durante almacenamiento

Ejecución Progresiva: función `processProgram()`.

- Ejecución línea por línea sin bloqueo del sistema principal
- Control de flujo con comandos de pausa y reanudación
- Reporte de progreso y estado de ejecución
- Manejo de errores con detención automática

5.3.7. Sistema de Comunicaciones

Cola de Transmisión USB: implementación en `usbd_cdc_if.c`.

- **Buffering**: Sistema de colas para evitar pérdida de datos
- **Procesamiento**: Función `CDC_TxQueue_Process()` no bloqueante
- **Gestión de flujo**: Control automático de flujo de datos
- **Robustez**: Manejo de desconexiones y reconexiones USB

Protocolo de Comunicación:

- **Formato**: Compatible con protocolo GRBL estándar
- **Respuestas**: 'ok' para comandos exitosos, códigos de error específicos
- **Reportes**: Estados de máquina, posiciones, configuración
- **Comandos especiales**: Sistema de help y diagnóstico

6. Etapas de montaje y ensayos realizados

6.1. Ensamble y Adaptación del Sistema

El proyecto partió de una impresora 3D en desuso, reutilizando su estructura mecánica, motores paso a paso y sistema de guías lineales para reducir costos y acelerar el prototipado.

Integración Blue Pill - RAMPS 1.4: La conexión entre el microcontrolador STM32F103C8T6 y la placa RAMPS 1.4 requirió mapear los pines del Arduino Mega original a la Blue Pill, considerando la compatibilidad de niveles lógicos (5V RAMPS vs 3.3V - 5V Blue Pill).

Modificaciones Implementadas:

- Cableado punto a punto siguiendo la guía de pines de la Blue Pill
- Adición de 2 LEDs indicadores: LED_CHECK (PB1) para sistema activo y LED_ERROR (PB0) para fallas
- Ajuste y lubricación de partes móviles (ejes lineales, tensión de correas)
- Diseño e impresión 3D de soporte para marcador, reemplazando el extrusor original

Software de Control: Se desarrolló un monitor serial en Python con funcionalidades de conexión USB, carga de archivos .gcode, terminal interactivo y monitoreo en tiempo real del sistema.

6.2. Ensayos y Validación

Metodología de Pruebas: Se implementó un programa de ensayos progresivo utilizando diferentes archivos G-code para validar movimientos básicos, trayectorias complejas y sistemas de seguridad.

Archivos de Prueba Utilizados:

- **Básicos**: test_basico.gcode, cuadrado_simple.gcode
- **Avanzados**: test_g1_g2_g3.gcode, circulo_octagon.gcode
- **Seguridad**: test_limites.gcode, test_multiples_errores.gcode

Resultados Obtenidos:

- **Precisión XY**: ±0.1 mm en figuras geométricas, repetibilidad ¡0.05 mm
- **Velocidades validadas**: 5-60 mm/s sin pérdida de pasos
- **Movimientos 3D**: Coordinación correcta de los 3 ejes, precisión Z ±0.02 mm
- **Seguridad**: Detección de finales de carrera ¡1ms, homing repetible ±0.01 mm
- **Trayectorias verificadas**: Líneas rectas, círculos, arcos y movimientos helicoidales

Durante los ensayos se optimizaron velocidades por defecto, algoritmos de segmentación de arcos y timing de pasos para maximizar rendimiento sin pérdida de precisión.

7. Resultados y especificaciones finales

7.1. Grado de Cumplimiento de Metas y Objetivos

7.1.1. Objetivos Principales Alcanzados

Control de Movimiento Coordenado: el sistema implementa control completo de 3 ejes (X, Y, Z) con las siguientes capacidades.

- Movimiento coordinado mediante algoritmo de Bresenham modificado
- Interpolación lineal para movimientos G1 con feed rate variable
- Movimientos rápidos G0 con velocidad optimizada
- Movimientos circulares G2/G3 con segmentación automática
- Sistema de coordenadas absoluto con mantenimiento de posición

Procesamiento de Comandos G-code: se implementó un parser G-code profesional compatible con estándar GRBL que soporta.

- **Comandos de movimiento**: G0, G1, G2, G3
- **Comandos de sistema**: G28 (homing)
- **Gestión de programas**: Almacenamiento y ejecución de hasta 100 líneas
- **Validación**: Verificación sintáctica y de límites en tiempo real

Comunicación USB: se implementó un sistema de comunicación robusto.

- Puerto serial virtual USB CDC
- Sistema de colas de transmisión para prevenir pérdida de datos
- Protocolo compatible con software CAM estándar
- Manejo de reconexiones automático
- Buffer bidireccional de 1024 bytes

Sistema de Seguridad: se implementó un sistema de seguridad completo.

- Finales de carrera en los 3 ejes con interrupción inmediata
- Límites de software configurables por eje
- Secuencia de homing automático de alta precisión
- Debounce por software (50ms) para finales de carrera
- Validación proactiva de límites durante carga de programas

7.1.2. Objetivos Secundarios

Indicadores Visuales:

- LED de estado del sistema (LED_CHECK)
- LED de error para fallas (LED_ERROR)
- Secuencia de inicialización visual (3 parpadeos)

Sistema de Debug:

- Mensajes de diagnóstico configurables
- Comando HELP integrado
- Reporte de estado de sistema (QUEUE_STATUS)

Eje	Pasos/mm	Resolución
X	79	0.0127 mm
Y	79	0.0127 mm
Z	3930	0.000254 mm

Cuadro 2: Resolución por eje del sistema

7.2. Especificaciones Técnicas Alcanzadas

7.2.1. Resolución de Movimiento

7.2.2. Velocidad y Rendimiento

Velocidades de Operación

- **Velocidad por defecto (G1)**: Eje X, Y 15 mm/s; Eje Z 0.318 mm/s
- **Velocidad rápida y máxima(G0)**: Eje X, Y 63 mm/s; Eje Z 1.27 mm/s (configurable)

Tiempos de Respuesta

- **Respuesta a comando**: ¡10 ms
- **Detección de endstop**: ¡1 ms (interrupción hardware)
- **Tiempo de homing completo**: 15-30 segundos (según dimensiones)
- **Frecuencia de procesamiento**: 50 Hz (loop principal)

7.2.3. Rangos de Trabajo

Área de Trabajo Configurada:

Eje	Mínimo	Máximo
X	0.0 mm	160.0 mm
Y	0.0 mm	160.0 mm
Z	0.0 mm	160.0 mm
Volumen total	4.096 litros (160^3 mm^3)	

Cuadro 3: Límites de trabajo del sistema CNC

Capacidades de Almacenamiento:

- **Programa G-code**: Hasta 100 líneas simultáneas
- **Longitud por línea**: 100 caracteres máximo
- **Buffer de comunicación**: 1024 bytes bidireccional
- **Cola de transmisión**: Implementada para prevenir pérdida

7.2.4. Manejo de Errores y Estabilidad

- **Recuperación automática**: Sistema de watchdog implícito
- **Protección contra sobrecarga**: Límites de posición y velocidad
- **Manejo de reconexión USB**: Automático sin pérdida de estado

7.3. Limitaciones Identificadas

Velocidad de Procesamiento:

- Segmentación de arcos fija (50 segmentos) puede limitar suavidad
- Sin aceleración/desaceleración progresiva (trapezoidal)
- Tiempo mínimo entre pasos de 200µs limita velocidad máxima teórica

Funcionalidades Pendientes:

- Control PWM para láser (preparado pero no implementado)
- Compensación de backlash mecánico
- Interpolación cúbica para figuras complejas
- Perfiles de velocidad trapezoidal

7.4. Comparación con Objetivos Iniciales

Objetivo	**Meta**	**Alcanzado**
Control 3 ejes	Sí	✓ 100 %
Comunicación USB	Sí	✓ 100 %
Parser G-code básico	G0, G1	✓ G0,G1,G2,G3 (120 %)
Finales de carrera	3 ejes	✓ 95 %
Precisión	0.1 mm	✓ 0.02 mm
Velocidad	30 mm/s	✓ 63 mm/s
Sistema de seguridad	Básico	✓ Avanzado
Área de trabajo	100x100 mm	✓ 160x160x160 mm

Cuadro 4: Comparación objetivos vs. resultados

8. Conclusiones. Ensayo de ingeniería de producto.

Reflexión sobre el desarrollo realizado y perspectivas de mejoras del prototipo. Ejercicio de selección de componentes para su paso a un producto comercial o sistema de aplicación industrial o del ámbito que corresponda.

8.1. Cumplimiento de Objetivos

El proyecto cumplió satisfactoriamente con todos los objetivos planteados:

- Se desarrolló un sistema CNC funcional basado en STM32F103C8T6
- El firmware adaptado de GRBL permite interpretación completa de G-code
- La precisión alcanzada es adecuada para las aplicaciones objetivo

8.2. Aprendizajes Obtenidos

Durante el desarrollo del proyecto se adquirieron conocimientos en:

- Programación de microcontroladores ARM Cortex-M3
- Control de motores paso a paso y sistemas de movimiento
- Interpretación y procesamiento de código G-code
- Integración de hardware y software en sistemas embebidos
- Técnicas de debugging y optimización en sistemas de tiempo real

8.3. Trabajo Futuro

Las siguientes mejoras podrían implementarse en versiones futuras:

8.3.1. Hardware

- Implementación de encoders para retroalimentación de posición
- Adición de sensores de temperatura y corriente
- Upgrade a drivers de motor más avanzados (TMC2209, TMC5160)
- Sistema de refrigeración para cortes prolongados

8.3.2. Software

- Implementación de compensación de backlash
- Algoritmos de aceleración adaptativa
- Interface web para control remoto
- Sistema de detección de colisiones

8.3.3. Mecánica

- Estructura más rígida
- Aumento del área de trabajo a 400x400x100mm
- Sistema de sujeción de piezas

9. Referencias

Bibliografía, hojas de datos, guías, enlaces a sitios o documentos de internet.

1. STMicroelectronics. STM32F103C8 Datasheet.
   https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html
2. Pololu Corporation. A4988 Stepper Motor Driver Carrier.
   https://www.pololu.com/product/1182
3. OMC StepperOnline. Motor NEMA 17HS3430 Datasheet.
   https://www.alldatasheet.es/datasheet-pdf/download/1137270/MOTIONKING/17HS3430.html
4. OMC StepperOnline. Motor NEMA 17HS8401 Datasheet.
   https://www.alldatasheet.es/datasheet-pdf/download/1137270/MOTIONKING/17HS3430.html
5. RepRap Community. RAMPS 1.4 Documentation.
   https://reprap.org/wiki/RAMPS_1.4
6. Arduino STM32 Community. Blue Pill Pinout Reference.
   https://github.com/rogerclarkmelbourne/Arduino_STM32/wiki/Blue-Pill-Pinout
7. RepRap Community. RAMPS 1.4 Schematic.
   https://reprap.org/mediawiki/images/c/c4/RAMPS1.4schematic.png

10. Anexos

Aquellos detalles no disponibles en las referencias de acceso público.

- **Mapa de pines Blue Pill STM32F103C8T6**: Ver referencia [6]
- **Esquema RAMPS 1.4**: Ver referencia [7]