



# INFORME TECNICO

UPFEET

# Indice

1. Introduccion.....	2
a. Integrantes.....	2
b. Esfuerzos.....	2
c. Descripción General.....	2
2. Escaner.....	5
a. Estructura.....	5
i. Materiales y componentes.....	5
1. Estructura Principal.....	5
2. Estructura de apoyo.....	5
ii. Dimensiones.....	6
1. Estructura Principal.....	6
2. Estructura de apoyo.....	7
iii. Planos.....	8
1. Estructura Principal.....	8
2. Estructura de apoyo.....	9
b. Sistema de ejes.....	9
i. Componentes.....	9
ii. Planos de soportes caseros.....	11
c. Placa Controladora.....	12
i. Drivers.....	12
ii. Raspberry pi 4.....	15
1. Código para drivers.....	15
2. Código para sensor Lidar.....	16
3. Código de procesamiento de datos.....	17
4. Código para archivos.....	18
iii. Esquemático y PCB.....	25
iv. Alimentación.....	26
1. Alimentación de los drivers.....	26
2. Alimentación de la raspberry pi.....	27
3. Justificación del diseño de alimentación.....	28
d. Obstáculos encontrados y soluciones aplicadas.....	28
i. Obstáculos estructurales.....	28
ii. Obstáculos electrónicos.....	29
e. Conclusiones.....	30
i. Resultados técnicos alcanzados.....	30
ii. Ideas a implementar en un futuro.....	30
3. Impresora.....	31
a. Estructura mecánica.....	31
i. Estructura metálica.....	31
ii. Justificación de las dimensiones y disposición estructural.....	32
iii. Materiales utilizados y fundamentos de selección.....	32
b. Sistema de ejes.....	32
i. Configuración general de ejes (X, Y, Z).....	32
ii. Componentes mecánicos de transmisión.....	33
iii. Medidas, relaciones de movimiento y guías lineales.....	33

c. Electrónica de control.....	34
i. Placa madre SKR V1.4.....	34
ii. Distribución de pines (pinout general y específico).....	34
iii. Funcionamiento del sistema de control.....	34
iv. Implementación en la impresora desarrollada.....	35
d. Limitaciones técnicas y soluciones aplicadas.....	35
i. Problemas estructurales.....	35
ii. Problemas electrónicos.....	36
iii. Problemas de software.....	36
e. Firmware.....	37
i. Firmware base utilizado (Biqu / Marlin modificado).....	37
ii. Configuración y personalización para la SKR V1.4.....	37
iii. Cambios implementados en la versión final.....	37
iv. Resultados obtenidos tras la modificación.....	37
f. Conclusiones.....	37
i. Resultados técnicos alcanzados.....	37
ii. Posibles mejoras futuras.....	37

# 1.Introduccion

## 1.a. Integrantes

- Melina Alfonso
- Agustin Azorin
- Eugenio Herrera
- Tiago Lopez
- Victoria Viva
- Valentin Zaccari

## 1.b. Esfuerzos

Inicio de actividades:

- 20 de marzo 2025
- 35 semanas trabajadas
- 600 horas de trabajo total
- 17 horas aproximadas de trabajo por semana

## 1.c. Descripción general

El sistema UPFEET fue desarrollado como una solución tecnológica innovadora destinada a aliviar los dolores plantares y el cansancio físico del personal aeronáutico, en especial pilotos, tripulantes y técnicos que permanecen largas horas de pie o en posición estática dentro de la aeronave.

El proyecto integra un escáner plantar tridimensional con una impresora 3D Maker Pi Pro modificada, conformando una herramienta capaz de capturar, procesar y fabricar plantillas ortopédicas personalizadas de manera rápida, precisa y de bajo costo.

## **Objetivo principal**

El objetivo de UPFEET es mejorar el confort postural y reducir las dolencias musculoesqueléticas del personal de vuelo mediante la digitalización exacta del pie y la fabricación directa de plantillas ortopédicas a medida.

El sistema busca brindar una alternativa más económica, portable y eficiente frente a los estudios ortopédicos tradicionales, los cuales suelen requerir equipos costosos, personal especializado y largos tiempos de análisis.

UPFEET, en cambio, permite realizar el escaneo, procesar los datos y obtener una plantilla personalizada en el mismo entorno de trabajo, optimizando recursos sin comprometer la precisión.

Así, el dispositivo contribuye a:

- Disminuir la fatiga y el dolor durante vuelos prolongados.
- Optimizar la postura corporal, reduciendo sobrecargas en tobillos, rodillas y columna.
- Democratizar el acceso a herramientas de corrección postural mediante un sistema simple, reproducible y de bajo costo.

## **Desarrollo y componentes principales**

El sistema UPFEET se compone de dos módulos principales diseñados para funcionar de manera integrada:

### **1. Escáner plantar UPFEET:**

- Captura la geometría del pie mediante un sensor LIDAR montado sobre un sistema de ejes motorizado con drivers TMC2208 y motores NEMA 17.
- Genera una matriz tridimensional de alturas que representa la superficie plantar con alta resolución.
- Los datos procesados se transforman en un archivo STL imprimible mediante código desarrollado específicamente para el proyecto.

### **2. Impresora 3D Maker Pi Pro modificada:**

- Fue reacondicionada para adaptarse al proceso de fabricación de plantillas, con ajustes en firmware, volumen de impresión y calibración del extrusor.

- Permite imprimir directamente las plantillas ortopédicas generadas por el escáner, manteniendo la continuidad digital del proceso.
- Esta modificación formó parte integral del proyecto, asegurando compatibilidad entre hardware, software y materiales de impresión.

El conjunto constituye un ecosistema digital completo, capaz de escanear, procesar e imprimir soluciones personalizadas sin requerir equipos externos ni servicios adicionales.

## Alcance del proyecto

Si bien el enfoque inicial se orienta al entorno aeronáutico, el sistema UPFEET posee un amplio potencial de expansión hacia otras áreas donde el confort postural y la salud laboral son esenciales:

- Salud y rehabilitación: creación de plantillas terapéuticas para fisioterapia y ortopedia.
- Deportes y alto rendimiento: análisis biomecánico y prevención de lesiones.
- Industria y logística: soporte ergonómico para trabajadores que permanecen de pie durante extensas jornadas.
- Calzado técnico y ortopédico: integración con procesos digitales de diseño y producción aditiva.

La arquitectura modular del sistema permite ajustar sus dimensiones, sensores y algoritmos según el uso final, manteniendo los principios de portabilidad, economía y accesibilidad que definen el proyecto.

## Síntesis

UPFEET representa la convergencia entre biomecánica, manufactura digital y accesibilidad tecnológica.

Su desarrollo demuestra que es posible recrear el proceso de escaneo y fabricación ortopédica de forma compacta, económica y transportable, sin necesidad de un laboratorio especializado.

Gracias a la Maker Pi Pro modificada y al escáner plantar automatizado, el sistema logra reducir costos, tiempos y complejidad operativa, ofreciendo una alternativa viable para mejorar la salud postural del personal aeronáutico y de otros sectores que demanden ergonomía y confort prolongado.

## 2. Escaner

### 2.a. Estructura

#### 2.a.i. Componentes y materiales

##### 2.a.i.1. Estructura Principal

La estructura principal del sistema UPFEET está compuesta por una caja rectangular de acero fijada mediante 18 tornillos hexagonales distribuidos alrededor de los límites externos, 12 tornillos adicionales que aseguran la superficie transparente de acrílico de policarbonato, y 4 tornillos finales ubicados en las esquinas de los lados frontal y posterior para reforzar la rigidez general.

El material principal seleccionado fue acero debido a su alta resistencia mecánica, estabilidad dimensional y durabilidad ante esfuerzos repetitivos, especialmente considerando que la estructura soporta la presión del usuario al apoyar el pie.

Se optó por una geometría rectangular cerrada para favorecer la distribución homogénea de tensiones y simplificar la alineación del sistema óptico y electrónico, garantizando precisión en las mediciones.

El uso de tornillos hexagonales (en lugar de tornillos autorroscantes o cabezas ranuradas) responde a criterios de mantenimiento y modularidad:

- Permiten desmontar fácilmente las tapas o paneles para el mantenimiento interno sin deformar el material.
- Mejoran el par de apriete y reducen el riesgo de dañar las cabezas con herramientas comunes.
- Favorecen una apariencia profesional y robusta, coherente con el estándar de equipamiento médico-técnico que busca proyectar el sistema UPFEET.

Asimismo, el acrílico de policarbonato transparente se eligió por su alta resistencia al impacto y excelente transmisión lumínica, permitiendo la correcta lectura del escáner 3D sin interferencias ópticas ni riesgo de rotura ante uso intensivo. Otras alternativas como el vidrio o metacrilato fueron descartadas: el primero por su peso y fragilidad, y el segundo por su menor resistencia térmica.

En síntesis, esta configuración logra el equilibrio entre rigidez estructural, facilidad de mantenimiento y precisión del sistema óptico, tres condiciones indispensables para el funcionamiento confiable y seguro del equipo.

##### 2.a.i.2. Estructura de Apoyo

La estructura de apoyo también está fabricada en acero, pero presenta una forma en “U” cuadrada, diseñada específicamente para encastrarse y sostener firmemente a la estructura principal. Esta elección responde a criterios tanto funcionales como ergonómicos:

- La forma en “U” permite una fijación estable y autoalineada, evitando desplazamientos laterales del conjunto durante el transporte.
- Facilita el montaje y desmontaje rápido del sistema, útil para el traslado.
- Proporciona una superficie de apoyo estable para el pie del usuario, evitando cargas innecesarias sobre la estructura principal y mejorando la estabilidad postural durante el escaneo.

Se mantuvo la coherencia de materiales (acero) para garantizar una compatibilidad térmica y mecánica entre ambas estructuras, evitando deformaciones diferenciales por cambios de temperatura o humedad. Además, se descartaron materiales alternativos como aluminio o plástico reforzado, ya que:

- El aluminio, aunque más liviano, presenta menor rigidez frente a torsiones repetitivas y puede deformarse con el tiempo.
- Los plásticos estructurales, por su parte, no ofrecen la misma estabilidad dimensional ni resistencia al impacto que requiere un entorno técnico de laboratorio o consultorio.

## **2.a.ii. Dimensiones**

### **2.a.ii.1. Estructura Principal**

La estructura principal del sistema UPFEET presenta unas dimensiones generales de 500 × 300 × 300 mm, configurando un volumen compacto que permite alojar los componentes ópticos, electrónicos y mecánicos del escáner sin comprometer la estabilidad del conjunto.

En su parte superior, la superficie transparente de acrílico de policarbonato mide 300 × 150 × 10 mm, dimensiones que fueron seleccionadas de forma estratégica:

- El largo y ancho superan levemente las medidas promedio del pie humano (entre 260 y 280 mm de largo y 100 a 110 mm de ancho), lo que garantiza una superficie de apoyo cómoda y segura para usuarios de distintos tamaños.
- El espesor de 10 mm ofrece rigidez suficiente para resistir el peso del usuario sin flexión visible ni deformaciones ópticas que pudieran afectar la calidad del escaneo.
- Esta proporción permite, además, mantener un campo visual completo del sensor 3D, optimizando el enfoque y la precisión de captura.

En conjunto, las dimensiones fueron determinadas para lograr el equilibrio entre ergonomía, precisión de escaneo y facilidad de transporte, manteniendo el peso total del sistema dentro de límites razonables para su manipulación manual.

### **2.a.ii.2. Estructura de Apoyo**

La estructura de apoyo fue diseñada con dimensiones ligeramente superiores a las de la estructura principal, permitiendo un encastre preciso y estable entre ambas piezas. Esta diferencia dimensional garantiza:

- Un ajuste firme, que evita holguras o movimientos durante el uso.
- Una correcta distribución de cargas, reduciendo el esfuerzo sobre los tornillos de fijación.
- Un montaje simple y repetible, que facilita la alineación automática del escáner respecto al plano del suelo.

El sobredimensionado se limita a unos pocos milímetros por lado, suficiente para asegurar el acople funcional sin interferencias y manteniendo la estética compacta del equipo.

De esta manera, las dimensiones conjuntas de ambas estructuras responden tanto a criterios ergonómicos (comodidad del usuario) como a criterios de ingeniería (estabilidad, modularidad y resistencia), garantizando la integridad física del escáner y la exactitud en sus mediciones tridimensionales.

### **2.b.i. Componentes**

El sistema de ejes del escáner UPFEET está conformado por tres ejes lineales que permiten el desplazamiento controlado del sensor de medición: dos ejes Z y un eje X.

Los dos ejes Z son los responsables de mover el eje X en sentido longitudinal (adelante y atrás), mientras que el eje X desplaza el sensor LIDAR de derecha a izquierda, cubriendo así toda la superficie de escaneo.

Esta disposición de ejes fue seleccionada por su estabilidad, precisión y simplicidad mecánica, además de permitir un mantenimiento más accesible que otras configuraciones como el sistema cartesiano clásico con eje Y independiente o el pórtico invertido.

#### **Configuración de motores y acoples**

Cada eje está accionado por un motor paso a paso independiente, garantizando movimientos precisos y sincronizados.

- Los motores de los ejes Z están atornillados a las paredes internas traseras de la estructura principal, lo que proporciona un anclaje rígido y reduce la transmisión de



vibraciones al resto del chasis.

- El motor del eje X se encuentra sostenido por un soporte impreso en 3D, diseñado a medida, que se fija a la varilla de uno de los ejes Z. Esta solución “casera” permite ajustar la posición del motor con alta flexibilidad, adaptándose al recorrido real del sensor sin necesidad de mecanizados complejos en el bastidor.

Cada motor se acopla a su respectiva varilla mediante un acople flexible de 5 mm a 8 mm, que conecta el eje del motor con la varilla roscada de acero inoxidable de 8 mm de diámetro.

La elección de acoples flexibles se justifica porque:

- Absorben ligeras desalineaciones entre el motor y la varilla, evitando esfuerzos radiales que podrían dañar los rodamientos.
- Reducen vibraciones y prolongan la vida útil de los componentes.
- Facilitan el mantenimiento, ya que se pueden desmontar sin desajustar toda la transmisión.

### **Varillas y soportes estructurales**

Cada varilla roscada (en acero inoxidable) está acompañada por dos varillas lisas de guía, también de acero, que garantizan la rectitud del movimiento y la estabilidad del eje móvil.

En los ejes Z, las varillas roscadas se sostienen mediante soportes frontales atornillados a la pared interna de la estructura principal, lo que evita que el peso del eje X provoque flexión o caídas durante la operación.

En el eje X, las varillas lisas están unidas a un soporte móvil que se apoya sobre una de las varillas roscadas del eje Z, logrando así una interconexión mecánica sólida y sincronizada entre los ejes.

Cada soporte móvil incorpora una tuerca antibacklash, lo que compensa el juego mecánico entre la rosca y la varilla, garantizando:

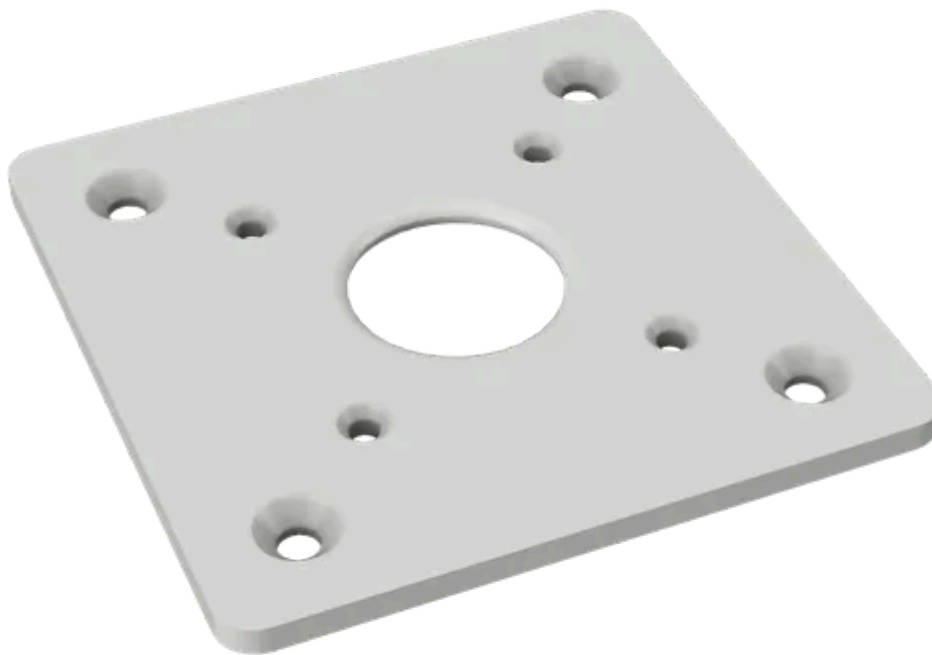
- **Mayor precisión en el desplazamiento** del sensor LIDAR.
- **Ausencia de vibraciones** en el sentido inverso de movimiento.
- **Repetibilidad de posición**, fundamental para obtener nubes de puntos limpias y sin errores de registro.

### **Justificación del diseño del sistema de ejes**

Se optó por este sistema de tres ejes con motores independientes por las siguientes razones:

1. **Precisión y repetibilidad:** permite controlar con exactitud la posición del sensor, condición esencial para reconstrucciones tridimensionales de alta fidelidad.
2. **Simplicidad constructiva:** elimina la necesidad de mecanismos de transmisión por correas o engranajes, reduciendo el mantenimiento y los puntos de falla.
3. **Modularidad y reparación accesible:** los componentes (varillas, tuercas, acoples, soportes) son estándar y fáciles de reemplazar, lo cual facilita las tareas de ajuste o actualización.
4. **Compatibilidad con impresión 3D:** el uso de soportes impresos a medida permite adaptar el diseño a variaciones dimensionales y optimizar el espacio interno del escáner sin requerir mecanizado industrial.

## 2.b.ii. Planos de Soportes caseros



## 2.c. Placa Controladora

### 2.c.i. Drivers

El sistema UPFEET utiliza drivers TMC2208 para el control de los motores paso a paso de los ejes X y Z. Estos módulos se encargan de traducir las señales digitales de la placa controladora en impulsos eléctricos precisos, regulando la corriente que llega a cada bobina del motor. De este modo, permiten un movimiento suave, silencioso y exacto del sistema de ejes.

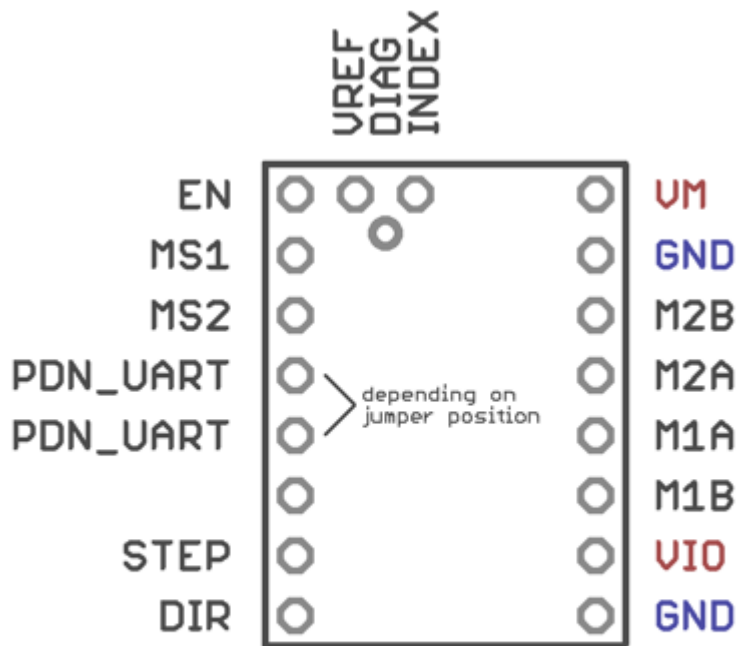
#### Principio de funcionamiento electrónico

El TMC2208 funciona como un controlador de micropasos (hasta 1/256 de paso) que emplea una tecnología denominada StealthChop, capaz de generar una corriente sinusoidal extremadamente estable.

Esto reduce el ruido electromecánico típico de los motores paso a paso convencionales, eliminando vibraciones y mejorando la precisión del posicionamiento.

En el esquema mostrado (ver figura), se observan las principales conexiones del driver:

- **VM (Voltaje de motor):** alimentación principal de los motores, 12V
- **VIO:** alimentación lógica, conectada a 3.3 V desde la placa controladora.
- **M1A / M1B / M2A / M2B:** salidas hacia las dos bobinas del motor paso a paso.
- **STEP / DIR:** señales digitales provenientes del microcontrolador, que determinan los impulsos de paso y el sentido de giro.
- **EN:** señal de habilitación, utilizada para activar o desactivar el driver, siempre en 0 para que esté activado el driver.
- **PDN\_UART:** interfaz de comunicación que permite configurar parámetros avanzados como corriente, microstepping o modo de funcionamiento.
- **VREF:** pin de referencia de voltaje, utilizado para ajustar la corriente máxima que circulará por las bobinas del motor.



### Por qué no utilizamos los pines MS1 y MS2 del driver TMC2208

Los pines MS1 y MS2 en los controladores de motores paso a paso tradicionalmente se utilizan para definir el modo de micropasos del motor.

En drivers más antiguos, como el A4988 o el DRV8825, estos pines se conectaban a niveles lógicos fijos (HIGH o LOW) para seleccionar entre 1,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$ ,  $\frac{1}{16}$  o  $\frac{1}{32}$  de paso, configurando así la resolución mecánica del motor.

Sin embargo, en el caso del TMC2208, esos pines ya no se utilizan para configurar el microstepping.

Este modelo de driver incorpora un sistema de control interno digital más avanzado, lo que hace que la configuración de micropasos se realice automáticamente o mediante software, no por hardware.

En el **TMC2208**, el número de micropasos se puede definir de tres maneras:

1. **Por comunicación UART:** el firmware (por ejemplo, Marlin o el código en Raspberry Pi) envía comandos digitales para configurar la resolución.
2. **Por resistencias internas preconfiguradas:** cuando no se usa UART, el driver arranca por defecto en **1/16 de paso con interpolación a 1/256**, lo que da movimientos ultrasuaves.
3. **Por pines CFG (según versión):** en algunas placas, los pines **CFG1 y CFG2** reemplazan a MS1/MS2, pero se dejan desconectados o a GND para usar la

configuración por defecto.

En el proyecto **UPFEET**, no se utilizaron los pines **MS1** y **MS2** porque:

- **No son necesarios para el control del microstepping en el TMC2208.**  
El driver ya cuenta con interpolación interna **StealthChop** que convierte automáticamente los micropasos a 1/256 sin necesidad de configuración externa.
- **Simplifica el cableado.**  
Evitar el uso de esos pines reduce la cantidad de conexiones entre la Raspberry Pi y los drivers, disminuyendo posibles interferencias o errores de montaje.
- **Compatibilidad con el firmware.**  
Al usar la configuración por defecto, el sistema puede comunicarse con los drivers **sin requerir comandos UART adicionales** ni librerías de configuración, lo que se adapta mejor a un entorno de desarrollo experimental y estable.
- **Estabilidad en el movimiento.**  
El modo interpolado del TMC2208 ofrece movimientos suaves, silenciosos y precisos, ideales para el escáner 3D del sistema UPFEET, sin necesidad de modificar manualmente la resolución.

#### **Ventajas de no usar MS1/MS2**

- **Menor complejidad eléctrica:** menos cables y menor riesgo de cortocircuitos o errores de pinout.
- **Funcionamiento automático y silencioso:** el TMC2208 gestiona la corriente y los micropasos internamente mediante el algoritmo **StealthChop**.
- **Compatibilidad con código genérico:** el driver funciona correctamente con los mismos comandos básicos (STEP y DIR) que un A4988, sin necesidad de configuración adicional.
- **Mayor precisión efectiva:** aunque se trabaja en 1/16 de paso, el driver interpola digitalmente hasta **1/256**, logrando movimientos más finos que los configurables por hardware.

#### **Ajuste del voltaje de referencia (Vref)**

El Vref se utiliza para establecer la corriente máxima que suministrará el driver al motor. En el caso de los motores NEMA 17 genéricos, típicos en aplicaciones de escaneo o impresión 3D, la corriente nominal suele ser de 1 A a 1.2 A por fase.

La corriente máxima se calcula según la fórmula aproximada del TMC2208:

$$I_{max} = \frac{V_{ref}}{1.77 \times R_{sense}}$$

Donde  $R_{sense}$  suele ser de  $0.11 \Omega$  en los módulos más comunes.  
Para obtener una corriente de 1 A:

$$V_{ref} = 1.77 \times 0.11 \times 1 \approx 0.19V$$

En la práctica, se recomienda ajustar el Vref entre 0.18 V y 0.22 V, lo cual:

- Evita sobrecalentamientos del motor o del driver.
- Proporciona suficiente torque para mover el sistema sin pérdida de pasos.
- Extiende la vida útil del motor y reduce el ruido operativo.

Este rango fue validado experimentalmente en el sistema UPFEET, donde se busca movimiento suave y preciso, priorizando la estabilidad del escaneo sobre la velocidad máxima.

### **Ventajas y desventajas del TMC2208**

#### **Ventajas:**

- Funcionamiento ultrasilencioso (StealthChop): ideal para entornos clínicos o de laboratorio donde se requiere bajo nivel de ruido.
- Alta resolución de microstepping (hasta 1/256): mejora la precisión de posicionamiento del sensor LIDAR.
- Protección térmica y de sobrecorriente integrada.
- Compatibilidad directa con drivers A4988 o DRV8825, facilitando el reemplazo.
- Configuración por UART: permite ajustar corriente, pasos y modos de trabajo sin modificar hardware.

#### **Desventajas:**

- Requiere una correcta ventilación si se trabaja a corrientes elevadas.
- Menor torque a bajas velocidades en comparación con el DRV8825, si no se calibra el Vref adecuadamente.

- Configuración UART opcional puede complicar la instalación si no se dispone de firmware compatible.

### Justificación de elección

El TMC2208 fue seleccionado como driver principal del sistema UPFEET por las siguientes razones:

1. **Compatibilidad directa:** se adapta perfectamente a la topología de la placa controladora utilizada, sin necesidad de rediseñar el sistema eléctrico.
2. **Rendimiento acústico:** la tecnología StealthChop reduce significativamente el ruido mecánico del escáner, aportando una experiencia más profesional.
3. **Costo-beneficio:** ofrece prestaciones similares a modelos más avanzados (como el TMC2209 o TMC2130) a un precio considerablemente menor.
4. **Disponibilidad:** eran los mismos drivers utilizados por la impresora 3D del laboratorio, por lo que se aprovecharon unidades existentes, evitando costos de envío adicionales y garantizando compatibilidad de repuestos.
5. **Mantenimiento sencillo:** el ajuste de  $V_{ref}$  es accesible y los parámetros son fácilmente replicables en caso de sustitución.

(La figura a continuación muestra el pinout de conexión del TMC2208 utilizado en el sistema UPFEET.)

## 2.c.ii. Raspberry pi

La Raspberry Pi 4 actúa como el cerebro principal del sistema UPFEET, coordinando el control de los motores, la adquisición de datos del sensor LIDAR, el procesamiento de la información tridimensional y el almacenamiento de los resultados.

A continuación, se describen las principales secciones de código utilizadas en el sistema.

### 2.c.ii.1. Código para drivers

Este código se encarga de enviar las señales de paso (STEP) y dirección (DIR) a los drivers TMC2208 que controlan los motores NEMA 17.

Permite realizar movimientos precisos y sincronizados en los ejes X y Z.

```

import RPi.GPIO as GPIO
import time

# Pines asignados a Los drivers
DIR_X = 20
STEP_X = 21
DIR_Z1 = 16
STEP_Z1 = 12

# Configuración inicial de pines
GPIO.setmode(GPIO.BCM)
GPIO.setup([DIR_X, STEP_X, DIR_Z1, STEP_Z1], GPIO.OUT)

def mover_motor(dir_pin, step_pin, pasos, direccion):
    GPIO.output(dir_pin, direccion)
    for i in range(pasos):
        GPIO.output(step_pin, GPIO.HIGH)
        time.sleep(0.001)
        GPIO.output(step_pin, GPIO.LOW)
        time.sleep(0.001)

# Ejemplo: mover eje X hacia adelante
mover_motor(DIR_X, STEP_X, 500, GPIO.HIGH)

```

### Explicación:

El programa genera pulsos digitales que determinan la rotación de cada motor.

El número de pasos y la dirección son ajustables, permitiendo calibrar la velocidad y distancia de desplazamiento del sensor.

### 2.c.ii.2. Código para LIDAR

Este bloque se comunica con el sensor LIDAR para leer las distancias medidas.

Utiliza una interfaz UART (o I2C, dependiendo del modelo), procesando los datos de forma continua mientras el sensor se desplaza.



```

import serial
import time

# Configuración del puerto serie del LIDAR
lidar = serial.Serial("/dev/ttyUSB0", 115200, timeout=1)

def leer_lidar():
    if lidar.in_waiting > 0:
        data = lidar.readline().decode().strip()
        return data
    return None

# Lectura continua
while True:
    distancia = leer_lidar()
    if distancia:
        print("Distancia medida:", distancia, "mm")
    time.sleep(0.05)

```

### Explicación:

Cada lectura devuelve una distancia en milímetros correspondiente al punto detectado por el sensor.

Estos datos se envían luego al módulo de procesamiento para construir la nube de puntos del escaneo.

### 2.c.ii.3. Código pra procesamiento de datos

Esta parte del sistema organiza los datos obtenidos del sensor y los convierte en una matriz tridimensional (nube de puntos) que representa la forma del pie.

```

import numpy as np

# Simulación de puntos obtenidos del LIDAR
datos_lidar = [(x, y, np.random.randint(0, 200)) for x in range(30) for y in range(10)]

# Procesamiento básico
def generar_matriz(datos):
    matriz = np.zeros((30, 10))
    for x, y, z in datos:
        matriz[x][y] = z
    return matriz

matriz_alturas = generar_matriz(datos_lidar)
print("Matriz de alturas generada correctamente")

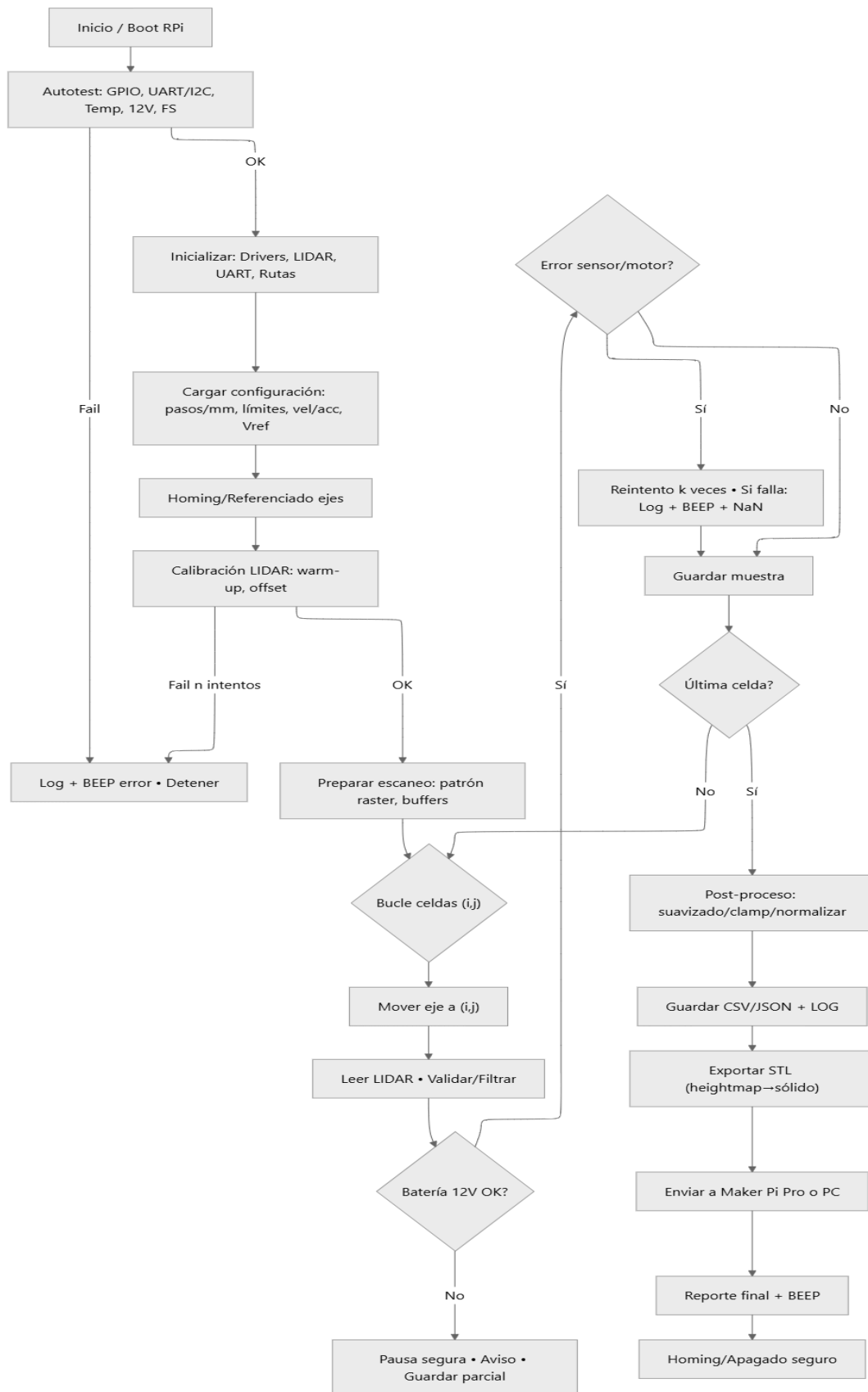
```

### Explicación:

Cada punto (x, y, z) representa una coordenada de la superficie escaneada.

En versiones futuras, esta matriz podrá exportarse a formato STL para la impresión 3D de la plantilla.

## 2.c.ii.4. Código para archivos



# **Explicación detallada de los bloques del flujo de procesos**

## **1. Inicio / Boot de la Raspberry Pi**

Al encender el sistema, la Raspberry Pi ejecuta su secuencia de arranque y carga el entorno operativo.

Se inicia el servicio principal del escáner (por ejemplo, un script Python o un daemon) que coordina los motores, sensores y registros de datos.

El software verifica la integridad de los archivos de configuración y la disponibilidad del sistema de archivos.

## **2. Autotest del sistema**

Antes de habilitar cualquier movimiento, el sistema ejecuta un autodiagnóstico:

- Comprueba la conectividad de los puertos GPIO, UART y I<sup>2</sup>C.
- Verifica el voltaje de alimentación de 12 V para los drivers y la disponibilidad de la fuente de 5 V.
- Controla la temperatura de la CPU y el estado de las pilas/baterías.
- Si detecta alguna anomalía, registra el evento en un archivo de log y emite un código acústico (beep) correspondiente.

Esto previene que el escáner inicie con una falla eléctrica o térmica.

## **3. Inicialización de periféricos**

Una vez completado el autotest, la Raspberry Pi inicializa:

- Los drivers TMC2208 (habilita pines STEP/DIR y señales ENABLE).
- El sensor LIDAR (establece comunicación UART y velocidad en baudios).
- El reloj interno o temporizador de muestreo.
- Las rutas de guardado para los datos del escaneo.

En esta etapa también se limpia la memoria de buffer y se prepara el espacio para la matriz de alturas.

## **4. Carga de configuración**

Se leen los parámetros almacenados en un archivo de configuración, entre ellos:

- Pasos por milímetro (relación del motor con el desplazamiento real).
- Límites de recorrido de los ejes.
- Velocidades de desplazamiento y aceleraciones permitidas.
- Valor de  $V_{ref}$  objetivo (corriente máxima de los motores).  
Esto garantiza que el escáner trabaje bajo los mismos parámetros en cada sesión.

## **5. Homing / Referenciado**

El sistema realiza un movimiento controlado hasta los puntos de referencia de los ejes (inicio físico o virtual).

Esto permite definir el punto (0, 0, 0) del sistema de coordenadas y evitar choques mecánicos.

Si se detectan finales de carrera, se detiene automáticamente.

El resultado del homing se almacena para sincronizar el movimiento posterior.

## **6. Calibración del sensor LIDAR**

El sensor realiza una fase de estabilización (warm-up) y se calibra la distancia cero:

- Se toman varias lecturas con el sensor apuntando a una superficie plana conocida.
- Se calcula un promedio y se guarda un offset de corrección.
- Si la variabilidad de los datos es alta, se repite la calibración hasta obtener estabilidad.

Si después de varios intentos la calibración falla, se registra el error y el sistema no continúa hasta resolverlo.

## **7. Preparación del escaneo**

El software define el patrón de movimiento (raster o serpenteado) y crea los buffers de almacenamiento para las lecturas.

Se establecen las dimensiones de la grilla (por ejemplo,  $60 \times 30$  puntos) y el orden de desplazamiento de los ejes X y Z.

También se activan los indicadores visuales o sonoros que confirman que el sistema está listo para iniciar el escaneo.

## **8. Bucle de escaneo**

Esta es la parte central del proceso:

1. El eje Z mueve el sensor a la posición correspondiente a la fila actual.
2. El eje X se desplaza paso a paso, y en cada posición el LIDAR realiza una lectura.
3. El valor de distancia medido se almacena en la matriz junto con sus coordenadas (x, y, z).
4. Se aplica una validación (verifica que la lectura sea numérica y esté dentro del rango útil).
5. Si alguna muestra falla, se repite la medición o se marca como NaN (no válida).

Durante el proceso:

- Si el voltaje de 12 V desciende, el escaneo se pausa automáticamente, guarda el progreso y emite un aviso acústico.
- Si ocurre un error de comunicación o motor, intenta un reintento local antes de abortar.

## **9. Post-proceso**

Cuando finaliza el recorrido completo:

- Se aplica un filtro de suavizado (por ejemplo, media 3×3) para reducir ruido.
- Se normalizan las alturas (ajuste de escala y eliminación de valores anómalos).
- Se convierte la matriz a milímetros reales según la calibración previa.

El resultado es una matriz tridimensional precisa del perfil plantar.

## **10. Guardado de datos (CSV / JSON)**

El sistema genera archivos de respaldo:

- **CSV o JSON** con todas las lecturas del escaneo.
- **Archivo de log** con hora, fecha, duración, temperatura y errores detectados. Estos archivos permiten repetir el análisis sin tener que volver a escanear.

## 11. Exportación a STL

Usando el algoritmo de triangulación, la matriz procesada se transforma en un sólido 3D (STL ASCII).

El modelo incluye la superficie plantar, la base inferior y las paredes laterales, asegurando que sea manifold (cerrado) y apto para impresión 3D.

La Raspberry Pi crea el archivo y lo deja listo para la siguiente etapa.

## 12. Envío a la Maker Pi Pro

El archivo STL se envía a la Maker Pi Pro modificada (ya sea por USB o red local).

La impresora inicia la fabricación de la plantilla ortopédica personalizada directamente desde el modelo obtenido.

En esta etapa puede verificarse la temperatura del hotend y la cama antes de imprimir.

## 13. Reporte final y señal acústica (BEEP)

Una vez completado el proceso, el sistema emite un **código sonoro** para indicar el estado final:

- 1 beep corto → Proceso exitoso.
- 2 beeps cortos → Advertencia o lectura parcial.
- 3 beeps cortos → Error crítico.  
También se genera un resumen de resultados (tiempo total, lecturas válidas, archivo STL generado).

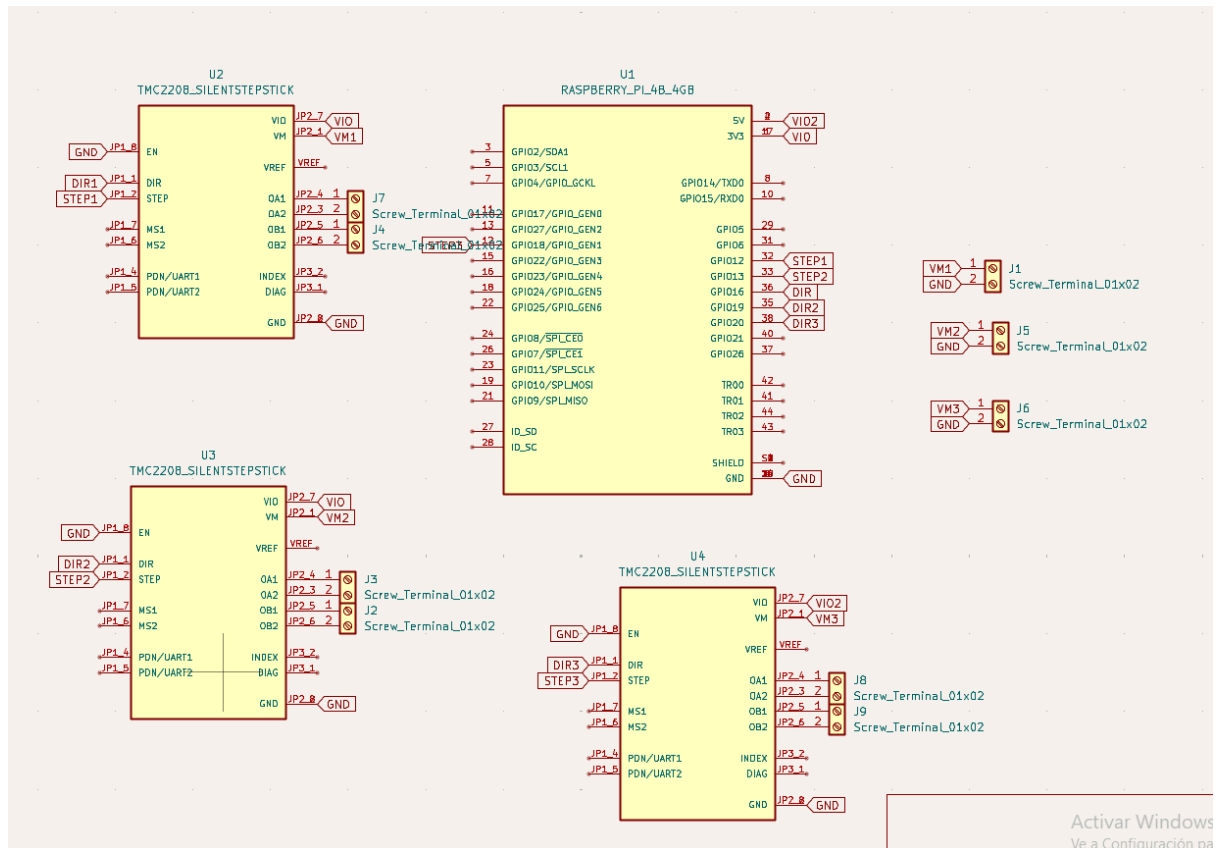
## 14. Homing final y apagado seguro

Finalmente, los ejes regresan a la posición de reposo y se deshabilitan los motores para reducir consumo.

El sistema cierra los puertos UART y GPIO, guarda el log final y apaga la Raspberry Pi de forma segura.

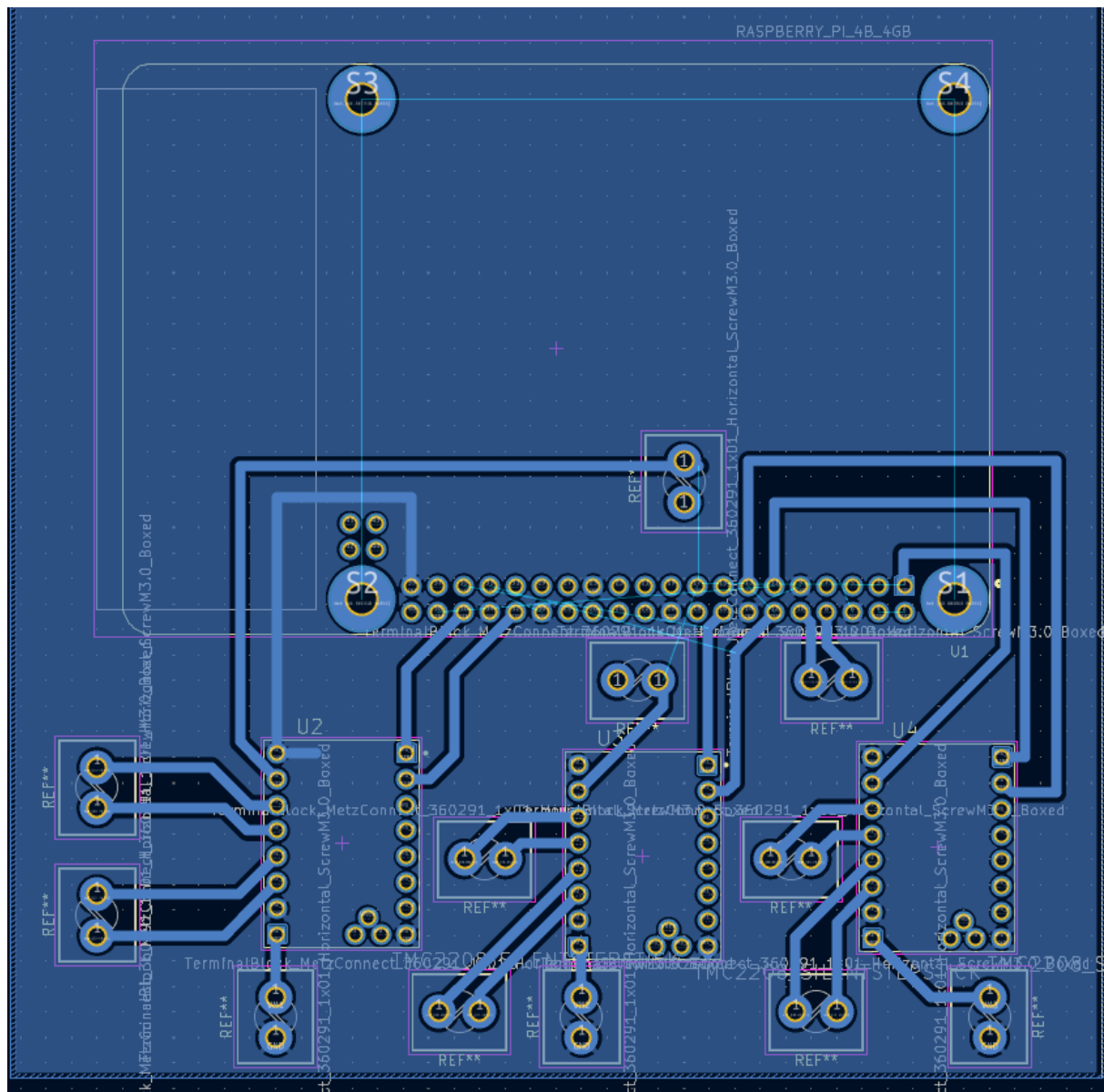
El escáner queda listo para el próximo uso.

## 2.c.iii. Esquemático y PCB



Activar Windows  
Ve a Configuración pa





## 2.c.iv. Alimentacion

El sistema UPFEET requiere dos fuentes de alimentación diferenciadas para garantizar un funcionamiento estable y seguro de todos sus componentes electrónicos:

- Una fuente de 12 V destinada a los drivers(TMC2208).
- Una fuente de 5 V dedicada a la Raspberry Pi

### 2.c.ii.1. Alimentación de los drivers

La alimentación de 12 V se obtiene mediante un banco de pilas AA conectadas en paralelo, lo que permite aumentar la corriente total disponible manteniendo el mismo nivel de tensión nominal.

Cada pila AA entrega aproximadamente 1,5 V, por lo que se emplea un arreglo previo en serie de 8 unidades ( $1,5 \text{ V} \times 8 = 12 \text{ V}$ ) para alcanzar el voltaje requerido, y luego se replica este conjunto en paralelo para multiplicar la capacidad de corriente (mAh) y así alimentar simultáneamente los tres motores NEMA 17 controlados por los drivers TMC2208.

Esta configuración fue elegida por los siguientes motivos:

- **Autonomía y portabilidad:** permite operar el sistema sin depender de una fuente externa o toma de corriente, facilitando pruebas y calibraciones fuera del entorno de laboratorio.
- **Seguridad eléctrica:** las pilas ofrecen una corriente limitada, reduciendo el riesgo de sobrecorriente o cortocircuitos que pudieran dañar los drivers.
- **Simplicidad de implementación:** evita la necesidad de fuentes conmutadas o reguladores adicionales, simplificando el cableado interno.
- **Compatibilidad:** los TMC2208 admiten un rango de operación entre 5,5 V y 36 V, por lo que 12 V representa un punto de equilibrio ideal entre torque, eficiencia y bajo nivel de ruido.

Sin embargo, se consideraron también las limitaciones inherentes a este tipo de alimentación:

- La capacidad total de corriente depende del número de conjuntos en paralelo, por lo que una autonomía prolongada requiere más grupos de pilas.
- Las pilas alcalinas pueden presentar caídas de tensión en uso prolongado, afectando el rendimiento de los motores si no se reemplazan periódicamente.
- A pesar de ello, la solución resultó económica, segura y fácilmente reemplazable, adecuada para el prototipo funcional del sistema.

## 2.c.ii.2. Alimentación de la Raspberry Pi

La Raspberry Pi se alimenta mediante su cargador oficial de 5 V / 3 A, conectado directamente al puerto USB-C del dispositivo.

Esta fuente fue seleccionada por las siguientes razones:

- **Estabilidad eléctrica garantizada:** el cargador oficial incluye regulación precisa de tensión, evitando caídas de voltaje que podrían causar reinicios o errores de lectura en los periféricos USB o GPIO.
- **Protección integrada:** cuenta con sistemas de protección contra sobrecorriente y cortocircuitos.

- **Compatibilidad directa:** elimina la necesidad de adaptadores o convertidores adicionales, asegurando una integración limpia con la electrónica general del sistema.

La separación física y eléctrica entre la alimentación de la Raspberry Pi y la de los motores evita ruidos eléctricos y picos de corriente que podrían interferir en las comunicaciones con el sensor LIDAR o en el control de los drivers.

### 2.c.ii.3. Justificación del diseño de alimentación

El diseño mixto —baterías para la sección de potencia y cargador dedicado para la sección lógica— fue adoptado para equilibrar autonomía, seguridad y modularidad, cumpliendo con los siguientes criterios:

1. **Independencia de circuitos:** el sistema de potencia y el sistema lógico se encuentran eléctricamente aislados, evitando interferencias cruzadas.
2. **Facilidad de mantenimiento:** ambos sistemas pueden desconectarse o reemplazarse de forma independiente sin afectar la integridad del circuito principal.
3. **Costo y disponibilidad:** se aprovecharon recursos accesibles (pilas y cargador original), reduciendo costos de fabricación y logística.
4. **Escalabilidad:** la fuente de 12 V puede ser reemplazada en el futuro por una batería recargable Li-ion o una fuente regulada, sin alterar la topología general del circuito.

## 2.d. Obstáculos encontrados y Soluciones aplicadas

### 2.d.i. Obstáculos Estructurales

Durante el desarrollo del sistema UPFEET se presentaron diversos desafíos vinculados a la fabricación y ensamblaje de la estructura mecánica, principalmente asociados a limitaciones de recursos, disponibilidad de materiales y adaptación de componentes.

En primer lugar, si bien el diseño estructural se encontraba definido y modelado correctamente, el costo y nivel de conocimiento técnico necesarios para su construcción superaban nuestras posibilidades de fabricación directa.

El trabajo requería equipos de soldadura, mecanizado y corte de precisión, además de materiales metálicos específicos.

Frente a esta dificultad, se logró establecer un vínculo con la empresa *Metalurgia Cronos*, que patrocinó el proyecto y se encargó de fabricar la estructura principal y de apoyo según nuestras especificaciones técnicas.

Este apoyo externo fue fundamental para avanzar con el prototipo manteniendo los estándares de calidad y precisión requeridos.

En segundo lugar, se detectó la imposibilidad de conseguir tuercas antibacklash en el mercado local del Gran Buenos Aires, y las opciones disponibles en línea presentaban costos de envío excesivos.

Ante esta situación, se decidió fabricar las tuercas antibacklash manualmente, utilizando materiales disponibles en el laboratorio y aplicando un sistema de resorte de compresión ajustado.

Si bien este procedimiento demandó pruebas iterativas, permitió obtener un resultado funcional y estable, resolviendo el problema sin incrementar el presupuesto.

Finalmente, durante las pruebas de ensamblaje del sistema de ejes, se detectó que los acoples flexibles utilizados para conectar los motores a las varillas roscadas provocaban un desplazamiento mínimo en el recorrido total del eje X.

Como consecuencia, el sensor LIDAR no alcanzaba a escanear completamente un extremo de la superficie transparente, dejando una zona sin digitalizar.

Para corregir este inconveniente, se diseñó y montó un soporte auxiliar en forma de “L”, impreso en 3D, que permitió retrasar el sensor unos centímetros y recuperar el área completa de escaneo.

Esta solución, aunque sencilla, mejoró significativamente la cobertura del sistema y demostró la capacidad de adaptación del equipo ante problemas de montaje real.

## **2.d.ii. Obstáculos electrónicos**

Durante el desarrollo del sistema electrónico del escáner UPFEET se presentaron diversos inconvenientes relacionados con la conectividad de los componentes, la estabilidad del sistema de control y las limitaciones de hardware disponibles.

El primer desafío se presentó con el sensor LIDAR, cuyo cableado original resultó insuficiente para permitir el desplazamiento completo a lo largo del eje de escaneo.

Dado que el recorrido del sensor requería una extensión mayor, se procedió a aumentar el alcance de los cables mediante extensiones Dupont, asegurando la continuidad eléctrica mediante cinta aislante de protección para evitar falsos contactos o desconexiones durante el movimiento.

Esta solución, aunque sencilla, resultó efectiva y permitió mantener la integridad de las señales de datos y alimentación sin interferir en la precisión del escaneo.

El segundo inconveniente surgió durante las pruebas finales, cuando la Raspberry Pi principal presentó fallas intermitentes de arranque (boot), lo que afectaba la estabilidad general del sistema.

Ante la falta de tiempo y recursos para una reparación completa, se decidió sustituirla temporalmente por una Raspberry Pi Pico, dispositivo de menor capacidad de procesamiento pero mayor confiabilidad operativa.

Esta modificación implicó delegar parte del procesamiento a una computadora externa, la cual asumió las tareas más demandantes del sistema, como la gestión del escaneo y la reconstrucción 3D.

Si bien esta solución redujo la autonomía del equipo, permitió completar las pruebas funcionales y validar el diseño general del sistema electrónico, priorizando la estabilidad y continuidad del proyecto frente a las restricciones de tiempo y presupuesto.

## 2.e. Conclusiones

### 2.e.i. Resultados técnicos alcanzados

El desarrollo del sistema UPFEET permitió integrar con éxito los principios de mecánica de precisión, electrónica aplicada y digitalización tridimensional, logrando un dispositivo funcional capaz de capturar y procesar la morfología plantar de manera precisa, portable y económica.

Desde el punto de vista estructural, se alcanzó una rigidez y estabilidad óptima, gracias al empleo de acero en la estructura principal y de apoyo, lo que asegura la correcta alineación del sistema óptico y la durabilidad del conjunto.

En el plano electrónico, la incorporación de drivers TMC2208 y motores NEMA 17 proporcionó un movimiento suave, silencioso y de alta resolución, adecuado para la calidad de escaneo requerida. La alimentación separada entre la sección de potencia (12 V) y la lógica de control (5 V) demostró ser una solución efectiva para evitar interferencias y garantizar la estabilidad del sistema.

Finalmente, el sistema alcanzó una coherencia integral entre diseño estructural, electrónica y software, consolidando una plataforma base lista para futuras mejoras de precisión y automatización.

### 2.e.ii. Ideas a implementar en un futuro

El proyecto UPFEET presenta un gran potencial de mejora, tanto en su precisión técnica como en la experiencia del usuario. Se proponen las siguientes líneas de evolución para futuras versiones:

1. **Implementación de un código acústico de diagnóstico (beep):**

Incorporar un pequeño buzzer o altavoz controlado por la Raspberry Pi para emitir códigos sonoros diferenciados que indiquen errores de conexión, sobrecorriente o falta de calibración.

Esto simplificará el proceso de diagnóstico y permitiría detectar fallas sin necesidad de interfaz visual, ideal para mantenimiento en campo.

2. **Módulo adicional de escaneo corporal (postura y rodillas):**

Integrar un segundo sistema de sensores que analice la alineación postural y el ángulo de las rodillas, mejorando la personalización del modelo 3D de las plantillas.

Este complemento permitiría correlacionar la forma plantar con la distribución biomecánica del peso, aumentando la precisión de las recomendaciones

ortopédicas.

3. **Optimización de la alimentación de los drivers:**

Sustituir el actual banco de pilas por una batería recargable de litio con circuito de protección y regulación de 12 V, o bien una fuente conmutada DC-DC de salida estabilizada.

Esta mejora reduciría el mantenimiento, estabilizaría la tensión de los motores y extendería la autonomía del equipo.

4. **Integración de control automatizado de corriente (Smart Vref):**

Incorporar una regulación dinámica de corriente según la carga de los motores, mejorando la eficiencia energética y reduciendo el calentamiento del sistema.

5. **Interfaz de usuario mejorada:**

Desarrollar un panel de control gráfico o aplicación complementaria que permita visualizar en tiempo real los valores de escaneo, estado del sistema y errores detectados por el código acústico.

## 3. Impresora

### 3.a. Estructura mecanica

#### 3.a.i. Estructura metálica

La estructura de la impresora 3D MakerPi P3 Pro se compone de un bastidor rígido de perfiles de aluminio extruido tipo V-Slot, en secciones 2020 y 2040. La configuración es ortogonal, formando un marco autoportante que sostiene los subsistemas de movimiento y extrusión.

Las uniones entre perfiles se efectúan mediante escuadras mecanizadas de aluminio y tornillería M5 con tuercas T-slot, asegurando acoplamientos rígidos y distribuyendo uniformemente los esfuerzos mecánicos.

La base principal constituye un marco rectangular que soporta los perfiles verticales, los cuales se fijan mediante escuadras internas y tornillería pasante, garantizando paralelismo y estabilidad torsional.

Los perfiles superiores cierran la estructura en pórtico, reduciendo la deflexión en los planos XY y Z bajo cargas dinámicas durante el funcionamiento. Todos los elementos metálicos se encuentran anodizados, incrementando la resistencia a la abrasión y a la corrosión.

### **3.a.ii. Justificación de las dimensiones y disposición estructural**

Las dimensiones finales de la impresora modificada son 560 mm (ancho) × 540 mm (profundidad) × 400 mm (altura total), ajustadas para la producción de plantillas tridimensionales y la estabilidad estructural del bastidor.

- Los 560 mm de ancho permiten ubicar los perfiles horizontales del pórtico y soportar de manera estable el desplazamiento del extrusor en el eje X, reduciendo deflexión y garantizando la perpendicularidad con el eje Z.
- La profundidad de 540 mm asegura soporte robusto para el eje Y y la cama caliente, minimizando torsiones bajo las cargas dinámicas del movimiento de la plataforma y la deposición de filamento.
- La altura total de 400 mm fue reducida respecto al diseño original debido a que la aplicación principal —impresión de plantillas— no requiere un recorrido vertical amplio. Esta modificación elimina espacio innecesario sobre el área de impresión, disminuye el momento de inercia de la estructura y mantiene la rigidez torsional necesaria para movimientos dinámicos. La sección vertical de los perfiles 2040 sigue siendo suficiente para soportar el recorrido del eje Z y la instalación del portabobinas sobre el bastidor sin comprometer la funcionalidad ni la estabilidad de la impresora.

La disposición ortogonal cerrada de los perfiles y la selección de estas dimensiones optimizan la rigidez torsional, la alineación de ejes y la precisión dimensional, asegurando la correcta deposición de filamento flexible (TPU) dentro de las tolerancias requeridas para plantillas ortopédicas.

### **3.a.iii. Materiales utilizados y fundamentos de selección**

La selección de materiales se realizó bajo criterios de resistencia mecánica, rigidez, estabilidad térmica y durabilidad:

- Aluminio extruido Al6063-T5: utilizado en perfiles principales por su módulo de elasticidad que permite mantener la geometría bajo cargas dinámicas, su baja densidad, y su resistencia a la corrosión. Su anodizado incrementa la dureza superficial, previene desgaste en ranuras de acoplamiento y contribuye a la disipación térmica.
- Tornillería M5 de acero al carbono: provee resistencia a la tracción y compresión en puntos de unión críticos, evitando deslizamientos o deformaciones de las conexiones bajo cargas alternantes de los ejes y la cama.
- Escuadras mecanizadas de aluminio: aseguran la alineación precisa de los perfiles y distribuyen uniformemente los esfuerzos en las esquinas, manteniendo estabilidad torsional y limitando deformaciones por fatiga.

La combinación de estos materiales asegura que la estructura metálica cumpla con los requisitos de rigidez torsional, estabilidad dimensional y repetibilidad necesarios para la impresión de filamentos flexibles como TPU, minimizando vibraciones, desviaciones y errores dimensionales.

## **3.b. Sistema de ejes**

### **3.b.i. Configuración general de ejes (X, Y, Z)**

La impresora adopta una arquitectura cartesiana, con tres ejes lineales ortogonales:

- Eje X: Movimiento lateral del extrusor a lo largo del pórtico superior. Controlado mediante un motor paso a paso acoplado a correa GT2, que transmite el desplazamiento al carro del extrusor.
  - Eje Y: Movimiento longitudinal de la cama caliente, desplazándose sobre guías lineales. La cama se mueve sobre rodamientos LM8UU instalados en perfiles 2020, con transmisión mediante correa GT2 accionada por motor paso a paso.
  - Eje Z: Movimiento vertical del extrusor y, en algunos casos, de la cama (dependiendo de la configuración). Utiliza husillos trapezoidales T8 acoplados a tuercas de bronce auto lubricadas, con soporte superior e inferior para minimizar excentricidad.
- La configuración cartesiana permite controlar de manera independiente cada eje, asegurando la coherencia dimensional de la pieza impresa y la repetibilidad de posicionamiento requerida para filamentos flexibles como TPU.

### **3.b.ii. Componentes mecánicos de transmisión**

El sistema de transmisión combina correas, poleas y husillos, seleccionados para optimizar precisión, rigidez y suavidad de movimiento:

- Correas GT2: Empleadas en los ejes X e Y, con paso de 2 mm, tensión controlada mediante sensores ajustables, garantizando desplazamientos precisos y mínima holgura.
- Poleas dentadas: Montadas sobre los ejes de los motores paso a paso, acopladas a las correas, permiten transmitir torque sin deslizamientos.
- Husillos trapezoidales T8: Utilizados en el eje Z, con paso de 2 mm por revolución y tuercas de bronce auto lubricadas, garantizan desplazamientos verticales precisos y soportan la carga de la estructura superior y el extrusor.
- Guías lineales y rodamientos LM8UU: Instalada en los ejes X e Y para mantener alineación y rectitud, reduciendo fricción lateral y minimizando desviaciones en la trayectoria del extrusor y de la cama caliente.
- Acoplamientos flexibles: Entre motores y husillos del eje Z, absorben desalineaciones mínimas y evitan transmisión de vibraciones a la estructura.

### **3.b.iii. Medidas, relaciones de movimiento y guías lineales**

El sistema de movimiento está compuesto por tres ejes cartesianos:

- Eje X: recorrido mecánico de 400 mm, con límite de desplazamiento configurado en firmware en 200 mm.
- Eje Y: recorrido mecánico de 300 mm, con límite de desplazamiento configurado en firmware en 155 mm.
- Eje Z: recorrido mecánico de 260 mm, con límite de desplazamiento configurado en firmware en 260 mm.

Cada eje utiliza guías lineales y varillas roscadas acopladas a motores paso a paso, garantizando el desplazamiento controlado en los tres ejes. Las relaciones de movimiento fueron definidas según el paso de rosca de los husillos y la cantidad de pasos por milímetro establecidos en el firmware base.

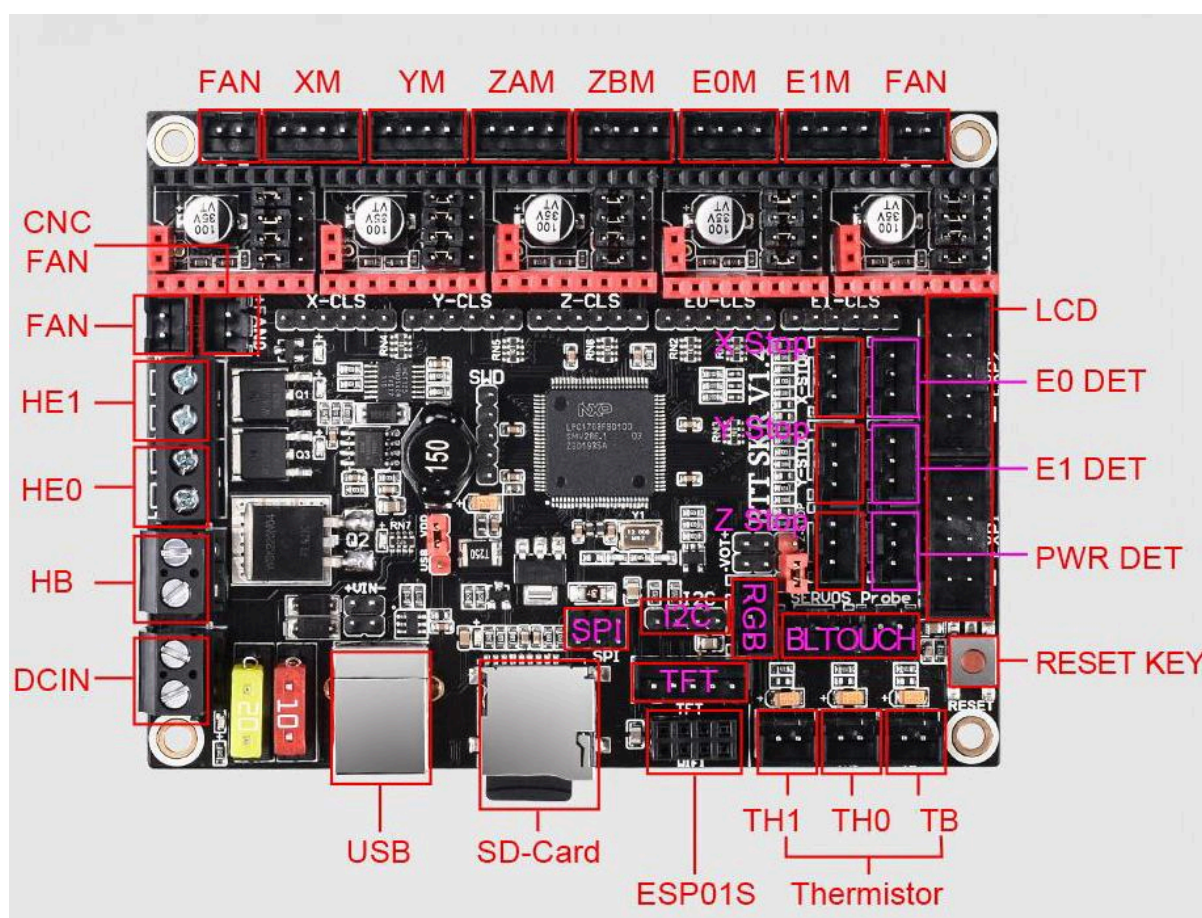


### 3.c. Electronica de control

### 3.c.i. Placa madre SKR V1.4

El sistema de control de la impresora está basado en una placa SKR V1.4, la cual incorpora un microcontrolador LPC1768 de 32 bits. Esta arquitectura permite una mayor capacidad de procesamiento y una gestión eficiente de los movimientos simultáneos de los ejes y extrusores. La placa admite controladores de motor TMC y dispone de una interfaz de comunicación estándar compatible con el firmware Marlin.

### 3.c.ii. Distribución de pines (pinout general y específico)



### 3.c.iii. Funcionamiento del sistema de control

El microcontrolador interpreta las instrucciones G-code provenientes del software de laminado y las traduce en señales eléctricas que activan los actuadores correspondientes. El firmware Marlin coordina la cinemática de los ejes, la regulación térmica de los extrusores y la modulación de velocidad de los ventiladores.

Además, gestiona la lectura de finales de carrera, la protección ante fallos térmicos y la sincronización entre extrusores independientes, garantizando la precisión en la ejecución de cada ciclo de impresión.

### 3.c.iv Implementación en la impresora desarrollada

La placa SKR V1.4 se configuró como unidad principal de control, utilizando un firmware derivado del Biqu V1, adaptado a la topología y requerimientos del sistema.

Se deshabilitó el canal de extrusión E0, manteniendo activo únicamente E1 como extrusor operativo.

Se ajustaron los parámetros de firmware correspondientes a los límites de desplazamiento y altura del eje Z, con un recorrido máximo de 400 mm.

Los controladores TMC2208 operan en modo UART, con corriente de referencia configurada en 0.9 A RMS y control de microstepping dinámico.

Los finales de carrera se configuraron en modo NC, con activación en posición MIN para los ejes X, Y y Z.

El subsistema térmico incluye un hotend activo asociado al canal E1, equipado con sensor NTC 100k (EPCOS 3950), y una cama calefactora controlada mediante MOSFET externo.

El cableado se implementó con conductores AWG 22 para señales lógicas y AWG 16 para potencia, con rutas separadas para minimizar interferencias electromagnéticas.

El firmware mantiene control PID independiente para el hotend y la cama calefactora, protección térmica redundante y control de ventiladores mediante salidas digitales dedicadas.

La interfaz de comunicación opera por USB (CDC virtual) y tarjeta SD, priorizando el modo SD para operación autónoma.

### 3.c.v. Conexiones

La SKR 1.4 recibe la alimentación desde la fuente de poder mediante un conector de tornillo de 2 pines en la placa;

**DCIN+** = polo positivo (12 V ).

**DCIN-** = tierra / negativo (GND).

La placa tiene salidas para hasta 5 motores, NEMA 17, repartidos en los siguientes conectores: **X, Y, Z, E0, E1**;

- Conectá en X → motor del eje X.
- Conectá en Y → motor del eje Y.
- Conectá en Z → motor del eje Z.
- Conectá en E0 → motor del extrusor 1.
- Hay un extrusor extra, E1 → extrusor 2.

Los motores están controlados por drivers TMC2208, que se instalan en los zócalos. Estos drivers reciben las señales STEP, DIR, EN, junto con la alimentación VMOT, VIO y GND, y

con esa información regulan cómo se energizan las dos bobinas internas del motor paso a paso, permitiendo que el motor gire con precisión y silenciosa.

Hay fines de carreras (endstops) que se conectan a los puertos X-STOP, Y-STOP y Z-STOP: Cada puerto posee tres pines (S, GND, +5V), pero únicamente se utilizan S y GND. El terminal NO (Normally Open) del switch se conecta al pin S (señal), mientras que el terminal COM se conecta al pin GND. El pin +5V no se utiliza.

Sensores de temperatura y calentadores

- **TH0** → entrada para termistor del “hotend”.
- **TB** → entrada para termistor de la cama caliente (heatbed).

Para calefacción (calentar la cama):

- **HE0** → salida de potencia para el hotend (resistencia calefactora).
- **HB** → salida para la cama caliente.

Ventiladores (fans): FAN0 y FAN1:

- **FAN0** → Este ventilador enfría el filamento de la primera capa.
- **FAN1** → Este ventilador enfría el disipador del hotend, para que a partir de la segunda capa, se enfrie mas rapido, ya que va mas rapida.

Reset Key:

El Reset Key es un botón físico que reinicia completamente la placa.  
Su función es:

- Recargar la configuración actual de la memoria del controlador.

Puerto para tarjeta microSD, que se utiliza para:

- Guardar y leer el archivo firmware.bin para actualizar Marlin.
- Ejecutar archivos G-code impresos directamente sin computadora.
- Almacenar configuraciones o pruebas de diagnóstico.

Por último, se utilizó un cable DB25 para llevar todas las conexiones del extrusor hacia la fuente. Para el cartucho calefactor se emplearon los pines 24 y 12 como línea positiva (+V) y los pines 12 y 9 como línea negativa (GND), utilizando pines duplicados para soportar la corriente y evitar sobrecalentamiento.

## 3.d. Limitaciones Técnicas y soluciones aplicadas

### 3.d.i. Problemas estructurales

Se identificaron las siguientes limitaciones mecánicas en la impresora:

- Reducción de altura del chasis: La altura original de la impresora se redujo 200 mm, quedando en aproximadamente 400 mm, para optimizar el volumen de trabajo destinado a

la fabricación de plantillas. Esta modificación requirió recalibración de los límites del eje Z y ajuste de recorridos máximos en firmware.

- Soporte de placa madre: La ubicación original de la placa SKR V1.4 no era compatible con la nueva disposición del chasis. Se implementó un soporte impreso en 3D, diseñado para ser atornillado a la base, asegurando alineación precisa, sujeción mecánica estable y disipación térmica adecuada.
- Posición de finales de carrera: Se incorporó una pieza impresa en 3D que permite que los finales de carrera contacten antes, estableciendo correctamente la referencia de posición inicial del eje Z sin necesidad de desplazar los sensores.
- Eliminación del segundo extrusor: El canal de extrusión E0 se retiró debido a que no era requerido para la función de la impresora y por la limitación de pines disponibles en la placa SKR V1.4, dejando únicamente activo el extrusor E1. Esta modificación simplificó la estructura del cabezal, redujo masa y eliminó posibles interferencias mecánicas.
- Pin de eje Z doblado y falso contacto: El eje del sensor de fin de carrera del eje Z sufrió deformación. Se enderezó el pin y se aseguraron conexiones firmes para eliminar falsos contactos.

### **3.d.ii. Problemas electrónicos**

Se detectaron las siguientes limitaciones y fallas en el sistema electrónico de la impresora:

- Selección de placa madre: Se evaluó la sustitución de la placa original de la MakerPi P3 Pro para compatibilidad con el nuevo esquema de control, seleccionando finalmente la SKR V1.4 por capacidad de integración y soporte de drivers TMC2208.
- Paralelización de finales de carrera del eje Z: Debido a la limitación de pines disponibles en la placa, los finales de carrera del eje Z se conectaron en paralelo, permitiendo mantener la funcionalidad de detección de límites sin saturar entradas de la placa.
- Reemplazo del cable DB25: El cable original presentaba pines cruzados, generando errores intermitentes de señal. Se diseñó y ensambló un nuevo cable internamente, asegurando correspondencia correcta de todas las señales de control y alimentación.
- Conector del eje Z2: El conector original de 8 pines se sustituyó por el cable sobrante del eje X, garantizando conexión adecuada con los drivers y el sensor de fin de carrera.
- Problema de termistor: El sensor de temperatura del hotend no fue conectado mediante el cable DB25 fabricado, sino que se utilizó un cable separado estándar, asegurando lectura de temperatura estable y evitando interferencias en la señal.

### **3.d.iii. Problemas de software**

Se identificaron las siguientes limitaciones y ajustes necesarios en la capa de firmware y control de la impresora:

- Reemplazo de firmware: Se sustituyó el firmware original por uno basado en Biqu V1, adaptando parámetros de control, drivers TMC2208 y configuración de extrusor único E1.
- Ajuste de volumen de impresión: Se modificó el firmware para reflejar la reducción de altura y soporte de chasis, ajustando los límites de desplazamiento en eje Z y estableciendo un volumen de trabajo compatible con la fabricación de plantillas.

- Fin de carrera invertido en eje Y: El sensor de fin de carrera del eje Y requería inversión de lógica, configurándose correctamente en firmware para garantizar detección precisa de posición inicial y límites de recorrido.
- Puertos de termistor: Ambos puertos disponibles en la placa para termistor fueron asignados al único extrusor activo (E1), garantizando redundancia de lectura y seguridad térmica.
- Escalado de medidas de impresión: Se corrigieron en firmware las dimensiones del volumen de impresión, ya que inicialmente las medidas mostradas eran aproximadamente la mitad del tamaño real, asegurando correspondencia correcta entre software de slicing y desplazamiento mecánico.

### 3.e. Firmware

#### 3.e.i. Firmware base utilizado (Biqu / Marlin modificado)

Se empleó como base Biqu V1, derivado de Marlin, para su compatibilidad con la placa SKR V1.4 y control de drivers TMC2208.

#### 3.e.ii. Configuración y personalización para la SKR V1.4

Se realizaron ajustes de asignación de extrusor único, configuración de drivers TMC2208 en modo UART, y redefinición del volumen de impresión según el chasis modificado.

#### 3.e.iii. Cambios implementados en la versión final

- Inversión de lógica del fin de carrera del eje Y.
- Consolidación de puertos de termistor para el extrusor activo.
- Corrección de escala de impresión para coincidencia con las dimensiones reales.

#### 3.e.iv Resultados obtenidos tras la modificación

El firmware final permite control de extrusor y ejes, lectura de sensores y fin de carrera, y ajuste del volumen de impresión conforme a las dimensiones finales de la impresora.

### 3.f. Conclusiones

#### 3.f.ii. Resultados técnicos alcanzados

Se completó la integración del extrusor único controlado por la placa SKR V1.4, la configuración de drivers TMC2208, la conexión y lectura de sensores y finales de carrera, y la adaptación del firmware basado en Biqu V1 para el volumen de impresión final.

#### 3.f.iii. Posibles mejoras futuras

- Incorporación de monitoreo avanzado para control de temperatura y desplazamiento.

- Optimización de rigidez estructural para minimizar vibraciones.
- Desarrollo de firmware adaptado a multi extrusión en futuras versiones si se requiere expansión funcional.