

# INSTITUTO SUPERIOR TECNOLÓGICO SUDAMERICANO



## TECNOLOGÍA SUPERIOR EN ELECTRÓNICA

“DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE  
FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA  
BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO  
ABRIL - SEPTIEMBRE 2023”

INFORME DEL PROYECTO DE INVESTIGACIÓN PREVIO A LA OBTENCIÓN  
DEL TÍTULO DE TECNÓLOGO EN LA TECNOLOGÍA SUPERIOR EN  
ELECTRÓNICA

### AUTORES:

Ramón González Marco Rolando

Vivanco Celi Roosevelt Iván

### DIRECTOR:

Ing. David Paul Rosales Herrera

Loja, 07 de noviembre del 2023

## **Certificación del Director del Proyecto de Investigación de Fin de Carrera**

**Ing.**

David Paúl Rosales Herrera

### **DIRECTOR DE INVESTIGACIÓN**

#### **CERTIFICA:**

Que ha supervisado el presente proyecto de investigación titulado “DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023” el mismo que cumple con lo establecido por el Instituto Superior Tecnológico Sudamericano; por consiguiente, autorizo su presentación ante el tribunal respectivo.

Loja, 07 de noviembre del 2023

.....

**Firma**

**Ing. David Paúl Rosales Herrera**

**Autoría**

Yo MARCO ROLANDO RAMÓN GONZÁLEZ con C.I. N° 1104105257 declaro ser el autor del presente trabajo de tesis titulado **DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023**, es original e inédito, dejando establecido que aquellos aportes intelectuales de otros autores se han referenciado debidamente en el proyecto de investigación.

Loja, 07 de noviembre del 2023

.....

**Firma**  
**C.I. 1104105257**

**Autoría**

Yo ROOSEVELT IVÁN VIVANCO CELI con C.I. N° 1717619439 declaro ser el autor del presente trabajo de tesis titulado **DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023**, es original e inédito, dejando establecido que aquellos aportes intelectuales de otros autores se han referenciado debidamente en el proyecto de investigación.

Loja, 07 de noviembre del 2023

.....

**Firma**  
**C.I. 1717619439**

## **Dedicatoria**

En primer lugar, quiero agradecer a mis padres que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Pero de manera especial este trabajo va dedicado a mi padre, que cuando inicié la carrera él fue quien me incentivó para retomar los estudios, pero lamentablemente falleció en el año 2022 y quiero dedicar unas palabras en su memoria. No hay palabras suficientes para expresar el dolor y la tristeza que siento desde que te fuiste de este mundo, sin embargo, quiero que sepas que, aunque ya no estes físicamente presente, siempre estarás en mi corazón. Tu memoria vive en cada rincón de mi ser y cada latido de mi corazón. Quiero agradecerte por a ver sido mi padre. Tú me enseñaste el verdadero significado del amor incondicional. Prometo honrar tu memoria y vivir mi vida de la mejor manera posible, llevando conmigo los valores y el amor que me enseñaste. Cada paso que dé, cada logro que alcance, será un tributo a tu espíritu valiente mi guerrero. Eres y serás por siempre un orgullo para mí porque fuiste el mejor papá del mundo, eres mi motor y mi inspiración.

**Marco Rolando Ramón González**

## **Dedicatoria**

Al creador de todas las cosas, el que me ha dado su mano y me ha llenado de fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad que de mi corazón puede emanar, dedico primeramente mi trabajo a Dios.

A mis padres que han sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles de mi vida, ellos han sido el pilar fundamental en este largo camino, que en todo momento me apoyaron con sus buenos consejos y por eso este logro es de ellos porque sin su ayuda no hubiese sido posible.

A mis hermanos que siempre han sido mi ejemplo a seguir, gracias por su apoyo material y moral.

A mi mujer por su apoyo en todo momento y buenos consejos.

**Roosevelt Iván Vivanco Celi**

## **Agradecimiento**

Agradezco a Dios por haberme brindado la sabiduría para culminar mis estudios. A mi familia, esposa e hijas por todo el apoyo que me supieron brindar durante todos estos años de preparación académica. A mis docentes quienes sembraron en mí conocimientos nuevos para poder así sacar adelante este proyecto. También, mi agradecimiento a mi tutor por su dedicación y paciencia.

A mi compañero de proyecto, por todas las horas compartidas durante el desarrollo de este proyecto y como no también, al ITS Sudamericano por permitirme ser parte de esta gran institución educativa.

**Marco Rolando Ramón González**

## **Agradecimiento**

“En primer lugar le agradezco a Dios por regalarme la vida, la salud y la valentía para seguir adelante en todos mis propósitos profesionales y personales.

A mis padres que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. También son los que me han brindado el soporte moral para poder superarme cada día más, teniendo siempre en cuenta que los límites y los obstáculos no existen”.

Al Instituto Sudamericano que por medio de sus docentes quienes han sido parte de mi formación, a todos ellos les quiero agradecer por transmitirme los conocimientos necesarios para hoy poder estar aquí. Sin ustedes los conceptos serían solo palabras, y las palabras se las lleva, el viento”.

**Roosevelt Iván Vivanco Celi**

**Acta de cesión de derechos****ACTA DE CESIÓN DE DERECHOS DE PROYECTO DE INVESTIGACIÓN DE FIN DE CARRERA**

Conste por el presente documento la Cesión de los Derechos de proyecto de investigación de fin de carrera, de conformidad con las siguientes cláusulas:

**PRIMERA.** - Por sus propios derechos; el Ing. David Paúl Rosales Herrera, en calidad de Director del proyecto de investigación de fin de carrera; Marco Rolando Ramón González, y Roosevelt Iván Vivanco Celi en calidad de autores del proyecto de investigación de fin de carrera; mayores de edad emiten la presente acta de cesión de derechos.

**SEGUNDA.** - Marco Rolando Ramón González, y Roosevelt Iván Vivanco Celi realizaron la Investigación titulada “Diseño e implementación de una máquina extrusora de filamento de uso general usando como materia prima botellas de plástico PET en la ciudad de Loja para el periodo abril - septiembre 2023”; para optar por el título de Tecnólogo en Electrónica, en el Instituto Superior Tecnológico Sudamericano de Loja, bajo la dirección de la Ing. David Paúl Rosales Herrera.

**TERCERA.** - Es política del Instituto que los proyectos de investigación de fin de carrera se apliquen y materialicen en beneficio de la comunidad.

**CUARTA.** - Los comparecientes Ing. David Paúl Rosales Herrera, en calidad de Director del proyecto de investigación de fin de carrera, Marco Rolando Ramón Gonzáles, y Roosevelt Iván Vivanco Celi como autores, por medio del presente instrumento, tienen a bien ceder en forma gratuita sus derechos de proyecto de investigación de fin de carrera titulado “Diseño e implementación de una máquina extrusora de filamento de uso general usando como materia prima botellas de plástico PET en la ciudad de Loja para el periodo abril - septiembre 2023” a favor del Instituto Superior Tecnológico Sudamericano de Loja; y, conceden autorización para que el Instituto pueda utilizar esta investigación en su beneficio y/o de la comunidad, sin reserva alguna.

**QUINTA.** - Aceptación. - Las partes declaran que aceptan expresamente todo lo estipulado en la presente cesión de derechos.

Para constancia suscriben la presente cesión de derechos, en la ciudad de Loja, en el mes de noviembre del año 2023.

F. \_\_\_\_\_  
Ing. David Paúl Rosales Herrera  
C.I. 1104222557

F. \_\_\_\_\_  
Marco Rolando Ramón González  
C.I. 1104105257

F. \_\_\_\_\_  
Roosevelt Iván Vivanco Celi  
C.I. 1717619439

## Declaración juramentada

Loja, 07 de noviembre del 2023

**Nombres:** Marco Rolando

**Apellidos:** Ramón González

**Cédula de Identidad:** 1104105257

**Carrera:** Electrónica

**Semestre de ejecución del proceso de titulación:** Abril – Octubre 2023

**Tema de proyecto de investigación de fin de carrera con fines de titulación:**

“DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023”

En calidad de estudiante del Instituto Superior Tecnológico Sudamericano de la ciudad de Loja;

Declaro bajo juramento que:

1. Soy autor del trabajo intelectual y de investigación del proyecto de fin de carrera.
2. El trabajo de investigación de fin de carrera no ha sido plagiado ni total ni parcialmente, para la cual se han respetado las normas internacionales de citas y referencias para las fuentes consultadas.
3. El trabajo de investigación de fin de carrera presentado no atenta contra derechos de terceros.
4. El trabajo de investigación de fin de carrera no ha sido publicado ni presentado anteriormente para obtener algún grado académico previo o título profesional.
5. Los datos presentados en los resultados son reales, no han sido falsificados, ni duplicados, ni copiados. Las imágenes, tablas, gráficas, fotografías y demás son de mi autoría; y en el caso contrario aparecen con las correspondientes citas o fuentes.

Por lo expuesto; mediante la presente asumo frente al INSTITUTO cualquier responsabilidad que pudiera derivarse por la autoría, originalidad y veracidad del contenido del trabajo de investigación de fin de carrera.

En consecuencia, me hago responsable frente al INSTITUTO y frente a terceros, de cualquier daño que pudiera ocasionar al INSTITUTO o a terceros, por el incumplimiento de lo declarado o que pudiera encontrar causa en el trabajo de investigación de fin de carrera presentado, asumiendo todas las cargas pecuniarias que pudieran derivarse de ello.

Asimismo, por la presente me comprometo a asumir además todas las cargas pecuniarias que pudieran derivarse para EL INSTITUTO en favor de terceros por motivo de acciones, reclamaciones o conflictos derivados del incumplimiento de lo declarado o las que encontraren causa en el contenido del trabajo de investigación de fin de carrera.

De identificarse fraude, piratería, plagio, falsificación o que el trabajo de investigación haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a la normatividad vigente dispuesta por la LOES y sus respectivos reglamentos y del Instituto Superior Tecnológico Sudamericano de la ciudad de Loja.

Firma .....  
Nro. Cédula 1104105257

## Declaración juramentada

Loja, 07 de noviembre del 2023

**Nombres:** Roosevelt Iván

**Apellidos:** Vivanco Celi

**Cédula de Identidad:** 1717619439

**Carrera:** Electrónica

**Semestre de ejecución del proceso de titulación:** Abril – Octubre 2023

**Tema de proyecto de investigación de fin de carrera con fines de titulación:**

“DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023”

En calidad de estudiante del Instituto Superior Tecnológico Sudamericano de la ciudad de Loja;

Declaro bajo juramento que:

1. Soy autor del trabajo intelectual y de investigación del proyecto de fin de carrera.
2. El trabajo de investigación de fin de carrera no ha sido plagiado ni total ni parcialmente, para la cual se han respetado las normas internacionales de citas y referencias para las fuentes consultadas.
3. El trabajo de investigación de fin de carrera presentado no atenta contra derechos de terceros.
4. El trabajo de investigación de fin de carrera no ha sido publicado ni presentado anteriormente para obtener algún grado académico previo o título profesional.
5. Los datos presentados en los resultados son reales, no han sido falsificados, ni duplicados, ni copiados. Las imágenes, tablas, gráficas, fotografías y demás son de mi autoría; y en el caso contrario aparecen con las correspondientes citas o fuentes.

Por lo expuesto; mediante la presente asumo frente al INSTITUTO cualquier responsabilidad que pudiera derivarse por la autoría, originalidad y veracidad del contenido del trabajo de investigación de fin de carrera.

En consecuencia, me hago responsable frente al INSTITUTO y frente a terceros, de cualquier daño que pudiera ocasionar al INSTITUTO o a terceros, por el incumplimiento de lo declarado o que pudiera encontrar causa en el trabajo de investigación de fin de carrera presentado, asumiendo todas las cargas pecuniarias que pudieran derivarse de ello.

Asimismo, por la presente me comprometo a asumir además todas las cargas pecuniarias que pudieran derivarse para EL INSTITUTO en favor de terceros por motivo de acciones, reclamaciones o conflictos derivados del incumplimiento de lo declarado o las que encontraren causa en el contenido del trabajo de investigación de fin de carrera.

De identificarse fraude, piratería, plagio, falsificación o que el trabajo de investigación haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, someténdome a la normatividad vigente dispuesta por la LOES y sus respectivos reglamentos y del Instituto Superior Tecnológico Sudamericano de la ciudad de Loja.

Firma .....  
Nro. Cédula 1717619439

## Índice de Contenidos

Certificación del Director del Proyecto de Investigación de Fin de Carrera .....	I
Autoría.....	II
Dedicatoria .....	IV
Agradecimiento .....	VI
Acta de cesión de derechos .....	VIII
Declaración juramentada.....	X
Índice de Figuras .....	17
Índice de Tablas.....	19
Resumen.....	20
Abstract .....	21
Problema .....	22
Tema.....	24
Justificación.....	25
Objetivos .....	28
Objetivo General .....	28
Objetivos Específicos .....	28
Marco Teórico.....	29
Marco Referencial .....	29
Marco Conceptual .....	32
Clasificación de los Plásticos .....	32
Tereftalato de polietileno.....	33
Propiedades Físicoquímicas del PET .....	34
Estabilidad térmica .....	34
Densidad.....	35
Resistencia al Ataque Químico y Solubilidad.....	35
Permeabilidad.....	35
Conductividad Térmica .....	36
Propiedades Ópticas .....	36
Métodos de Conformado PET.....	36
Aplicaciones del PET Reutilizo o rPET .....	39
Prendas de Vestir.....	39
Láminas de PET 100% Reciclado.....	40

Diseño Metodológico .....	42
Métodos de Investigación.....	42
Técnicas de investigación.....	44
Búsqueda Bibliográfica.....	44
Experimento.....	45
Propuesta Práctica de Acción.....	46
Hardware .....	46
Arduino Mega 2560.....	46
Shiel RepRap Arduino Mega Pololu RAMPs.....	47
Pantalla GLCD ST7920 de 128x64 Pixeles.....	48
Bloque Calefactor con Termistor para Impresora 3D Oopensource.....	49
Motor a pasos Nema 17 .....	51
Driver de control A4988.....	52
Fuente de Alimentación Conmutada 12V 33A.....	53
Software.....	54
El IDE de Arduino .....	56
Fusion360 para Diseño del Sistema de Enrollado .....	57
Desarrollo de la Propuesta.....	58
Diseño y Construcción del Prototipo.....	58
Configuración del software.....	59
Configuración del Código G.....	64
Etapa de Adecuación e Interconexión de Hardware .....	67
La pantalla y opciones de inicio. ....	67
Requerimientos de Potencia Eléctrica. ....	68
Esquema de Conexiones Eléctrico.....	69
Desarrollo e implementación de mecanismos .....	70
Funcionamiento general del prototipo .....	75
Arquitectura del sistema .....	75
Proceso general .....	77
Esquema de conexiones electrónico .....	78
Pruebas de Funcionamiento y Resultados .....	80
Prueba de Funcionamiento.....	80
Resultados .....	82
Conclusiones .....	89

Recomendaciones.....	90
Bibliografía .....	91
Anexos .....	97
Anexo I: Certificado de aprobación .....	97
Anexo III: Certificado de implementación.....	100
Anexo IV: Certificado de traducción .....	101
Anexo V: Presupuesto .....	102
Anexo VI: Cronograma .....	105
Anexo VII: Programación .....	106
Anexo VII: Evidencias fotográficas .....	150

## Índice de Figuras

<b>Figura 1.</b> Estructura química del PET.....	34
<b>Figura 2.</b> Moldes para soplado de botellas plásticas.....	37
<b>Figura 3.</b> Matriz de botellas PET (soplado previo).....	38
<b>Figura 4.</b> Esquema de Peletizadora y producto final (PELLET) .....	39
<b>Figura 5.</b> Prendas de vestir a partir de rPET .....	40
<b>Figura 6.</b> Aplicaciones de Lámina PET CERSA .....	41
<b>Figura 7.</b> Placa de desarrollo Arduino Mega2560. ....	47
<b>Figura 8.</b> Capacidad de puertos a interfaces de Shield RAMPs 1.4.....	48
<b>Figura 9.</b> Pantalla de inicio en GLCD 128x64.....	49
<b>Figura 10.</b> Bloque calefactor ensamblado con termistor y niquelina.....	50
<b>Figura 11.</b> Resistencia eléctrica de 24W y Termistor NTC de 100kΩ .....	50
<b>Figura 12.</b> Motor a pasos Nema 17 1.5A .....	51
<b>Figura 13.</b> Driver A4988. Posicionamiento en la placa Ramps 1.4 .....	52
<b>Figura 14.</b> Fuente de alimentación.....	53
<b>Figura 15.</b> Ícono del Firmware OpenSource Marlin. ....	55
<b>Figura 16.</b> Página oficial de Marlin ORG. ....	55
<b>Figura 17.</b> IDE Arduino y sketch base .....	56
<b>Figura 18.</b> Toolkit Autodesk para la creación de engranajes en Fusion360. ....	57
<b>Figura 19.</b> Vista isométrica en la interfaz de Fusion360.....	58
<b>Figura 20.</b> Selección y descarga de Firmware .....	60
<b>Figura 21.</b> IDE Arduino y sketch de Marlin 1.1.x con sus subprogramas .....	60
<b>Figura 22.</b> Selección de la placa Arduino Mega 2560 para compilación. ....	61
<b>Figura 23.</b> Instalación de la biblioteca U8glib en el entorno (se instalan las 3 dependencias).....	62
<b>Figura 24.</b> Configuración de la pantalla de inicio .....	63
<b>Figura 25.</b> Resumen de compilación en la placa de desarrollo Arduino Mega2560. ....	64
<b>Figura 26.</b> Modificación de temperatura en código G. ....	65
<b>Figura 27.</b> Modificación de velocidad en código G.....	66
<b>Figura 28.</b> Pantalla de inicio del sistema.....	67
<b>Figura 29.</b> Pantalla de operación en GLCD 128x64 .....	68
<b>Figura 30.</b> Driver A4988. Posicionamiento en la placa Ramps 1.4.....	69
<b>Figura 31.</b> Diagrama de conexiones del sistema Freeplastic. ....	70
<b>Figura 32.</b> Levantamiento 3D del prototipo.....	71
<b>Figura 33.</b> Implementación del sistema de enrollado. Especifica 47 h de impresión 3D. ....	72
<b>Figura 34.</b> Impresión 3D del sistema de engranajes. ....	73
<b>Figura 35.</b> Bloque calefactor adaptado para extruir filamento de 1.5mm.....	74
<b>Figura 36.</b> Ensamble del sistema de reducción por engranajes.....	74
<b>Figura 37.</b> Pantalla de operación de máquina filamentadora. ....	75
<b>Figura 38.</b> Arquitectura del sistema. ....	76
<b>Figura 39.</b> Flujograma de proceso de máquina filamentadora.....	78
<b>Figura 40.</b> Diagrama electrónico.....	79
<b>Figura 41.</b> Cintas de PET llevadas a ensayar en las pruebas de extrusión y fusión.....	80
<b>Figura 42.</b> Botella estándar de 2 litros y su incorporación al sistema.....	82

<b>Figura 43.</b> Perfiles y diámetro de impresión obtenidos tras la termo formación. ....	83
<b>Figura 44.</b> Pruebas de calidad y de alimentación del sistema. ....	84
<b>Figura 45.</b> Botella estándar limpia y lista para ser procesada. ....	85
<b>Figura 46.</b> Botella estándar limpia y lista para ser procesada. ....	86
<b>Figura 47.</b> Pruebas de resistencia mecánica: Primeros signos de elongación.....	87
<b>Figura 48.</b> Pruebas de resistencia mecánica: Punto de corte y colapso. ....	88
<b>Figura 49.</b> Ensamblaje del sistema.....	150
<b>Figura 50.</b> Armado de placa de Arduino al proyecto. ....	150
<b>Figura 51.</b> Conexión de placa al sistema.....	151
<b>Figura 52.</b> Pruebas del sistema de Calefacción.....	151
<b>Figura 53.</b> Corte de cinta Enrollado del filamento.....	152

**Índice de Tablas**

<b>Tabla 1.</b> Clasificación de plásticos.....	32
<b>Tabla 2.</b> Componentes para el prototipo .....	102
<b>Tabla 3.</b> Recursos del proyecto .....	103
<b>Tabla 4.</b> Presupuesto del proyecto .....	104
<b>Tabla 5.</b> Cronograma de actividades .....	105

## Resumen

Las cifras alarmantes de producción de plástico en 2018, alcanzando un asombroso total de 359 millones de toneladas y estimaciones de la ONU, de que en el futuro habrá más plástico en nuestros océanos que peces. Este sombrío panorama se ve agravado ya que 8 millones de toneladas aproximadamente de ese plástico terminan en nuestros océanos, donde se convierten en parte del hábitat de los peces y, de manera preocupante, ingresan a nuestra dieta a través del consumo de micro plásticos. Adicionalmente, la fabricación de envases de bebidas está mayormente compuesto por tereftalato de polietileno (PET), caracterizado por su larga vida útil de hasta 1000 años y excelentes propiedades mecánicas. Estas cifras alarmantes han motivado este proyecto, el mismo que pretende enfocar el impacto del PET a ser usado como filamento PET de propósito general en forma de filamento obtenido de una máquina termo formadora que fue diseñada en este trabajo investigativo.

Este proyecto comienza con la adaptación del firmware de control "Marlin", al que se le incorpora hardware para implementar un sistema de control de temperatura mediante termistores y resistencias eléctricas. Además, se integra un sistema de embobinado que regula el movimiento de un motor a pasos Nema 17. Ambos sistemas dependen de un tercer componente, un sistema de corte de cintas que proporciona cintas con un ancho uniforme para su posterior termo formación y embobinado. Cada uno de estos sistemas es ajustable, lo que implica configurar la temperatura de extrusión, la velocidad angular de embobinado y el ancho del corte de la cinta. Estos parámetros se sometieron a pruebas y como resultados se constató que tras la termo formación y aun con una boquilla de 1.5mm el plástico tiende a dilatarse por lo que el diámetro final del filamento llega a enfriarse a 1.65mm sometido a un proceso de ventilación.

En el proceso, se aplicaron métodos de investigación, incluyendo el hermenéutico para obtener información de fuentes bibliográficas y el fenomenológico para establecer premisas cualitativas. También se utilizó el método práctico proyectual, que determinó que el sistema requiere una temperatura de extrusión superior a 200 °C y una velocidad de embobinado de más de 70 mm/s para garantizar un rendimiento óptimo de los actuadores del sistema.

**Palabras clave:** Reciclaje, botellas PET, cintas, filamentadora, velocidad de embobinado, temperatura de extrusión.

## Abstract

Alarming figures of plastic production in 2018, reaching a staggering total of 359 million tons and UN estimates that in the future there will be more plastic in our oceans than fish. This bleak picture is compounded as approximately 8 million tons of that plastic ends up in our oceans, where it becomes part of the fish habitat and, worryingly, enters our diet through the consumption of microplastics. Furthermore, the manufacture of beverage containers is mostly composed of polyethylene terephthalate (PET), characterized by its long shelf life of up to 1000 years and excellent mechanical properties. These alarming figures have motivated this project, which aims to focus on the impact of PET being used as general-purpose PET filament in the form of filament obtained from a thermoforming machine that was designed in this research work.

This project begins with the adaptation of the control firmware "Marlin", which incorporates hardware to implement a temperature control system using thermistors and electrical resistors. In addition, a winding system that regulates the movement of a Nema 17 stepper motor is integrated. Both systems are dependent on a third component, a ribbon slitting system that provides ribbons with a uniform width for subsequent thermoforming and winding. Each of these systems is adjustable, which involves setting the extrusion temperature, winding angular velocity and tape slitting width. These parameters were tested and the results showed that after thermoforming and even with a 1.5mm nozzle, the plastic tends to expand, causing the final diameter of the filament cools down to 1.65mm under a ventilation process.

In the process, research methods were applied, including the hermeneutic method to obtain information from bibliographic sources, the phenomenological method to establish qualitative premises. The practical design method was also used, which determined that the system requires an extrusion temperature above 200 °C and a winding speed of more than 70 mm/s to ensure optimum performance of the system's actuators.

**Keywords:** Recycling, PET bottles, tapes, filamenter, winding speed, extrusion temperature.

## Problema

Para la National Geographic España en su entrega *20 datos sobre el problema del plástico en el mundo*, especifica el larguísimo tiempo de degradación del plástico (450 años si se encuentra en la intemperie) y en general polietileno tereftalato PET del cual están conformadas las botellas de agua y refrescos (42% del plástico del mundo que se usa y desecha en minutos). En el mismo estudio detalla que cada minuto, se vende un millón de botellas y que aun el 25% va a parar en vertederos y fuentes hídricas. (Miranda D, 2023)

Solo en 2018, la producción mundial de plásticos fue de 359 millones de toneladas que equivale al peso aproximado de la población mundial. En ese mismo año se calcula que unos 8 millones de toneladas de residuos plásticos acabaron en los océanos. Para 2050 existirá más plástico que peces según datos de la ONU. El gran reto mundial implica la reducción en la producción de este comoditi y una forma de hacerlo es volverlo un recurso circular. Su gran presencia, así como su maleabilidad ofrece una gran opción para cumplir la “regla de las 3 R” del reducir, reciclar y reutilizar. Colocándonos en el punto del reciclar y luego reutilizándolo con un cambio de forma, entramos en el ciclo del a la reducción en su producción. Como conclusión nos encontraríamos dentro de la denominada “economía circular”. (CAUCE, 2022)

Un recurso muy barato y de gran presencia es justamente de donde parte el presente proyecto. La economía circular es un modelo de producción que se basa en el modelo cíclico de la naturaleza. La nueva producción se aprovecha de un recurso de gran presencia que cumplió su fase de utilidad. De este modo y por medio de la termo transformación se obtiene el medio de una nueva fase de utilidad, creando un curso cerrado y de autosuficiencia. (Edo, 2021)

En el caso de Ecuador, según Alarcón (2017), al año se producen cerca de 4 millones de desechos y de toda esa cantidad solo entre un 15% y 25% son residuos reciclados, mientras que el resto llegan a ser desechos que fueron depositados en los contenedores equivocados, lo que provoca que se mezclan entre residuos reutilizables y

los no reutilizables. En Loja la planta de reciclaje del Centro Integral de Manejo de Residuos Sólidos se ha registrado un crecimiento del 43% entre el año 2015 al 2016 (450 mil kg al año a 560 mil kg). Para el día de hoy esta sobrepasaría los 1330 mil kg por año. Aunque exista este nivel de reciclaje solo indica un incremento gigantesco del 300% en el consumo de plásticos nuevos (vírgenes) sin suponer valor agregado más que el de reciclaje (no se procesan plásticos en mencionado Centro). (Municipio de Loja, 2017)

**Tema**

“DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE  
FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA  
BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO  
ABRIL - SEPTIEMBRE 2023”

### **Justificación**

El presente proyecto pretende aportar con un medio investigativo motivado por las siguientes líneas y sub-líneas de investigación como: sistema de gestión ambiental y conflictos socio ecológicos sostenibles para la ciudad de Loja, alineado con La Agenda 2030 de los Objetivos de Desarrollo Sostenible de la CEPAL. Por su gran impacto tras su recolección, lavado, procesado y conformado pertenece a la línea de transición ecológico -energética, automatización y economía circular. El producto obtenido tras el conformado es un filamento de diámetro variable que puede ser usado en sistemas de corte por abrasión, materia prima de Pellets inclusive como filamento de impresión 3D; con lo que se obtiene un derivado enfilado con la línea de desarrollo y gestión del emprendimiento e innovación.

El desarrollo de un sistema que transforme plástico de subproceso pone a prueba el conjunto de enseñanzas adquiridos a lo largo de la carrera por parte del cuerpo docente del Instituto Superior Tecnológico Sudamericano. Así, la adaptación hardware, modificación de software de control, el análisis de circuitos y el manejo de distintos niveles de voltaje validan a los responsables de ejecutar el presente trabajo de titulación; siempre en búsqueda de una solución integral económicamente asequible y funcional.

El plástico PET será la materia prima que cierra el círculo de reciclaje reutilización y reducción de plásticos del presente proyecto. Supone un constante problema por su degradación lenta de alrededor de 1000 años y por ser inoportuna localización en vertederos que arrastrados por las aguas llegan al mar. El polietileno

tereftalato (PET) es un polímero termoplástico con importantes aplicaciones industriales, el 70% de todas las fibras sintéticas son de PET y se emplean para fabricar cuerda para llantas, diversos artículos de piel y textiles, pero sobre todo en envases para la industria alimenticia. Por las condiciones mecánicas y el relativo bajo costo se ha convertido en uno de los plásticos más producidos. (Flores, 2017)

Si bien no todas las plantas de tratamiento de residuos tienen la capacidad de tratar este plástico muchas sencillamente lo recolectan. Uno de los inconvenientes al ser reutilizado se asocia a la exposición a las temperaturas pues emite vapores tóxicos. De contar con una planta del suficiente control y capacidad de termo tratado se cuenta con un recurso sumamente barato y fácil de obtener. En el presente proyecto se pretende formular una miniplanta de tratamiento de este plástico en de cintas obtenidas de botellas a ser convertidas en filamentos de diámetro variable.

Una producción que posee materia prima prácticamente gratuita solo implica costos de manufactura por depreciación de la maquinaria (Extrusora) junto con el consumo de energético. Al reducir esto de la ecuación variados propósitos de uso como fibras para ropa, elementos agropecuarios termoformados, generación de pellets hasta incluso prótesis biónicas producidas con este plástico se vuelven asequibles a cualquier persona; prótesis diseñadas a medida e impresas en 3D. Yendo hacia lo general las aplicaciones del filamento podrían ir desde cabezas de escobas y filamento de corte en máquinas como guadañas de combustión, también usarse como alambre de amarre en construcciones o producción agrícolas y también como Pellet a combinarse con plástico

virgen. Las prestaciones mecánicas, así como químicas se conjugan en el polietileno tereftalato, volviéndolo un plástico adecuado para la reutilización con mínimos procesos. (Chemical Safety Facts, 2023)

## Objetivos

### Objetivo General

- Diseñar e implementar una máquina extrusora de cinta utilizando botellas de polietileno tereftalato (PET) para la elaboración de filamento de propósito general.

### Objetivos Específicos

- Adecuar el hardware de una impresora 3D para lograr el control de velocidad de un motor a pasos, el control automático de temperatura y la visualización de esas variables.
- Desarrollar el sistema monitoreo de velocidad de embobinado y temperatura de extrusión para un mejor control del producto final, por medio de la modificación del Firmware Marlin de una impresora 3D.
- Implementar el sistema de fileteado de cintas homogéneas a través de un sistema de rodamientos y una arista cortante, para obtener anchos de cintas variables.
- Evaluar el prototipo final bajo niveles de temperatura y de velocidad de embobinado para obtener detalles sobre el diámetro del filamento y sus posibles aplicaciones.

## **Marco Teórico**

### **Marco Referencial**

#### **Datos Generales del Cantón Loja**

La Ciudad de Loja es un valle semiplano que se encuentra a 2,204 metros sobre el nivel del mar. Ciudad bordeada por dos ríos, el Malacatos y Zamora. Posee una población creciente de 254000 habitantes. En Loja es notorio la variedad de catedrales que adornan cada cuadra del centro de la ciudad contando con una obra reciente de regeneración del casco urbano efectuado en 2016 que permitió el desalojo del cableado aéreo y volverlo subterráneo además de añadir aceras de hasta 2,8 metros (Abad, 2023).

El casco central se está delimitado desde la calle Colón y hacia el sur de la calle Lourdes con límite al este con la calle Olmedo y al Occidente la Av. Iberoamérica. Resulta el cuadrante de mayor actividad comercial con más de 2500 puestos (Abad, 2023). Su población económicamente activa se dedica a la agricultura y la ganadería en un 19%, seguida del comercio un 17%, como personal docente en un 17%. Las actividades del siguiente 30% PEA se coloca en actividades tales como construcción, administración pública, industria manufacturera y transporte y comunicaciones. (EcuRed, 2019)

La Constitución de la República del Ecuador reconoce el derecho al disfrute pleno de la ciudad y de sus espacios públicos (Art. 31), así como al acceso y participación del espacio como ámbito de deliberación, cohesión social y promoción de la igualdad en la diversidad (Art. 23). En el cantón Loja se promueve por medio del reciclaje organizado desde el gobierno descentralizado. Partiendo de ello y como

referencias La Agenda 2030 de los Objetivos de Desarrollo Sostenible de la CEPAL en el cantón Loja se reconoce la importancia de incrementar los espacios públicos verdes en las ciudades como parte de las metas del ODS 11: “A 2030, proporcionar acceso universal a zonas verdes y espacios públicos seguros, inclusivos y accesibles” logrando así cumplir la propuesta de un volverla a Loja todo un Sistema Verde como base estructurante de la ciudad. (Segarra et al., 2021)

### **El Reciclaje de Productos Plásticos en el Cantón Loja.**

Por iniciativa del Municipio de Loja se crea el Centro de Gestión Integral de Residuos Sólidos que administra la planta de Planta de Reciclaje y acopio de materiales inorgánicos y producto de la clasificación domiciliaria para su reciclado y comercialización. (Municipio de Loja, 2018)

Adicionalmente, se indagó que más de 200 toneladas de basura son producidas actualmente de forma diaria. Podríamos hacer una relación desde 2014 donde la producción de plástico reciclado era de 30 toneladas por mes para 2018 superaba las 80 y actualmente hablaríamos de una producción de 100 toneladas por mes. La óptica de esta situación únicamente especifica que el cantón Loja únicamente se acopia, clasifica y empacan para su posterior comercialización como producto reciclado sin ningún otro añadido. Este insumo se comercializa principalmente a ciudades como Quito, Guayaquil y Cuenca, que poseen la capacidad instalada de procesar el producto. (Municipio de Loja, 2018)

Cifras que en esta investigación son consideradas como importantes y orientan a la ejecución de una posible alternativa para contribuir a nuestro medio.

### **El Reciclaje en Cantones de la Provincia de Loja**

Para el 2023, cantones como Pindal y Zapotillo se unen al proceso de acopio, clasificación y empaquetado de productos plásticos. Con el propósito de contribuir a la reducción de la contaminación ambiental realizando un adecuado proceso de reciclado y disposición final de envases vacíos de agroquímicos, el Ministerio de Agricultura y Ganadería (MAG), a través de la Dirección Distrital de Loja, capacita a estos cantones (MAG, 2023). La iniciativa últimamente ha tomado vigor por el impacto medioambiental que se da con el aumento de plagas en contenedores de uso agroquímico. El centro de acopio temporal se encuentra implementado con la Asociación de la Industria de Protección de Cultivos y Salud Animal (APCSA), en la parroquia El Tambo, del cantón Catamayo. (MAG, 2023)

## Marco Conceptual

### Clasificación de los Plásticos

Los plásticos son materiales poliméricos Petro-derivados en su mayoría. Por sus prestaciones de suavidad, ligereza y transparencia se han utilizado ampliamente en variedad de empaques y envases. Su consumo se ha extendido globalmente generando serios problemas ecológicos pues su degradación es demasiado lenta (las enzimas de los microorganismos no son capaces de romper los enlaces de carbono C-C). Tenemos algunos ejemplos de termoplásticos como el polietileno (PE), polipropileno (PP), poliestireno (PS), cloruro de polivinilo (PVC) y tereftalato de polietileno (PET) (Flores, 2017).

En la Tabla 1 se clasifica a los tipos de plástico.

**Tabla 1.**

*Clasificación de plásticos.*

Símbolo	Nombre	Usos	Reusable	Reciclable
	Polietileno Tereftalato PET	Envases Contenedores y Souvenirs	Textiles Almohadas Chalecos Salvavidas Velas de barco Abrigos	Sí
	Polietileno De Alta Densidad HDPE	Tapas de envases	Cajones De Plástico, Vallas	Sí
	Policloruro De Vinilo PVC	Tuberías Tarjetas De Crédito	Suelos, Partes De Casas Móviles, Tuberías, Cables	Sí en porcentaje
	Polietileno De Baja Densidad	Bolsas de comida	Papeleras	Sí

	LDPE	Tapas Flexibles		
	Polipropileno PP	Pañales Tuppers Vasos Desechables	Rastrillos, Rascadores	Difícilmente
	Poliestireno PS	Vasos de café	Aislamiento, Reglas	Sí
	Otros Plásticos (Policarbonato, Estireno, Nylon, ABS, PLA)	Discos Láminas de techos Biberones Cases de computadoras	Construcción	Con Dificultad

Imagen tomada de: <https://www.aristegui.info/identificacion-de-los-plasticos-por-sus-codigos/>

### **Tereftalato de polietileno**

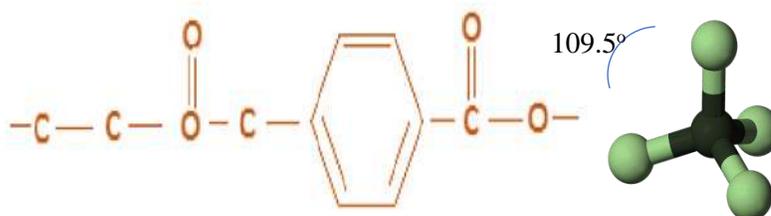
El tereftalato de polietileno o “PET” es un termoplástico producido por la polimerización del etilenglicol y el ácido tereftálico. Por esta razón forma parte de la familia de los poliésteres. En su estructura es amorfo y en algunos casos cristalizado dependiendo de la velocidad de enfriamiento después del conformado. Una de sus propiedades últimamente estudiadas es la disipación de cargas estáticas, esta propiedad permite crear sensores y circuitos inmunes a tensiones estáticas (Elias & Jurado, 2012).

En su estructura atómica sus 3 átomos de carbono se orientan formando un tetraedro alrededor del átomo de carbono de ángulo de enlace entre dos carbonos

próximos de  $109^\circ$  aproximadamente. La introducción en la cadena principal de grupos aromáticos más voluminosos y polarizables como muestra la figura 1, aumenta la rigidez. El anillo bencénico proporciona rigidez además mayor resistencia química a la saponificación de reacción con un hidróxido alcalino, por ejemplo. (Blanco, 2012)

### Figura 1.

*Estructura química del PET.*



*Nota:* Composición química del tereftalato de polietileno. Tomado de Materiales poliméricos y compuestos, por (Blanco, 2012).

## Propiedades Fisicoquímicas del PET

### Estabilidad térmica

Modifica su viscosidad en la fase de fusión ( $275^\circ\text{C}$ ). Debido al aumento de temperatura se provoca cambios químicos en la masa plástica, es decir se reducen, despolimerizan, descomponen, fragmentan y reacciones en los grupos funcionales. La temperatura máxima a la que el PET amorfo alcanza a resistir es de  $71^\circ\text{C}$ , pero el PET cristalizado alcanza a resistir temperaturas de hasta  $230^\circ\text{C}$ . (Mariano, 2011)

### **Densidad**

Se relaciona directamente con el método de elaboración. Es un factor económico de alto valor en la industria plástica ya que es determinante al momento de elegir el uso de un plástico con otro. El PET en estado amorfo tiene una densidad entre 1.33-1.37 g/cm<sup>3</sup> y en estado cristalino entre 1.45-1.51 g/cm<sup>3</sup>. (Beltrán, 2012)

### **Resistencia al Ataque Químico y Solubilidad**

Este comportamiento obedece a la naturaleza de los grupos funcionales que conforman los monómeros y a su estructura. El PET presenta una resistencia química a los hidrocarburos, alcoholes, grasas y aceites, éter, bases diluidas y ácidos (Elias & Jurado, 2012). El PET es insoluble en solventes orgánicos, aunque presenta una solubilidad en mayor o menor grado a algunos solventes como cetonas, compuestos clorados y alcoholes de cuatro u ocho carbonos, solventes halogenados, aromáticos, cetonas de bajo peso molecular y bases. Durante el procesamiento del PET se da el factor higroscópico donde muestra sensibilidad a la hidrólisis. (Martínez, 2013)

### **Permeabilidad**

Se define como la cantidad de ocupación de un determinado gas, (habitualmente vapor de agua y oxígeno, nitrógeno y dióxido de carbono), que traspasa el material por una determinada unidad de superficie. En la industria alimenticia es requerible un material con excelentes propiedades de barrera. El PET presenta muy buena barrera al dióxido de carbono, al oxígeno y al agua (0.7 % por día). (Sánchez & Paredes, 2014)

### **Conductividad Térmica**

A la cantidad de calor que transmite un cuerpo se la define como conductividad térmica (Pérez, 2016), en este caso, es la cantidad de calor que transmiten los materiales plásticos a través de ellos. Con esta propiedad se conoce si el plástico tiene un buen aislamiento térmico, siendo mejores los que presentan valores bajos. El PET presenta un valor de conductividad térmica de 0.24 W/m.K. (Mariano, 2011)

### **Propiedades Ópticas**

Están relacionadas con la morfología y estructura del material, las principales son: índice de refracción y transmisión de la luz. Los materiales amorfos cuando no contienen impurezas poseen mayor transparencia, por el contrario, las formaciones con estructuras cristalinas. El PET tiene una transmisión de luz del 89% y un índice de refracción de 1.576. (Sánchez & Paredes, 2014)

### **Métodos de Conformado PET**

Un molde hembra de aluminio permite emplazar una plancha de termoplástico y fijarla de modo que el hueco entre molde y pieza fuese estanco, calentarla hasta su temperatura de fusión y hacer el vacío. El material se adapta y una vez fría la pieza, se extrae y recorta. Como alternativa al vacío entre el molde y lámina, puede aplicarse presión sobre ésta para obtener un resultado similar. (Blanco, 2012)

**Figura 2.**

*Moldes para soplado de botellas plásticas.*



*Nota:* El gráfico representa moldes en aluminio para soplado de botellas plásticas.

Imagen tomada de: <http://bottle-blows.com/5-2-plastic-bottle-blowing-mold-manufacturing.html>

Puede darse un estiramiento no homogéneo. Es por esto que se desarrollado técnicas de pre-estirado por punzón o soplado previo, que permiten obtener mayor regularidad de espesor. Una vez termoformado y conformado el PET, tras su vida útil en posible reutilizar envases de forma casi directa volviéndolo cintas de un ancho regular a ser transformado en otra materia prima como es el caso del presente trabajo. Convertiremos cintas PET en filamento (hilo de sección transversal variable).

**Figura 3.**

Matriz de botellas PET (soplado previo).



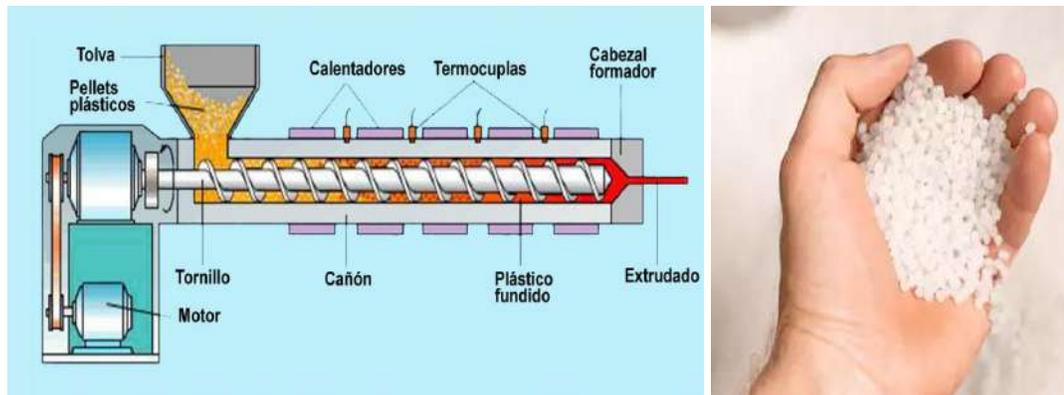
*Nota:* Representan botellas aplicada un soplado con moldes de aluminio. Imagen tomada de: <https://www.gestiondecompras.com/es/productos/piezas-plasticas/preformas-pet/>

Un PELLET es la materia prima que permite una reutilización por conformación (MAXI PET, 2020). Los **pellets** de plástico se producen en una cortadora de placa y el proceso se realiza en 4 pasos:

- Dosificación: Se suministra la materia prima
- Mezclado: Se agrega el vapor de agua y los aditivos necesarios
- Peletización: Se preforma un filamento por presión y se corta en porciones
- Secado: Se elimina el agua contenida en el producto

**Figura 4.**

*Esquema de Peletizadora y producto final (PELLET)*



*Nota:* Se muestra de pellets. Imagen tomada de: <http://blog.fdtecsl.com/peletizadoras-pequenas-pruebas-plasticos> y de (MAXPET, 2020)

## **Aplicaciones del PET Reutilizo o rPET**

### **Prendas de Vestir**

En industria textil cada vez son más las marcas que apuestan por la creación de prendas rPET. Por ejemplo, Nike desde 2010 fabrica camisetas del FC Barcelona con poliéster reciclado. Nike ha reutilizado más de 6 billones de botellas hasta la fecha. Se le suman marcas como Patagonia, The NorthFace, Decathlon, Target, Skunkfunk, H&M, G-Star Raw, Billabong o Zara. Durante el proceso de reciclado se reconstituye la fibra a nivel molecular por tanto el poliéster mantiene las mismas propiedades. La diferencia es en su impacto medioambiental. (Teefactory, 2023)

**Figura 5.**

*Prendas de vestir a partir de rPET*



*Nota:* Productos obtenidos para vestir de material reciclado. Imagen tomada de: <https://teefactory.es/sostenibilidad/poliester-reciclado>. Se muestra una chaqueta de la marca Stanley Stella.

**Láminas de PET 100% Reciclado**

Una lámina de PET usualmente se usa para termo formar y ser parte del empaque de juguetes y herramientas, así mismo se puede conformar como envase o contenedor desechable para alimentos. La empresa ecuatoriana CERSA ofrece todo un abanico de prestación con su producto de PET en forma laminas en rollos; ofreciendo aplicaciones en farmacéutica, alimentos, florícolas y protectores de Bioseguridad. (CERSA, 2023)

**Figura 6.***Aplicaciones de Lámina PET CERSA*

Imágenes tomadas de: <https://cersa.ec>. Las láminas de PET por medio de termoformado constituyen las aplicaciones mostradas debajo.

## **Diseño Metodológico**

### **Métodos de Investigación**

Se adoptó las metodologías fenomenológicas, hermenéutica y práctico proyectual proponiendo el presente proyecto como un caso de estudio de carácter experimental basado en búsquedas bibliográficas relativas.

### **Método Hermenéutico**

Es un método diseñado por Edmund Husserl como un paradigma que pretende explicar la naturaleza esencial de las cosas y la veracidad de los fenómenos. Se parte del estudio de las experiencias de vida, respecto de un suceso, desde la perspectiva y del sujeto. Por lo que es una metodología usada para el cualitativo por medio de observación e interpretación de los datos o hechos, especialmente en el pasado (historia, escritos, conductas, etc.). Conocer las vivencias por medio de los relatos, las historias y las anécdotas es fundamental porque permite comprender la naturaleza de la dinámica del contexto e incluso transformarla para obtener variantes. (Guillen, F., & Elida, D. (2019)

Partiendo de la experiencia de la indagación en fuentes bibliográficas especialmente en el diseño de máquinas adaptadas con el fin de reutilizar plásticos brindó las posibilidades enormes de cumplir con el objetivo principal del proyecto. Logrando un sistema con un gran mecanismo para mitigar la contaminación ambiental por plásticos que terminarían en vertederos. Así mismo, en cuanto al impacto social diríamos que se ofrece una gran oportunidad de negocios al especificar una máquina de pocos materiales que genera materia prima de obtención domestica inmediata.

### **Método Fenomenológico.**

Es una herramienta para el establecimiento y resolución de tesis de base cualitativa. Está orientado a la observación descriptiva e interpretación de las estructuras fundamentales de la experiencia vivida. Compone un acercamiento coherente y estricto al análisis de las dimensiones éticas, relacionales y prácticas (Fuster, 2019).

En este proyecto se aplicó este método para el análisis, indagación y observación de varios estudios previamente realizados en el ámbito técnico y electrónico los mismos que permitieron aplicar un mecanismo, variables, hardware y software. Así como, en base a la experiencia y aplicación de varios ensayos previos se pudo optimizar el control de temperaturas, accionamiento de actuadores, procesamiento del plástico hasta poder reducir o mitigar el impacto medioambiental por medio del proyecto ejecutado.

### **Método Práctico Proyectual**

Este método consiste en seguir de manera ordenada una serie de instrucciones o guías para poder llegar a un resultado cuantitativo. El sentido lógico se añade por parte del investigador y tiene sustento experiencial. De esta forma en combinación se conseguirá un máximo resultado en un mínimo esfuerzo. Este método propuesto por Bruno Munari propone los siguientes pasos: definición del problema, ideas, elementos del problema, recopilación de datos, análisis de datos, encuestas, materiales tecnológicos, experimentación, modelos, verificación, bocetos constructivos y solución (Castro, 2016).

Desde la problemática hasta la resolución de objetivos esta metodología nos brindó un sistema para obtener una solución de forma ordenada. Sirvió de gran ayuda para la validación de resultados:

- El filamento plástico ¿será un medio de propósito general (como filamento de amarre ornamental, agropecuario, herramientas de corte, mobiliario)?
- El producto obtenido ¿será útil para impresión 3D?

Así mismo en el proceso de boceteo, construcción hasta dar con la solución.

- ¿El sistema es funcional?
- ¿El sistema es mínimo y viable?
- ¿El sistema puede comercializarse?

## **Técnicas de investigación**

### **Búsqueda Bibliográfica**

La búsqueda bibliográfica es una técnica de investigación documental válida para cualquier tipo de investigación. Consiste en indagar las fuentes de información relacionadas con el problema a desarrollar. Esta fuente puede ser de diversos tipos, como libros, revistas, tesis, artículos de periódicos o científicos, entre otros. A partir de esta los investigadores pueden conducir con mejor precisión la realización del estudio. (Miño & Molina, 2019)

Por cuanto respecta a esta investigación la búsqueda bibliográfica nos presentó la base teórica para la manipulación térmica del PET (temperatura de vitrificación de 200°C que permite el termoformado de las cintas PET). Así mismo, nos ofreció bases sobre el control de velocidad y de temperatura de lazo cerrado y en un mismo sistema

basado en el firmware de una impresora 3D, permitiéndonos manipular el enrollamiento y la facilidad para manipular este plástico sin incurrir en la emisión de gases nocivos . Esta búsqueda bibliográfica también nos ha puesto en contexto sobre el acopio y tratamiento previo al proceso de extrusión para eliminar impurezas en el plástico.

### **Experimento**

Esta es la técnica de investigación predilecta en ciencias que consiste en escoger un sistema. A este se le realizaran modificaciones específicas bajo algunas condiciones para luego estudiar el efecto. En un experimento es ideal mantener la mayor cantidad de condiciones fijas y solo variar un factor, de forma tal que cualquier cambio observado pueda ser atribuido a ese factor (López & Sandoval, 2016).

El experimento como técnica fue crucial en esta investigación. Lo que permitió probar las variables de temperatura, velocidad angular, diámetros de sección, ancho de cinta. Dentro de esta misma técnica se aplicó la observación con lo que se pretende particularizar los efectos frente a cada “experimento” pues en esta técnica fue posible establecer relaciones de acción reacción. Para cerrar el círculo de experimentación y únicamente con la base teórica de la técnica de búsqueda bibliográfica, cada experimento fue parte de una técnica de prueba y error tratando de encontrar diferentes conclusiones a los experimentos planteados.

## **Propuesta Práctica de Acción**

El desarrollo de este proyecto parte de una modificación tanto de software como de hardware de una impresora 3D Open Source del proyecto denominado Rep Rap. Existe además un componente mecánico basado en un sistema de reducción de velocidad que permite mejorar prestaciones del prototipo obtenido. El software y hardware implícitos en el sistema se encuentran trabajados por medio del software también libre Arduino que en conjunto permiten una rápida implementación del presente proyecto. En los resultados se ve la transformación de cintas plásticas obtenidas de botellas de refresco PET termoformadas en filamento de diámetros variable en función de la temperatura de vitrificación, así como de la velocidad de embobinado del sistema mecánico. Se detalla lo mencionado a continuación.

### **Hardware**

#### **Arduino Mega 2560**

Como su nombre lo indica esta placa se encuentra basada en el microcontrolador ATmega2560 el mismo que se muestra en la figura 8. La mencionada placa posee hasta 54 pines digitales tanto de entrada como de salida; de estos, 15 producen salidas de modulación de ancho de pulso PWM, 16 se pueden usar como entradas analógicas, 4 puerto seriales de hardware embebido UART una de ellas (Tx0-Rx0) se encuentra conectada físicamente a un puerto USB. El bootloader del microcontrolador especifica un funcionamiento a 16 MHz. La placa también posee puertos ICSP con protocolos de comunicación I2C. Esta placa se encuentra acompañada de una Shield o placa capa

(RAMPS 1.4) que conecta el resto de las interfaces del sistema al Arduino Mega 2560.  
(Arduino CC, 2023).

### **Figura 7.**

*Placa de desarrollo Arduino Mega2560.*



*Nota:* Representa a una placa de Arduino basada en microcontroladores. Imagen tomada de: <https://acortar.link/EAEpM>. Arduino Mega 2560.

### **Shiel RepRap Arduino Mega Pololu RAMPs**

Probablemente en el mercado existan diferentes placas para controlar una impresora 3D doméstica, pero sin duda, la combinación Arduino Mega 2560 más RAMPS 1.4 es de las más usadas entre la jerga de desarrolladores REP RAP. En la figura 15 se muestra una tarjeta Ramps con todas las interfaces posibles a conectar. Para el presente proyecto usaremos cierta parte de esta placa como se detalla en el apartado de diagrama eléctrico, así como de sus interfaces. Esto último implica posibles mejoras a futuro como una placa especializada, así como un firmware de control. Usualmente

esta placa posee las interfaces mostradas en la Figura 9 para conectar periféricos a usarse en una impresora 3D. (RepRap ORG, 2023)

### Figura 8.

*Capacidad de puertos a interfaces de Shield RAMPs 1.4*

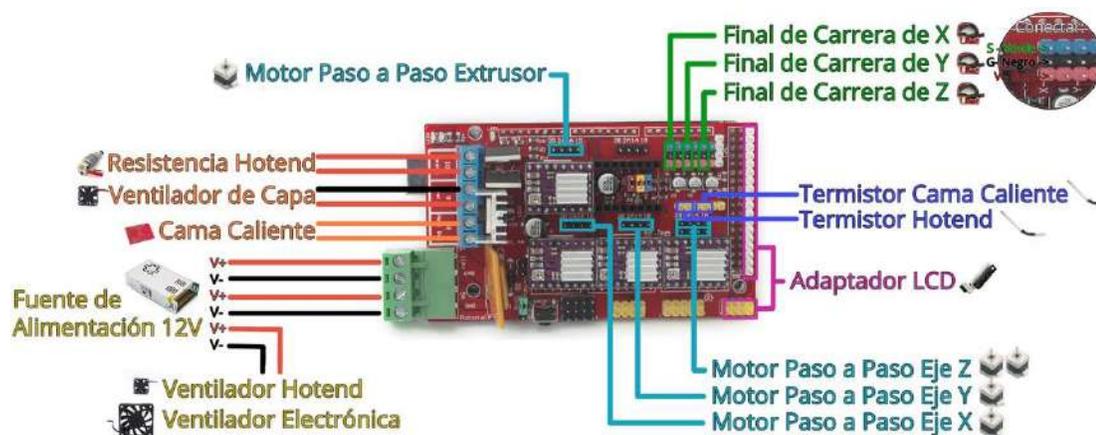


Imagen tomada de <https://www.hta3d.com/es/blog/Como-Conectar-RAMPS1.4>

### Pantalla GLCD ST7920 de 128x64 Píxeles

Una pantalla GLCD o gráfica de cristal líquido (del inglés, Graphic Liquid Crystal Display) es un monitor monocromático formada por una matriz de píxeles que presenta su programación una vez una luz choque por su parte trasera. Actualmente se ha vuelto muy común verlas entre impresoras 3D desde una gama básica a una media ya que utiliza cantidades muy pequeñas de energía eléctrica y su costo es muy asequible comparada con una pantalla OLED. En el presente proyecto usaremos una GLCD de 128x64 píxeles que permite realizar visualizaciones por medio del controlador ST7920 que crea una interfaz de conexión con un protocolo de comunicación en paralelo SPI. El

firmware del sistema posee la programación requerida para que en la pantalla de inicio muestre el mensaje que se ve en la figura 10 (FreePlastic). (RepRap ORG, 2023)

### **Figura 9.**

*Pantalla de inicio en GLCD 128x64*



*Nota:* Representa la pantalla para inicio que crea interfaz. Tomada del proyecto

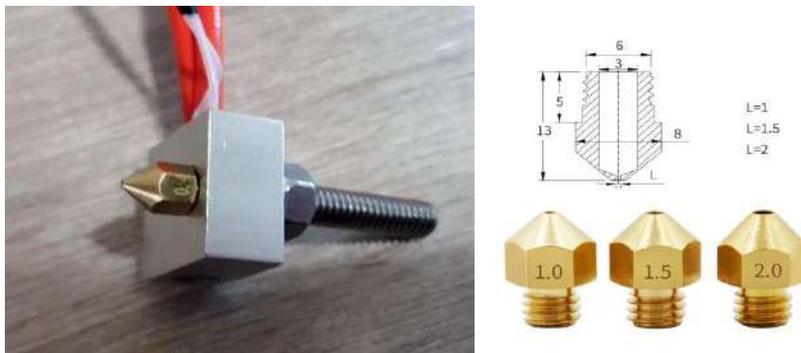
### **Bloque Calefactor con Termistor para Impresora 3D Oopensource**

Un bloque calefactor es un mecanismo por el cual el sistema de control calienta un metal por el cual ha de interponerse un material dúctil y maleable. Hablando de plástico se vuelve blanco y al atravesar una boquilla se transforma en un segmento más fino extruido. Este bloque permite una termo formación del plástico pues alberga un sensor termistor y la niquelina en una pieza de aluminio. Como se ve en la figura 10 se trata de un cubo metálico dotado de roscas para la boquilla M6 y un orificio roscado M3 para asegurar el mencionado sensor. El bloque calefactor en conjunto con el termistor y niquelina se lo puedo hallar comercialmente como bloque MK8.

(COMPUTORREFUERTE, 2023)

**Figura 10.**

*Bloque calefactor ensamblado con termistor y niquelina*



*Nota:* Representa un cubo metálico calefactor. Imagen tomada de:

<https://es.aliexpress.com/item/1005002571228195.html>

El conjunto termistor y niquelina posibilitan un control de temperatura automatizado donde el actuador es una resistencia eléctrica de 40W que puede funcionar a 12V o 24V y el sensor que cierra el lazo es un termistor NTC de 100k $\Omega$ . Este actuador está protegido por un recubrimiento de acero inoxidable (cartucho) que permite la mejor distribución de calor. Está preparada para el desgaste térmico pues la oxidación es el agente de mayor preocupación. Esta resistencia permite llegar a temperaturas de más de 300 °C que probablemente no sean seguras ni útiles para este efecto. (Hackduino, 2023)

**Figura 11.**

*Resistencia eléctrica de 24W y Termistor NTC de 100k $\Omega$*



Imagen tomada de: <https://www.instructables.com/Ejemplo-básico-de-termistor-NTC-y-Arduino/>

Inherente y formando parte del mismo sistema se encuentra el sensor de temperatura o termistor 100k NTC. Se trata de una unión de dos metales como cobre y acero que al reaccionar a un impacto mecánico como el aumento de temperatura tienden a modificar su diferencia de potencial pudiéndose realizar una traducción de este cambio de voltaje en función de la temperatura. Usualmente el firmware limita el control de temperatura a 275 °C; para vitrificar el plástico PET es suficiente tan solo 200 °C. Este dispositivo funciona a 12V. (Hackduino, 2023)

### **Motor a pasos Nema 17**

Un motor paso a paso **Nema 17 es de tipo bipolar**, es decir posee dos pares de bobinas que deben ser alimentadas con cierta lógica para generar rotaciones. El usado en el presente es de 1.8° por paso, es decir, cada revolución posee 200 pasos. Cada bobinado posee una inductancia de 3 a 8mH requiere de entre 1.5A a 1.8A una tensión de alimentación suministrada de 1-4V y es capaz de efectuar un torque de 4 kg/cm o 44Ncm. (RepRap ORG, 2023)

### **Figura 12.**

*Motor a pasos Nema 17 1.5A*



*Nota:* Representa motor bipolar. Imagen tomada de:

[https://reprap.org/wiki/NEMA\\_17\\_Stepper\\_motor](https://reprap.org/wiki/NEMA_17_Stepper_motor)

### Driver de control A4988

Como es común se debe realizar adecuaciones para que el medio de control maniobre el medio de potencia. De esta forma es que se usa el Driver A4988 que recibe la lógica proveniente del microcontrolador y efectúa su activación en potencia hacia el motor. Este pequeño driver posee protección para sobre corrientes y sobre temperaturas y a la capacidad de reducir los micro pasos del 1 al 1/32 división por paso. En su placa sobresale un potenciómetro tipo timer que permite regular la corriente. (RepRap ORG, 2023)

### Figura 13.

*Driver A4988. Posicionamiento en la placa Ramps 1.4*



*Nota:* Representa un driver con protección para sobre cargas. Imagen tomada de: <https://reprap.org/wiki/A4988>. A4988 y su disipador de calor

### Fuente de Alimentación Conmutada 12V 33A

Para el presente trabajo se realiza un cálculo de potencias en donde el bloque calefactor consumo 40W (5A a 12V), el motor a pasos puede llegar a consumir hasta 1.5A el resto de la electrónica consumirían alrededor de 1A. por lo que una fuente debe ser superior a 15A. El uso de una fuente de alimentación de mayor potencia se debe a que se poseía como parte de materiales previos al inicio del desarrollo del presente.

(CREATEC4 S.C.A., 2023)

### Figura 14.

*Fuente de alimentación.*



*Nota:* Representa fuente de alimentación superior a 15A. Tomada del proyecto

## **Software**

Es bien sabido que el componente de software de un sistema provee control de arbitraje en el accionamiento de los actuadores conectados al mismo en función de las condiciones de entrada y de la lógica de programación. Para el presente proyecto y aunque es intangible juega un papel importantísimo pues se encarga de gestionar todo el conjunto electrónico embebido. A nivel general el software se encuentra desarrollado en C++ además de ser requerible para la personalización y configuración el IDE de Arduino. Este programa que toma el nombre de “firmware” permitirá al usuario interactuar dentro del marco de la programación predefinida que se aloja en una placa de desarrollo como Arduino Mega 2560. (Votofeli, 2023).

El mencionado programa se basa en la utilización y configuración del firmware de gestión de impresoras 3D Marlin, el mismo que es de código abierto. Por cuanto el diseño de partes mecánicas, así como chasis de los diferentes componentes se usó el programa Fusion360; ambos detallados a continuación:

## **Marlin**

Marlin es un firmware de código abierto del movimiento RepRap para creación de máquinas autorreplicables como son impresoras 3D y máquinas de control numérico computacional en general CNC y vigente desde 2005. Posee una licencia GPLv3 y es de libre acceso. Puede ser embebido en microcontroladores (de 8 a 32 Bits) así como lo es una placa de desarrollo Arduino donde su código se encuentra optimizado para gestionar el microcontrolador y convertirlo en el cerebro de un sistema de lazo cerrado (IDEA 1.61, 2023). Este Firmware permite la lectura de extensión GCODE que especifica

movimientos y acciones con actuadores como niquelinas, ventiladores, motores, etc. Posiblemente desde una tarjeta SD (Votofeli, 2023).

### Figura 15.

*Ícono del Firmware OpenSource Marlin.*



Nota: Tomado de <https://marlinfw.org>

Marlin tiene la capacidad de gobernar Impresoras 3D, máquina de control numérico CNC inclusive plotters. Es el firmware más usado para los sistemas de impresión por deposición pues cuenta con una licencia GPL de desarrollo libre con el requerimiento de mantenerlo libre aun si se ha modificado como producto. Sus homónimos son Klipper, Octoprint, Mach3, entre otros. (KolMite, 2023)

### Figura 16.

*Página oficial de Marlin ORG.*



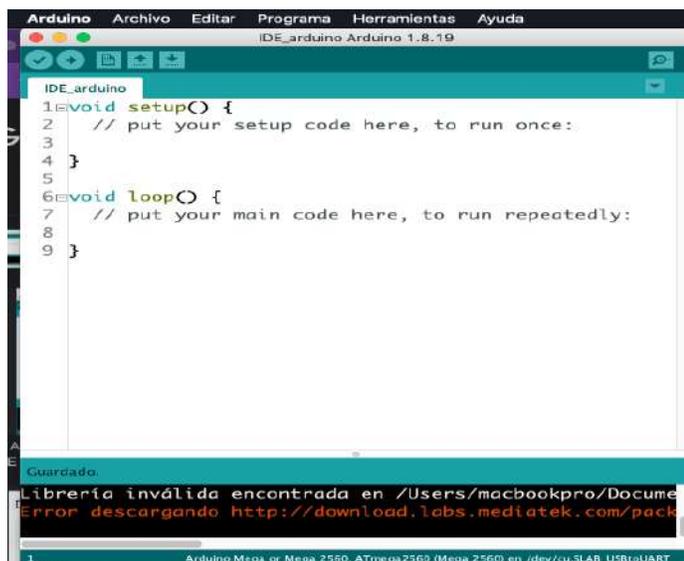
Imagen capturada de: <https://marlinfw.org/>.

## El IDE de Arduino

Arduino como tal es un conjunto de plataforma de programación y un sistema embebido también llamado así. Se admite varias plataformas de hardware y software, así como sistemas operativos, lo que permite que casi cualquier persona en el mundo pueda acceder al avance tecnológico del proyecto Arduino. Se puede decir que es una plataforma de código abierto flexible y de millones de adeptos porque su soporte también es alto. Arduino permite el acceso a diagramas y especificaciones de proyectos de forma libre que se encuentran en acceso público con el fin de replicarlo o mejorarlo. En el caos del software también es de libre acceso permitiéndose modificar el código enfocado a casi cualquier propósito. El proyecto Arduino integra software y hardware por medio de su IDE (Entorno de Desarrollo Integrado de programación). Este entorno es un laboratorio completo de experiencias basadas en un programa y su retroalimentación directa implementada en la placa de desarrollo. (Arduino CC, 2023)

### Figura 17.

*IDE Arduino y sketch base*



```
Arduino  Archivo  Editar  Programa  Herramientas  Ayuda
IDE_arduino Arduino 1.8.19
IDE_arduino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
Guardado.
Librería inválida encontrada en /Users/macbookpro/Docume
Error descargando http://download.labs.mediatek.com/pack
1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) en /dev/cu.SLAB_USBtoUART
```

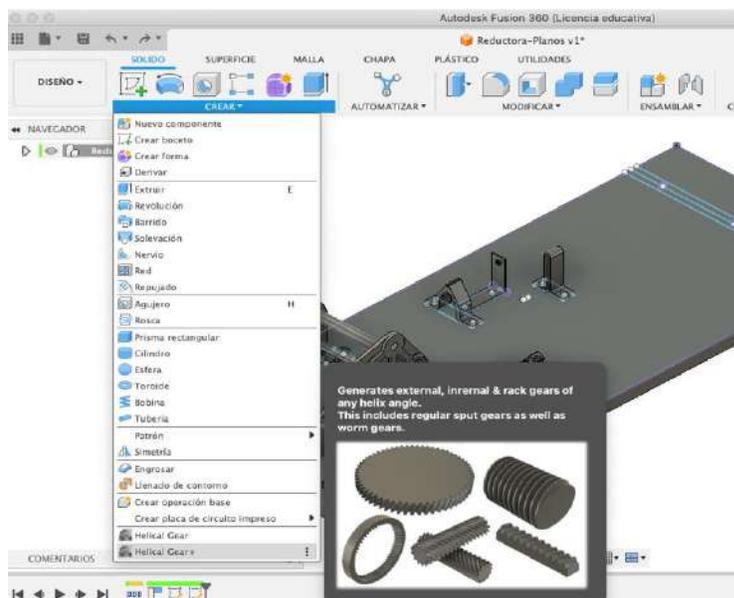
## Fusion360 para Diseño del Sistema de Enrollado

El *sistema de enrollado de filamento* parte del control de un motor a pasos Nema 17 que es activado por modulación de ancho de pulso PWM y energizado por el driver A4988. Este sistema necesita de un aumento mecánico de potencia logrado por un tren de engranajes. Mencionados mecanismos fueron desarrollados en el software de diseño CAD/CAM/CAE **Fusion360** de la compañía Autodesk bajo una licencia de estudiante. Es necesario añadir un Toolkit llamado Helicoidal Gear +, descargable de forma gratuita de la tienda Autodesk Store. (Autodesk, 2023)

De acuerdo con el mismo Autodesk es la mejor plataforma de software profesional de diseño y fabricación de productos con procesamiento en la nube. Este poderoso programa permite el modelamiento computación en diseño manufactura y aplicaciones de ingeniería CAD, CAM, CAE. Posee todo un conjunto de extensiones para el análisis, gestión y manufactura del producto diseñado. (Autodesk, 2023)

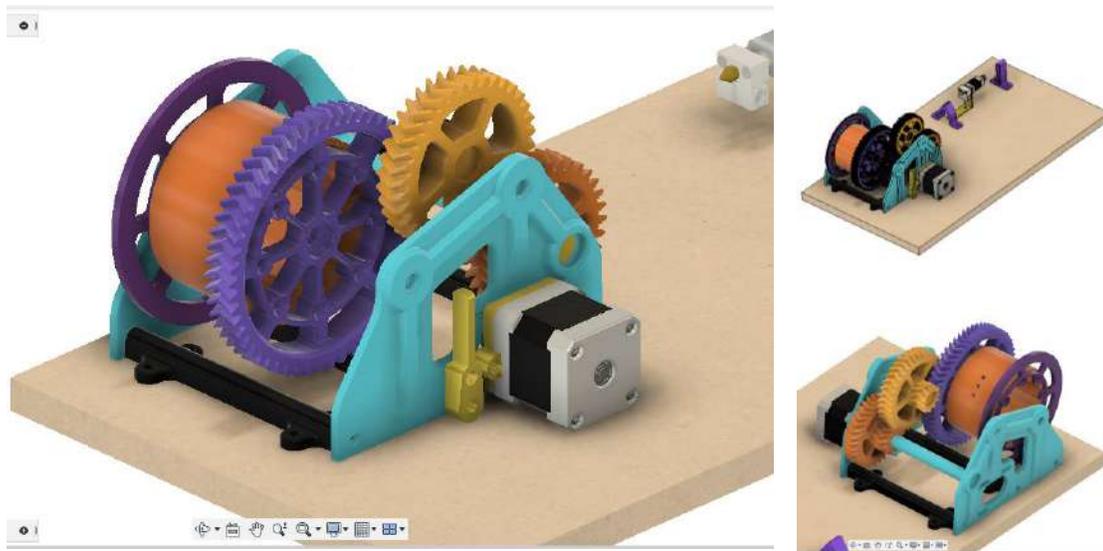
### Figura 18.

*Toolkit Autodesk para la creación de engranajes en Fusion360.*



**Figura 19.**

*Vista isométrica en la interfaz de Fusion360*



*Nota:* Puede verse el modelo en: <https://a360.co/3NbuJZC>

## **Desarrollo de la Propuesta**

### **Diseño y Construcción del Prototipo**

Para llevar a efecto la construcción del prototipo se creyó conveniente realizar un levantamiento 3D que nos permitirá tener mejor óptica del producto final. Si bien es cierto con esto anterior ya es posible su implementación para llevarse a pruebas de ejecución. Se parte de la construcción de un sistema de reducción de velocidad adaptada al motor Nema 17. Esto mejorara significativamente el aumento de torque para enrollar el filamento producto. Además, es necesario realizar modificaciones bloque calefactor comercial de una impresora 3D con lo que se permitirá obtener plástico termoformado a diferentes diámetros. Para este proyecto la boquilla tendrá una envergadura de 1.5mm. Para una mejor compresión a la construcción se ha dividido en 3 etapas. Etapa de

configuración de software, etapa de adecuación e interconexión de hardware y finalmente la etapa de adecuación de mecanismos.

### **Configuración del software**

Gracias a la versatilidad de Marlin junto con la facilidad para su modificación a través del entorno de programación Arduino IDE las variantes de máquinas de control numérico computacional CNC son ilimitadas; implementables incluso a nivel doméstico (en casa).

Marlin nos ha de prestar un sencillo control de temperatura por medio del acondicionamiento de un termistor y una niquelina: en función a la necesidad accionará este actuador para calentar el *bloque de calefacción* por donde atravesará la cinta de plástico PET. Así mismo el mismo firmware se encargará de gestionar el movimiento de un motor a pasos con el que pretendemos enrollar el filamento generado; todo monitorizado a través de una pantalla GLCD también personalizable. Al visitar la página es probable que existan varias versiones, pero para efectos del presente proyecto usamos la versión 1.1.x compatible con microcontroladores AVR, así como con el Shield (RAMPS 1.4) por ir montado en una ATMEGA 2560.

Figura 20.

### Selección y descarga de Firmware



Description	Version	Download	Configurations
<b>Current Marlin Release</b> Supports AVR and ARM Arduino and PlatformIO	<a href="#">2.1.2.1</a>	<a href="#">2.1.2.1.zip</a>	<a href="#">View / Download</a>
<b>Marlin 2.0 LTS</b> Supports AVR and ARM Arduino and PlatformIO	<a href="#">2.0.9.7</a>	<a href="#">2.0.9.7.zip</a>	<a href="#">View / Download</a>
<b>Marlin 1.1 LTS</b> Supports AVR Arduino and PlatformIO	<a href="#">1.1.9.1</a>	<a href="#">1.1.9.1.zip</a>	<a href="#">View / Download</a>
<b>Marlin 1.0 LTS</b> Supports Arduino 1.6.8 and up	<a href="#">1.0.2-3</a>	<a href="#">1.0.2-3.zip</a>	(included)

*Nota:* Representa la descarga del sistema desde la web.

Dependiendo del compilador usado el sistema tiende a tener ciertas mínimas variaciones en su comportamiento. Es por eso que se menciona cierta versión del software Arduino como es la **1.8.16** donde hemos probado el proyecto e inclusive hemos realizado la personalización. Arduino IDE se encuentra actualmente en la distribución 1.8.19 (la versión previa muestras mayor estabilidad).

Figura 21.

### IDE Arduino y sketch de Marlin 1.1.x con sus subprogramas

```

Marlin - CONFIGURATION.h [Arduino 1.8.16]
Arduino - Editor Properties: Herramientas: Ayuda
// #ifndef CONFIGURATION_H_
// #define CONFIGURATION_H_

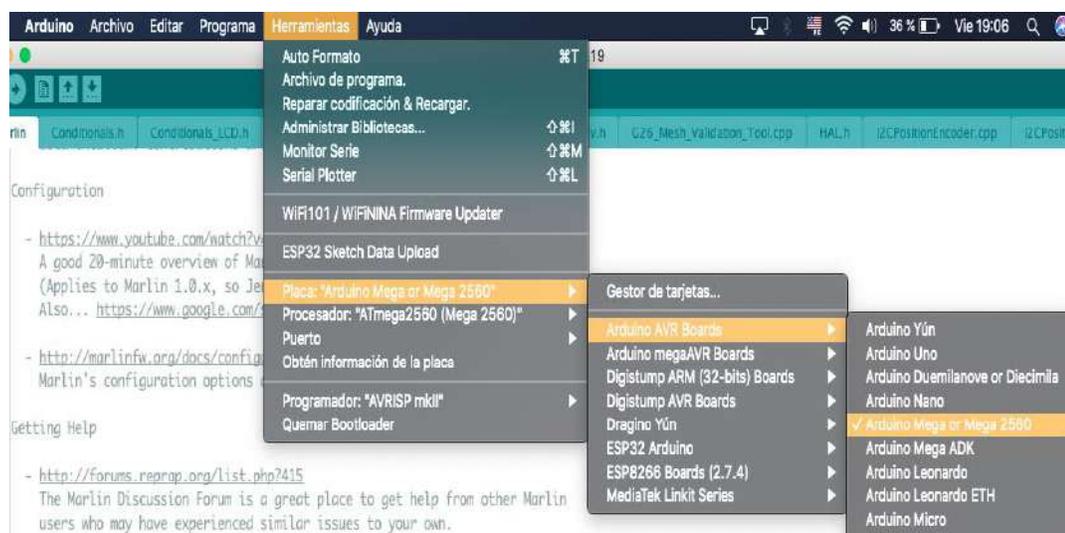
34
35 #define DOGLCD
36 #define ULTIPANEL
37 #define DEFAULT_LCD_CONTRAST 90
38 #define LCD_CONTRAST_MIN 50
39 #define LCD_CONTRAST_MAX 140
40
41 #define ENABLED(MARKUPANEL)
42
43 #define DUEGLIB_ST7565_64128N
44
45 #define ENABLED(ISSONESTAR_LCD)
46
47 #define REPRAPWORLD_KEYPAD
48 #define REPRAPWORLD_KEYPAD_MOVE_STEP 10.0
49 #define AIN2_KEYPAD
50 #define AIN2_KEY_NUM 5
51 #define ULTIPANEL
52
53 // #define helpers to implement AIN2_KEYPAD macros
54 #define ENCODER_PULSES_PER_STEP 1
55 #define ENCODER_PPRPPS_PER_MM 1000

```

Una vez descargado todo un comprimido de Marlin se debe ejecutar en único archivo de la carpeta con extensión. ino (buscar Marlin.h). Al abrir despliega y llama una serie de otros subprogramas donde podremos realizar las configuraciones respectivas en cuanto a la pantalla de inicio, la interfaz, el idioma, el límite máximo de temperatura inclusive ajustar el sistema de control automático PID para obtener respuestas más rápidas, así como se muestra en la Figura 22. A continuación se muestra algunas configuraciones sobre este firmware en el IDE de Arduino.

**Figura 22.**

*Selección de la placa Arduino Mega 2560 para compilación.*

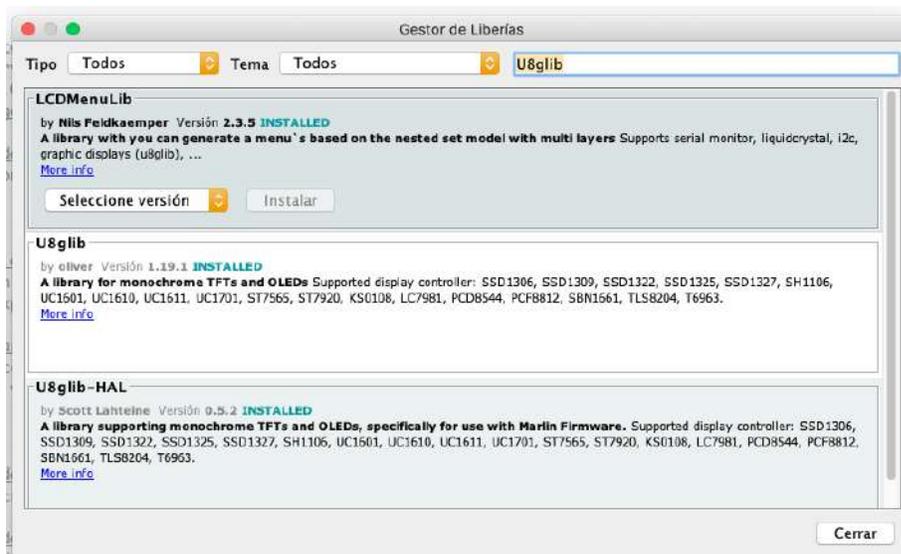


*Nota:* Representa las configuraciones del sistema Arduino.

Hecho esto y antes de compilar o subir el programa se debe seleccionar la placa a utilizar en este caso Arduino Mega 2560. Además, es requerible instalar una biblioteca que gestiona el sistema de archivos a ser monitorizados en la pantalla GLCD cuyo nombre es **U8glib** y es parte del repositorio del Gestor de bibliotecas.

Figura 23.

*Instalación de la biblioteca U8glib en el entorno (se instalan las 3 dependencias)*



En la pestaña de *Configuración.h* se debe ratificar que el SERIAL\_PORT 0 que garantiza usar el puerto nativo de la placa; así mismo la velocidad del puerto es 2500000 baudios. Se usa un solo extrusor o EXTRUDERS 1 para controlar el bloque de calefacción. Se activa el puerto TEMP\_SENSOR\_0 1 para tomar la medida de sensor de temperatura. Se coloca ENDSTOP\_INVERTING **false**, pues no se utiliza sensores de final de carrera. El motor del extrusor debe girar en sentido antihorario INVERT\_X\_DIR **true**. En cuanto a las temperaturas preestablecidas se activa de 180 a 240 grados dependiendo del espesor de la cinta PET (PREHEAT\_1\_TEMP\_HOTEND 180 / 240). El lenguaje por defecto será es Ingles, aunque es modificable a español (LCD\_LANGUAGE **en**).

La placa de control de potencia son Ramps 1.4 o BOARD\_RAMPS\_14\_EFB deben estar de-comentadas (borrar los //) todas incluso para darle mayor compatibilidad se activa las del 13. De misma forma se quita los comentarios en BOARD\_GEN7\_14. Para finalizar en la pestaña **Driver** se debe constatar la activación de toda la gama de drivers, aunque en el presente se use el A4988 probablemente posmos optar por alguno de la serie TMC y brindar control sobre las frecuencias que ocasionan ruido en los motores a pasos (define A4988 0x001). El PID o parámetros del sistema de control automático se mantiene por default y únicamente se recomienda modificación bajo cierta metodología de calibración. En la pestaña **Version.h** se coloca el nombre de nuestra máquina (FreePlastic).

#### Figura 24.

##### *Configuración de la pantalla de inicio*



```

1
2 #if ENABLED(USE_AUTOMATIC_VERSIONING)
3
4 #include "_Version.h"
5
6 #else
7
8 #define SHORT_BUILD_VERSION ""
9 #define DETAILED_BUILD_VERSION SHORT_BUILD_VERSION " (Github)"
10 #define STRING_DISTRIBUTION_DATE "2023-06-20"
11 #define REQUIRED_CONFIGURATION_H_VERSION 010109
12 #define REQUIRED_CONFIGURATION_ADV_H_VERSION 010109
13 #define PROTOCOL_VERSION "1.0"
14 #define MACHINE_NAME "Filamentadora" //Defines a generic printer name to be output to
15 #define SOURCE_CODE_URL "https://github.com/MarlinFirmware/Marlin" //The SOURCE_CODE_
16 #define DEFAULT_MACHINE_UUID "cede2a2f-41a2-4748-9b12-c55c62f367ff" // The WEBSITE_URI
17 #define WEBSITE_URL "FreePlastic"
18
19 #endif // USE_AUTOMATIC_VERSIONING

```

Se guarda y procede a compilar el resultado. Al cargar el firmware de control se obtiene el siguiente resumen; de 256 KB de memoria flash, 8 KB de SRAM y 4 KB de EEPROM se ocupa el 43% del almacenamiento y el 52% de la memoria dinámica, esto último indica un funcionamiento óptimo en plena operación (véase figura 25).

### Figura 25.

*Resumen de compilación en la placa de desarrollo Arduino Mega2560.*

```

Compilado
/Users/macbookpro/Library/Arduino15/packages/arduino/tools/avr-gcc/7.3.0
/Users/macbookpro/Library/Arduino15/packages/arduino/tools/avr-gcc/7.3.0
Usando librería U8glib con versión 1.19.1 en la carpeta: /Users/macbookpro/Library/Arduino15/packages/arduino/tools/avr-gcc/7.3.0
El Sketch usa 105188 bytes (41%) del espacio de almacenamiento de progr
Las variables Globales usan 4315 bytes (52%) de la memoria dinámica, de
  
```

### Configuración del Código G

En este caso y por medio de una tarjeta SD se procede a buscar y seleccionar a través de la perilla de control un archivo con extensión GCODE. EL mismo que se encuentra dotado de las instrucciones *G90 G1 E10 F65 G91*. La letra G de cada instrucción hace referencia a movimientos en este caso iniciamos con un 90 describiendo un sistema de escala absoluto y finaliza con un G91 que es un comando de movimiento incremental. G1 hace referencia a un movimiento controlado y continuo; el comando E10 a un tramo de 10mm y una velocidad de avance de 65mm/s.

El valor de temperatura puede ser procesado por el grupo de instrucciones *M104 S205 T0* y *M109 S205 T0* la primera que resetea el extrusor E0 (T0) y define la temperatura en 205C y la siguiente instrucción es una de espera hasta lograr esa

temperatura para finalmente proceder con el encendido del motor. De requerir aumentar la temperatura únicamente variaremos esas dos instrucciones a 220C por ejemplo que se vería como se muestra en la figura 40. De requerirse aumentar la velocidad angular del motor bastaría con modificar en “bloque” la instrucción F45 por un F100 por ejemplo, se sugiere usar la función reemplazar de cualquier editor de texto para conseguir que todas las instrucciones F45 se modifiquen a F100, así como se ve en la figura 41.

### **Figura 26.**

*Modificación de temperatura en código G.*

```
1 G90
2 M82
3 M104 S220 T0
4 M109 S220 T0
5 ;G28
6 G21 ;metric values
7 G92 E0
8 M106
9 G91
10 G1 E10 F45
```

**Figura 27.**

Modificación de velocidad en código G.

```
1  G90
2  M82
3  M104 S220 T0
4  M109 S220 T0
5  ;G28
6  G21          ;metric values
7  G92 E0
8  M106
9  G91
10 G1 E10 F45
11 G90
12 G91
13 G1 E10 F45
14 G90
15 G91
16 G1 E10 F45
17 G90
18 G91
19 G1 E10 F45
20 G90
21 G91
22 G1 E10 F45
23 G90
24 G91
25 G1 E10 F45
```

## Etapa de Adecuación e Interconexión de Hardware

### La pantalla y opciones de inicio.

Tras cargar y correr el firmware, la pantalla de inicio mostrará capturas del sensor, así como se visualizará de la posición relativa X, Y y Z como se muestra en la figura 17. La perilla integrada será el medio de navegación entre opciones. Además, cuenta con un botón de reset y un indicador sonoro Buzzer activado en cada selección.

### Figura 28.

*Pantalla de inicio del sistema*



*Nota:* Representa la pantalla para inicio que crea interfaz. Tomada del proyecto

Cada vez que se seleccione una opción en pantalla existirá una alerta sonora. Esta puede aparecer al seleccionar, deslizar (al rotar la perilla) y al salir atrás. Los pines más importantes de esta placa son P1 de GND, P2 de voltaje (5V típicamente). El P3

que ajusta el contraste, P4 (D/I), P5(R/W), P6 (E), P7(8 bits), PIN15 -P16 (CS) son parte de la comunicación SPI desde el controlador hasta la pantalla.

**Figura 29.**

*Pantalla de operación en GLCD 128x64*



*Nota:* Representa la pantalla de operación del proyecto

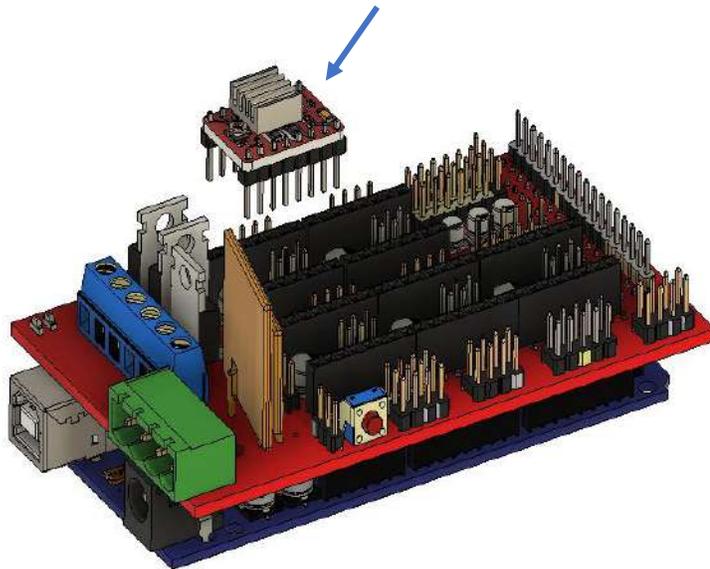
**Requerimientos de Potencia Eléctrica.**

La alimentación del sistema lo provee una fuente de 33A a 12V. La placa Shield RAMPs se encuentra preparada para soportar y regular este suministro. En las entradas V+ y V- se encuentran aisladas por pares. La línea de 11A proporcionar la intensidad para el bloque de calefacción y movimientos de motor mientras que la línea de 5A alimentará el controlador y sensor e incluso un ventilador; tal como se verá en el apartado de conexiones eléctricas.

La interfaz **HEAT 1** es donde se conecta la resistencia del bloque de calefacción; esta salida está preparada con un sistema de activación MOSFET. El señalado **Termistor de hotend** es donde se conecta el sensor del sistema (sin polaridad) que retroalimenta a la niquelina (en T0). Todo el último conjunto definido como **Adaptador de LCD** es donde se conecta la pantalla GLCD. El motor Nema 17 se conecta al puerto **E0 (extrusor 0)**. Esto último especifica una correcta polarización del driver A4988 y esto es como se ve en la figura 28.

### Figura 30.

*Driver A4988. Posicionamiento en la placa Ramps 1.4*



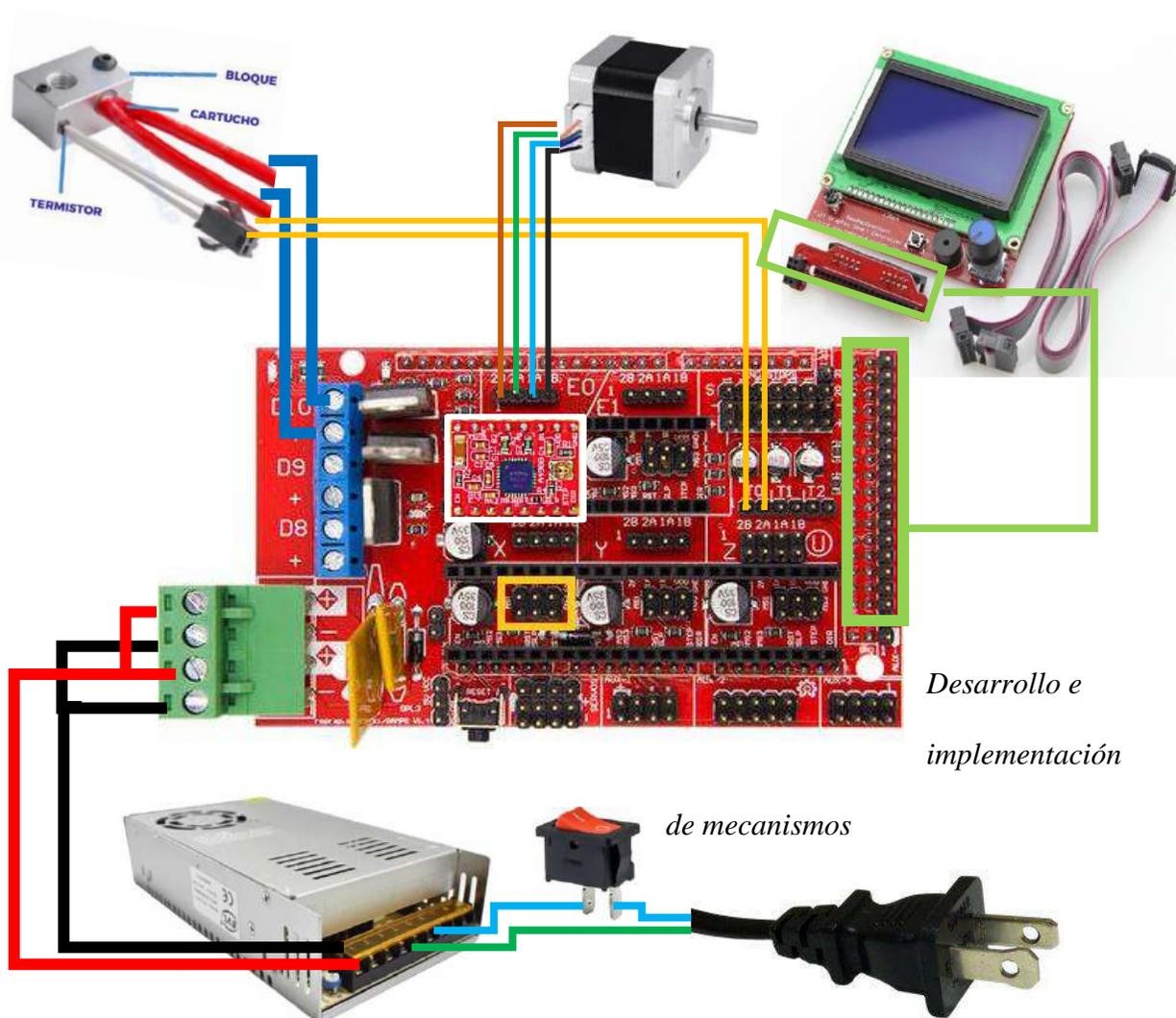
### Esquema de Conexiones Eléctrico

En el diagrama de la figura 29 se muestran las conexiones físicas a la Shield Ramps. Desde una perspectiva eléctrica, todo comienza desde la toma de corriente proponiendo un sistema de 110VAC. Estos son transformados a 12VDC por medio de la fuente de alimentación que puede suministrar unos 33A. El prototipo requiere unos 15 A

para un funcionamiento integral y distribuyen para todos los sensores, actuadores, pantalla LCD y para el microcontrolador. El controlador captura medidas del sensor y se da la condición activa el proceso de enrollamiento mientras nos encontramos calentado la cinta por medio del bloque. Todos los procesos se visualizan en pantalla.

**Figura 31.**

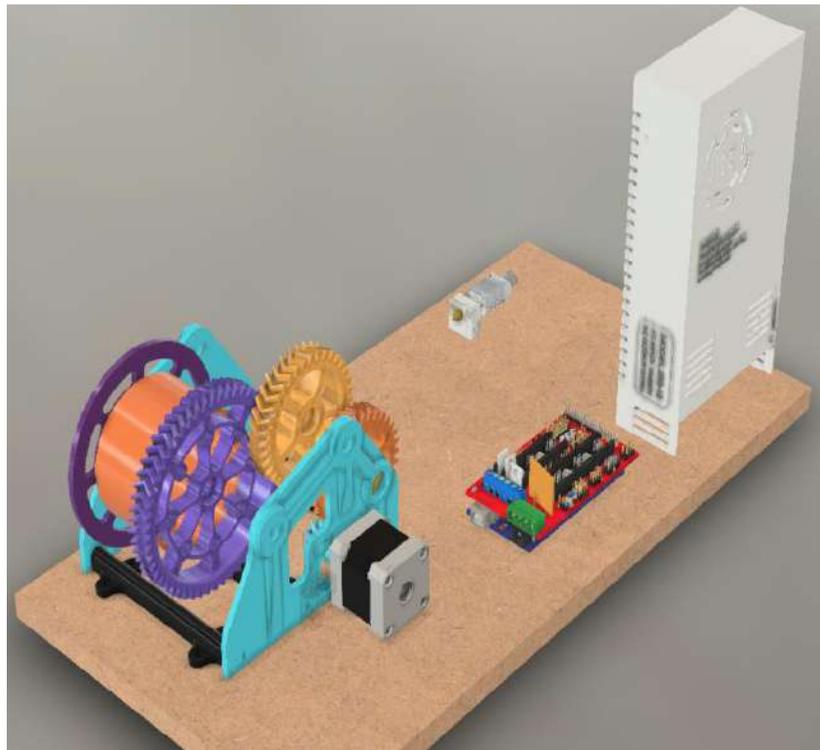
*Diagrama de conexiones del sistema Freeplastic.*



Poseer un modelo levantado en 3D permite tener una mejor óptica del resultado final. Así se procedió a diseñar el sistema de enrollamiento en el software Fusion360. Aunque el motor tenga una capacidad de efectuar de 4kg/cm es necesario prever posibles mayores torques que eviten su sobrecalentamiento; por lo que se efectúa una conexión a sistema de engranajes que permitirá la reducción de velocidad y el aumento del torque. Inicialmente se conecta un engranaje de adaptación de 7 dientes. Seguido se conecta con un engranaje mayor de 32 dientes lo que produce una reducción de velocidad de casi 4 veces. Se le añade un nivel más de 12 dientes a 36 con lo que se consigue triplicar esta reducción. Finalmente, un nivel más de 11 a 55 con lo que se quintuplica el resultado. En total se obtiene una reducción 1 a 60.

**Figura 32.**

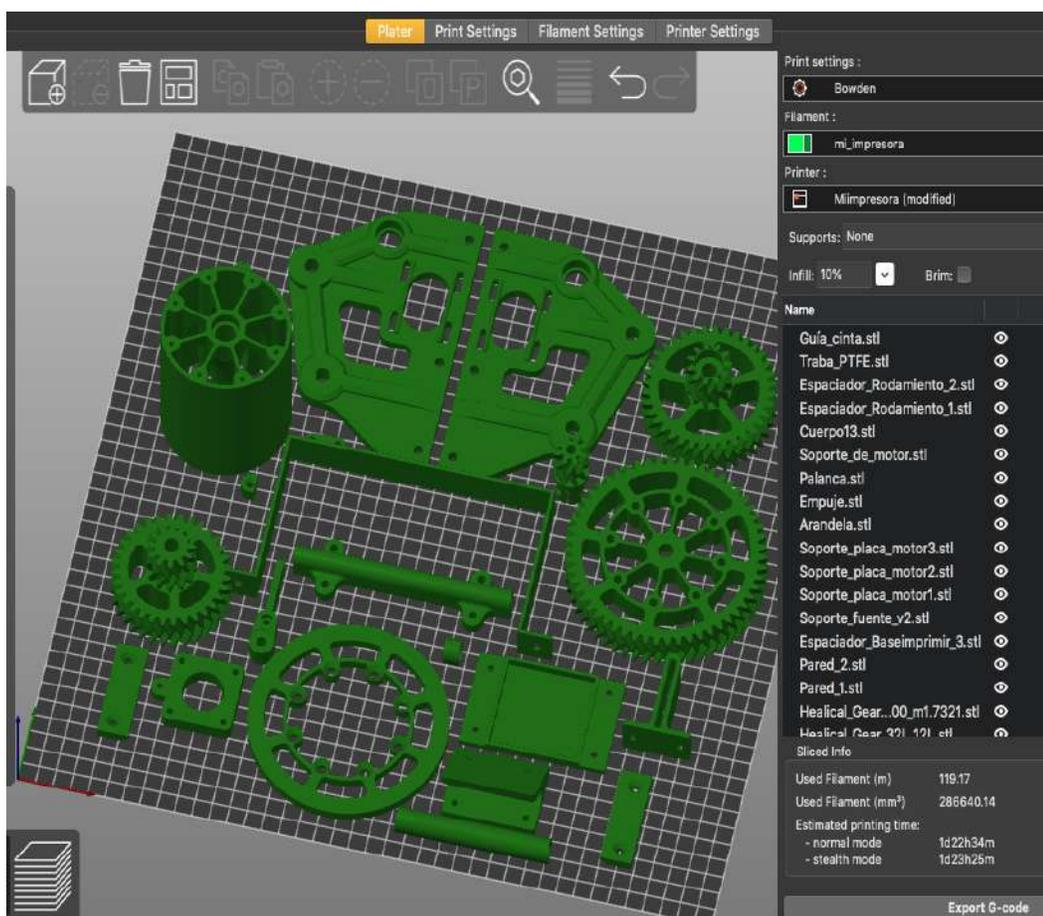
*Levantamiento 3D del prototipo.*



Las partes móviles como el sistema de engranajes se implementaron por medio de tecnología de impresión 3D con lo que el sistema de enrollado prácticamente depende de este paso previo. Como se puede ver en la figura 32 el tiempo de impresión estimado para estas piezas es de 47 horas aproximadamente. El plástico utilizado es PLA+ pues posee una resistencia básica y lo suficiente para este proyecto.

### Figura 33.

*Implementación del sistema de enrollado. Especifica 47 h de impresión 3D.*



El resultado tras el proceso de impresión 3D se visualiza en la figura 33. El ensamblaje completo se lo puede visualizar en las pruebas de funcionamiento así mismo se acompaña de una imagen en el software de diseño por presentar mejor óptica del armado.

**Figura 34.**

*Impresión 3D del sistema de engranajes.*



*Nota:* Representa los engranajes previamente diseñados y posteriormente impresos en 3D proyecto.

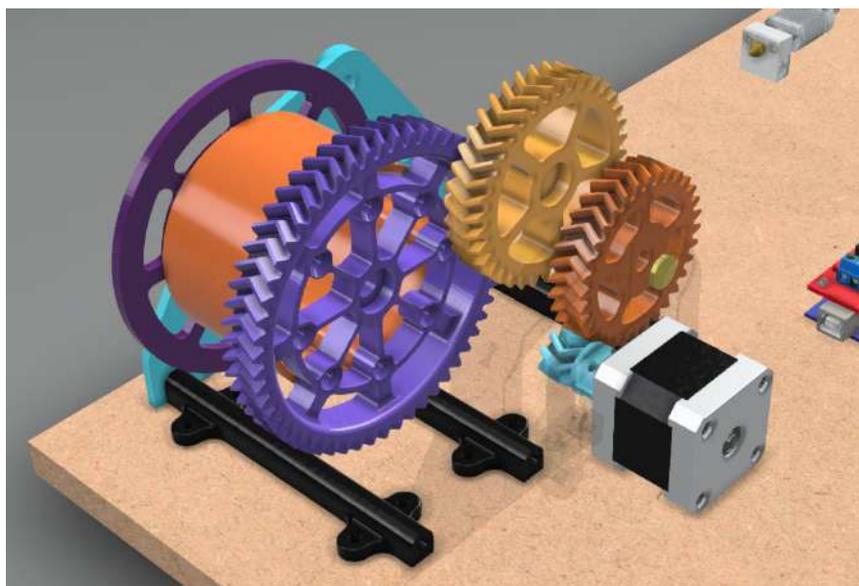
Otro mecanismo que requiere de adaptación es el bloque calefactor tradicional. Usualmente este posee una rosca para un barrel M6. A esta se le realiza un biselado que permite el ingreso de la cinta. Seguido de ello a la boquilla estándar de 0.4mm también se debe agrandar el agujero a 1.5mm o cualquier otro dependiendo la finalidad del filamento a obtener.

**Figura 35.**

*Bloque calefactor adaptado para extruir filamento de 1.5mm.*

**Figura 36.**

*Ensamble del sistema de reducción por engranajes.*



*Nota: Montaje y armado de engranajes del proyecto.*

## Funcionamiento general del prototipo

### Arquitectura del sistema

El prototipo se compone de tres sistemas interconectados en una línea de producción, donde cada uno depende del anterior. El sistema mecánico, en primer lugar, tiene la capacidad de cortar las cintas en función de la altura de la cuchilla de corte. Luego, el sistema de calefacción y embobinado se activa tras encender la fuente de alimentación, que suministra los voltajes necesarios para los distintos actuadores, sensores y controladores. La pantalla GLCD muestra lecturas de temperatura y posición de forma predeterminada, así como el flujo de material y el tiempo de operación.

La operación se inicia únicamente cuando se lee una tarjeta SD que contiene los datos de temperatura y velocidad en formato .GCODE. Una vez seleccionado el archivo, el prototipo inicia un proceso de calentamiento. Una vez que la temperatura programada se alcanza, el motor entra en funcionamiento. En este punto, se espera que la botella esté preparada y embonada a través del bloque de calefacción, además de anclada al tambor del sistema de embobinado.

### Figura 37.

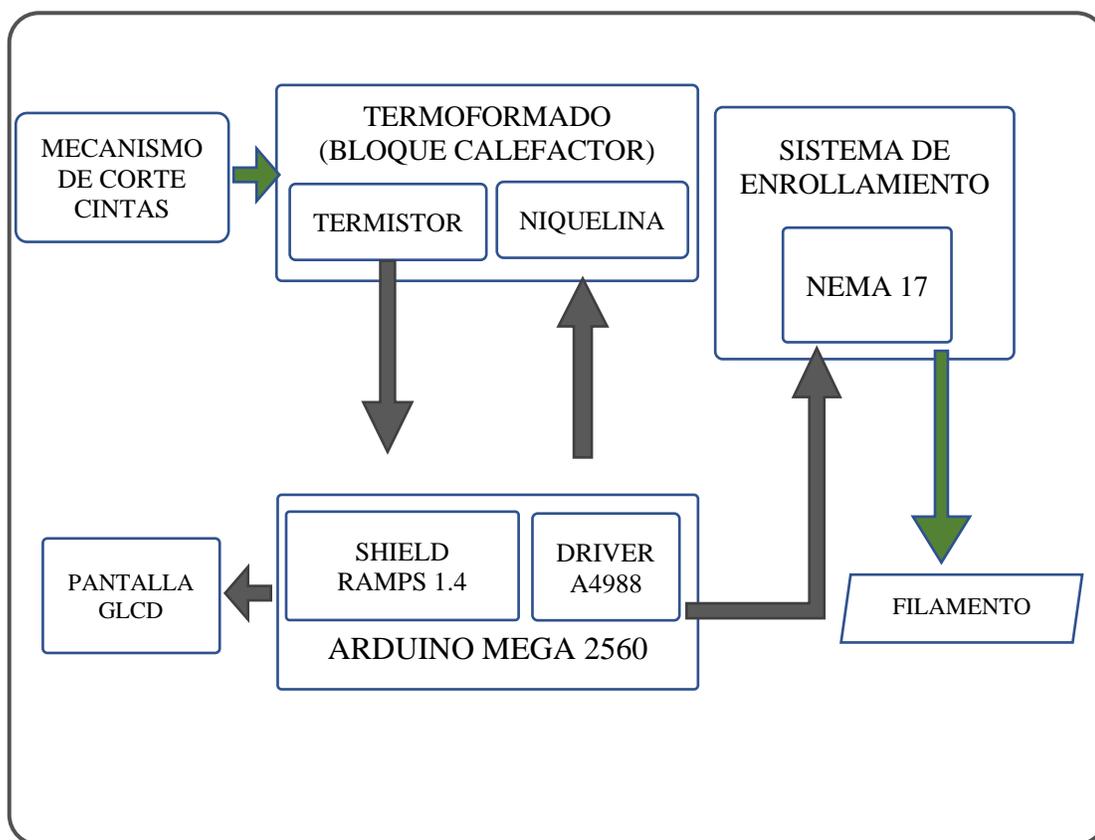
Pantalla de operación de máquina filamentadora.



Para una comprensión más clara del sistema y su funcionamiento, hemos organizado una subdivisión de sistemas y distribuido los componentes que los componen. Como se muestra en la figura 37, la arquitectura incluye los siguientes un *bloque de calefacción* que emplea una resistencia térmica y posee una trote alimentación por medio de un termistor; *bloque de enrollado*, encargado de controlar un motor Nema 17; y un *bloque de monitoreo* que se presenta el medio de seguimiento y la visualización de datos mediante una pantalla GLCD. Todos estos componentes se conectan al Arduino Mega2560 a través del Shield RAMPs, lo que facilita la comunicación y coordinación entre ellos.

**Figura 38.**

Arquitectura del sistema.



## Proceso general

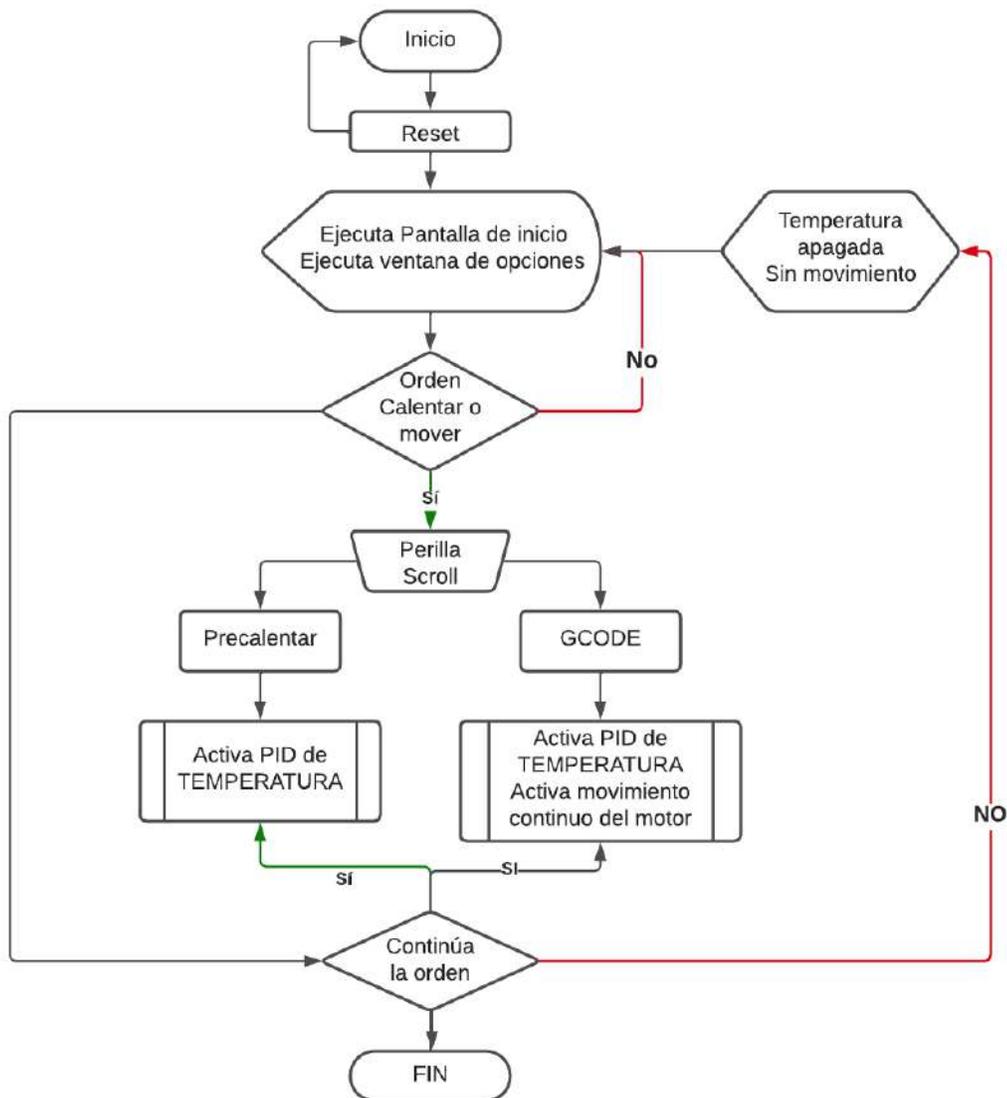
En esta sección, nos enfocaremos en detallar el proceso de adquisición (a través del sensor de temperatura) y el proceso de activación (del motor paso a paso y la niquelina). Es importante destacar que el firmware podría ofrecer funcionalidades adicionales que, en este momento, no consideraremos y que podrían considerarse como mejoras futuras.

Cada proceso de activación está respaldado por su propia subrutina de control, la cual utiliza un controlador PID. Como se ilustra en el diagrama de la figura 38, la pantalla muestra un mensaje de "Inicialización Correcta" o "Filamentadora Lista" junto con los valores de temperatura actuales detectados. El proceso comienza al seleccionar la orden de precalentar el bloque de extrusión, lo que se puede hacer utilizando el periférico de scroll o seleccionando un archivo GCODE con la configuración previamente guardada.

A modo de ejemplo, si el GCODE preconfigurado establece una temperatura objetivo de 200°C, el sistema se esforzará por alcanzar y mantener esa temperatura a lo largo del tiempo. Una vez que se alcanza la temperatura adecuada, se activa el motor para comenzar el proceso de embobinado. Este proceso se repite hasta que se detecte la ausencia de suministro o se cancele el proceso. El motor no se activará si la temperatura requerida no se ha alcanzado. En caso de cancelación del proceso se retorna a la pantalla de inicio se apaga el motor y la temperatura comienza a descender.

**Figura 39.**

*Flujograma de proceso de máquina filamentadora.*



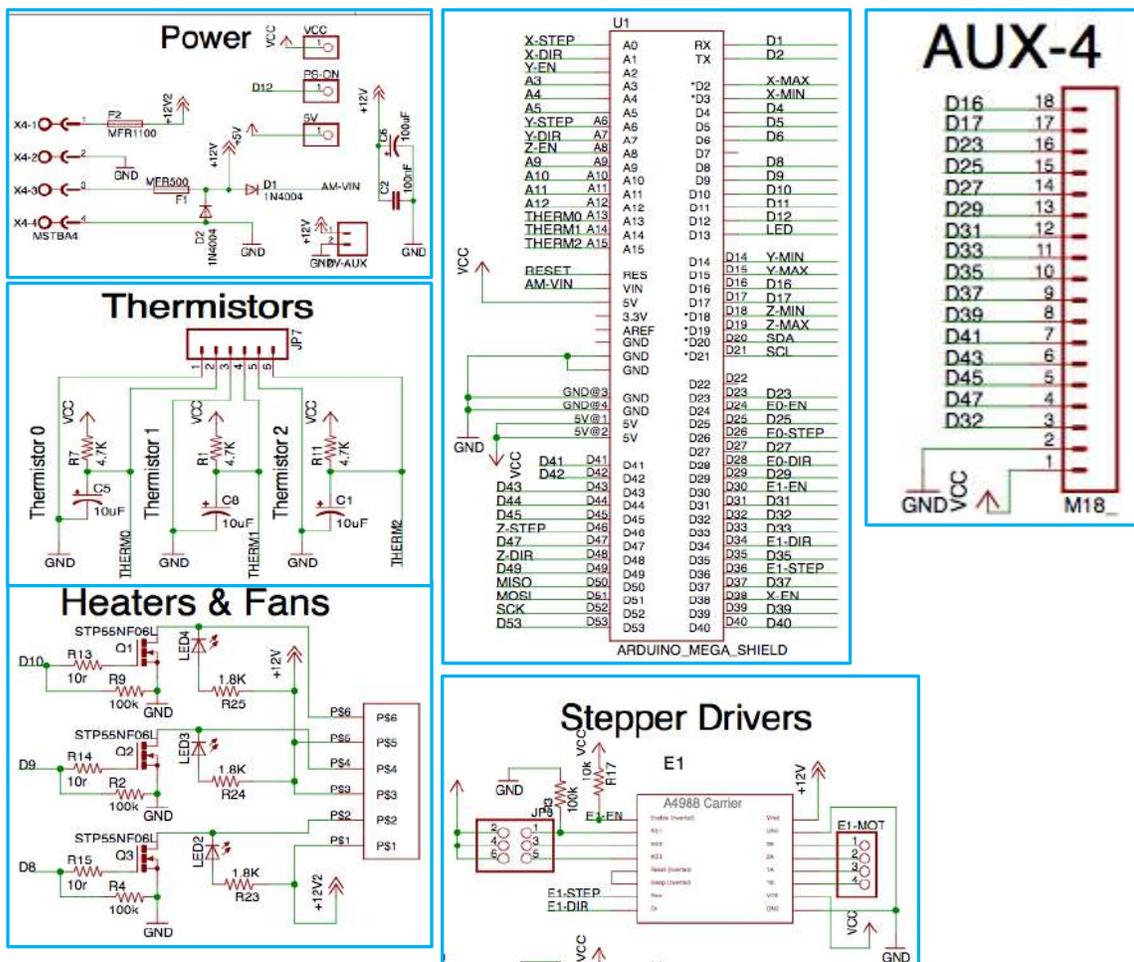
### ***Esquema de conexiones electrónico***

El diagrama de la figura 40 ilustra las conexiones hacia la placa de desarrollo Arduino Mega 2560 a través de la Shield Ramps 1.4, facilitando una organización ordenada de los periféricos del sistema. Está dividido en los siguientes bloques:

"POWER" para la regulación de voltaje, "TERMISTOR" para la adaptación del sensor de temperatura, "HEATERS" para el suministro de potencia al bloque calefactor, "STEPPER DRIVER" que conecta con el motor Nema 17, y "AUX-4" para la interfaz que enlaza la pantalla GLCD con el Arduino MEGA.

**Figura 40.**

*Diagrama electrónico.*



Imágenes tomadas de: <https://acortar.link/qMpavh>

## Pruebas de Funcionamiento y Resultados

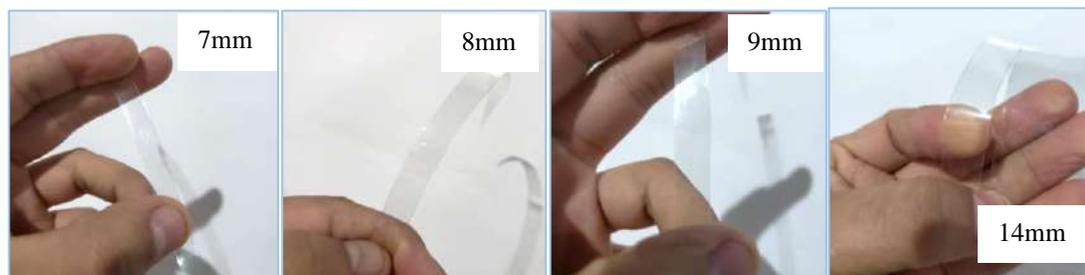
### Prueba de Funcionamiento

Se especificó inicialmente cuatro ensayos utilizando cintas de diversos anchos que se cortan inmediatamente después de embobinar el filamento. Las dimensiones de las cintas empleadas son 7 mm, 8 mm, 9 mm y 14 mm. El objetivo principal era lograr que el diámetro del filamento resultante se asemeje al de la boquilla, que es de 1.5 mm.

Durante estos experimentos, se configuró la temperatura de termoformado del bloque o boquilla en 205°C, y se ajustó la velocidad de embobinado a 70 mm/s. La configuración de la velocidad se especifica en el código G, que se almacena en la tarjeta SD y se lee posteriormente por la máquina. Estos ensayos se denominan '*Prueba de Extrusión y Fusión*'. Al concluir cada proceso, se llevó a cabo una evaluación visual conocida como '*Prueba de Calidad del filamento resultante*', en la que se verificó la uniformidad, la presencia de rugosidades, imperfecciones, así como la detección de impurezas o burbujas en el filamento."

#### Figura 41.

*Cintas de PET llevadas a ensayar en las pruebas de extrusión y fusión.*



También, se estableció la *Prueba de control de velocidad* y temperatura donde se evaluó la comodidad de aumentar o disminuir tanto en temperatura como en velocidad de forma muy sencilla y al mover la perilla de opciones justo debajo de la pantalla, esto en caso de requerirse calibrar esos parámetros directamente.

Se ejecutó, además, la *Prueba de alimentación del sistema*, donde se usaron cuchillas estándar para “cúter”. Se propuso cortar 20m de material y evaluar el corte inicial y final de manera que el ancho de la cinta sea homogéneo y no presente rebabas por el desgaste de la mencionada cuchilla.

Para finalizar, y con el objetivo de que el producto final fuese de propósito general fue requerible contar con una noción respecto a las condiciones mecánicas de resistencia. Así se estableció la *Prueba de resistencia mecánica*, donde se evaluó el filamento en función al peso que resiste al ser tensionado. Además, se verificó que si se produce alguna elongación por lo que se marca 6cm con un marcador azul. Para esta prueba se usaron materiales domésticos como una balanza, y se generó pesos de referencia con botellas llenas de arena. El filamento se sujetaba a una prenda de banco; con 50cm de largo se anclaba un extremo y se cargaba el otro extremo.

Cada prueba realizada y los resultados que a continuación son descritos se efectuaron por medio de botellas lisas entre 1 a 3 litros de botellas estándar. Se aclara que el procesamiento del plástico se debe dar con envases previamente preparados limpios, cortados el haciendo e iniciados unos centímetros en corte para ser colocados en la entrada de corte del sistema donde también es foco de atención la capacidad del prototipo para una “*Producción continua*”.

**Figura 42.**

*Botella estándar de 2 litros y su incorporación al sistema.*



*Nota:* Prueba para iniciar el corte. Tomado del proyecto máquina.

## **Resultados**

Por una botella estándar lisa de 2 litros se obtuvieron entre 20g y por una de 3 litros 30g de filamento donde no se ocupó la base ni la boquilla de cada botella. El segmento de botellas restante, así como botellas rugosas presentaron un nuevo reto en procesamiento que no es abordado en el presente trabajo.

Las *Prueba de Extrusión y Fusión* con la boquilla de 1.5mm (205°C a 70 mm/s) arrojaron 4 resultados en la morfología del producto. Puntualizando, una cinta de 7mm termo-forma un filamento cuya sección transversal describe un perfil U. En el caso

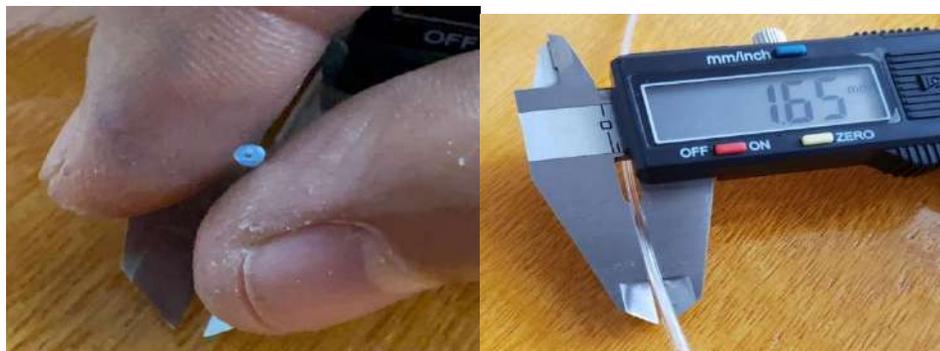
de la cinta de ancho 8mm el perfil termoformado resultante es una C. Cuando se prueba un ancho de cinta de 9mm a 10mm el resultado fue un perfil tipo O. con lo que este último es perfectamente recomendable con filamento de propósito general. En el caso de la cinta de 14mm para este sistema no se recomienda su uso, pues demanda demasiada potencia para el motor y bajo las condiciones especificadas no fue posible termo formar el material.

El diámetro obtenido de este filamento de perfil de O es de 1.7mm que guarda cierta diferencia con la boquilla de 1.5mm y esto fue porque el plástico vitrificado tiende a expandirse a la salida de la boquilla por lo que se procedió a implementar un sistema de ventilación tipo turbina.

### **Figura 43.**

Perfiles y diámetro de impresión obtenidos tras la termo formación.





**Perfil tipo O  
9 a 10 mm**

**Diámetro  
resultante.**

Por cuando a la *Prueba de Calidad del filamento resultante* se caracteriza únicamente el filamento obtenido a 9mm. Como la botella se encontraba limpia y se aprovechó únicamente su parte lisa no presentó rugosidades o heterogeneidad, dado que no se fundió la materia prima resulta improbable añadir burbujas. Por otro lado, y por la existencia de polvo o restos de material residual si fue posible añadir impurezas de material carbonizado. Esta prueba estaba seriamente ligada respecto a la observación de la *Prueba de alimentación del sistema* establecida pues, una cuchilla desgastada presenta pequeñas rebabas en el producto, así como un aumento en el requerimiento mecánico por la falta de uniformidad. Con una cuchilla estándar sin uso y con una prueba de corte de 20m no se vio afectaciones en el corte de cintas ni en la integridad del perfil O.

#### **Figura 44.**

Pruebas de calidad y de alimentación del sistema.

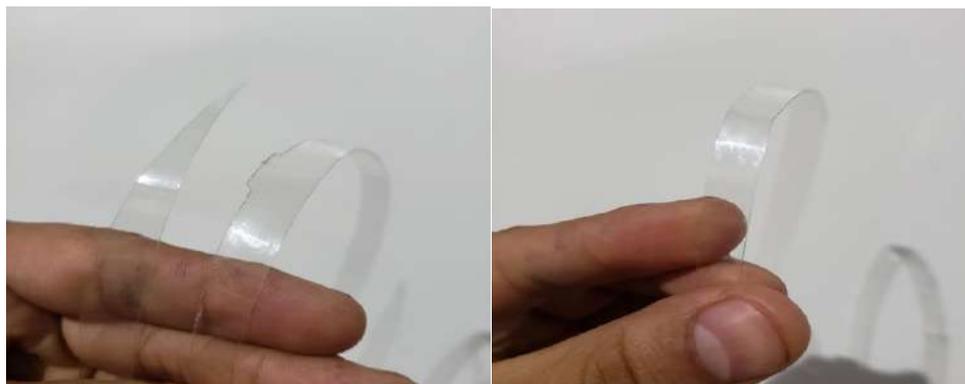


En cuanto a irregularidades se menciona que presentaron dificultades al enrollarlo y al mantener un ancho de cinta constante. En el caso de botellas lisas y con los mismos parámetros un ancho de cinta superior a 10mm y con cierto cambio en su ancho provoca que el motor no consiga extruirlo. Otro posible fallo para una *Producción continua* implicó que el material obtenido de una botella tiene una extensión de unos 9 metros es por lo que también fue necesario implementar un medio de fusión de extremos de cintas así se obtuvo la materia prima sin cortes.

Finalmente, y bajo el razonamiento se pudo evidenciar que: tanto la fuente se encontró con un sistema de control de temperatura que activó su propia ventilación, así como el prototipo mismo que estaba controlado por un lazo retroalimentado de temperatura, el sistema en teoría pudo mantenerse funcionando de manera indefinida. Aunque se recomienda que en caso de existir un operador a cargo la maquina trabaje bajo este mismo lapso evitando fallos por ausencia del componente humano (no es una maquina automática).

**Figura 45.**

*Botella estándar limpia y lista para ser procesada.*



Para *las pruebas de control de velocidad* se partió del hecho que la velocidad se encuentra en mm/s aunque su aumento o disminución se presentaron en porcentaje, es decir, si configuramos el código G para que se dé movimiento constante de 100mm/s al colocar el marcador en 125% la velocidad aumentó a 125mm/s de otra forma si se configura a 50mm/s y se realiza el mismo porcentaje de incremento la velocidad aumentaría a 62,5m/s. En el caso de la temperatura la escala es unitaria y al escoger la opción **TUNE** y modificar **Nozzle** podremos con el movimiento del Scroll o perilla aumentar o disminuir esta. No se recomienda disminuir la temperatura de a menos de 200C pues el motor tiende a realizar sobre torque y su movimiento se ve afectado.

Como este filamento será un producto de uso general se vio la necesidad de tener una noción respecto a la resistencia mecánica del mismo (*Prueba de resistencia mecánica*). Así detallamos que usando una balanza SI se midió porciones de arena en kilogramos. Se cargo el primer bidón con 9kg. En ese punto el filamento no presenta elongación o alguna falla estructural.

#### **Figura 46.**

*Botella estándar limpia y lista para ser procesada.*



*Nota:* Botella previo a ser procesada.

Para los 16kg el sedal aún no se mostró elongación y tampoco ninguna falla estructural como micro-rupturas o cortes. Aumentando 4K más es cuando comenzaron a aparecer signos de elongación, es decir, una parte del marcado comienza a volverse translucido. Esta zona es de alrededor de 4mm

**Figura 47.**

*Pruebas de resistencia mecánica: Primeros signos de elongación.*



*Nota:* Sedal en prueba de elongación.

Desde aquí en adelante el aumento de peso fueron progresivos de 1k en 1k hasta llegar a los 28K donde el filamento colapsó y mostró su punto máximo de corte. Aunque no existió una elongación mayor a 6mm el sedal mostró un corte brusco al suscitarse ese peso. Se concluyó que el producto podría resistir hasta 20kg de sometimiento mecánico (unos 196N).

**Figura 48.**

*Pruebas de resistencia mecánica: Punto de corte y colapso.*



*Nota:* Pruebas aplicada en el proyecto para verificar su resistencia mecánica.

## Conclusiones

Se consiguió levantar un diseño de una máquina termo formadora de cintas de botellas plásticas PET, a partir de la personalización del firmware Marlin y hardware de una impresora 3D doméstica, además el diseño del sistema permite el corte con la fuerza del enrollamiento. Se pudo obtener un sistema que consume 43% del almacenamiento y el 52% de la memoria dinámica del Atmega 2650 con una capacidad para controlar velocidad de embobinado y de un control de temperatura de laso cerrado en un conjunto termistor/resistencia eléctrica.

La pantalla personalizada presenta una interfaz para que el usuario por medio de un enconder rotativo pueda navegar y seleccionar entre las distintas opciones otorgadas por el firmware Marlin. Las opciones por seleccionar son códigos G preconfigurados.

Por medio de mecanismo de espaciadores y una hoja de cutter se consiguió implementar un sistema mecánico para el corte homogéneo de cintas mientras el sistema se encuentra enrollando el filamento extruido. Estos espaciadores son cilindros con una altura que pueden ir desde los 7mm hasta los 10mm dentro del sistema.

En la evaluación del prototipo final se constató que tras la termo formación y aun con una boquilla de 1.5mm el plástico tiende a dilatarse por lo que el diámetro final del filamento llega a enfriarse a 1.65mm logrado al colocar un ventilador a la salida. En cuanto a la velocidad de mejores prestaciones se podría mencionar unos 200m/s a una temperatura de 220C para conseguir mencionado diámetro de filamento.

## Recomendaciones

Se recomienda la construcción de una placa especializada bajo los requerimientos del presente proyecto. El uso de un Arduino Mega, así como de toda una Shield RAMPs implica un 40% de hardware desperdiciado. Además, se debe utilizar una fuente de alimentación superior a 15 A de 12V pues la Shield RAMPs lo requiere.

Conservar la uniformidad en la cinta a termo formar. Caso contrario el diámetro, así como el sistema mecánico y luego el suministro eléctrico se verán afectados pues anchos superiores a 1cm implican mayor requerimiento de torque, así como de corriente.

Mantener el motor a pasos desconectado mecánicamente del sistema y una vez en operación deslizar hacia el sistema de reducción de velocidad. Un arranque brusco podrá dañar el sistema mecánico del prototipo, así como implica un pico alto de corriente.

Cuando el motor a pasos se encuentra alimentado, así como conectado al sistema mecánico, no es recomendable jalar el filamento pues esto puede ocasionar que el driver del motor se extralimite, también puede producir voltajes de fuerza contraelectromotriz que se devuelve al resto de la electrónica del sistema.

## Bibliografía

- Abad, J. (2023). Desarrollo De Un Prototipo Robótico Apoyado Con Tecnología Holográfica 3d Para Aplicaciones Publicitarias En El Período Octubre 2022 – Febrero 2023.  
<http://dspace.tecnologicosudamericano.edu.ec/jspui/bitstream/123456789/767/1/TESIS%20OLIVER%20ABAD%20J.pdf>
- Arduino CC. (2023). *Arduino Mega 2560 Rev3*. Arduino Online Shop.  
<https://storeusa.arduino.cc/products/arduino-mega-2560-rev3>
- Arduino CC. (2023). *About Arduino*. <https://www.arduino.cc/en/about>
- Arrieta, E. (2022a). *Diferencia entre método inductivo y deductivo*. Diferenciador.  
<https://www.diferenciador.com/diferencia-entre-metodo-inductivo-y-deductivo/>
- Arrieta, E. (2022b). *Los 15 métodos de investigación*. Diferenciador.  
<https://www.diferenciador.com/metodos-de-investigacion/>
- Blanco, F. (8 de Octubre de 2012). *Materiales poliméricos y compuestos. Materiales poliméricos de mayor interés industrial. Obtenido de sitio web de Materiales poliméricos de mayor interés industrial*.  
[https://www.upv.es/materiales/Fcm/Fcm15/fcm15\\_3.html](https://www.upv.es/materiales/Fcm/Fcm15/fcm15_3.html)
- CAUCE. (2022). Economía circular. CAUCE. <https://cauce.eu/economia-circular/>
- Castro, J. A. (2021). Metodología Proyectual De Diseño Para Impulsar La Educación Primaria Rural En Chile. *Interciencia*, 46(6), 265-271.  
<https://www.redalyc.org/journal/339/33968022005/movil/>
- Chemical Safety Facts. (2023). *¿Qué es el poliestireno? | Usos, beneficios e información*

- sobre la seguridad. <https://es.chemicalsafetyfacts.org/es/poliestireno/>
- CERSA. (2023). *CERSA - Fabricante de Láminas Plásticas PET*. CERSA - Fabricante de Láminas Plásticas PET. <http://cersa.ec//>
- COMPU TORRE FUERTE. (2023). *Kit Impresora 3d Bloque Aluminio Garganta Boquilla Mk8*. <https://computorrefuerte.mercadoshops.com.mx/MLM-1305757752-kit-impresora-3d-bloque-aluminio-garganta-boquilla-mk8-175-JM>
- CREATEC4 S.C.A. (2023). *Fuente de alimentación 12V 30A Electrónica Productos- Createc 3D Shop*. CREATEC4 S.C.A. <https://createc3d.com/es/comprar-electronica-impresora-3d/24-comprar-fuente-de-alimentacion-12v-30a-precio-oferta.html>
- Decoesfera. (2020). *La fibra para relleno de edredones y almohadas hecha con plástico reciclado ayuda a cuidar el océano*. Decoesfera. <https://decoracion.trendencias.com/dormitorio/fibra-para-relleno-edredones-almohadas-hecha-plastico-reciclado-ayuda-a-cuidar-oceano>
- Elias, X., & Jurado, L. (2012). *Reciclaje de residuos industriales*. Madrid: Dias de Santos S.A.
- Edo, J. A. (2021, enero 12). Impresión 3D de pellets y Economía Circular | IT3D. *IT3D Group*. <https://it3d.com/impresion-3d-economia-circular/>
- EcuRed. (14 de 12 de 2019). EcuRed. Obtenido de [https://www.ecured.cu/Loja\\_\(Ecuador\)\\_enciclopediaonline](https://www.ecured.cu/Loja_(Ecuador)_enciclopediaonline). (13 de 11 de 2018). Obtenido de [enciclopediaonline](https://www.ecured.cu/Loja_(Ecuador)_enciclopediaonline): <https://enciclopediaonline.com/es/ensayo-y-error/>
- Fabbri, M. S. (2020). *Las técnicas de investigación: La observación*.

<http://institutocienciashumanas.com/wp-content/uploads/2020/03/Las-t%C3%A9cnicas-de-investigaci%C3%B3n.pdf>

Flores, D. F. S. (2017). *Degradación de materiales plásticos “PET” (polyethylene terephthalate), como alternativa para su gestión.*

Fuster Guillen, D. E. (2019). Investigación cualitativa: Método fenomenológico hermenéutico. *Propósitos y Representaciones*, 7(1), 201. <https://doi.org/10.20511/pyr2019.v7n1.267>

Guillen, F., & Elida, D. (2019). Investigación cualitativa: Método fenomenológico hermenéutico. *Propósitos y Representaciones*, 7(1), 201-229. <https://doi.org/10.20511/pyr2019.v7n1.267>

Hackduino. (2023). *Ejemplo Básico De Termistor NTC Y Arduino*. Instructables.

<https://www.instructables.com/Ejemplo-básico-de-termistor-NTC-y-Arduino/>

IDEA161. (2022, noviembre 3). *10 buenas razones para utilizar el Firmware de Marlin.*

En Amezcua, M. (2015). La Búsqueda Bibliográfica en diez pasos. *Index de Enfermería*, 24(1-2), 14-14. <https://doi.org/10.4321/S1132-12962015000100028>

KolMitE (2022, diciembre 8). *Firmware de la impresora 3D Marlin*. Recuperado 31 de mayo de 2023, de <https://kolmite.com/firmware-de-la-impresora-3d-marlin/>

López, N., & Sandoval, I. (2016). *Métodos y técnicas de investigación cuantitativa y cualitativa.*

<http://biblioteca.udgvirtual.udg.mx/jspui/bitstream/123456789/176/3/M%C3%A9todos%20y%20t%C3%A9cnicas%20de%20investigaci%C3%B3n%20cuantitativa%20y%20cualitativa.pdf>

- Martínez, M. (2014). *Ciencia y arte en la metodología cualitativa*. (2ª ed.). México: Trillas.
- MAXI PET. (2020). *Pellet y el procesamiento de envases de PET*. Maxipet. <https://maxipet.net/blog/pellet-y-envases-de-pet>
- Maya, E. (2014). *Métodos y técnicas de investigación*. [http://www.librosoa.unam.mx/bitstream/handle/123456789/2418/metodos\\_y\\_tecnicas.pdf?sequence=3&isAllowed=y](http://www.librosoa.unam.mx/bitstream/handle/123456789/2418/metodos_y_tecnicas.pdf?sequence=3&isAllowed=y)
- Miño, R., & Molina, R. (2019). *Estudio de mercado para determinar la producción y comercialización de filamentos para impresión 3D a base de Polietileno Tereftalato (PET) en la ciudad de Guayaquil*. <https://www.dspace.espol.edu.ec/bitstream/123456789/52960/1/T-111214.pdf>
- Miranda, D. (2023, enero 10). *20 datos sobre el problema del plástico en el mundo*. Nacional Geographic España. [https://www.nationalgeographic.com.es/mundo-ng/20-datos-sobre-problema-plastico-mundo\\_15282](https://www.nationalgeographic.com.es/mundo-ng/20-datos-sobre-problema-plastico-mundo_15282)
- Muñoz, L. (2012). *Estudio del uso del polietileno tereftalato como material de restitución en suelos de baja capacidad de carga*. México: Universidad Nacional Autónoma de México.
- Municipio de Loja*. (2017). *Planta de reciclaje aumenta nivel de producción*. Recuperado 30 de mayo de 2023, de <https://www.loja.gob.ec/noticia/2016-11/planta-de-reciclaje-aumenta-nivel-de-produccion>
- ONU. (2018, agosto 24). *Un problema doble: El plástico también emite potentes gases de efecto invernadero*. UNEP. <http://www.unep.org/es/noticias-y-reportajes/reportajes/un-problema-doble-el-plastico-tambien-emite-potentes->

gases-de

Suasnavas D. (2017). *Degradación de materiales plásticos “PET” (polyethylene terephthalate), como alternativa para su gestión*. PUCE Quito  
<http://repositorio.puce.edu.ec/bitstream/handle/22000/13224/Degradación%20Qu%20de%20materiales%20plásticos%20de%20PET.pdf?sequence=1&isAllowed=y>

Tecnológico de Monterrey Blog. (2022). *Poka-Yoke, un método a prueba de errores*.  
<https://blog.maestriasydiplomados.tec.mx/poka-yoke-un-metodo-a-prueba-de-errores>

*Planta de reciclaje aumenta nivel de producción / Municipio de Loja*. (2018). Recuperado 30 de mayo de 2023, de <https://www.loja.gob.ec/noticia/2016-11/planta-de-reciclaje-aumenta-nivel-de-produccion>

Red emprenderverde: *B-PET Filament, desarrollo de filamento de PET reciclado para impresoras 3D*. (2022).  
<https://www.redemprenderverde.es/pg/entrevistas/admin/read/66415/bpet-filament-desarrollo-de-filamento-de-pet-reciclado-para-impresoras-3d>

RepRap ORG. (2023). A4988—*RepRap*. <https://reprap.org/wiki/A4988>

RepRap ORG. (2023). *RAMPS 1.4/es*—*RepRap*. [https://reprap.org/wiki/RAMPS\\_1.4/es](https://reprap.org/wiki/RAMPS_1.4/es)

RepRap ORG. (2023). *RepRapDiscount Full Graphic Smart Controller*—*RepRap*.  
[https://reprap.org/wiki/RepRapDiscount\\_Full\\_Graphic\\_Smart\\_Controller](https://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller)

Segarra-Morales, Galina, Torres-Gutiérrez, Mercedes, & Roldán, Claudia González. (2021). *Sistema Verde Urbano de Loja como base estructurante de la ciudad*. *Estoa*. Revista de la Facultad de Arquitectura y Urbanismo de la

Universidad de Cuenca, 10(20), 81-102. <https://doi.org/10.18537/est.v010.n020.a05>

Teefactory. (2023). PET reciclado, una segunda vida para el poliéster. <https://teefactory.es/sostenibilidad/poliester-reciclado>

Vitofeli. *Qué es Marlin y para qué sirve*. (2023). Recuperado 31 de mayo de 2023, de <http://www.vitofeli.com.ar/es/?view=article&id=7:que-es-marlin-y-para-que-sirve&catid=17>

## Anexos

### Anexo I: Certificado de aprobación



VICERRECTORADO ACADÉMICO

Loja, 21 de Julio del 2023  
Of. N° 846 -VDIN-ISTS-2023

Sr.(ita). VIVANCO CELI ROOSEVELT IVAN  
**ESTUDIANTE DE LA CARRERA DE TECNOLOGÍA SUPERIOR EN ELECTRONICA**

Ciudad

De mi consideración:

Por medio de la presente me dirijo a ustedes para comunicarles que una vez revisado el anteproyecto de investigación de fin de carrera de su autoría titulado **"DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023"**, el mismo cumple con los lineamientos establecidos por la institución; por lo que se autoriza su realización y puesta en marcha, para lo cual se nombra como director de su proyecto de fin de carrera (el/la) Ing. DAVID PAUL ROSALES HERRERA.

Particular que le hago conocer para los fines pertinentes.

Atentamente,

  
Ing. Germán Patricio Villamarín Coronel Mgs.

**VICERRECTOR DE DESARROLLO E INNOVACION DEL ISTS**





---

**VICERRECTORADO ACADÉMICO**

---

Loja, 21 de Julio del 2023  
Of. N° 850 -VDIN-ISTS-2023

Sr.(ita). RAMON GONZALEZ MARCO ROLANDO  
**ESTUDIANTE DE LA CARRERA DE TECNOLOGÍA SUPERIOR EN ELECTRONICA**

Ciudad

De mi consideración:

Por medio de la presente me dirijo a ustedes para comunicarles que una vez revisado el anteproyecto de investigación de fin de carrera de su autoría titulado **DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023**, el mismo cumple con los lineamientos establecidos por la institución; por lo que se autoriza su realización y puesta en marcha, para lo cual se nombra como director de su proyecto de fin de carrera (el/la) Ing. DAVID PAUL ROSALES HERRERA.

Particular que le hago conocer para los fines pertinentes.

Atentamente,

Ing. Germán Patricio Villamañán Coronel Mgs.  
**VICERRECTOR DE DESARROLLO E INNOVACION DEL ISTS**



## Anexo II: Autorización para la ejecución



Yo, Ing. Leydi Maribel Mingo Morocho, Mgs. con documento de identidad 1105653792, coordinadora de la carrera de Electrónica del Instituto Superior Tecnológico Sudamericano de la ciudad de Loja a petición verbal del interesado.

### **AUTORIZO**

A Marco Rolando Ramón González con cédula de identidad Nro. 1104105257 y a Rossevelt Ivan Vivanco Celi con cédula de identidad Nro. 1717619439, estudiantes del sexto ciclo de la carrera de Electrónica del “Instituto Superior Tecnológico Sudamericano”; para que realicen su proyecto de investigación de fin de carrera titulado: **“DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023”** para lo cual nos comprometemos en entregar a los estudiantes la información necesaria hasta que culmine dicho proceso.

Loja, 07 de noviembre del 2023

Ing. Leydi Maribel Mingo Morocho, Mgs.  
C.I. 1105653792

**Anexo III: Certificado de implementación**

Loja, 07 de noviembre del 2023

Ing. David Paúl Rosales Herrera

**TUTOR DEL SEMINARIO DE PROYECTOS DE INVESTIGACIÓN DE FIN DE CARRERA- ELECTRÓNICA**, a petición verbal por parte del interesado.

## **CERTIFICO**

*Que el Sr Marco Rolando Ramón González con cédula 1104105257 y Sr Roosevelt Ivan Vivanco Celi con cédula 1717619439 han venido trabajando en el Proyecto de fin de carrera titulado “DISEÑO E IMPLEMENTACIÓN DE UNA MÁQUINA EXTRUSORA DE FILAMENTO DE USO GENERAL USANDO COMO MATERIA PRIMA BOTELLAS DE PLÁSTICO PET EN LA CIUDAD DE LOJA PARA EL PERIODO ABRIL - SEPTIEMBRE 2023”; el mismo que se encuentra a la presente fecha en un 100% culminado según los requerimientos funcionales planteados. Lo certifico en honor a la verdad para los fines pertinentes y a solicitud del interesado.*

-----  
Ing. David Paul Rosales Herrera

**TUTOR SEMINARIO DE PROYECTOS DE INVESTIGACIÓN DE FIN DE CARRERA**  
Semestre Abril– Octubre 2023

## Anexo IV: Certificado de traducción



CERTIF. N°. 009-JP-ISTS-2023

Loja, 28 de Octubre de 2023

El suscrito, Lic. Juan Pablo Quezada Rosales., **DOCENTE DEL ÁREA DE INGLÉS - CIS DEL INSTITUTO SUPERIOR TECNOLÓGICO "SUDAMERICANO"**, a petición de la parte interesada y en forma legal,

**C E R T I F I C A:**

Que el apartado **ABSTRACT** del Proyecto de Investigación de Fin de Carrera de los señores **MARCO ROLANDO RAMÓN GONZÁLEZ & ROOSEVELT IVAN VIVANCO CELI** estudiantes en proceso de titulación periodo Abril – Noviembre 2023 de la carrera de **ELECTRÓNICA**; está correctamente traducido, luego de haber ejecutado las correcciones emitidas por mi persona; por cuanto se autoriza la impresión y presentación dentro del empastado final previo a la disertación del proyecto.

Particular que comunico en honor a la verdad para los fines académicos pertinentes.

*English is a piece of cake.*



Lic. Juan Pablo Quezada Rosales  
**DOCENTE DEL ÁREA DE INGLÉS ISTS - CIS**

Checked by:  
Juan Pablo Quezada R.  
E.F.L. Teacher

## Anexo V: Presupuesto

En la tabla 2 se detalla el costo de componentes electrónicos y materiales que se van a utilizar en el prototipo, en la tabla 3 se describen recursos humanos, tecnológicos y logísticos, finalmente en la tabla 4 se presenta el presupuesto total del proyecto.

**Tabla 2.**

*Componentes para el prototipo*

<b>Cantidad</b>	<b>Componentes</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
1	Arduino Mega 2560	\$30.00	\$ 30.00
1	Shield Ramps 1.4	\$11.00	\$ 11.00
1	Driver A4988	\$2.00	\$ 2.00
1	Bloque calefactor	\$8.00	\$ 8.00
1	Niquelina de 12V	\$12.00	\$12.00
1	Termistor de 12V	\$ 3.00	\$ 3.00
1	Motor a pasos Nema 17	\$ 17.00	\$17.00
1	Glcd 12864 para Shield Ramps 1.4	\$22.00	\$ 22.00
1	Juego de cables para motor y pantalla	\$ 4.00	\$ 4.00
1	Plancha de Melamínico 30x50	\$15.0	\$15.00
1	Juego de tornillos	\$3.00	\$ 3.00
4	Ángulos pequeños	\$2.00	\$ 8.00
1	Conjunto de piezas impresas en 3D	\$70.00	\$70.00
1	Kilogramo de botellas PET recicladas	\$5.00	\$5.00
1	Fuente de 12V-33A conmutada	\$23.00	\$23.00
		<b>TOTAL</b>	<b>\$ 233.00</b>

**Tabla 3.***Recursos del proyecto*

<b>Recursos Humanos</b>				
<b>Cantidad</b>	<b>Nombre del recurso</b>	<b>Descripción</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
2	Desarrollador del proyecto	Estudiante que document el proyecto	\$0.00	\$0.00
1	Directora del proyecto	Tutor del proyecto	\$0.00	\$0.00
1	Propietario inmueble	Donde se implementará el prototipo	\$0.00	\$0.00
<b>Recursos Tecnológicos</b>				
<b>Cantidad</b>	<b>Nombre del recurso</b>	<b>Descripción</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
6 (meses)	Internet	Búsqueda de información	\$24.5	\$147.00
<b>Hardware</b>				
<b>Cantidad</b>	<b>Nombre del recurso</b>	<b>Descripción</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
1	Celular	Capturas y pruebas	\$300.00	\$150.00 (depreciado)
1	Computador	Búsqueda de información y codificación	\$1000.00	\$500.00 (depreciado)
<b>Software</b>				
<b>Cantidad</b>	<b>Nombre del recurso</b>	<b>Descripción</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
1	Paquete Office	Word, Excel, PowerPoint	\$0.00	\$0.00
1	Arduino	Desarrollo de código fuente	\$0.00	\$0.00
<b>Herramientas</b>				
<b>Cantidad</b>	<b>Nombre del recurso</b>	<b>Descripción</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
2	Cúter	Armado de la máquina	\$2.50	\$5.00
2	Alicate universal	Armado	\$3.50	\$7.00
2	Alicate de corte	Armado	\$4.50	\$9.00
2	Destornilladores	Armado	\$4.00	\$8.00
2	Cautín y estaño	Armado	\$12.00	24.00
1	Multímetro	Medición de voltaje	\$42.00	\$42.00
<b>Recursos Logísticos</b>				
<b>Cantidad</b>	<b>Nombre del recurso</b>	<b>Descripción</b>	<b>VALOR UNITARIO</b>	<b>VALOR TOTAL</b>
1	Resma de hojas	Impresión de documentos varios	\$5.00	\$5.00

**Tabla 4.***Presupuesto del proyecto*

<b>Presupuesto del proyecto</b>	
<b>Recursos Humanos</b>	\$0.00
<b>Recursos Tecnológicos</b>	\$147.00
<b>Hardware</b>	\$650.00
<b>Software</b>	\$0.00
<b>Herramientas</b>	\$95.00
<b>Recursos Logísticos</b>	\$5.00
<b>Componentes para el prototipo</b>	\$233.00
<b>TOTAL</b>	<b>\$1130.00</b>



## Anexo VII: Programación

Las configuraciones se encuentran los siguientes segmentos de código. Vale recalcar que únicamente se debe abrir desde la aplicación Arduino IDE el ARCHIVO Marlin.ino. Este fichero se encargará de gestionar el resto de los subcódigos. A continuación, se presentan los códigos donde se realizaron las configuraciones “*Configuration.h*”, “*Boards.h*” y “*Version.h*”. Se puede encontrar el resto de los subprogramas en:

<https://drive.google.com/file/d/1PDDc9RzuEFZYeeSPFWjTtqZgi19FkeYX/view?usp=sharing>

```
/** Marlin 3D Printer Firmware
 * Copyright (C) 2016 MarlinFirmware [https://github.com/MarlinFirmware/Marlin]
 *
 * Based on Sprinter and grbl.
 * Copyright (C) 2011 Camiel Gubbels / Erik van der Zalm
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/**
 * Configuration.h
 *
 * Basic settings such as:
 *
 * - Type of electronics
 * - Type of temperature sensor
 * - Printer geometry
 * - Endstop configuration
 * - LCD controller
 * - Extra features
 *
 * Advanced settings can be found in Configuration_adv.h
 */
```

```

*/
#ifndef CONFIGURATION_H
#define CONFIGURATION_H
#define CONFIGURATION_H_VERSION 010109

//===== Getting Started =====

/**
 * Here are some standard links for getting your machine calibrated:
 *
 * http://reprap.org/wiki/Calibration
 * http://youtu.be/wAL9d7FgInk
 * http://calculator.josefprusa.cz
 * http://reprap.org/wiki/Triffid\_Hunter%27s\_Calibration\_Guide
 * http://www.thingiverse.com/thing:5573
 * https://sites.google.com/site/repraplogphase/calibration-of-your-reprap
 * http://www.thingiverse.com/thing:298812
 */

//=====DELTA Printer =====

// For a Delta printer start with one of the configuration files in the
// example_configurations/delta directory and customize for your machine.
//

//=====SCARA Printer=====
// For a SCARA printer start with the configuration files in
// example_configurations/SCARA and customize for your machine.
//

//===== HANGPRINTER=====
// For a Hangprinter start with the configuration file in the
// example_configurations/hangprinter directory and customize for your machine.
//

// @section info

// User-specified version info of this build to display in [Pronterface, etc] terminal window
// during
// startup. Implementation of an idea by Prof Braino to inform user that any changes made
// to this
// build by the user have been successfully uploaded into firmware.
#define STRING_CONFIG_H_AUTHOR "(none, default config)" // Who made the changes.
#define SHOW_BOOTSCREEN
#define STRING_SPLASH_LINE1 SHORT_BUILD_VERSION // will be shown during bootup in
line 1
#define STRING_SPLASH_LINE2 WEBSITE_URL // will be shown during bootup in line 2

/**
 * *** VENDORS PLEASE READ ***
 *

```

```

* Marlin allows you to add a custom boot image for Graphical LCDs.
* With this option Marlin will first show your custom screen followed
* by the standard Marlin logo with version number and web URL.
*
* We encourage you to take advantage of this new feature and we also
* respectfully request that you retain the unmodified Marlin boot screen.
*/

// Enable to show the bitmap in Marlin/_Bootscreen.h on startup.
//#define SHOW_CUSTOM_BOOTSCREEN

// Enable to show the bitmap in Marlin/_Statusscreen.h on the status screen.
//#define CUSTOM_STATUS_SCREEN_IMAGE

// @section machine
/**
 * Select the serial port on the board to use for communication with the host.
 * This allows the connection of wireless adapters (for instance) to non-default port pins.
 * Serial port 0 is always used by the Arduino bootloader regardless of this setting.
 *
 * :[0, 1, 2, 3, 4, 5, 6, 7]
 */
#define SERIAL_PORT 0

/**
 * This setting determines the communication speed of the printer.
 *
 * 250000 works in most cases, but you might try a lower speed if
 * you commonly experience drop-outs during host printing.
 * You may try up to 1000000 to speed up SD file transfer.
 *
 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
 */
#define BAUDRATE 250000

// Enable the Bluetooth serial interface on AT90USB devices
//#define BLUETOOTH

// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_RAMPS_14_EFB
#endif

// Optional custom name for your RepStrap or other custom machine
// Displayed in the LCD "Ready" message
//#define CUSTOM_MACHINE_NAME "3D Printer"

// Define this to set a unique identifier for this printer, (Used by some programs to
// differentiate between machines)

```

```

// You can use an online service to generate a random UUID. (eg
http://www.uuidgenerator.net/version4)
//#define MACHINE_UUID "00000000-0000-0000-0000-000000000000"

// @section extruder

// This defines the number of extruders
// :[1, 2, 3, 4, 5]
#define EXTRUDERS 1

// Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used for Volumetric, Filament
Width Sensor, etc.
#define DEFAULT_NOMINAL_FILAMENT_DIA 3.0

// For Cyclops or any "multi-extruder" that shares a single nozzle.
//#define SINGLENOZZLE

/**
 * Průša MK2 Single Nozzle Multi-Material Multiplexer, and variants.
 *
 * This device allows one stepper driver on a control board to drive
 * two to eight stepper motors, one at a time, in a manner suitable
 * for extruders.
 *
 * This option only allows the multiplexer to switch on tool-change.
 * Additional options to configure custom E moves are pending.
 */
//#define MK2_MULTIPLEXER
#if ENABLED(MK2_MULTIPLEXER)
  // Override the default DIO selector pins here, if needed.
  // Some pins files may provide defaults for these pins.
  //#define E_MUX0_PIN 40 // Always Required
  //#define E_MUX1_PIN 42 // Needed for 3 to 8 steppers
  //#define E_MUX2_PIN 44 // Needed for 5 to 8 steppers
#endif

// A dual extruder that uses a single stepper motor
//#define SWITCHING_EXTRUDER
#if ENABLED(SWITCHING_EXTRUDER)
  #define SWITCHING_EXTRUDER_SERVO_NR 0
  #define SWITCHING_EXTRUDER_SERVO_ANGLES { 0, 90 } // Angles for E0, E1[, E2, E3]
  #if EXTRUDERS > 3
    #define SWITCHING_EXTRUDER_E23_SERVO_NR 1
  #endif
#endif

// A dual-nozzle that uses a servomotor to raise/lower one of the nozzles
//#define SWITCHING_NOZZLE
#if ENABLED(SWITCHING_NOZZLE)
  #define SWITCHING_NOZZLE_SERVO_NR 0
  #define SWITCHING_NOZZLE_SERVO_ANGLES { 0, 90 } // Angles for E0, E1

```

```

    // #define HOTEND_OFFSET_Z { 0.0, 0.0 }
#endif

/**
 * Two separate X-carriages with extruders that connect to a moving part
 * via a magnetic docking mechanism. Requires SOL1_PIN and SOL2_PIN.
 */
// #define PARKING_EXTRUDER
#if ENABLED(PARKING_EXTRUDER)
    #define PARKING_EXTRUDER_SOLENOIDS_INVERT // If enabled, the solenoid is NOT
    magnetized with applied voltage
    #define PARKING_EXTRUDER_SOLENOIDS_PINS_ACTIVE LOW // LOW or HIGH pin signal
    energizes the coil
    #define PARKING_EXTRUDER_SOLENOIDS_DELAY 250 // Delay (ms) for magnetic field.
    No delay if 0 or not defined.
    #define PARKING_EXTRUDER_PARKING_X { -78, 184 } // X positions for parking the
    extruders
    #define PARKING_EXTRUDER_GRAB_DISTANCE 1 // mm to move beyond the
    parking point to grab the extruder
    #define PARKING_EXTRUDER_SECURITY_RAISE 5 // Z-raise before parking
    #define HOTEND_OFFSET_Z { 0.0, 1.3 } // Z-offsets of the two hotends. The first
    must be 0.
#endif

/**
 * "Mixing Extruder"
 * - Adds G-codes M163 and M164 to set and "commit" the current mix factors.
 * - Extends the stepping routines to move multiple steppers in proportion to the mix.
 * - Optional support for Repetier Firmware's 'M164 S<index>' supporting virtual tools.
 * - This implementation supports up to two mixing extruders.
 * - Enable DIRECT_MIXING_IN_G1 for M165 and mixing in G1 (from Pia Taubert's
    reference implementation).
 */
// #define MIXING_EXTRUDER
#if ENABLED(MIXING_EXTRUDER)
    #define MIXING_STEPPERS 2 // Number of steppers in your mixing extruder
    #define MIXING_VIRTUAL_TOOLS 16 // Use the Virtual Tool method with M163 and M164
    // #define DIRECT_MIXING_IN_G1 // Allow ABCDHI mix factors in G1 movement
    commands
#endif

// Offset of the extruders (uncomment if using more than one and relying on firmware to
// position when changing).
// The offset has to be X=0, Y=0 for the extruder 0 hotend (default extruder).
// For the other hotends it is their distance from the extruder 0 hotend.
// #define HOTEND_OFFSET_X { 0.0, 20.00 } // (in mm) for each extruder, offset of the
// hotend on the X axis
// #define HOTEND_OFFSET_Y { 0.0, 5.00 } // (in mm) for each extruder, offset of the hotend
// on the Y axis

// @section machine

```

```

/**
 * Select your power supply here. Use 0 if you haven't connected the PS_ON_PIN
 *
 * 0 = No Power Switch
 * 1 = ATX
 * 2 = X-Box 360 203Watts (the blue wire connected to PS_ON and the red wire to VCC)
 *
 * :{ 0:'No power switch', 1:'ATX', 2:'X-Box 360' }
 */
#define POWER_SUPPLY 0

#if POWER_SUPPLY > 0
// Enable this option to leave the PSU off at startup.
// Power to steppers and heaters will need to be turned on with M80.
// #define PS_DEFAULT_OFF

// #define AUTO_POWER_CONTROL // Enable automatic control of the PS_ON pin
#if ENABLED(AUTO_POWER_CONTROL)
#define AUTO_POWER_FANS // Turn on PSU if fans need power
#define AUTO_POWER_E_FANS
#define AUTO_POWER_CONTROLLERFAN
#define POWER_TIMEOUT 30
#endif

#endif

// @section temperature

// ===== Thermal Settings =====

/**
 * --NORMAL IS 4.7kohm PULLUP!-- 1kohm pullup can be used on hotend sensor, using
 * correct resistor and table
 *
 * Temperature sensors available:
 *
 * -4 : thermocouple with AD8495
 * -3 : thermocouple with MAX31855 (only for sensor 0)
 * -2 : thermocouple with MAX6675 (only for sensor 0)
 * -1 : thermocouple with AD595
 * 0 : not used
 * 1 : 100k thermistor - best choice for EPCOS 100k (4.7k pullup)
 * 2 : 200k thermistor - ATC Semitec 204GT-2 (4.7k pullup)
 * 3 : Mendel-parts thermistor (4.7k pullup)
 * 4 : 10k thermistor !! do not use it for a hotend. It gives bad resolution at high temp. !!
 * 5 : 100K thermistor - ATC Semitec 104GT-2/104NT-4-R025H42G (Used in ParCan & J-
 * Head) (4.7k pullup)
 * 501 : 100K Zonestar (Tronxy X3A) Thermistor
 * 6 : 100k EPCOS - Not as accurate as table 1 (created using a fluke thermocouple) (4.7k
 * pullup)
 * 7 : 100k Honeywell thermistor 135-104LAG-J01 (4.7k pullup)

```

```

* 71 : 100k Honeywell thermistor 135-104LAF-J01 (4.7k pullup)
* 8 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)
* 9 : 100k GE Sensing AL03006-58.2K-97-G1 (4.7k pullup)
* 10 : 100k RS thermistor 198-961 (4.7k pullup)
* 11 : 100k beta 3950 1% thermistor (4.7k pullup)
* 12 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup) (calibrated for Makibox
hot bed)
* 13 : 100k Hisens 3950 1% up to 300°C for hotend "Simple ONE " & "Hotend "All In
ONE"
* 15 : 100k thermistor calibration for JGAurora A5 hotend
* 20 : the PT100 circuit found in the Ultimainboard V2.x
* 60 : 100k Maker's Tool Works Kapton Bed Thermistor beta=3950
* 66 : 4.7M High Temperature thermistor from Dyze Design
* 70 : the 100K thermistor found in the bq Hephestos 2
* 75 : 100k Generic Silicon Heat Pad with NTC 100K MGB18-104F39050L32 thermistor
*
* 1k ohm pullup tables - This is atypical, and requires changing out the 4.7k pullup for
1k.
*
* (but gives greater accuracy and more stable PID)
* 51 : 100k thermistor - EPCOS (1k pullup)
* 52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
* 55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)
*
* 1047 : Pt1000 with 4k7 pullup
* 1010 : Pt1000 with 1k pullup (non standard)
* 147 : Pt100 with 4k7 pullup
* 110 : Pt100 with 1k pullup (non standard)
*
* Use these for Testing or Development purposes. NEVER for production machine.
* 998 : Dummy Table that ALWAYS reads 25°C or the temperature defined below.
* 999 : Dummy Table that ALWAYS reads 100°C or the temperature defined below.
*
* :{ '0': "Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC Semitec 204GT-2",
'3':"Mendel-parts / 4.7k", '4':"10k !! do not use for a hotend. Bad resolution at high temp.
!!", '5':"100k / 4.7k - ATC Semitec 104GT-2 (Used in ParCan & J-Head)", '501':"100K
Zonestar (Tronxy X3A)", '6':"100k / 4.7k EPCOS - Not as accurate as Table 1", '7':"100k /
4.7k Honeywell 135-104LAG-J01", '8':"100k / 4.7k 0603 SMD Vishay NTCS0603E3104FXT",
'9':"100k / 4.7k GE Sensing AL03006-58.2K-97-G1", '10':"100k / 4.7k RS 198-961", '11':"100k
/ 4.7k beta 3950 1%", '12':"100k / 4.7k 0603 SMD Vishay NTCS0603E3104FXT (calibrated
for Makibox hot bed)", '13':"100k Hisens 3950 1% up to 300°C for hotend 'Simple ONE ' &
hotend 'All In ONE'", '20':"PT100 (Ultimainboard V2.x)", '51':"100k / 1k - EPCOS", '52':"200k
/ 1k - ATC Semitec 204GT-2", '55':"100k / 1k - ATC Semitec 104GT-2 (Used in ParCan & J-
Head)", '60':"100k Maker's Tool Works Kapton Bed Thermistor beta=3950", '66':"Dyze
Design 4.7M High Temperature thermistor", '70':"the 100K thermistor found in the bq
Hephestos 2", '71':"100k / 4.7k Honeywell 135-104LAF-J01", '147':"Pt100 / 4.7k",
'1047':"Pt1000 / 4.7k", '110':"Pt100 / 1k (non-standard)", '1010':"Pt1000 / 1k (non
standard)", '-4':"Thermocouple + AD8495", '-3':"Thermocouple + MAX31855 (only for
sensor 0)", '-2':"Thermocouple + MAX6675 (only for sensor 0)", '-1':"Thermocouple +
AD595", '998':"Dummy 1", '999':"Dummy 2" }
*/
#define TEMP_SENSOR_0 1

```

```

#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_BED 0
#define TEMP_SENSOR_CHAMBER 0

// Dummy thermistor constant temperature readings, for use with 998 and 999
#define DUMMY_THERMISTOR_998_VALUE 25
#define DUMMY_THERMISTOR_999_VALUE 100

// Use temp sensor 1 as a redundant sensor with sensor 0. If the readings
// from the two sensors differ too much the print will be aborted.
// #define TEMP_SENSOR_1_AS_REDUNDANT
#define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10

// Extruder temperature must be close to target for this long before M109 returns success
#define TEMP_RESIDENCY_TIME 10 // (seconds)
#define TEMP_HYSTERESIS 3 // (degC) range of +/- temperatures considered "close" to
the target one
#define TEMP_WINDOW 1 // (degC) Window around target to start the residency
timer x degC early.

// Bed temperature must be close to target for this long before M190 returns success
#define TEMP_BED_RESIDENCY_TIME 10 // (seconds)
#define TEMP_BED_HYSTERESIS 3 // (degC) range of +/- temperatures considered
"close" to the target one
#define TEMP_BED_WINDOW 1 // (degC) Window around target to start the
residency timer x degC early.

// The minimal temperature defines the temperature below which the heater will not be
enabled It is used
// to check that the wiring to the thermistor is not broken.
// Otherwise this would lead to the heater being powered on all the time.
#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define HEATER_3_MINTEMP 5
#define HEATER_4_MINTEMP 5
#define BED_MINTEMP 5

// When temperature exceeds max temp, your heater will be switched off.
// This feature exists to protect your hotend from overheating accidentally, but *NOT*
from thermistor short/failure!
// You should use MINTEMP for thermistor short/failure protection.
#define HEATER_0_MAXTEMP 275
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define HEATER_3_MAXTEMP 275
#define HEATER_4_MAXTEMP 275
#define BED_MAXTEMP 150

```

```

//=====PID Settings =====
//=====
// PID Tuning Guide here: http://reprap.org/wiki/PID\_Tuning

// Comment the following line to disable PID and enable bang-bang.
#define PIDTEMP
#define BANG_MAX 255 // Limits current to nozzle while in bang-bang mode; 255=full
current
#define PID_MAX BANG_MAX // Limits current to nozzle while PID is active (see
PID_FUNCTIONAL_RANGE below); 255=full current
#define PID_K1 0.95 // Smoothing factor within any PID loop
#if ENABLED(PIDTEMP)
  //#define PID_AUTOTUNE_MENU // Add PID Autotune to the LCD "Temperature" menu to
run M303 and apply the result.
  //#define PID_DEBUG // Sends debug data to the serial port.
  //#define PID_OPENLOOP 1 // Puts PID in open loop. M104/M140 sets the output power
from 0 to PID_MAX
  //#define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.125Hz=8s)
and minimum state time of approximately 1s useful for heaters driven by a relay
  //#define PID_PARAMS_PER_HOTEND // Uses separate PID parameters for each extruder
(useful for mismatched extruders)
      // Set/get with gcode: M301 E[extruder number, 0-2]
  #define PID_FUNCTIONAL_RANGE 10 // If the temperature difference between the target
temperature and the actual temperature
      // is more than PID_FUNCTIONAL_RANGE then the PID will be shut off
and the heater will be set to min/max.

  // If you are using a pre-configured hotend then you can use one of the value sets by
uncommenting it

  // Ultimaker
  #define DEFAULT_Kp 22.2
  #define DEFAULT_Ki 1.08
  #define DEFAULT_Kd 114

  // MakerGear
  //#define DEFAULT_Kp 7.0
  //#define DEFAULT_Ki 0.1
  //#define DEFAULT_Kd 12

  // Mendel Parts V9 on 12V
  //#define DEFAULT_Kp 63.0
  //#define DEFAULT_Ki 2.25
  //#define DEFAULT_Kd 440

#endif // PIDTEMP

//===== PID > Bed Temperature Control =====

/**

```

```

* PID Bed Heating
*
* If this option is enabled set PID constants below.
* If this option is disabled, bang-bang will be used and BED_LIMIT_SWITCHING will enable
hysteresis.
*
* The PID frequency will be the same as the extruder PWM.
* If PID_dT is the default, and correct for the hardware/configuration, that means 7.689Hz,
* which is fine for driving a square wave into a resistive load and does not significantly
* impact FET heating. This also works fine on a Fotek SSR-10DA Solid State Relay into a
250W
* heater. If your configuration is significantly different than this and you don't understand
* the issues involved, don't use bed PID until someone else verifies that your hardware
works.
*/
//#define PIDTEMPBED

//#define BED_LIMIT_SWITCHING

/**
* Max Bed Power
* Applies to all forms of bed control (PID, bang-bang, and bang-bang with hysteresis).
* When set to any value below 255, enables a form of PWM to the bed that acts like a
divider
* so don't use it unless you are OK with PWM on your bed. (See the comment on enabling
PIDTEMPBED)
*/
#define MAX_BED_POWER 255 // limits duty cycle to bed; 255=full current

#if ENABLED(PIDTEMPBED)

  // #define PID_BED_DEBUG // Sends debug data to the serial port.

  // 120V 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
  // from FOPDT model - kp=.39 Tp=405 Tdead=66, Tc set to 79.2, aggressive factor of .15 (vs
  .1, 1, 10)
  #define DEFAULT_bedKp 10.00
  #define DEFAULT_bedKi .023
  #define DEFAULT_bedKd 305.4

  // 120V 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
  // from pidautotune
  // #define DEFAULT_bedKp 97.1
  // #define DEFAULT_bedKi 1.41
  // #define DEFAULT_bedKd 1675.16

  // FIND YOUR OWN: "M303 E-1 C8 S90" to run autotune on the bed at 90 degreesC for 8
  cycles.
#endif // PIDTEMPBED

// @section extruder

```



```

// @section homing

// Specify here all the endstop connectors that are connected to any endstop or probe.
// Almost all printers will be using one per axis. Probes will use one or more of the
// extra connectors. Leave undefined any used for non-endstop and non-probe purposes.
#define USE_XMIN_PLUG
#define USE_YMIN_PLUG
#define USE_ZMIN_PLUG
//#define USE_XMAX_PLUG
//#define USE_YMAX_PLUG
//#define USE_ZMAX_PLUG

// Enable pullup for all endstops to prevent a floating state
#define ENDSTOPPULLUPS
#if DISABLED(ENDSTOPPULLUPS)
  // Disable ENDSTOPPULLUPS to set pullups individually
  //#define ENDSTOPPULLUP_XMAX
  //#define ENDSTOPPULLUP_YMAX
  //#define ENDSTOPPULLUP_ZMAX
  //#define ENDSTOPPULLUP_XMIN
  //#define ENDSTOPPULLUP_YMIN
  //#define ENDSTOPPULLUP_ZMIN
  //#define ENDSTOPPULLUP_ZMIN_PROBE
#endif

// Mechanical endstop with COM to ground and NC to Signal uses "false" here (most
// common setup).
#define X_MIN_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Y_MIN_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Z_MIN_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // set to true to invert the logic of the
// probe.

/**
 * Stepper Drivers
 *
 * These settings allow Marlin to tune stepper driver timing and enable advanced options
 * for
 * stepper drivers that support them. You may also override timing options in
 * Configuration_adv.h.
 *
 * A4988 is assumed for unspecified drivers.
 *
 * Options: A4988, DRV8825, LV8729, L6470, TB6560, TB6600, TMC2100,
 *          TMC2130, TMC2130_STANDALONE, TMC2208, TMC2208_STANDALONE,
 *          TMC26X, TMC26X_STANDALONE, TMC2660, TMC2660_STANDALONE,
 *          TMC5130, TMC5130_STANDALONE

```

```

* :['A4988', 'DRV8825', 'LV8729', 'L6470', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130',
'TMC2130_STANDALONE', 'TMC2208', 'TMC2208_STANDALONE', 'TMC26X',
'TMC26X_STANDALONE', 'TMC2660', 'TMC2660_STANDALONE', 'TMC5130',
'TMC5130_STANDALONE']
*/
#define X_DRIVER_TYPE A4988
#define Y_DRIVER_TYPE A4988
#define Z_DRIVER_TYPE A4988
#define X2_DRIVER_TYPE A4988
#define Y2_DRIVER_TYPE A4988
#define Z2_DRIVER_TYPE A4988
#define E0_DRIVER_TYPE A4988
#define E1_DRIVER_TYPE A4988
#define E2_DRIVER_TYPE A4988
#define E3_DRIVER_TYPE A4988
#define E4_DRIVER_TYPE A4988

// Enable this feature if all enabled endstop pins are interrupt-capable.
// This will remove the need to poll the interrupt pins, saving many CPU cycles.
#define ENDSTOP_INTERRUPTS_FEATURE

/**
 * Endstop Noise Filter
 *
 * Enable this option if endstops falsely trigger due to noise.
 * NOTE: Enabling this feature means adds an error of +/-0.2mm, so homing
 * will end up at a slightly different position on each G28. This will also
 * reduce accuracy of some bed probes.
 * For mechanical switches, the better approach to reduce noise is to install
 * a 100 nanofarads ceramic capacitor in parallel with the switch, making it
 * essentially noise-proof without sacrificing accuracy.
 * This option also increases MCU load when endstops or the probe are enabled.
 * So this is not recommended. USE AT YOUR OWN RISK.
 * (This feature is not required for common micro-switches mounted on PCBs
 * based on the Makerbot design, since they already include the 100nF capacitor.)
 */
#define ENDSTOP_NOISE_FILTER

//=====Movement Settings=====
// @section motion

/**
 * Default Settings
 * These settings can be reset by M502
 * Note that if EEPROM is enabled, saved values will override these.
 */

/**
 * With this option each E stepper can have its own factors for the
 * following movement settings. If fewer factors are given than the
 * total number of extruders, the last value applies to the rest.

```

```

*/
//#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *           X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 4000, 500 }

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 *           X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_MAX_FEEDRATE      { 300, 300, 5, 25 }

/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 *           X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_MAX_ACCELERATION  { 3000, 3000, 100, 10000 }

/**
 * Default Acceleration (change/s) change = mm/s
 * Override with M204
 *
 * M204 P Acceleration
 * M204 R Retract Acceleration
 * M204 T Travel Acceleration
 */
#define DEFAULT_ACCELERATION      3000 // X, Y, Z and E acceleration for printing
moves
#define DEFAULT_RETRACT_ACCELERATION 3000 // E acceleration for retracts
#define DEFAULT_TRAVEL_ACCELERATION 3000 // X, Y, Z acceleration for travel (non
printing) moves

/**
 * Default Jerk (mm/s)
 * Override with M205 X Y Z E
 *
 * "Jerk" specifies the minimum speed change that requires acceleration.
 * When changing speed and direction, if the difference is less than the
 * value set here, it may happen instantaneously.
 */
#define DEFAULT_XJERK             10.0
#define DEFAULT_YJERK             10.0
#define DEFAULT_ZJERK             0.3
#define DEFAULT_EJERK             5.0

```

```

/**
 * S-Curve Acceleration
 *
 * This option eliminates vibration during printing by fitting a Bézier
 * curve to move acceleration, producing much smoother direction changes.
 *
 * See https://github.com/synthetos/TinyG/wiki/Jerk-Controlled-Motion-Explained
 */
// #define S_CURVE_ACCELERATION

// =====Z Probe Options =====
// @section probes

//
// See http://marlinfw.org/docs/configuration/probes.html
//

/**
 * Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
 *
 * Enable this option for a probe connected to the Z Min endstop pin.
 */
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN

/**
 * Z_MIN_PROBE_ENDSTOP
 *
 * Enable this option for a probe connected to any pin except Z-Min.
 * (By default Marlin assumes the Z-Max endstop pin.)
 * To use a custom Z Probe pin, set Z_MIN_PROBE_PIN below.
 *
 * - The simplest option is to use a free endstop connector.
 * - Use 5V for powered (usually inductive) sensors.
 *
 * - RAMPS 1.3/1.4 boards may use the 5V, GND, and Aux4->D32 pin:
 * - For simple switches connect...
 * - normally-closed switches to GND and D32.
 * - normally-open switches to 5V and D32.
 *
 * WARNING: Setting the wrong pin may have unexpected and potentially
 * disastrous consequences. Use with caution and do your homework.
 */
// #define Z_MIN_PROBE_ENDSTOP

/**
 * Probe Type
 *
 * Allen Key Probes, Servo Probes, Z-Sled Probes, FIX_MOUNTED_PROBE, etc.
 * Activate one of these to use Auto Bed Leveling below.

```

```

*/

/**
 * The "Manual Probe" provides a means to do "Auto" Bed Leveling without a probe.
 * Use G29 repeatedly, adjusting the Z height at each point with movement commands
 * or (with LCD_BED_LEVELING) the LCD controller.
 */
#define PROBE_MANUALLY
#define MANUAL_PROBE_START_Z 0.2

/**
 * A Fix-Mounted Probe either doesn't deploy or needs manual deployment.
 * (e.g., an inductive probe or a nozzle-based probe-switch.)
 */
#define FIX_MOUNTED_PROBE

/**
 * Z Servo Probe, such as an endstop switch on a rotating arm.
 */
#define Z_PROBE_SERVO_NR 0 // Defaults to SERVO 0 connector.
#define Z_SERVO_ANGLES {70,0} // Z Servo Deploy and Stow angles

/**
 * The BLTouch probe uses a Hall effect sensor and emulates a servo.
 */
#define BLTOUCH

/**
 * Enable one or more of the following if probing seems unreliable.
 * Heaters and/or fans can be disabled during probing to minimize electrical
 * noise. A delay can also be added to allow noise and vibration to settle.
 * These options are most useful for the BLTouch probe, but may also improve
 * readings with inductive probes and piezo sensors.
 */
#define PROBING_HEATERS_OFF // Turn heaters off when probing
#if ENABLED(PROBING_HEATERS_OFF)
  #define WAIT_FOR_BED_HEATER // Wait for bed to heat back up between probes (to
  improve accuracy)
#endif
#define PROBING_FANS_OFF // Turn fans off when probing
#define DELAY_BEFORE_PROBING 200 // (ms) To prevent vibrations from triggering
  piezo sensors

// A probe that is deployed and stowed with a solenoid pin (SOL1_PIN)
#define SOLENOID_PROBE

// A sled-mounted probe like those designed by Charles Bell.
#define Z_PROBE_SLED
#define SLED_DOCKING_OFFSET 5 // The extra distance the X axis must travel to pickup
  the sled. 0 should be fine but you can push it further if you'd like.

```

```

//
// For Z_PROBE_ALLEN_KEY see the Delta example configurations.
//

/**
 * Z Probe to nozzle (X,Y) offset, relative to (0, 0).
 * X and Y offsets must be integers.
 *
 * In the following example the X and Y offsets are both positive:
 * #define X_PROBE_OFFSET_FROM_EXTRUDER 10
 * #define Y_PROBE_OFFSET_FROM_EXTRUDER 10
 *
 *   +- BACK ---+
 *   |           |
 * L | (+) P | R <-- probe (20,20)
 * E |       | I
 * F | (-) N (+) | G <-- nozzle (10,10)
 * T |       | H
 *   | (-) | T
 *   |       |
 *   O-- FRONT --+
 *   (0,0)
 */
#define X_PROBE_OFFSET_FROM_EXTRUDER 10 // X offset: -left +right [of the nozzle]
#define Y_PROBE_OFFSET_FROM_EXTRUDER 10 // Y offset: -front +behind [the nozzle]
#define Z_PROBE_OFFSET_FROM_EXTRUDER 0 // Z offset: -below +above [the nozzle]

// Certain types of probes need to stay away from edges
#define MIN_PROBE_EDGE 10

// X and Y axis travel speed (mm/m) between probes
#define XY_PROBE_SPEED 8000

// Feedrate (mm/m) for the first approach when double-probing (MULTIPLE_PROBING ==
2)
#define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z

// Feedrate (mm/m) for the "accurate" probe of each point
#define Z_PROBE_SPEED_SLOW (Z_PROBE_SPEED_FAST / 2)

// The number of probes to perform at each point.
// Set to 2 for a fast/slow probe, using the second probe result.
// Set to 3 or more for slow probes, averaging the results.
// #define MULTIPLE_PROBING 2

/**
 * Z probes require clearance when deploying, stowing, and moving between
 * probe points to avoid hitting the bed and other hardware.
 * Servo-mounted probes require extra space for the arm to rotate.
 * Inductive probes need space to keep from triggering early.
 *

```

```

* Use these settings to specify the distance (mm) to raise the probe (or
* lower the bed). The values set here apply over and above any (negative)
* probe Z Offset set with Z_PROBE_OFFSET_FROM_EXTRUDER, M851, or the LCD.
* Only integer values >= 1 are valid here.
*
* Example: `M851 Z-5` with a CLEARANCE of 4 => 9mm from bed to nozzle.
* But: `M851 Z+1` with a CLEARANCE of 2 => 2mm from bed to nozzle.
*/
#define Z_CLEARANCE_DEPLOY_PROBE 10 // Z Clearance for Deploy/Stow
#define Z_CLEARANCE_BETWEEN_PROBES 5 // Z Clearance between probe points
#define Z_CLEARANCE_MULTI_PROBE 5 // Z Clearance between multiple probes
// #define Z_AFTER_PROBING 5 // Z position after probing is done

#define Z_PROBE_LOW_POINT -2 // Farthest distance below the trigger-point to go
before stopping

// For M851 give a range for adjusting the Z probe offset
#define Z_PROBE_OFFSET_RANGE_MIN -20
#define Z_PROBE_OFFSET_RANGE_MAX 20

// Enable the M48 repeatability test to test probe accuracy
// #define Z_MIN_PROBE_REPEATABILITY_TEST

// For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
// :{ 0:'Low', 1:'High' }
#define X_ENABLE_ON 0
#define Y_ENABLE_ON 0
#define Z_ENABLE_ON 0
#define E_ENABLE_ON 0 // For all extruders

// Disables axis stepper immediately when it's not being used.
// WARNING: When motors turn off there is a chance of losing position accuracy!
#define DISABLE_X false
#define DISABLE_Y false
#define DISABLE_Z false
// Warn on display about possibly reduced accuracy
// #define DISABLE_REDUCE_ACCURACY_WARNING

// @section extruder

#define DISABLE_E false // For all extruders
#define DISABLE_INACTIVE_EXTRUDER true // Keep only the active extruder enabled.

// @section machine

// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the
wrong way.
#define INVERT_X_DIR true
#define INVERT_Y_DIR false
#define INVERT_Z_DIR false

```

```

// @section extruder

// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false

// @section homing

// #define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been
homed

// #define UNKNOWN_Z_NO_RAISE // Don't raise Z (lower the bed) if Z is "unknown." For
beds that fall when Z is powered off.

// #define Z_HOMING_HEIGHT 4 // (in mm) Minimal z height before homing (G28) for Z
clearance above the bed, clamps, ...
// Be sure you have this distance over your Z_MAX_POS in case.

// Direction of endstops when homing; 1=MAX, -1=MIN
// :[-1,1]
#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1

// @section machine

// The size of the print bed
#define X_BED_SIZE 200
#define Y_BED_SIZE 200

// Travel limits (mm) after homing, corresponding to endstop positions.
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS X_BED_SIZE
#define Y_MAX_POS Y_BED_SIZE
#define Z_MAX_POS 200

/**
 * Software Endstops
 *
 * - Prevent moves outside the set machine bounds.
 * - Individual axes can be disabled, if desired.
 * - X and Y only apply to Cartesian robots.
 * - Use 'M211' to set software endstops on/off or report current state
 */

// Min software endstops constrain movement within minimum coordinate bounds

```

```

#define MIN_SOFTWARE_ENDSTOPS
#if ENABLED(MIN_SOFTWARE_ENDSTOPS)
  #define MIN_SOFTWARE_ENDSTOP_X
  #define MIN_SOFTWARE_ENDSTOP_Y
  #define MIN_SOFTWARE_ENDSTOP_Z
#endif

// Max software endstops constrain movement within maximum coordinate bounds
#define MAX_SOFTWARE_ENDSTOPS
#if ENABLED(MAX_SOFTWARE_ENDSTOPS)
  #define MAX_SOFTWARE_ENDSTOP_X
  #define MAX_SOFTWARE_ENDSTOP_Y
  #define MAX_SOFTWARE_ENDSTOP_Z
#endif

#if ENABLED(MIN_SOFTWARE_ENDSTOPS) || ENABLED(MAX_SOFTWARE_ENDSTOPS)
  // #define SOFT_ENDSTOPS_MENU_ITEM // Enable/Disable software endstops from the
  LCD
#endif

/**
 * Filament Runout Sensors
 * Mechanical or opto endstops are used to check for the presence of filament.
 *
 * RAMPS-based boards use SERVO3_PIN for the first runout sensor.
 * For other boards you may need to define FIL_RUNOUT_PIN, FIL_RUNOUT2_PIN, etc.
 * By default the firmware assumes HIGH=FILAMENT PRESENT.
 */
// #define FILAMENT_RUNOUT_SENSOR
#if ENABLED(FILAMENT_RUNOUT_SENSOR)
  #define NUM_RUNOUT_SENSORS 1 // Number of sensors, up to one per extruder.
  Define a FIL_RUNOUT#_PIN for each.
  #define FIL_RUNOUT_INVERTING false // set to true to invert the logic of the sensor.
  #define FIL_RUNOUT_PULLUP // Use internal pullup for filament runout pins.
  #define FILAMENT_RUNOUT_SCRIPT "M600"
#endif

//=====
//===== Bed Leveling =====
//=====
// @section calibrate

/**
 * Choose one of the options below to enable G29 Bed Leveling. The parameters
 * and behavior of G29 will change depending on your selection.
 *
 * If using a Probe for Z Homing, enable Z_SAFE_HOMING also!
 *
 * - AUTO_BED_LEVELING_3POINT

```

```

* Probe 3 arbitrary points on the bed (that aren't collinear)
* You specify the XY coordinates of all 3 points.
* The result is a single tilted plane. Best for a flat bed.
*
* - AUTO_BED_LEVELING_LINEAR
* Probe several points in a grid.
* You specify the rectangle and the density of sample points.
* The result is a single tilted plane. Best for a flat bed.
*
* - AUTO_BED_LEVELING_BILINEAR
* Probe several points in a grid.
* You specify the rectangle and the density of sample points.
* The result is a mesh, best for large or uneven beds.
*
* - AUTO_BED_LEVELING_UBL (Unified Bed Leveling)
* A comprehensive bed leveling system combining the features and benefits
* of other systems. UBL also includes integrated Mesh Generation, Mesh
* Validation and Mesh Editing systems.
*
* - MESH_BED_LEVELING
* Probe a grid manually
* The result is a mesh, suitable for large or uneven beds. (See BILINEAR.)
* For machines without a probe, Mesh Bed Leveling provides a method to perform
* leveling in steps so you can manually adjust the Z height at each grid-point.
* With an LCD controller the process is guided step-by-step.
*/
#define AUTO_BED_LEVELING_3POINT
#define AUTO_BED_LEVELING_LINEAR
#define AUTO_BED_LEVELING_BILINEAR
#define AUTO_BED_LEVELING_UBL
#define MESH_BED_LEVELING

/**
 * Normally G28 leaves leveling disabled on completion. Enable
 * this option to have G28 restore the prior leveling state.
 */
#define RESTORE_LEVELING_AFTER_G28

/**
 * Enable detailed logging of G28, G29, M48, etc.
 * Turn on with the command 'M111 S32'.
 * NOTE: Requires a lot of PROGMEM!
 */
#define DEBUG_LEVELING_FEATURE

#if ENABLED(MESH_BED_LEVELING) || ENABLED(AUTO_BED_LEVELING_BILINEAR) ||
ENABLED(AUTO_BED_LEVELING_UBL)
  // Gradually reduce leveling correction until a set height is reached,
  // at which point movement will be level to the machine's XY plane.
  // The height can be set with M420 Z<height>
  #define ENABLE_LEVELING_FADE_HEIGHT

```

```

// For Cartesian machines, instead of dividing moves on mesh boundaries,
// split up moves into short segments like a Delta. This follows the
// contours of the bed more closely than edge-to-edge straight moves.
#define SEGMENT_LEVELLED_MOVES
#define LEVELLED_SEGMENT_LENGTH 5.0 // (mm) Length of all segments (except the last
one)

/**
 * Enable the G26 Mesh Validation Pattern tool.
 */
// #define G26_MESH_VALIDATION
#if ENABLED(G26_MESH_VALIDATION)
  #define MESH_TEST_NOZZLE_SIZE 0.4 // (mm) Diameter of primary nozzle.
  #define MESH_TEST_LAYER_HEIGHT 0.2 // (mm) Default layer height for the G26 Mesh
Validation Tool.
  #define MESH_TEST_HOTEND_TEMP 205.0 // (°C) Default nozzle temperature for the
G26 Mesh Validation Tool.
  #define MESH_TEST_BED_TEMP 60.0 // (°C) Default bed temperature for the G26
Mesh Validation Tool.
#endif

#endif

#if ENABLED(AUTO_BED_LEVELING_LINEAR) || ENABLED(AUTO_BED_LEVELING_BILINEAR)

  // Set the number of grid points per dimension.
  #define GRID_MAX_POINTS_X 3
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  // Set the boundaries for probing (where the probe can reach).
  // #define LEFT_PROBE_BED_POSITION MIN_PROBE_EDGE
  // #define RIGHT_PROBE_BED_POSITION (X_BED_SIZE - MIN_PROBE_EDGE)
  // #define FRONT_PROBE_BED_POSITION MIN_PROBE_EDGE
  // #define BACK_PROBE_BED_POSITION (Y_BED_SIZE - MIN_PROBE_EDGE)

  // Probe along the Y axis, advancing X after each column
  // #define PROBE_Y_FIRST

  #if ENABLED(AUTO_BED_LEVELING_BILINEAR)

    // Beyond the probed grid, continue the implied tilt?
    // Default is to maintain the height of the nearest edge.
    // #define EXTRAPOLATE_BEYOND_GRID

    //
    // Experimental Subdivision of the grid by Catmull-Rom method.
    // Synthesizes intermediate points to produce a more detailed mesh.
    //
    // #define ABL_BILINEAR_SUBDIVISION
    #if ENABLED(ABL_BILINEAR_SUBDIVISION)

```

```

    // Number of subdivisions between probe points
    #define BILINEAR_SUBDIVISIONS 3
  #endif

#endif

#elif ENABLED(AUTO_BED_LEVELING_UBL)

  //===== Unified Bed Leveling =====

  // #define MESH_EDIT_GFX_OVERLAY // Display a graphics overlay while editing the
  mesh

  #define MESH_INSET 1 // Set Mesh bounds as an inset region of the bed
  #define GRID_MAX_POINTS_X 10 // Don't use more than 15 points per axis,
  implementation limited.
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  #define UBL_MESH_EDIT_MOVES_Z // Sophisticated users prefer no movement of
  nozzle
  #define UBL_SAVE_ACTIVE_ON_M500 // Save the currently active mesh in the current
  slot on M500

  // #define UBL_Z_RAISE_WHEN_OFF_MESH 2.5 // When the nozzle is off the mesh, this
  value is used
  // as the Z-Height correction value.

#elif ENABLED(MESH_BED_LEVELING)

  //=====Mesh =====

  #define MESH_INSET 10 // Set Mesh bounds as an inset region of the bed
  #define GRID_MAX_POINTS_X 3 // Don't use more than 7 points per axis,
  implementation limited.
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  // #define MESH_G28_REST_ORIGIN // After homing all axes ('G28' or 'G28 XYZ') rest Z at
  Z_MIN_POS

#endif // BED_LEVELING

/**
 * Points to probe for all 3-point Leveling procedures.
 * Override if the automatically selected points are inadequate.
 */
#if ENABLED(AUTO_BED_LEVELING_3POINT) || ENABLED(AUTO_BED_LEVELING_UBL)
  // #define PROBE_PT_1_X 15
  // #define PROBE_PT_1_Y 180
  // #define PROBE_PT_2_X 15
  // #define PROBE_PT_2_Y 20

```

```

//#define PROBE_PT_3_X 170
//#define PROBE_PT_3_Y 20
#endif

/**
 * Add a bed leveling sub-menu for ABL or MBL.
 * Include a guided procedure if manual probing is enabled.
 */
//#define LCD_BED_LEVELING

#if ENABLED(LCD_BED_LEVELING)
  #define MBL_Z_STEP 0.025 // Step size while manually probing Z axis.
  #define LCD_PROBE_Z_RANGE 4 // Z Range centered on Z_MIN_POS for LCD Z adjustment
#endif

// Add a menu item to move between bed corners for manual bed adjustment
//#define LEVEL_BED_CORNERS

#if ENABLED(LEVEL_BED_CORNERS)
  #define LEVEL_CORNERS_INSET 30 // (mm) An inset for corner leveling
  #define LEVEL_CORNERS_Z_HOP 4.0 // (mm) Move nozzle up before moving between
  corners
  // #define LEVEL_CENTER_TOO // Move to the center after the last corner
#endif

/**
 * Commands to execute at the end of G29 probing.
 * Useful to retract or move the Z probe out of the way.
 */
// #define Z_PROBE_END_SCRIPT "G1 Z10 F12000\nG1 X15 Y330\nG1 Z0.5\nG1 Z10"

// @section homing

// The center of the bed is at (X=0, Y=0)
// #define BED_CENTER_AT_0_0

// Manually set the home position. Leave these undefined for automatic settings.
// For DELTA this is the top-center of the Cartesian print volume.
// #define MANUAL_X_HOME_POS 0
// #define MANUAL_Y_HOME_POS 0
// #define MANUAL_Z_HOME_POS 0

// Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
//
// With this feature enabled:
//
// - Allow Z homing only after X and Y homing AND stepper drivers still enabled.
// - If stepper drivers time out, it will need X and Y homing again before Z homing.
// - Move the Z probe (or nozzle) to a defined XY point before Z Homing when homing all
  axes (G28).
// - Prevent Z homing when the Z probe is outside bed area.

```

```

//
//#define Z_SAFE_HOMING

#if ENABLED(Z_SAFE_HOMING)
  #define Z_SAFE_HOMING_X_POINT ((X_BED_SIZE) / 2) // X point for Z homing when
homing all axes (G28).
  #define Z_SAFE_HOMING_Y_POINT ((Y_BED_SIZE) / 2) // Y point for Z homing when
homing all axes (G28).
#endif

// Homing speeds (mm/m)
#define HOMING_FEEDRATE_XY (50*60)
#define HOMING_FEEDRATE_Z (4*60)

// @section calibrate

/**
 * Bed Skew Compensation
 *
 * This feature corrects for misalignment in the XYZ axes.
 *
 * Take the following steps to get the bed skew in the XY plane:
 * 1. Print a test square (e.g., https://www.thingiverse.com/thing:2563185)
 * 2. For XY_DIAG_AC measure the diagonal A to C
 * 3. For XY_DIAG_BD measure the diagonal B to D
 * 4. For XY_SIDE_AD measure the edge A to D
 *
 * Marlin automatically computes skew factors from these measurements.
 * Skew factors may also be computed and set manually:
 *
 * - Compute AB : SQRT(2*AC*AC+2*BD*BD-4*AD*AD)/2
 * - XY_SKEW_FACTOR : TAN(PI/2-ACOS((AC*AC-AB*AB-AD*AD)/(2*AB*AD)))
 *
 * If desired, follow the same procedure for XZ and YZ.
 * Use these diagrams for reference:
 *
 * Y          Z          Z
 * ^  B-----C  ^  B-----C  ^  B-----C
 * | / / /      | / / /      | / / /
 * | / / /      | / / /      | / / /
 * | A-----D   | A-----D   | A-----D
 * +----->X   +----->X   +----->Y
 * XY_SKEW_FACTOR  XZ_SKEW_FACTOR  YZ_SKEW_FACTOR
 */
//#define SKEW_CORRECTION

#if ENABLED(SKEW_CORRECTION)
  // Input all length measurements here:
  #define XY_DIAG_AC 282.8427124746
  #define XY_DIAG_BD 282.8427124746
  #define XY_SIDE_AD 200

```

```

// Or, set the default skew factors directly here
// to override the above measurements:
#define XY_SKEW_FACTOR 0.0

//#define SKEW_CORRECTION_FOR_Z
#if ENABLED(SKEW_CORRECTION_FOR_Z)
  #define XZ_DIAG_AC 282.8427124746
  #define XZ_DIAG_BD 282.8427124746
  #define YZ_DIAG_AC 282.8427124746
  #define YZ_DIAG_BD 282.8427124746
  #define YZ_SIDE_AD 200
  #define XZ_SKEW_FACTOR 0.0
  #define YZ_SKEW_FACTOR 0.0
#endif

// Enable this option for M852 to set skew at runtime
//#define SKEW_CORRECTION_GCODE
#endif

//=====
//===== Additional Features =====

// @section extras
//
// EEPROM
//
// The microcontroller can store settings in the EEPROM, e.g. max velocity...
// M500 - stores parameters in EEPROM
// M501 - reads parameters from EEPROM (if you need reset them after you changed them
temporarily).
// M502 - reverts to the default "factory settings". You still need to store them in EEPROM
afterwards if you want to.
//
//#define EEPROM_SETTINGS // Enable for M500 and M501 commands
//#define DISABLE_M503 // Saves ~2700 bytes of PROGMEM. Disable for release!
#define EEPROM_CHITCHAT // Give feedback on EEPROM commands. Disable to save
PROGMEM.

//
// Host Keepalive
//
// When enabled Marlin will send a busy status message to the host
// every couple of seconds when it can't accept commands.
//
#define HOST_KEEPALIVE_FEATURE // Disable this if your host doesn't like keepalive
messages
#define DEFAULT_KEEPALIVE_INTERVAL 2 // Number of seconds between "busy"
messages. Set with M113.
#define BUSY_WHILE_HEATING // Some hosts require "busy" messages even during
heating

```

```

//
// M100 Free Memory Watcher
//
// #define M100_FREE_MEMORY_WATCHER // Add M100 (Free Memory Watcher) to
// debug memory usage

//
// G20/G21 Inch mode support
//
// #define INCH_MODE_SUPPORT

//
// M149 Set temperature units support
//
// #define TEMPERATURE_UNITS_SUPPORT

// @section temperature

// Preheat Constants
#define PREHEAT_1_TEMP_HOTEND 180
#define PREHEAT_1_TEMP_BED 70
#define PREHEAT_1_FAN_SPEED 0 // Value from 0 to 255

#define PREHEAT_2_TEMP_HOTEND 240
#define PREHEAT_2_TEMP_BED 110
#define PREHEAT_2_FAN_SPEED 0 // Value from 0 to 255

/**
 * Nozzle Park
 *
 * Park the nozzle at the given XYZ position on idle or G27.
 *
 * The "P" parameter controls the action applied to the Z axis:
 *
 * P0 (Default) If Z is below park Z raise the nozzle.
 * P1 Raise the nozzle always to Z-park height.
 * P2 Raise the nozzle by Z-park amount, limited to Z_MAX_POS.
 */
// #define NOZZLE_PARK_FEATURE

#if ENABLED(NOZZLE_PARK_FEATURE)
  // Specify a park position as { X, Y, Z }
  #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
  #define NOZZLE_PARK_XY_FEEDRATE 100 // X and Y axes feedrate in mm/s (also used for
  // delta printers Z axis)
  #define NOZZLE_PARK_Z_FEEDRATE 5 // Z axis feedrate in mm/s (not used for delta
  // printers)
#endif

/**

```

```

* Clean Nozzle Feature -- EXPERIMENTAL
*
* Adds the G12 command to perform a nozzle cleaning process.
*
* Parameters:
* P Pattern
* S Strokes / Repetitions
* T Triangles (P1 only)
*
* Patterns:
* P0 Straight line (default). This process requires a sponge type material
*   at a fixed bed location. "S" specifies strokes (i.e. back-forth motions)
*   between the start / end points.
*
* P1 Zig-zag pattern between (X0, Y0) and (X1, Y1), "T" specifies the
*   number of zig-zag triangles to do. "S" defines the number of strokes.
*   Zig-zags are done in whichever is the narrower dimension.
*   For example, "G12 P1 S1 T3" will execute:
*
*   --
*   | (X0, Y1) | ^   ^   ^   | (X1, Y1)
*   |   | / \ / \ / \ |
*   A |   | / \ / \ / \ |
*   |   | / \ / \ / \ |
*   | (X0, Y0) | /   \   \   \ | (X1, Y0)
*   --   +-----+
*           |_____|_____|_____|
*           T1    T2    T3
*
* P2 Circular pattern with middle at NOZZLE_CLEAN_CIRCLE_MIDDLE.
*   "R" specifies the radius. "S" specifies the stroke count.
*   Before starting, the nozzle moves to NOZZLE_CLEAN_START_POINT.
*
* Caveats: The ending Z should be the same as starting Z.
* Attention: EXPERIMENTAL. G-code arguments may change.
*
*/
//#define NOZZLE_CLEAN_FEATURE

#if ENABLED(NOZZLE_CLEAN_FEATURE)
  // Default number of pattern repetitions
  #define NOZZLE_CLEAN_STROKES 12

  // Default number of triangles
  #define NOZZLE_CLEAN_TRIANGLES 3

  // Specify positions as { X, Y, Z }
  #define NOZZLE_CLEAN_START_POINT { 30, 30, (Z_MIN_POS + 1)}
  #define NOZZLE_CLEAN_END_POINT   {100, 60, (Z_MIN_POS + 1)}

  // Circular pattern radius

```

```

#define NOZZLE_CLEAN_CIRCLE_RADIUS 6.5
// Circular pattern circle fragments number
#define NOZZLE_CLEAN_CIRCLE_FN 10
// Middle point of circle
#define NOZZLE_CLEAN_CIRCLE_MIDDLE NOZZLE_CLEAN_START_POINT

// Moves the nozzle to the initial position
#define NOZZLE_CLEAN_GOBACK
#endif

/**
 * Print Job Timer
 *
 * Automatically start and stop the print job timer on M104/M109/M190.
 *
 * M104 (hotend, no wait) - high temp = none,    low temp = stop timer
 * M109 (hotend, wait)   - high temp = start timer, low temp = stop timer
 * M190 (bed, wait)     - high temp = start timer, low temp = none
 *
 * The timer can also be controlled with the following commands:
 *
 * M75 - Start the print job timer
 * M76 - Pause the print job timer
 * M77 - Stop the print job timer
 */
#define PRINTJOB_TIMER_AUTOSTART

/**
 * Print Counter
 *
 * Track statistical data such as:
 *
 * - Total print jobs
 * - Total successful print jobs
 * - Total failed print jobs
 * - Total time printing
 *
 * View the current statistics with M78.
 */
// #define PRINTCOUNTER

// =====LCD and SD support =====
// @section lcd

/**
 * LCD LANGUAGE
 *
 * Select the language to display on the LCD. These languages are available:
 *
 * en, an, bg, ca, cn, cz, cz_utf8, de, el, el-gr, es, es_utf8, eu,
 * fi, fr, fr_utf8, gl, hr, it, kana, kana_utf8, ko_KR, nl, pl, pt,

```

```

* pt_utf8, pt-br, pt-br_utf8, ru, sk_utf8, tr, uk, zh_CN, zh_TW, test
*
* :{'en':'English', 'an':'Aragonese', 'bg':'Bulgarian', 'ca':'Catalan', 'cn':'Chinese', 'cz':'Czech',
'cz_utf8':'Czech (UTF8)', 'de':'German', 'el':'Greek', 'el-gr':'Greek (Greece)', 'es':'Spanish',
'es_utf8':'Spanish (UTF8)', 'eu':'Basque-Euskera', 'fi':'Finnish', 'fr':'French', 'fr_utf8':'French
(UTF8)', 'gl':'Galician', 'hr':'Croatian', 'it':'Italian', 'kana':'Japanese', 'kana_utf8':'Japanese
(UTF8)', 'ko_KR':'Korean', 'nl':'Dutch', 'pl':'Polish', 'pt':'Portuguese', 'pt-br':'Portuguese
(Brazilian)', 'pt-br_utf8':'Portuguese (Brazilian UTF8)', 'pt_utf8':'Portuguese (UTF8)',
'ru':'Russian', 'sk_utf8':'Slovak (UTF8)', 'tr':'Turkish', 'uk':'Ukrainian', 'zh_CN':'Chinese
(Simplified)', 'zh_TW':'Chinese (Taiwan)', 'test':'TEST' }
*/
#define LCD_LANGUAGE en

/**
* LCD Character Set
*
* Note: This option is NOT applicable to Graphical Displays.
*
* All character-based LCDs provide ASCII plus one of these
* language extensions:
*
* - JAPANESE ... the most common
* - WESTERN ... with more accented characters
* - CYRILLIC ... for the Russian language
*
* To determine the language extension installed on your controller:
*
* - Compile and upload with LCD_LANGUAGE set to 'test'
* - Click the controller to view the LCD menu
* - The LCD will display Japanese, Western, or Cyrillic text
*
* See http://marlinfw.org/docs/development/lcd\_language.html
*
* :['JAPANESE', 'WESTERN', 'CYRILLIC']
*/
#define DISPLAY_CHARSET_HD44780 JAPANESE

/**
* SD CARD
*
* SD Card support is disabled by default. If your controller has an SD slot,
* you must uncomment the following option or it won't work.
*
*/
#define SDSUPPORT

/**
* SD CARD: SPI SPEED
*
* Enable one of the following items for a slower SPI transfer speed.
* This may be required to resolve "volume init" errors.

```

```

*/
//#define SPI_SPEED SPI_HALF_SPEED
//#define SPI_SPEED SPI_QUARTER_SPEED
//#define SPI_SPEED SPI_EIGHTH_SPEED

/**
 * SD CARD: ENABLE CRC
 *
 * Use CRC checks and retries on the SD communication.
 */
//#define SD_CHECK_AND_RETRY

/**
 * LCD Menu Items
 *
 * Disable all menus and only display the Status Screen, or
 * just remove some extraneous menu items to recover space.
 */
//#define NO_LCD_MENUS
//#define SLIM_LCD_MENUS

//
// ENCODER SETTINGS
//
// This option overrides the default number of encoder pulses needed to
// produce one step. Should be increased for high-resolution encoders.
//
//#define ENCODER_PULSES_PER_STEP 4

//
// Use this option to override the number of step signals required to
// move between next/prev menu items.
//
//#define ENCODER_STEPS_PER_MENU_ITEM 1

/**
 * Encoder Direction Options
 *
 * Test your encoder's behavior first with both options disabled.
 *
 * Reversed Value Edit and Menu Nav? Enable REVERSE_ENCODER_DIRECTION.
 * Reversed Menu Navigation only? Enable REVERSE_MENU_DIRECTION.
 * Reversed Value Editing only? Enable BOTH options.
 */

//
// This option reverses the encoder direction everywhere.
//
// Set this option if CLOCKWISE causes values to DECREASE
//
//#define REVERSE_ENCODER_DIRECTION

```

```

//
// This option reverses the encoder direction for navigating LCD menus.
//
// If CLOCKWISE normally moves DOWN this makes it go UP.
// If CLOCKWISE normally moves UP this makes it go DOWN.
//
// #define REVERSE_MENU_DIRECTION

//
// Individual Axis Homing
//
// Add individual axis homing items (Home X, Home Y, and Home Z) to the LCD menu.
//
// #define INDIVIDUAL_AXIS_HOMING_MENU

//
// SPEAKER/BUZZER
//
// If you have a speaker that can produce tones, enable it here.
// By default Marlin assumes you have a buzzer with a fixed frequency.
//
// #define SPEAKER

//
// The duration and frequency for the UI feedback sound.
// Set these to 0 to disable audio feedback in the LCD menus.
//
// Note: Test audio output with the G-Code:
// M300 S<frequency Hz> P<duration ms>
//
// #define LCD_FEEDBACK_FREQUENCY_DURATION_MS 2
// #define LCD_FEEDBACK_FREQUENCY_HZ 5000

//=====LCD / Controller Selection =====
//===== (Character-based LCDs) =====
//
// RepRapDiscount Smart Controller.
// http://reprap.org/wiki/RepRapDiscount\_Smart\_Controller
//
// Note: Usually sold with a white PCB.
//
// #define REPRAP_DISCOUNT_SMART_CONTROLLER

// ULTIMAKER Controller.
//
// #define ULTIMAKERCONTROLLER

//
// ULTIPANEL as seen on Thingiverse.
//

```

```

//#define ULTIPANEL

//
// PanelOne from T3P3 (via RAMPS 1.4 AUX2/AUX3)
// http://reprap.org/wiki/PanelOne
//
//#define PANEL_ONE

//
// GADGETS3D G3D LCD/SD Controller
// http://reprap.org/wiki/RAMPS_1.3/1.4_GADGETS3D_Shield_with_Panel
//
// Note: Usually sold with a blue PCB.
//
//#define G3D_PANEL

//
// RigidBot Panel V1.0
// http://www.inventapart.com/
//
//#define RIGIDBOT_PANEL

//
// Makeboard 3D Printer Parts 3D Printer Mini Display 1602 Mini Controller
// https://www.aliexpress.com/item/Micromake-Makeboard-3D-Printer-Parts-3D-Printer-
Mini-Display-1602-Mini-Controller-Compatible-with-Ramps-1/32765887917.html
//
//#define MAKEBOARD_MINI_2_LINE_DISPLAY_1602

//
// ANET and Tronxy 20x4 Controller
//
//#define ZONESTAR_LCD      // Requires ADC_KEYPAD_PIN to be assigned to an analog
pin.
                        // This LCD is known to be susceptible to electrical interference
                        // which scrambles the display.  Pressing any button clears it up.
                        // This is a LCD2004 display with 5 analog buttons.

//
// Generic 16x2, 16x4, 20x2, or 20x4 character-based LCD.
//
//#define ULTRA_LCD

//=====
//=====
//===== LCD / Controller Selection =====
//===== (I2C and Shift-Register LCDs) =====
//=====
//=====

//

```

```

// CONTROLLER TYPE: I2C
//
// Note: These controllers require the installation of Arduino's LiquidCrystal_I2C
// library. For more info: https://github.com/kiyoshigawa/LiquidCrystal\_I2C
//
//
// Elefu RA Board Control Panel
// http://www.elefu.com/index.php?route=product/product&product\_id=53
//
//#define RA_CONTROL_PANEL

//
// Sainsmart (YwRobot) LCD Displays
//
// These require F.Malpartida's LiquidCrystal_I2C library
// https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home
//
//#define LCD_SAINSMART_I2C_1602
//#define LCD_SAINSMART_I2C_2004

//
// Generic LCM1602 LCD adapter
//
//#define LCM1602

//
// PANELOLU2 LCD with status LEDs,
// separate encoder and click inputs.
//
// Note: This controller requires Arduino's LiquidTWI2 library v1.2.3 or later.
// For more info: https://github.com/lincomatic/LiquidTWI2
//
// Note: The PANELOLU2 encoder click input can either be directly connected to
// a pin (if BTN_ENC defined to != -1) or read through I2C (when BTN_ENC == -1).
//
//#define LCD_I2C_PANELOLU2

//
// Panucatt VIKI LCD with status LEDs,
// integrated click & L/R/U/D buttons, separate encoder inputs.
//
//#define LCD_I2C_VIKI

//
// CONTROLLER TYPE: Shift register panels
//
//
// 2 wire Non-latching LCD SR from https://goo.gl/aJ4sH
// LCD configuration: http://reprap.org/wiki/SAV\_3D\_LCD

```

```

//
//#define SAV_3DLCD

//=====
//===== LCD / Controller Selection =====
//===== (Graphical LCDs) =====
//=====
//=====

//
// CONTROLLER TYPE: Graphical 128x64 (DOGM)
//
// IMPORTANT: The U8glib library is required for Graphical Display!
//   https://github.com/olikraus/U8glib_Arduino
//

//
// RepRapDiscount FULL GRAPHIC Smart Controller
// http://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller
//
#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER

//
// ReprapWorld Graphical LCD
// https://reprapworld.com/?products_details&products_id/1218
//
//#define REPRAPWORLD_GRAPHICAL_LCD

//
// Activate one of these if you have a Panucatt Devices
// Viki 2.0 or mini Viki with Graphic LCD
// http://panucatt.com
//
//#define VIKI2
//#define miniVIKI

//
// MakerLab Mini Panel with graphic
// controller and SD support - http://reprap.org/wiki/Mini_panel
//
//#define MINIPANEL

//
// MaKr3d Makr-Panel with graphic controller and SD support.
// http://reprap.org/wiki/MaKr3d_MaKrPanel
//
//#define MAKRPANEL

//
// Adafruit ST7565 Full Graphic Controller.

```

```
// https://github.com/eboston/Adafruit-ST7565-Full-Graphic-Controller/  
//  
//#define ELB_FULL_GRAPHIC_CONTROLLER  
  
//  
// BQ LCD Smart Controller shipped by  
// default with the BQ Hephestos 2 and Witbox 2.  
//  
//#define BQ_LCD_SMART_CONTROLLER  
  
//  
// Cartesio UI  
// http://mauk.cc/webshop/cartesio-shop/electronics/user-interface  
//  
//#define CARTESIO_UI  
  
//  
// LCD for Melzi Card with Graphical LCD  
//  
//#define LCD_FOR_MELZI  
  
//  
// SSD1306 OLED full graphics generic display  
//  
//#define U8GLIB_SSD1306  
  
//  
// SAV OLEd LCD module support using either SSD1306 or SH1106 based LCD modules  
//  
//#define SAV_3DGLCD  
#if ENABLED(SAV_3DGLCD)  
  //#define U8GLIB_SSD1306  
  #define U8GLIB_SH1106  
#endif  
  
//  
// Original Ulticontroller from Ultimaker 2 printer with SSD1309 I2C display and encoder  
// https://github.com/Ultimaker/Ultimaker2/tree/master/1249_Ulticontroller_Board_(x1)  
//  
//#define ULTI_CONTROLLER  
  
//  
// TinyBoy2 128x64 OLED / Encoder Panel  
//  
//#define OLED_PANEL_TINYBOY2  
  
//  
// MKS MINI12864 with graphic controller and SD support  
// http://reprap.org/wiki/MKS_MINI_12864  
//  
//#define MKS_MINI_12864
```

```

//
// Factory display for Creality CR-10
// https://www.aliexpress.com/item/Universal-LCD-12864-3D-Printer-Display-Screen-With-Encoder-For-CR-10-CR-7-Model/32833148327.html
//
// This is RAMPS-compatible using a single 10-pin connector.
// (For CR-10 owners who want to replace the Melzi Creality board but retain the display)
//
// #define CR10_STOCKDISPLAY

//
// ANET and Tronxy Graphical Controller
//
// #define ANET_FULL_GRAPHICS_LCD // Anet 128x64 full graphics lcd with rotary encoder
// as used on Anet A6
// A clone of the RepRapDiscount full graphics display but with
// different pins/wiring (see pins_ANET_10.h).

//
// MKS OLED 1.3" 128 x 64 FULL GRAPHICS CONTROLLER
// http://reprap.org/wiki/MKS_12864OLED
//
// Tiny, but very sharp OLED display
//
// #define MKS_12864OLED // Uses the SH1106 controller (default)
// #define MKS_12864OLED_SSD1306 // Uses the SSD1306 controller

//
// Silvergate GLCD controller
// http://github.com/android444/Silvergate
//
// #define SILVER_GATE_GLCD_CONTROLLER

//=====
//===== Other Controllers =====
//=====

//
// CONTROLLER TYPE: Standalone / Serial
//

//
// LCD for Malyan M200 printers.
// This requires SDSUPPORT to be enabled
//
// #define MALYAN_LCD

//

```

```

// CONTROLLER TYPE: Keypad / Add-on
//

//
// RepRapWorld REPRAPWORLD_KEYPAD v1.1
// http://reprapworld.com/?products_details&products_id=202&cPath=1591_1626
//
// REPRAPWORLD_KEYPAD_MOVE_STEP sets how much should the robot move when a key
// is pressed, a value of 10.0 means 10mm per click.
//
//#define REPRAPWORLD_KEYPAD
//#define REPRAPWORLD_KEYPAD_MOVE_STEP 10.0

//=====Extra Features =====
// @section extras

// Increase the FAN PWM frequency. Removes the PWM noise but increases heating in the
// FET/Arduino
//#define FAST_PWM_FAN

// Use software PWM to drive the fan, as for the heaters. This uses a very low frequency
// which is not as annoying as with the hardware PWM. On the other hand, if this
// frequency
// is too low, you should also increment SOFT_PWM_SCALE.
//#define FAN_SOFT_PWM

// Incrementing this by 1 will double the software PWM frequency,
// affecting heaters, and the fan if FAN_SOFT_PWM is enabled.
// However, control resolution will be halved for each increment;
// at zero value, there are 128 effective control positions.
#define SOFT_PWM_SCALE 0

// If SOFT_PWM_SCALE is set to a value higher than 0, dithering can
// be used to mitigate the associated resolution loss. If enabled,
// some of the PWM cycles are stretched so on average the desired
// duty cycle is attained.
//#define SOFT_PWM_DITHER

// Temperature status LEDs that display the hotend and bed temperature.
// If all hotends, bed temperature, and target temperature are under 54C
// then the BLUE led is on. Otherwise the RED led is on. (1C hysteresis)
//#define TEMP_STAT_LEDS

// M240 Triggers a camera by emulating a Canon RC-1 Remote
// Data from: http://www.doc-diy.net/photo/rc-1_hacked/
//#define PHOTOGRAPH_PIN 23

// SkeinForge sends the wrong arc g-codes when using Arc Point as fillet procedure
//#define SF_ARC_FIX

// Support for the BariCUDA Paste Extruder

```

```

//#define BARICUDA

// Support for BlinkM/CyzRgb
//#define BLINKM

// Support for PCA9632 PWM LED driver
//#define PCA9632

/**
 * RGB LED / LED Strip Control
 *
 * Enable support for an RGB LED connected to 5V digital pins, or
 * an RGB Strip connected to MOSFETs controlled by digital pins.
 *
 * Adds the M150 command to set the LED (or LED strip) color.
 * If pins are PWM capable (e.g., 4, 5, 6, 11) then a range of
 * luminance values can be set from 0 to 255.
 * For Neopixel LED an overall brightness parameter is also available.
 *
 * *** CAUTION ***
 * LED Strips require a MOSFET Chip between PWM lines and LEDs,
 * as the Arduino cannot handle the current the LEDs will require.
 * Failure to follow this precaution can destroy your Arduino!
 * NOTE: A separate 5V power supply is required! The Neopixel LED needs
 * more current than the Arduino 5V linear regulator can produce.
 * *** CAUTION ***
 *
 * LED Type. Enable only one of the following two options.
 *
 */
//#define RGB_LED
//#define RGBW_LED

#if ENABLED(RGB_LED) || ENABLED(RGBW_LED)
  #define RGB_LED_R_PIN 34
  #define RGB_LED_G_PIN 43
  #define RGB_LED_B_PIN 35
  #define RGB_LED_W_PIN -1
#endif

// Support for Adafruit Neopixel LED driver
//#define NEOPIXEL_LED
#if ENABLED(NEOPIXEL_LED)
  #define NEOPIXEL_TYPE NEO_GRBW // NEO_GRBW / NEO_GRB - four/three channel
  driver type (defined in Adafruit_NeoPixel.h)
  #define NEOPIXEL_PIN 4 // LED driving pin on motherboard 4 => D4 (EXP2-5 on
  Printrboard) / 30 => PC7 (EXP3-13 on Rumba)
  #define NEOPIXEL_PIXELS 30 // Number of LEDs in the strip
  #define NEOPIXEL_IS_SEQUENTIAL // Sequential display for temperature change - LED by
  LED. Disable to change all LEDs at once.
  #define NEOPIXEL_BRIGHTNESS 127 // Initial brightness (0-255)

```

```

    // #define NEOPIXEL_STARTUP_TEST // Cycle through colors at startup
#endif

/**
 * Printer Event LEDs
 *
 * During printing, the LEDs will reflect the printer status:
 *
 * - Gradually change from blue to violet as the heated bed gets to target temp
 * - Gradually change from violet to red as the hotend gets to temperature
 * - Change to white to illuminate work surface
 * - Change to green once print has finished
 * - Turn off after the print has finished and the user has pushed a button
 */
#if ENABLED(BLINKM) || ENABLED(RGB_LED) || ENABLED(RGBW_LED) ||
    ENABLED(PCA9632) || ENABLED(NEOPIXEL_LED)
    #define PRINTER_EVENT_LEDS
#endif

/**
 * R/C SERVO support
 * Sponsored by TrinityLabs, Reworked by codexmas
 */

/**
 * Number of servos
 *
 * For some servo-related options NUM_SERVOS will be set automatically.
 * Set this manually if there are extra servos needing manual control.
 * Leave undefined or set to 0 to entirely disable the servo subsystem.
 */
// #define NUM_SERVOS 3 // Servo index starts with 0 for M280 command

// Delay (in milliseconds) before the next move will start, to give the servo time to reach its
// target angle.
// 300ms is a good value but you can try less delay.
// If the servo can't reach the requested position, increase it.
#define SERVO_DELAY { 300 }

// Only power servos during movement, otherwise leave off to prevent jitter
// #define DEACTIVATE_SERVOS_AFTER_MOVE

#endif // CONFIGURATION_H

```

Por cuenta respecta a la selección de la tarjeta Ramps 1.4 se implementa en el siguiente código.

```

#ifndef BOARDS_H
#define BOARDS_H

```

```

#define BOARD_UNKNOWN -1

// RAMPS 1.3 / 1.4 - ATmega1280, ATmega2560
#define BOARD_RAMPS_OLD 3 // MEGA/RAMPS up to 1.2

#define BOARD_RAMPS_13_EFB 33 // RAMPS 1.3 (Power outputs: Hotend, Fan, Bed)
#define BOARD_RAMPS_13_EEB 34 // RAMPS 1.3 (Power outputs: Hotend0, Hotend1,
Bed)
#define BOARD_RAMPS_13_EFF 35 // RAMPS 1.3 (Power outputs: Hotend, Fan0, Fan1)
#define BOARD_RAMPS_13_EEF 36 // RAMPS 1.3 (Power outputs: Hotend0, Hotend1,
Fan)
#define BOARD_RAMPS_13_SF 38 // RAMPS 1.3 (Power outputs: Spindle, Controller
Fan)

#define BOARD_RAMPS_14_EFB 43 // RAMPS 1.4 (Power outputs: Hotend, Fan, Bed)
#define BOARD_RAMPS_14_EEB 44 // RAMPS 1.4 (Power outputs: Hotend0, Hotend1,
Bed)
#define BOARD_RAMPS_14_EFF 45 // RAMPS 1.4 (Power outputs: Hotend, Fan0, Fan1)
#define BOARD_RAMPS_14_EEF 46 // RAMPS 1.4 (Power outputs: Hotend0, Hotend1,
Fan)
#define BOARD_RAMPS_14_SF 48 // RAMPS 1.4 (Power outputs: Spindle, Controller
Fan)

#define BOARD_RAMPS_PLUS_EFB 143 // RAMPS Plus 3DYMY (Power outputs: Hotend,
Fan, Bed)
#define BOARD_RAMPS_PLUS_EEB 144 // RAMPS Plus 3DYMY (Power outputs: Hotend0,
Hotend1, Bed)
#define BOARD_RAMPS_PLUS_EFF 145 // RAMPS Plus 3DYMY (Power outputs: Hotend,
Fan0, Fan1)
#define BOARD_RAMPS_PLUS_EEF 146 // RAMPS Plus 3DYMY (Power outputs: Hotend0,
Hotend1, Fan)
#define BOARD_RAMPS_PLUS_SF 148 // RAMPS Plus 3DYMY (Power outputs: Spindle,
Controller Fan)

// RAMPS Derivatives - ATmega1280, ATmega2560

#define BOARD_3DRAG 77 // 3Drag Controller
#define BOARD_K8200 78 // Velleman K8200 Controller (derived from 3Drag
Controller)
#define BOARD_K8400 79 // Velleman K8400 Controller (derived from 3Drag
Controller)
#define BOARD_BAM_DICE 401 // 2PrintBeta BAM&DICE with STK drivers
#define BOARD_BAM_DICE_DUE 402 // 2PrintBeta BAM&DICE Due with STK drivers
#define BOARD_MKS_BASE 40 // MKS BASE v1.0
#define BOARD_MKS_BASE_15 405 // MKS v1.5 with Allegro A4982 stepper drivers
#define BOARD_MKS_BASE_HEROIC 41 // MKS BASE 1.0 with Heroic HR4982 stepper
drivers
#define BOARD_MKS_GEN_13 47 // MKS GEN v1.3 or 1.4
#define BOARD_MKS_GEN_L 53 // MKS GEN L
#define BOARD_MKS_GEN_L_V2 54 // MKS GEN L V2
#define BOARD_ZRIB_V20 504 // zrib V2.0 control board

```

```

#define BOARD_FELIX2      37 // Felix 2.0+ Electronics Board (RAMPS like)
#define BOARD_RIGIDBOARD  42 // Invent-A-Part RigidBoard
#define BOARD_RIGIDBOARD_V2 52 // Invent-A-Part RigidBoard V2
#define BOARD_SAINSMART_2IN1 49 // Sainsmart 2-in-1 board
#define BOARD_ULTIMAKER   7 // Ultimaker
#define BOARD_ULTIMAKER_OLD 71 // Ultimaker
#define BOARD_AZTEEG_X3   67 // Azteeg X3
#define BOARD_AZTEEG_X3_PRO 68 // Azteeg X3 Pro
#define BOARD_ULTIMAIN_2  72 // Ultimainboard 2.x (Uses TEMP_SENSOR 20)
#define BOARD_RUMBA       80 // Rumba
#define BOARD_BQ_ZUM_MEGA_3D 503 // bq ZUM Mega 3D
#define BOARD_MAKEBOARD_MINI 431 // MakeBoard Mini v2.1.2 is a control board
sold by MicroMake
#define BOARD_TRIGORILLA_13 343 // TriGorilla Anycubic version 1.3 based on RAMPS
EFB
#define BOARD_TRIGORILLA_14 443 // TriGorilla Anycubic version 1.4 based on RAMPS
EFB
#define BOARD_RAMPS_ENDER_4 243 // Creality: Ender-4, CR-8

//Other ATmega1280, ATmega2560

#define BOARD_CNCONTROLS_11 111 // Cartesio CN Controls V11
#define BOARD_CNCONTROLS_12 112 // Cartesio CN Controls V12
#define BOARD_CHEAPTRONIC  2 // Cheaptronic v1.0
#define BOARD_CHEAPTRONIC_V2 21 // Cheaptronic v2.0
#define BOARD_MIGHTYBOARD_REVE 200 // Makerbot Mightyboard Revision E
#define BOARD_MEGATRONICS  70 // Megatronics
#define BOARD_MEGATRONICS_2 701 // Megatronics v2.0
#define BOARD_MEGATRONICS_3 703 // Megatronics v3.0
#define BOARD_MEGATRONICS_31 704 // Megatronics v3.1
#define BOARD_RAMBO        301 // Rambo
#define BOARD_MINIRAMBO    302 // Mini-Rambo
#define BOARD_MINIRAMBO_10A 303 // Mini-Rambo 1.0a
#define BOARD_EINSY_RAMBO  304 // Einsy Rambo
#define BOARD_EINSY_RETRO  305 // Einsy Retro
#define BOARD_ELEFU_3      21 // Elefu Ra Board (v3)
#define BOARD_LEAPFROG     999 // Leapfrog
#define BOARD_MEGACONTROLLER 310 // Mega controller
#define BOARD_SCOOVO_X9H   321 // abee Scoovo X9H
#define BOARD_GT2560_REV_A  74 // Geeetech GT2560 Rev. A
#define BOARD_GT2560_REV_A_PLUS 75 // Geeetech GT2560 Rev. A+ (with auto level
probe)

//ATmega1281, ATmega2561

#define BOARD_MINITRONICS  702 // Minitronics v1.0/1.1
#define BOARD_SILVER_GATE  25 // Silvergate v1.0

// Sanguinololu and Derivatives - ATmega644P, ATmega1284P

#define BOARD_SANGUINOLOLU_11 6 // Sanguinololu < 1.2

```

```

#define BOARD_SANGUINOLOLU_12 62 // Sanguinololu 1.2 and above
#define BOARD_MELZI 63 // Melzi
#define BOARD_MELZI_MAKR3D 66 // Melzi with ATmega1284 (MaKr3d version)
#define BOARD_MELZI_CREALITY 89 // Melzi Creality3D board (for CR-10 etc)
#define BOARD_MELZI_MALYAN 92 // Melzi Malyan M150 board
#define BOARD_MELZI_TRONXY 505 // Tronxy X5S
#define BOARD_STB_11 64 // STB V1.1
#define BOARD_AZTEEG_X1 65 // Azteeg X1
#define BOARD_ANET_10 69 // Anet 1.0 (Melzi clone)

// Other ATmega644P, ATmega644, ATmega1284P
#define BOARD_GEN3_MONOLITHIC 22 // Gen3 Monolithic Electronics
#define BOARD_GEN3_PLUS 9 // Gen3+
#define BOARD_GEN6 5 // Gen6
#define BOARD_GEN6_DELUXE 51 // Gen6 deluxe
#define BOARD_GEN7_CUSTOM 10 // Gen7 custom (Alfons3 Version)
"https://github.com/Alfons3/Generation_7_Electronics"
#define BOARD_GEN7_12 11 // Gen7 v1.1, v1.2
#define BOARD_GEN7_13 12 // Gen7 v1.3
#define BOARD_GEN7_14 13 // Gen7 v1.4
#define BOARD_OMCA_A 90 // Alpha OMCA board
#define BOARD_OMCA 91 // Final OMCA board
#define BOARD_SETHI 20 // Sethi 3D_1

// Teensyduino - AT90USB1286, AT90USB1286P

#define BOARD_TEENSYLU 8 // Teensylu
#define BOARD_PRINTRBOARD 81 // Printrboard (AT90USB1286)
#define BOARD_PRINTRBOARD_REVF 811 // Printrboard Revision F (AT90USB1286)
#define BOARD_BRAINWAVE 82 // Brainwave (AT90USB646)
#define BOARD_BRAINWAVE_PRO 85 // Brainwave Pro (AT90USB1286)
#define BOARD_SAV_MKI 83 // SAV Mk-I (AT90USB1286)
#define BOARD_TEENSY2 84 // Teensy++2.0 (AT90USB1286) - CLI compile:
HARDWARE_MOTHERBOARD=84 make
#define BOARD_5DPRINT 88 // 5DPrint D8 Driver Board

#define MB(board) (defined(BOARD_##board) && MOTHERBOARD==BOARD_##board)

#endif // __BOARDS_H

```

La interfaz de inicio debe configurarse en Version.h como sigue:

```

#if ENABLED(USE_AUTOMATIC_VERSIONING)

#include "_Version.h"

#else

/**

```

```

* Marlin release version identifier
*/
#define SHORT_BUILD_VERSION ""

/**
 * Verbose version identifier which should contain a reference to the location
 * from where the binary was downloaded or the source code was compiled.
 */
#define DETAILED_BUILD_VERSION SHORT_BUILD_VERSION " (Github)"

/**
 * The STRING_DISTRIBUTION_DATE represents when the binary file was built,
 * here we define this default string as the date where the latest release
 * version was tagged.
 */
#define STRING_DISTRIBUTION_DATE "2020-06-20"

/**
 * Required minimum Configuration.h and Configuration_adv.h file versions.
 *
 * You must increment this version number for every significant change such as,
 * but not limited to: ADD, DELETE RENAME OR REPURPOSE any directive/option on
 * the configuration files.
 */
#define REQUIRED_CONFIGURATION_H_VERSION 010109
#define REQUIRED_CONFIGURATION_ADV_H_VERSION 010109

#define PROTOCOL_VERSION "1.0"

/**
 * Defines a generic printer name to be output to the LCD after booting Marlin.
 */
#define MACHINE_NAME "Filamentadora lista"
#define SOURCE_CODE_URL "https://github.com/MarlinFirmware/Marlin"

/**
 * Default generic printer UUID.
 */
#define DEFAULT_MACHINE_UUID "cede2a2f-41a2-4748-9b12-c55c62f367ff"

/**
 * The WEBSITE_URL is the location where users can get more information such as
 * documentation about a specific Marlin release.
 */
#define WEBSITE_URL "FreePlastic"

#endif // USE_AUTOMATIC_VERSIONING

```

## Anexo VII: Evidencias fotográficas

**Figura 49.**

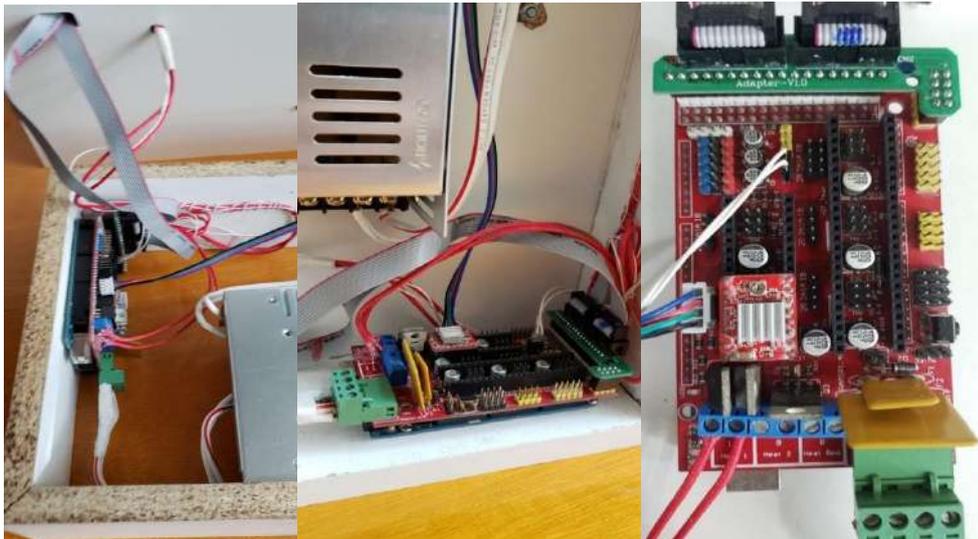
*Ensamblaje del sistema*



*Nota:* Partes de la máquina antes del ensamble.

**Figura 50.**

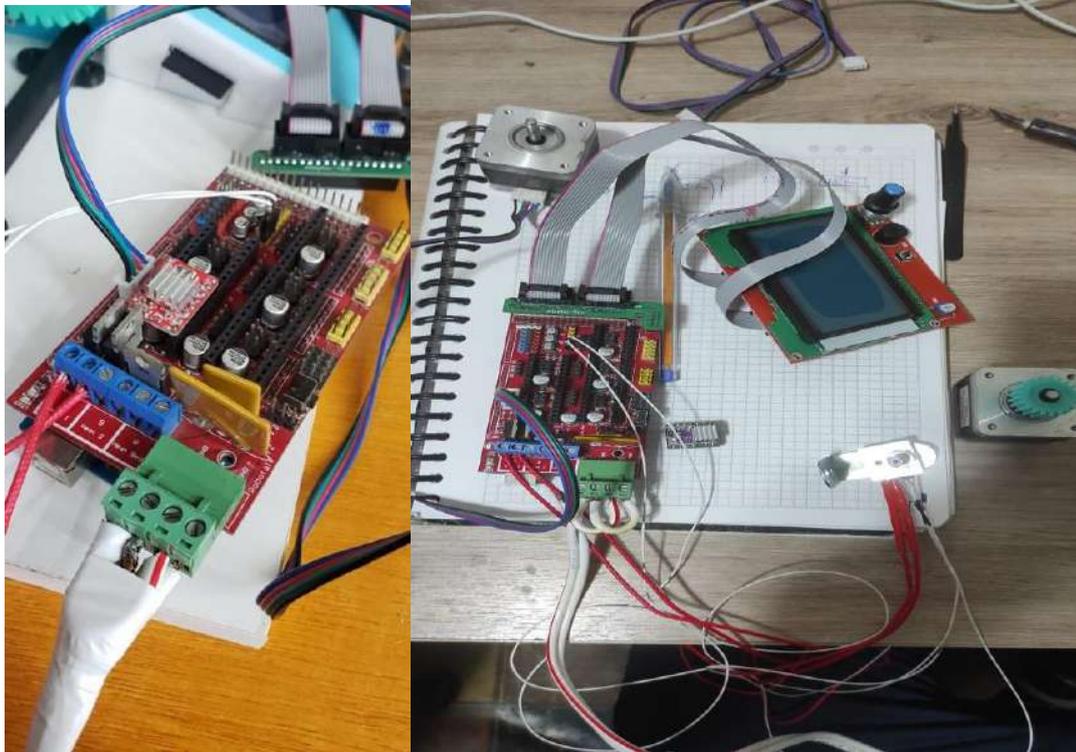
*Armado de placa de Arduino al proyecto.*



*Nota:* Representa el armado de la placa de Arduino bajo la base del proyecto.

**Figura 51.**

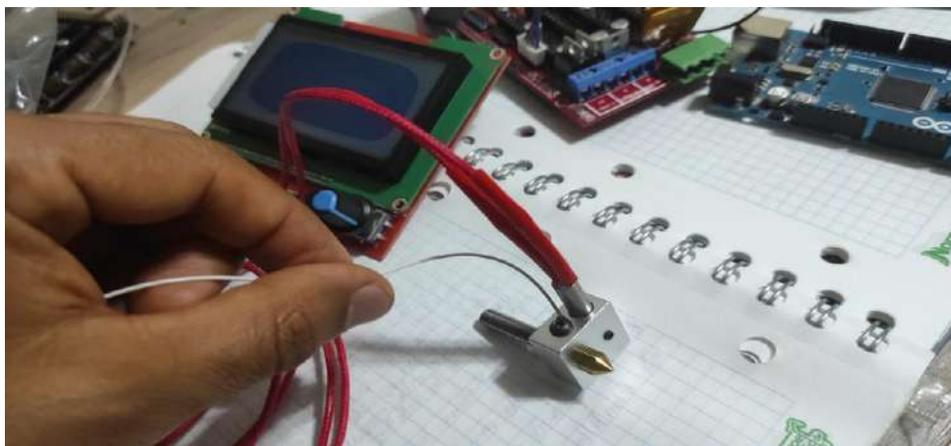
*Conexión de placa al sistema.*



*Nota:* Representa parte de las conexiones de la maquina al sistema.

**Figura 52.**

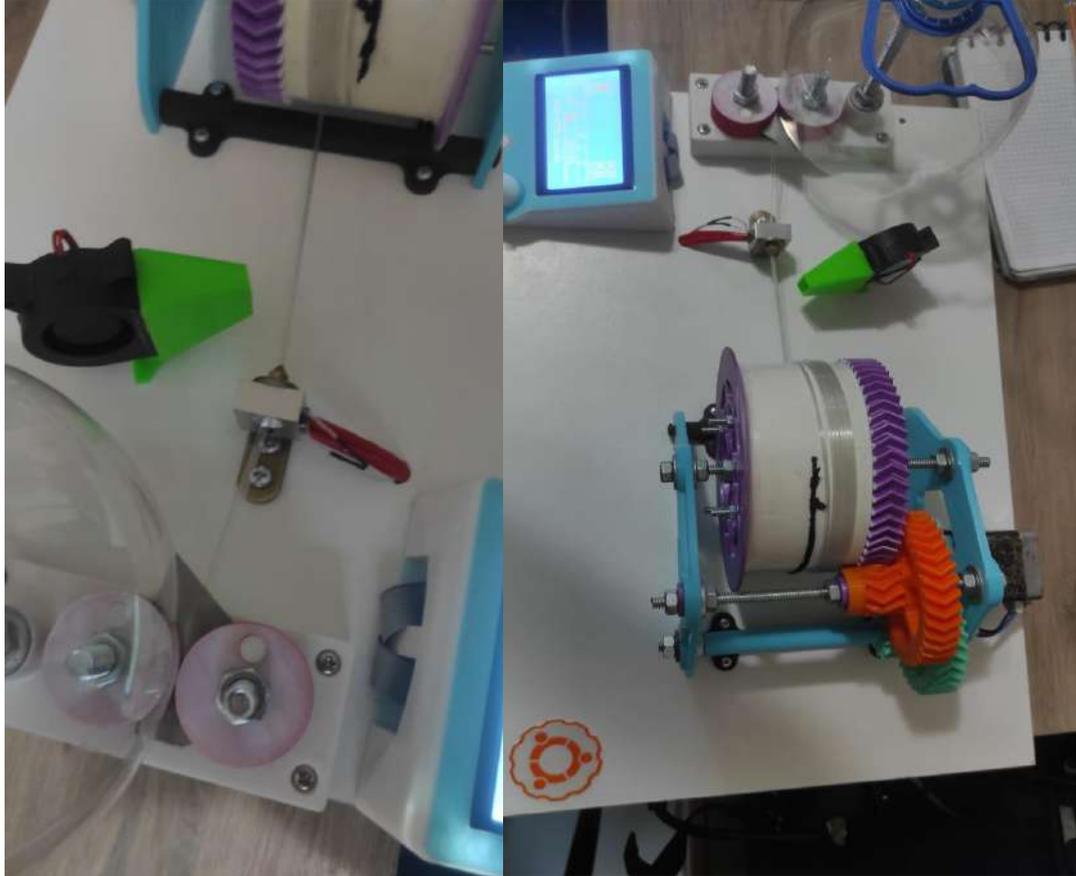
*Pruebas del sistema de Calefacción.*



*Nota:* Representa parte de las pruebas de su funcionamiento.

**Figura 53.**

Corte de cinta y Enrollado del filamento



*Nota:* Representa parte de las pruebas de su funcionamiento.