



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Diseño de sistema de control mediante un Sistema Arduino
empleado en el funcionamiento de las impresoras en tres
dimensiones con demostración práctica”

TESINA DE SEMINARIO

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA ESPECIALIZACIÓN TELECOMUNICACIONES

Presentado por:

MAX MICHAEL VALVERDE ERAZO

MILTON SAUL VILLACIS SOLIS

GUAYAQUIL – ECUADOR

2014

AGRADECIMIENTO

En primer lugar a Dios por acompañarme y guiarme siempre.

A mis padres que en todo momento me brindaron su apoyo para culminar mi carrera.

A mis amigos y compañeros que estuvieron en cada momento que necesitaba de su apoyo.

Max Valverde E.

Son profundas las palabras y mis eternos agradecimientos a quién da alegría a mi vida, a Dios dador de la sabiduría y la inteligencia. A

mi querida madre por sus sabios
consejos, mis hermanas, grandes
amigos y mi familia, por el aporte de
muchas ideas, sugerencias
técnicas, paciencia y su amistad
brindada.

Milton Villacis S.

DEDICATORIA

Dedico este trabajo a mis padres que con su ejemplo de valentía y sabiduría, lograron educarme, guiarme durante toda mi vida.

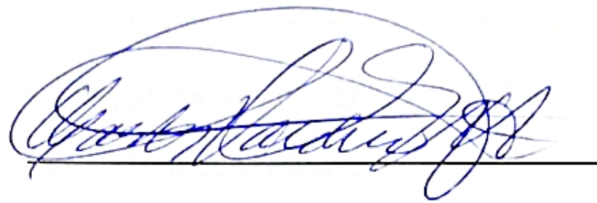
Max Valverde E.

A mí querida madre: Sra. María Isabel Solís Solís, por ser después de Dios el pilar fundamental en mi vida, la que con sus sabias enseñanzas me ha guiado por el camino del bien y a la realización de mis metas como una de ellas culminar con éxito mi carrera, que a pesar de las dificultades que se han presentado hemos cumplido con

cada meta propuesta. A mis hermanas que de alguna manera me han brindado su apoyo. Reciban este esfuerzo de parte de su hijo y hermano que las ama mucho.

Milton Villacis S.

TRIBUNAL DE SUSTENTACIÓN

A handwritten signature in blue ink, appearing to read 'Carlos Valdivieso A.', written over a horizontal line.

Ing. Carlos Valdivieso A.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

A handwritten signature in blue ink, appearing to read 'Hugo Villavicencio V.', written over a horizontal line.

Ing. Hugo Villavicencio V.

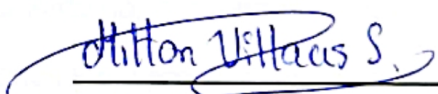
PROFESOR DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

La responsabilidad del contenido de esta tesina, me corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.

(Reglamento de Graduación de Pregrado de la ESPOL)



MAX VALVERDE ERAZO

MILTON VILLACIS SOLIS

RESUMEN

Este proyecto está basado en el diseño del sistema de control de una impresora 3D utilizando una placa con un microcontrolador llamado Arduino Mega 2560, por lo que el estudiante debe tener conocimientos técnicos, ya que se ve involucrado el avance de la tecnología.

En el capítulo 1, se encuentra la descripción del proyecto y se detalla los objetivos, limitaciones y motivación del proyecto, con la finalidad de fortalecer los conocimientos en el diseño de controladores.

En el capítulo 2, se encuentra el detalle de las características y funcionalidad del hardware y software que se utiliza en este proyecto.

En el capítulo 3, se especifica el diseño del proyecto, la instalación del entorno de desarrollo, configuración de la tarjeta Arduino, diagramas de bloques y códigos fuente de los ejercicios introductorios y del proyecto, detallando paso a paso el funcionamiento de cada uno de ellos.

En el capítulo 4, se encuentran las pruebas de los ejercicios introductorios del capítulo 3, el análisis de cada uno de ellos y las pruebas de calibración para la prueba final de la impresora.

Para finalizar nuestro trabajo realizamos las respectivas conclusiones y recomendaciones del proyecto que son necesarias para un buen funcionamiento de la impresora, además para tomar en cuenta ciertos correctivos en caso de fallas al momento de configurar el equipo.

INDICE GENERAL

RESUMEN	i
INDICE GENERAL.....	iii
ABREVIATURAS	vii
INDICE DE FIGURAS	viii
INTRODUCCION	10
CAPÍTULO 1	1
1. DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1. Descripción de la Planta.....	1
1.2. Motivación	3
1.3. Identificación del Problema	3
1.4. Objetivos Principales.....	4
1.5. Limitaciones	5
CAPÍTULO 2.....	7
2. FUNDAMENTO TEÓRICO	7
2.1. La impresora Prusa DIY	8
2.1.1. El Marco	10
2.1.2. Los ejes X, Y, Z.	10
2.1.3. El extrusor.....	11
2.1.4. Motor paso a paso	12
2.1.5. La mesa de impresión	13
2.2. La placa Arduino	14
2.2.1. Uso de la placa Arduino en el proyecto de impresión 3D. ..	15
2.3. Placa RAMPS	16
2.4. Drivers para los motores	17
2.5. Programación y funcionamiento.....	18
2.5.1. Firmware.....	19

2.5.1.2. Marlin	20
2.5.2. El Software	22
2.5.2.1. El Pronterface.....	22
2.6. El código G.....	25
2.6.1. Instrucciones en código G	26
CAPITULO 3.....	27
3. DISEÑO DEL PROYECTO	27
3.1. Características del sistema	28
3.2. Entorno de comunicación con Arduino	29
3.2.1. Instalacion del IDE	30
3.2.2. Carga del ejemplo Blink.....	36
3.2.3. Diagramas del ejercicio	37
3.2.4. Desarrollo del algoritmo	38
3.2.5. Código fuente	39
3.3. Desarrollo de ejemplos y ejecución del proyecto	39
3.3.1. Luces de auto fantástico	39
3.3.1.1. Diagramas	40
3.3.1.2. Desarrollo de algoritmos	42
3.3.1.3. Código fuente	42
3.3.2. Control de movimiento de un motor de paso	43
3.3.2.1. Diagramas	44
3.3.2.2. Desarrollo del algoritmo	46
3.3.2.3. Código fuente	46
3.3.3. Sistema de Control de Impresora 3D mediante la Plataforma Arduino 47	
3.3.3.1. Descripción del proceso	48
3.3.3.2. Diagramas	53
3.3.3.3. Desarrollo del algoritmo	54

CAPÍTULO 4.....	56
4. IMPLEMENTACIÓN, EJERCICIOS DE PRUEBA Y SIMULACIÓN DEL PROYECTO	56
4.1. Ejecución ejemplo Blink	57
4.1.1. Componentes necesarios	58
4.1.2. Implementación	58
4.1.3. Discusión del ejemplo Blink	59
4.2. Luces del auto fantástico.....	59
4.2.1. Componentes necesarios	59
4.2.2. Implementación	60
4.2.3. Discusión del ejemplo del auto fantástico.....	60
4.3. Control de movimiento de un motor de paso.....	61
4.3.1. Componentes necesarios	62
4.3.2. Implementación	62
4.3.3. Discusión del ejemplo de control de movimiento de un motor de paso.....	63
4.4. Sistema de control de impresora 3D mediante la plataforma Arduino.....	63
4.4.1. Implementación	64
4.4.2. Discusión sobre el proyecto de impresión 3D controlado por plataforma Arduino.	65
CONCLUSIONES	69
RECOMENDACIONES.....	72
ANEXOS.....	75
ANEXO 1.- Arduino Mega 2560 PIN diagram	76
ANEXO 2.- Arduino Mega 2560 PIN mapping table.....	77
ANEXO 3. Diagrama esquemático del pololu A4988	79
ANEXO 4.- Diagrama de pines RAMPS 1.4	80
ANEXO 5.- Diagrama de conexiones de la RAMPS	81

ANEXO 6.- Especificaciones Técnicas L293D.....	82
BIBLIOGRAFÍA.....	85

ABREVIATURAS

ABS	Acrilonitrilo Butadieno Estireno
PLA	Poliácido láctico
Vin	Voltaje de entrada
USB	Universal Serial Bus
DIY	Do It Yourself
IDE	Integrated Development Environment
RAMPS	RepRap Arduino Mega Pololu Shield
PID	Proporcional Integral Derivativo
SD	Secure Digital
LCD	Liquid Cristal Display
PC	Personal Computer
CNC	Control Numérico Computarizado
COM	Communications
LPT	Line Print Terminal
LED	Light Emitting Diodo
PID	Proporcional Integrador Derivativo

INDICE DE FIGURAS

Figura 2.1. Impresora Prusa DIY [26]	9
Figura 2.2. Partes de la impresora 3D Prusa [1]	10
Figura 2.3. Detalle de la extrusora en funcionamiento.....	12
Figura 2.4. Motor de paso [27]	13
Figura 2.5. Tarjeta Arduino Mega [7]	15
Figura 2.6. La Placa RAMPS [28]	17
Figura 2.7. Pololu A 4988 [29].....	17
Figura 2.8. Pieza 3D impresa con Marlin y Sprinter [30]	21
Figura 2.9. Control de movimientos de los ejes en Pronterface.....	23
Figura 2.10. Calibración y monitoreo de parámetros en Pronterface	23
Figura 2.11. Diagrama de bloques del funcionamiento de la impresora 3D..	24
Figura 2.12. Ejemplo de uso del Pronterface	25
Figura 2.13. Tabla de algunos comandos del código G	26
Figura 3.1. Flujo de trabajo de la impresión en 3D[1]	29
Figura 3.2. Archivo ejecutable Arduino	31
Figura 3.3. Ventana IDE	31
Figura 3.4. Administrador de dispositivos	32
Figura 3.5. Actualización de Software de controlador	33
Figura 3.6. Administrador de dispositivos	34
Figura 3.7. Puerto serial en Arduino	35
Figura 3.8. Selección de tarjeta Arduino MEGA.....	35
Figura 3.9. Carga del ejemplo Blink	36
Figura 3.10. Ejemplo Blink en IDE	37
Figura 3.11. Diagrama de bloques.....	37

Figura 3.12. Diagrama ASM	38
Figura 3.13. Diagrama de bloques.....	40
Figura 3.14. Diagrama ASM	40
Figura 3.15. Diagrama de conexiones	41
Figura 3.16. Diagrama esquemático	41
Figura 3.17. Diagrama de bloques.....	44
Figura 3.18. Diagrama ASM	44
Figura 3.19. Diagrama de conexiones con Arduino.	45
Figura 3.20. Esquemático de la conexión del driver SN754410 con motor de paso bipolar.	45
Figura 3.21. Elección del Baudrate en Pronterface.....	49
Figura 3.22. Elección del Baudrate en Marlin	50
Figura 3.23. Conexiones de los motores a la RAMPS	51
Figura 3.24. Configuración del centro de Impresión en Slic3r	52
Figura 3.25. Diagrama de bloques del proceso de impresión	53
Figura 3.26. Diagrama de conexiones de RAMPS [31]	53
Figura 3.27. Diagrama ASM	54
Figura 4.1. Diagrama interno de la tarjeta led pin 13.	58
Figura 4.2. Ejemplo Blink en ejecución	58
Figura 4.3. Luces del auto fantástico	60
Figura 4.4. Control de movimiento de motor de paso	62
Figura 4.5. Sistema completo de impresión 3D	64
Figura 4.6. Impresión 3D de un escorpión	64

INTRODUCCION

La impresora 3D es un dispositivo con la capacidad de crear un objeto tridimensional con el método de adición de material, se puede crear cualquier objeto que imaginemos, para esto debemos tener el modelo a imprimir en formato digital, que contenga las características volumétricas del modelo a imprimir. Diversos materiales se pueden utilizar para imprimir, depende del tipo de impresora, las de menor costo funcionan con termoplásticos como el PLA o ABS.

Hoy en día la impresión 3D no es fenómeno masivo, ya que aún no se ha convertido en un objeto cotidiano, son varias las personas que aún desconocen la funcionalidad de la impresión 3D, sin embargo día a día se ve la importancia de realizar este tipo de impresiones que va a aumentando debido a la versatilidad que tienen este tipo de impresora, las cuales están siendo utilizados en campos como: la medicina, la industria, arquitectura, ingeniería, educación, entre otros.

CAPÍTULO 1

1. DESCRIPCIÓN GENERAL DEL PROYECTO

1.1.Descripción de la Planta

Un sistema de control para el funcionamiento de una impresora 3D tiene como objetivo llevar a cabo las tareas dadas desde el ordenador para controlar con precisión los diferentes movimientos y trayectorias de las extrusoras que realizarán la impresión de un determinado objeto. El sistema de referencia es un sistema cartesiano tridimensional en los ejes X, Y, Z para el movimiento en línea recta de las extrusoras.

El sistema está compuesto en su parte física por un acople de tarjetas electrónicas RAMPS-Arduino. El microcontrolador se encuentra embebido en la placa Arduino, la cual permite de manera sencilla la programación del chip para que el programa realice las rutinas especificadas, el chip que se controla a través del sistema Arduino es un microcontrolador de la familia ATMega muy utilizado por su versatilidad y facilidad de programación. Se verifica en el proyecto la importancia de incorporar los conocimientos sobre microcontroladores.

[3]

En la actualidad la placa Arduino es la número uno en el mercado de tarjetas programables no solamente por la amplia gama de ventajas que posee frente a la programación del controlador sino también por la diversa cantidad de tipos de placas para aplicaciones específicas.

En nuestro caso la placa utilizada es Arduino MEGA la cual contiene un controlador ATmega2560 y nuestro propósito es desarrollar un sistema de control para el funcionamiento de la impresora 3D.

Las diversas aplicaciones de la impresión 3D en la última década y la oportunidad de poder modificar las impresoras para necesidades específicas han determinado un gran empuje en el desarrollo de esta

tecnología para ponerla al alcance de los usuarios no familiarizados con la programación o la electrónica.

1.2. Motivación

La motivación para realizar este proyecto es el deseo de investigar y conocer cómo funciona la impresora 3D. En este trabajo se detallará el diseño del sistema de control de la impresora utilizando el microcontrolador Arduino.

Algunas de las motivaciones que nos empujó en la investigación de este proyecto, es la curiosidad por conocer la funcionalidad del controlador, realizar el diseño del control de rutinas de impresión mediante el sistema Arduino, y de esta manera poner en práctica y desarrollar nuestros conocimientos adquiridos en el área de la electrónica durante nuestros estudios, para alcanzar un mejor desenvolvimiento en el ámbito profesional. [8]

1.3. Identificación del Problema

El diseño de un sistema de control implica la programación de un microcontrolador que realice tareas específicas y la utilización de programas que sirvan como interface al usuario para el manejo de la impresora 3D, la elección de un controlador apropiado que se acople

al programa de interface de usuario, sugiere una investigación minuciosa que permita seleccionar los componente más sencillos que se adapten a las expectativas requeridas. Existe una amplia gama de software para el manejo de impresoras 3D, la mayoría de licencia libre, sin embargo algunas podrían causar un poco de dificultad durante la configuración de sus parámetros de funcionamiento.

Para tener un efectivo proceso de impresión 3D se realizarán pruebas reales con motores de paso lo cual permitirá relacionarse de mejor manera con la programación del controlador y la graduación de los parámetros del software.

1.4. Objetivos Principales

El objetivo principal en este proyecto es obtener un adecuado diseño de un sistema de control de una impresora 3D empleando la plataforma Arduino.

Los objetivos específicos del proyecto son los siguientes:

- Analizar las componentes del sistema.
- Establecer la operatividad de la plataforma Arduino en el desarrollo de sistemas de control

- Emplear software que permitan la comunicación del controlador con la impresora 3D
- Aplicar la programación a los procesadores de las tarjetas Arduino.
- Probar el sistema de control a través de la tarjeta Arduino para estudiar el comportamiento aislado de cada rutina.

1.5.Limitaciones

Una de las limitaciones que se presentan en este proyecto es la escasa cantidad de programas simuladores además de la ausencia de instrumentos para pruebas de ensayo. Por otro lado la investigación se limita a los aspectos de programación y creación de rutinas en el controlador a través de softwares ya existentes, y no discute la construcción física de la impresora 3D.

Si bien existen limitaciones también en cuanto a las dimensiones de las piezas reproducidas respecto al tamaño de la impresora, el mayor de los inconvenientes es el acabado de la pieza el cual debe determinarse cuidadosamente dentro de los parámetros de impresión del software, sin embargo un análisis de prueba y error podría brindarnos mejores experiencias.

Las impresiones en varios colores sugieren otro reto, que aún no ha sido abordado por este tipo de impresoras, además de no poder utilizar otro material de impresión que el plástico ABS o PLA.

CAPÍTULO 2

2. FUNDAMENTO TEÓRICO

Para la realización de este proyecto se deberán considerar aspectos tales como un buen manejo del software, el firmware y los elementos controladores de los motores de paso.

Por tal motivo en este capítulo explicaremos acerca de las partes de la impresora, los programas necesarios para una adecuada consecución del flujo de trabajo del proceso de la impresión, así como conocer las características y generalidades de las distintas familias de placas Arduino.

La impresión 3D va evolucionando desde sus orígenes y también su importancia en el sector industrial ha aumentado, desde sus inicio a

principios de los 90 cuando se utilizaba la fabricación por capas o también llamado prototipado rápido ya se vislumbraba una nueva era tecnológica para la impresión y luego en el 2004 se creó el proyecto RepRap para la construcción de una máquina que se puede autoreplicar. [1].

La tecnología RepRap es de licencia libre, esto permite que haya clones y versiones mejoradas de manera que se van realizando actualizaciones periódicas, además permite que los usuarios tengan la facilidad de diseñar o encontrar un diseño ya establecido para elaborarlo en la impresora.

2.1. La impresora Prusa DIY

La impresora PRUSA es una de las variedades que surgieron del proyecto de replicación de prototipado rápido de impresión 3D RepRap, es una máquina de tercera generación muy útil por su facilidad de construcción y su bajo coste relativo. Refiere muchas ventajas por sobre sus predecesoras y está en constante evolución, esta variedad es ideal para el aprendizaje educativo en el ámbito de la programación, manejo de interface y funcionamiento de motores.

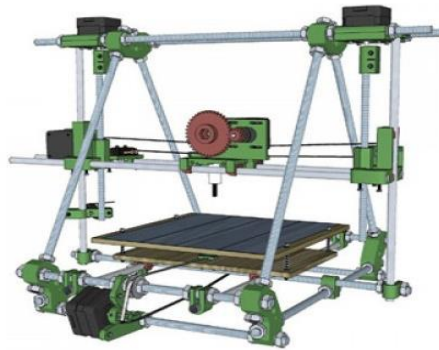


Figura 2.1. Impresora Prusa DIY [26]

La impresora Prusa forma parte del grupo de impresoras autoreplicables, por este motivo se puede encontrar en internet los archivos que contienen la información necesaria para la impresión de cada pieza plástica.

La impresora PRUSA consta de las siguientes partes principales: [1]

- Marco
- Ejes x, y ,z
- Motores de paso
- Uniones de marco
- Fines de carrera
- Extrusora
- Mesa de impresión
- Soportes

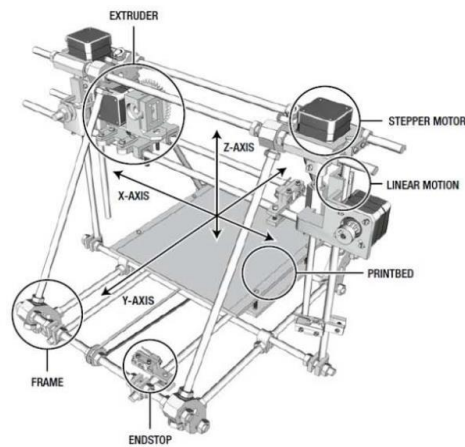


Figura 2.2. Partes de la impresora 3D Prusa [1]

2.1.1. El Marco

El marco es el bastidor de la impresora y la que le da rigidez a la misma, es donde se pondrán los tres ejes que serán movidos por motores de paso. El bastidor viene con roscas de interconexión en los vértices donde se colocaran también las partes para sostener los ejes Y y Z. [16]

2.1.2. Los ejes X, Y, Z.

Los ejes X y Y tienen un grado de libertad y son controlados cada uno por un motor de paso, cada uno de estos motores tiene una correa que mueve la mesa de impresión en la dirección determinada. El eje Z se controla a través de 2 motores de paso conectados a los ejes verticales. La placa inferior de la mesa de

impresión tiene cuatro casquillos unidos que se pegan a los ejes y les permiten su movilidad. [16]

2.1.3. El extrusor

Es el dispositivo que nos permite fundir el plástico, consta de una parte superior fría que suministra el hilo plástico a través de engranajes movidos por un motor de paso los cuales empujan este hilo de plástico a través de un husillo que se calienta a tal punto de fundirlo sin quemarlo, logrando un filamento muy consistente y delgado de plástico, para los materiales plásticos utilizados la temperatura de fundición es de 220 a 230 °C para ABS y de 170 a 180 °C para PLA, es recomendable que la mesa de impresión también se encuentre lo suficientemente caliente para que la lámina plástica quede fija en ella y se pueda ir acumulando las sucesivas capas de filamento durante la impresión, en algunos casos la falta de una adecuada temperatura en la mesa de impresión ocasiona que el mismo extrusor barra las capas que quedan e impidan el proceso de montaje de capas. [5]

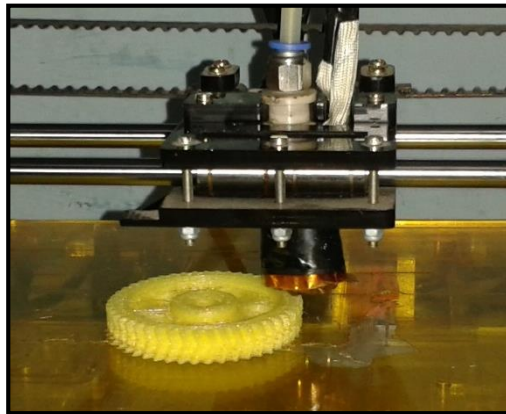


Figura 2.3. Detalle de la extrusora en funcionamiento.

2.1.4. Motor paso a paso

En esta clase de motores la excitación de las bobinas ubicadas en el estator depende del controlador, de este modo es posible controlar una amplia variedad de movimientos de grados por paso, permitiendo inclusive movimientos muy pequeños (desde $1,8^\circ$) hasta movimientos más grandes (de hasta 90°). La funcionalidad de estos motores es convertir una cantidad de pulsos eléctricos en desplazamiento angulares discretos, es decir que avanza una determinada cantidad de grados dependiendo de las entradas de control manejadas mediante un microcontrolador. La velocidad depende de la frecuencia de los pulsos aplicados.



Figura 2.4. Motor de paso [27]

Los drivers que se utilizan tienen un modo de control a corriente constante de manera que la fuerza que hace el motor depende de la corriente, y no de la tensión, de alimentación. Los valores de tensión y corriente que da el fabricante te permiten conocer los parámetros máximos del motor. Así pues, si operas a esa tensión y a esa corriente nominal, el motor dará el par nominal y su vida útil será larga; por el contrario, si lo alimentamos a más de eso se sobrecalentará y durará menos tiempo. [5].

2.1.5. La mesa de impresión

Es el lugar donde la extrusora colocara el material plástico caliente mientras realiza el barrido de la figura, esta superficie debe ser lo más liso posible para evitar desgaste y mala calidad de la impresión, se puede colocar una cama caliente para mejorar los resultados de la impresión y evitar errores de endurecimiento de plástico durante el proceso. [16]

2.2. La placa Arduino

La placa Arduino es una plataforma de hardware libre con entradas y salidas analógicas y digitales con un microcontrolador (Atmega168, Atmega328, Atmega1280, ATmega2560, depende del tipo de Arduino), que se encuentra embebida en la placa y puede ser programado mediante lenguaje de programación Arduino basado en Wiring y un entorno desarrollado basado en Processing, que permite crear prototipos en software y hardware de fácil manejo.

Su fácil programación y un entorno de trabajo agradable permiten que sea utilizada por aficionados a la electrónica, artistas o ingenieros en proyectos multidisciplinarios. Los proyectos realizados con Arduino pueden ejecutarse sin necesidad de conectar a una computadora, además que permite la comunicación con distintos softwares. [17]

Un punto importante de destacar es la energización de la placa Arduino la cual puede darse a través de un puerto USB o conexión externa mediante el pin VIN, la diferencia entre ambas, radica en la cantidad de corriente que se limita al sistema, y esta depende de las características de los dispositivos mencionados. Mediante la conexión al puerto USB se tiene limitada la intensidad de corriente con apenas 0,5A, por el contrario con la fuente de alimentación externa conectada

al pin VIN, se opera con una intensidad de corriente proveniente a la de la fuente. [4]

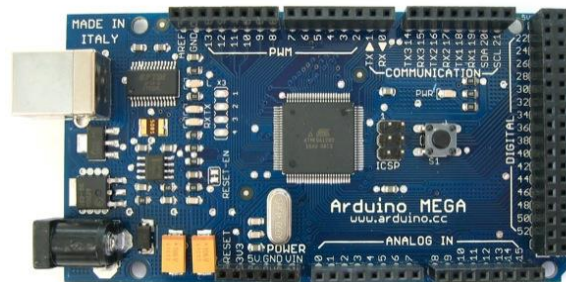


Figura 2.5. Tarjeta Arduino Mega [7]

2.2.1. Uso de la placa Arduino en el proyecto de impresión 3D.

Una gran parte de las impresoras 3D usan controladores de la familia Arduino por su fácil programación, su gran capacidad de procesamiento y su versatilidad prácticamente en todos los trabajos que se puedan requerir.

Todas las impresoras del proyecto RepRap pueden adaptarse para utilizar la plataforma Arduino en el del proceso un proyecto de impresión 3D, esta plataforma incluye además un entorno de desarrollo (IDE) con una interface amigable desde la cual cargaremos las órdenes específicas (firmware) que realizará el microcontrolador.

2.3.Placa RAMPS

RAMPS (RepRap Arduino Mega Pololu Shield), es una placa electrónica que sirve de interfaz entre una placa Arduino-Mega y los controladores de los motores de paso (Pololus) de la impresora 3D. La estructura de su diseño permite la colocación de la tarjeta Arduino en su parte superior, además de permitir la opción de conectarse hasta con cinco tarjetas drivers Pololus. El Arduino Mega representa el cerebro de la máquina, la RAMPS por su parte además de proporcionar el hardware de conmutación entre el heating y la cama caliente, permite el acople de los controladores de los motores de paso que manejarán los ejes y la impresora. [1].

Destacan entre las características de las RAMPS las siguientes:

- Utiliza el estándar de la electrónica Arduino, haciendo que la plataforma, sea más fácil de usar.
- Salida de voltaje adicional para ventiladores intercambiables
- Posee pines para futuras expansiones
- Las conexiones de los motores, finales de carrera y termistores son del tipo de las conexiones para servos, haciendo que la conexión/desconexión sea sencilla.

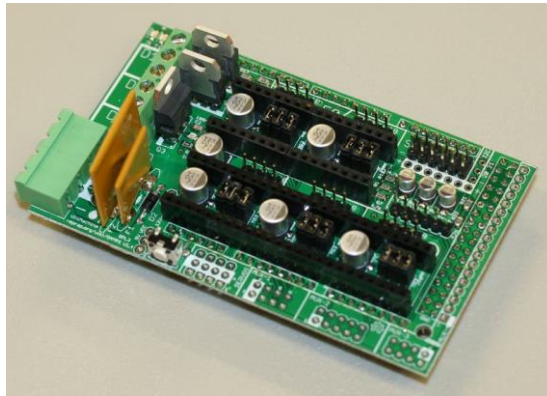


Figura 2.6. La Placa RAMPS [28]

2.4. Drivers para los motores

Los drivers son pequeños microcontroladores que van conectados a la placa RAMPS o la placa Arduino, tienen la capacidad de controlar los motores de paso, el driver más utilizado y más conocido es el modelo Pololu A4988.



Figura 2.7. Pololu A 4988 [29]

El modelo A4988 es un driver completo, con la capacidad de transmitir salidas desde 8V Y $\pm 1A$ hasta 35V y $\pm 2A$, diseñado para controlar los motores de paso en una amplia gama de resoluciones por ejemplo, revolución completa, media revolución, un cuarto de revolución, un octavo de revolución y un dieciseisavo de revolución por paso.[25]

Llevar integrados un potenciómetro para regular la intensidad de corriente que se va a suministrar al motor, que estará en función de la carga a mover.

Existen otra clase de modelos y entre ellos se encuentra el modelo Pololu A4983, que se diferencia del Pololu A4988 en que contiene un diodo de protección contra sobre-corrientes que no posee el primero.
[5]

2.5.Programación y funcionamiento

En esta sección se describe en primer lugar el firmware que debe ser cargado al microcontrolador y que contendrá el paquete de órdenes específicas para el proceso de impresión, es allí donde podremos elegir las diferentes configuraciones para los tipos de impresoras que son compatibles con Arduino, elección del tipo de placa Arduino utilizada, configuraciones del termistor, forma de trabajo de los fines de carrera,

velocidad de transferencia de datos, calibración de ejes, etc. En segundo lugar mencionaremos el software de control que debe ser instalado en el ordenador desde donde se maneja el hardware conformado por los motores de paso y la extrusora, este software envía también al procesador la información del objeto a imprimirse en formato Gcode.

2.5.1. Firmware

El firmware es el conjunto de órdenes específicas que debe cargarse a un microcontrolador para que los procesos electrónicos puedan ser llevados a cabo. En nuestro caso particular de la impresora 3D tiene la función principal de interpretar los comandos del código G provenientes del software de control y enviarlos a la electrónica lo que se traduce en movimientos específicos de los motores de paso.

Entre los firmware más utilizados en los proyectos de impresión 3D se encuentran el Sprinter y el Marlin.

2.5.1.1. Sprinter

Es uno de los firmwares más populares para los proyectos de impresión 3D, es muy fácil de configurar

aún para los no programadores, a través del archivo `configuration.h`. Está diseñado para controladores electrónicos como RAMPS, pero es compatible también con otras plataformas. Las características principales que destacan al Sprinter son:

- Aceleración de movimientos para impresiones rápidas y controladas.
- Soporte para fines de carrera.
- Control de temperatura PID.
- Carga del código G por SD. [1]

2.5.1.2. **Marlin**

Marlín es básicamente una evolución de Sprinter, que resulta en un manejo de configuración más complicado por la cantidad de opciones que permite, pero que presenta muchas mejoras entre las que se destacan impresiones más limpias y estampados más suaves. Posee también funciones de interpretación que anticipa los movimientos futuros para evitar interrupciones innecesarias. [1]

Entre otras características importantes en Marlin se tiene:

- Alto intervalo de pasos.
- Anticipación de movimientos y alta velocidad en curvas.
- Interrupción por protección de temperatura.
- Soporte completo para fines de carrera.
- Soporte para display LCD (ideal 20×4)
- Soporte para arcos.
- Sobremuestreo de temperatura.
- Selección de temperatura dinámica o “Auto Temperatura”.
- Ajuste PID. [24]



Figura 2.8. Pieza 3D impresa con Marlin y Sprinter [30]

2.5.2. El Software

Como hemos mencionado anteriormente además del firmware hace falta un software que sirva de interface entre el usuario y la impresora y desde donde se enviarán al firmware los archivos en código G del objeto a imprimir.

Existen muchos softwares de control gratuitos, sin embargo en este proyecto se optó por utilizar el software Pronterface por ser uno de los más sencillos en cuanto a su operación.

2.5.2.1. El Pronterface

Este programa nos permite controlar la impresora 3D a través de una PC, es muy utilizado por los usuarios de impresoras RepRap por su facilidad de manejo.

Dentro de su entorno gráfico existe un área de control de movimientos desde donde se puede manejar los movimientos de los motores con aumentos de pasos en múltiplos de 10, además de tener la posibilidad de volver a la posición inicial de referencia (home).

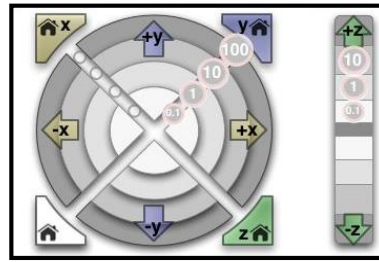


Figura 2.9. Control de movimientos de los ejes en Pronterface

El Pronterface permite también calibrar los valores de temperatura de extrusión y de la cama caliente y monitorearlos durante todo el proceso; así como modificar la velocidad de transmisión de datos y sincronizarla con la que se especificó en el firmware y permite realizar pruebas de extrusión especificando la cantidad de filamento que se desea extruir.



Figura 2.10. Calibración y monitoreo de parámetros en Pronterface

Pronterface puede trabajar con archivos G-code, sin embargo también admite archivos en formato .stl. Para poder convertirlo a G-code Pronterface utiliza a su vez un programa especial de capeado en este caso el Slic3r (aunque pueden ser otros también de licencia libre) que convierte el archivo .stl en archivos G-code que son interpretados por el firmware como movimientos específicos para los ejes y la extrusora. La relación entre el firmware y el software se la interpreta mejor en el siguiente diagrama de bloques:

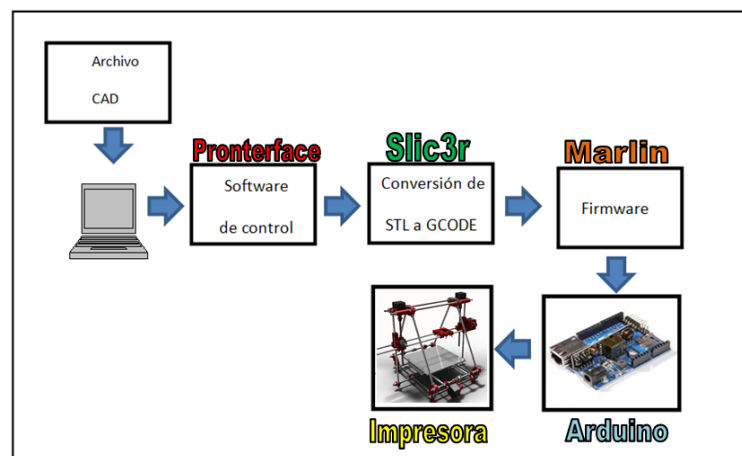


Figura 2.11. Diagrama de bloques del funcionamiento de la impresora 3D.

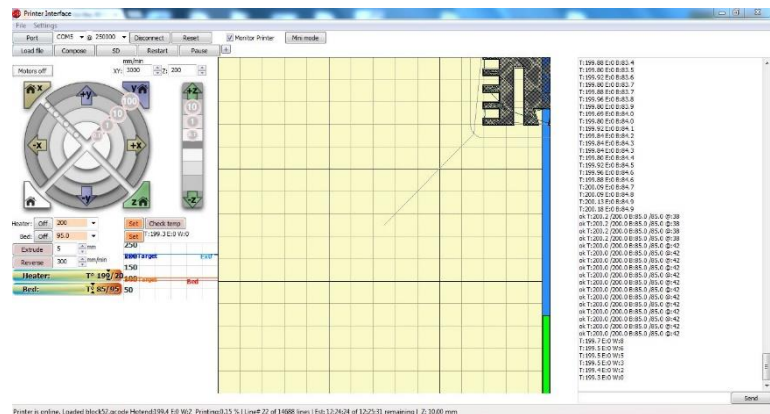


Figura 2.12. Ejemplo de uso del Pronterface

2.6.El código G

El código G es el lenguaje nativo que se utiliza en las máquinas de Control Numérico Computarizado (CNC). Al escribir un programa en este lenguaje lo que se crea listas de órdenes que generan acciones específicas de carácter geométrico.

Existen también instrucciones llamadas código M, sus nombres se derivan del hecho que son instrucciones G (generales) y M (misceláneas). Las funciones del tipo G sirven para general acciones de movimiento (giros, avances, arcos, etc.) y las de tipo M generan acciones que no son de movimiento. [20]

2.6.1. Instrucciones en código G

Las instrucciones de código G empiezan precisamente con la letra G seguida de dos números del 00 al 99.

A continuación se muestra una tabla donde se describirán algunas de estas instrucciones. [21]

Comando	Descripción
G00	Interpolación Lineal Rápida.
G01	Interpolación lineal a la velocidad programada en el registro F.
G02	Movimiento Circular en el sentido horario <u>Feedrate</u> .
G03	Movimiento Circular en el sentido anti-horario <u>Feedrate</u> .
G04	Es una demora o una pausa con un tiempo específico.
G17	Selección del Plano X-Y
G18	Selección del Plano X-Z
G19	Selección del Plano Y-Z
G40	Compensación anulada, o al centro de la línea de desplazamiento.
G41	Compensación a la Izquierda de la línea de desplazamiento.
G42	Compensación a la Derecha de la línea de desplazamiento.
G70	Unidad de Datos expresados en Pulgadas.
G71	Unidad de Datos expresados en Milímetros.
G90	Desplazamiento en Modo Absoluto.
G91	Desplazamiento en Modo Incremental o Relativo.

Figura 2.13. Tabla de algunos comandos del código G

CAPITULO 3

3. DISEÑO DEL PROYECTO

En este capítulo se presenta un grupo de ejemplos de programación y control de motores utilizando el Arduino MEGA que contiene el microcontrolador ATmega2560, trataremos la comunicación del Arduino con la PC y luego a través del Programa Pronterface evidenciaremos la correcta programación del micro.

Con la finalidad de prepararnos adecuadamente para la implementación y prueba de la impresora 3D, se realizarán los siguientes ejercicios:

- Comunicación e interface de Usuario-Arduino
- Control de frecuencia de luz en diodo.
- Control secuencial de encendido de luces.

- Control de movimientos de motores de paso

Para cada ejercicio se desarrollaran los respectivos diagramas de bloques y diagrama ASM explicando su algoritmo, además del código fuente que lo describe.

3.1.Características del sistema

El funcionamiento de una impresora 3D requiere de un algoritmo que verifique el cumplimiento de algunos procesos antes de obtener el resultado final.

Para saber cómo funciona este algoritmo debemos conocer algunas de las características del proceso de trabajo del sistema.

Para realizar la impresión de un modelo en 3D se carga al Pronterface un archivo en formato .stl de dicho modelo, el modelo a imprimir debe estar correctamente escalado, pues algunas impresiones podrían quedar fuera del rango de la base de impresión; este escalado se puede modificar en el Slic3r, que viene ya anexado en el Pronterface y desde este último se puede visualizar la posición del objeto a imprimirse en 2D o 3D.

Con el programa Slic3r el modelo .stl es convertido a rebanadas que vienen dadas en formato G-code que son interpretadas por el firmware Marlin y da las órdenes a la extrusora de como imprimir cada rebanada y cuanta cantidad de material de impresión debe colocar.

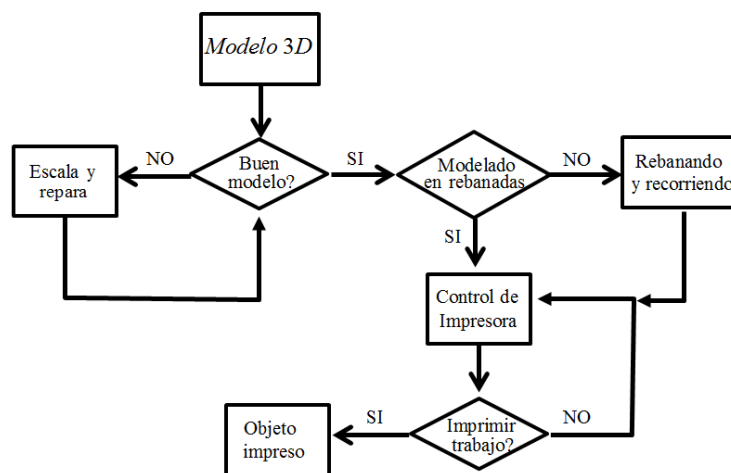


Figura 3.1. Flujo de trabajo de la impresión en 3D[1]

3.2. Entorno de comunicación con Arduino

IDE Integrated Development Environment (Entorno de Desarrollo Integrado) este entorno es un conjunto de programas que permite programar de una manera menos complicada la tarjeta Arduino, de tal forma que no se necesita ser un experto en desarrollo de software para crear los programas de control de la tarjeta.

Esta aplicación es diseñada en Java, es decir que no depende del sistema operativo, entonces es compatible para toda computadora con sistema operativo que soporte Java.

3.2.1. Instalacion del IDE

La instalación y configuración del IDE para establecer la comunicación entre el Arduino y PC, se la realiza de la siguiente manera:

Primero se descarga el archivo comprimido para la instalación, se puede realizar directamente desde la página oficial de Arduino.

Una vez descargado se procede a descomprimirlo y se selecciona el archivo que se encuentra dentro de la carpeta con el nombre Arduino.

Nombre	Fecha de modifica...	Tipo	Tamaño
drivers	21/05/2012 18:04	Carpeta de archivos	
examples	21/05/2012 18:05	Carpeta de archivos	
hardware	21/05/2012 18:05	Carpeta de archivos	
java	21/05/2012 18:06	Carpeta de archivos	
lib	21/05/2012 18:05	Carpeta de archivos	
libraries	21/05/2012 18:05	Carpeta de archivos	
reference	21/05/2012 18:05	Carpeta de archivos	
tools	21/05/2012 18:05	Carpeta de archivos	
arduino	21/05/2012 18:05	Aplicación	840 KB
cygiconv-2.dll	21/05/2012 18:04	Extensión de la apl...	947 KB
cygwin1.dll	21/05/2012 18:04	Extensión de la apl...	1.829 KB
libusb0.dll	21/05/2012 18:04	Extensión de la apl...	43 KB
revisions	21/05/2012 18:04	Documento de tex...	33 KB
ntbSerial.dll	21/05/2012 18:04	Extensión de la apl...	76 KB

Figura 3.2. Archivo ejecutable Arduino

Luego de ejecutar el archivo Arduino nos aparece el IDE [Fig. 3.3]

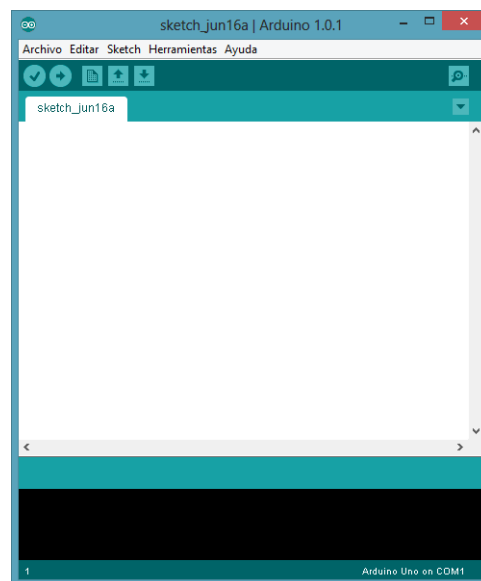


Figura 3.3. Ventana IDE

Como siguiente paso, debemos conectar la tarjeta Arduino a la computadora, dar clic derecho en Equipo → Administrar, luego

nos aparece una ventana en la cual se escoge la opción Administrador de dispositivos → Otros dispositivos → Arduino MEGA → Actualizar software de controlador.

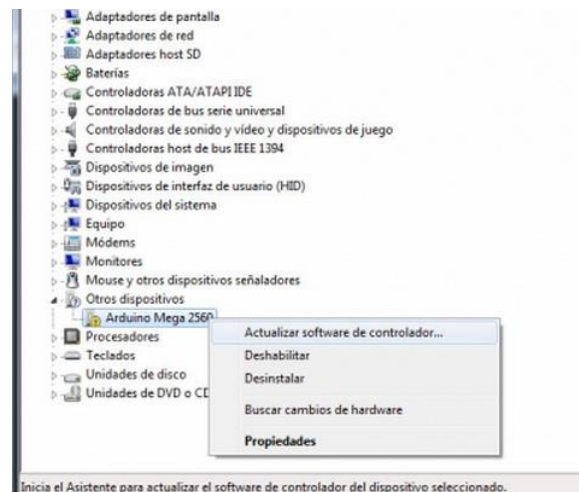


Figura 3.4. Administrador de dispositivos

Seleccionamos en la carpeta descomprimida en la sección Arduino la carpeta drivers, damos clic en siguiente, para culminar la instalación.

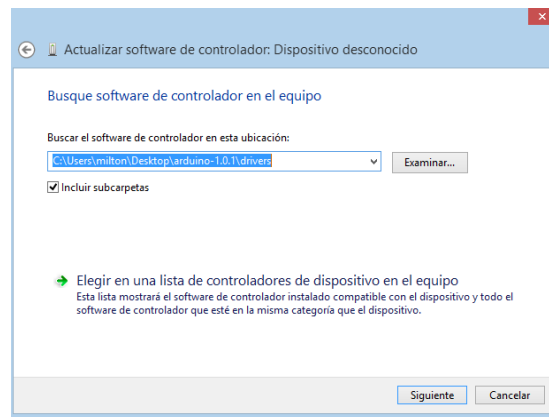


Figura 3.5. Actualización de Software de controlador

Para comprobar que la instalación se realizó con éxito entramos a: Administrador de dispositivos → Puertos (COM y LPT) → Arduino MEGA.

A lado de la opción Arduino MEGA, se encuentra el puerto que se ha configurado para establecer el enlace de comunicación con el Arduino, en nuestro caso salió (COM3).

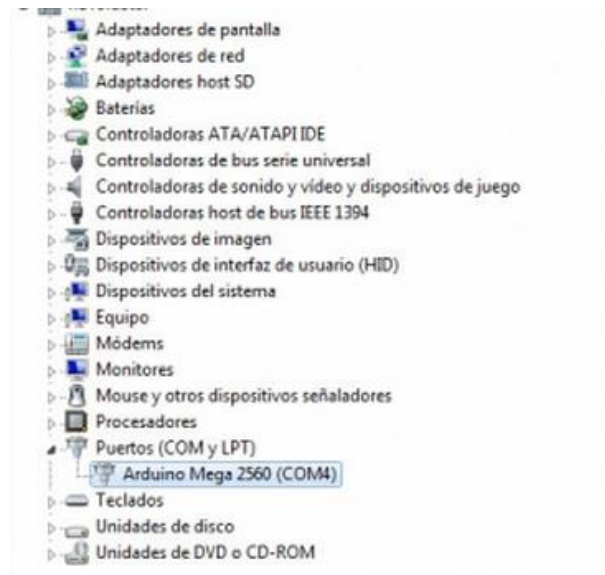


Figura 3.6. Administrador de dispositivos

La configuración en el IDE para la comunicación con la placa se realiza de la siguiente forma:

En el menú abrimos Herramientas → Puerto Serial → COM3 (puerto en el que se encuentra configurado la tarjeta).

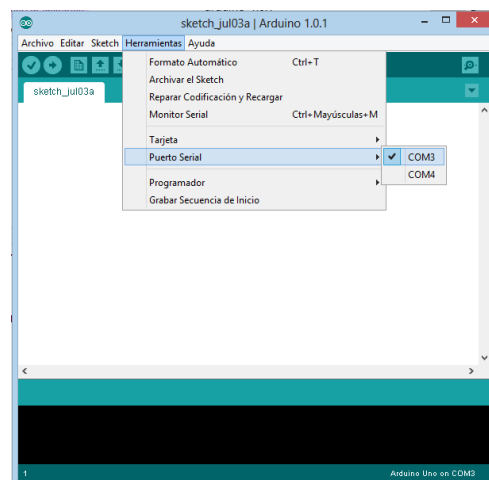


Figura 3.7. Puerto serial en Arduino

Luego se configura en el IDE la tarjeta que se utiliza entrando en el menú → Herramientas → Tarjeta → Arduino MEGA.

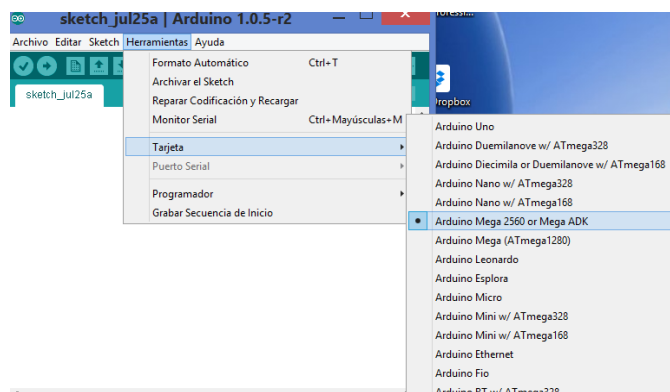


Figura 3.8. Selección de tarjeta Arduino MEGA

3.2.2. Carga del ejemplo Blink

Para lograr un buen diseño de una rutina de control, empezaremos con un ejemplo introductorio que nos permitirá identificar si la conexión entre la computadora y el Arduino fue realizada de manera correcta.

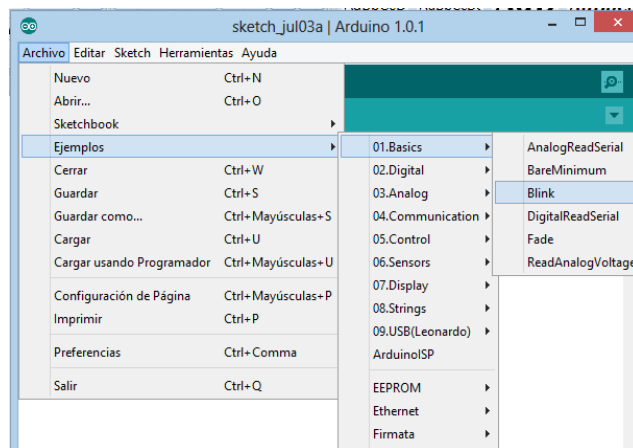
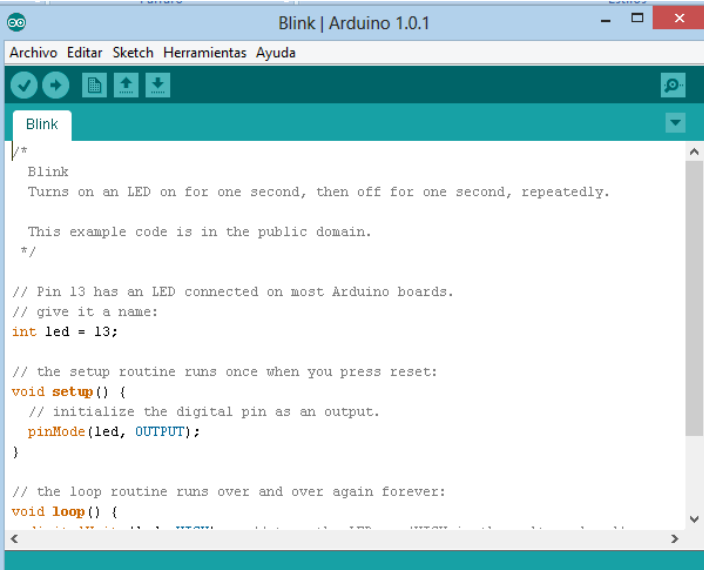


Figura 3.9. Carga del ejemplo Blink

El ejemplo Blink consiste en controlar el parpadeo del led que lleva incorporado la placa Arduino, mediante un código que se ejecuta en el IDE, dentro del código se puede determinar la frecuencia en la que se desea que prenda y apague el led.



The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The title bar reads 'Blink | Arduino 1.0.1'. The menu bar includes 'Archivo', 'Editar', 'Sketch', 'Herramientas', and 'Ayuda'. The toolbar contains icons for opening, saving, and running. The sketch editor displays the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Figura 3.10. Ejemplo Blink en IDE

3.2.3. Diagramas del ejercicio

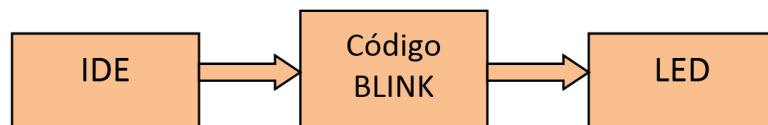


Figura 3.11. Diagrama de bloques

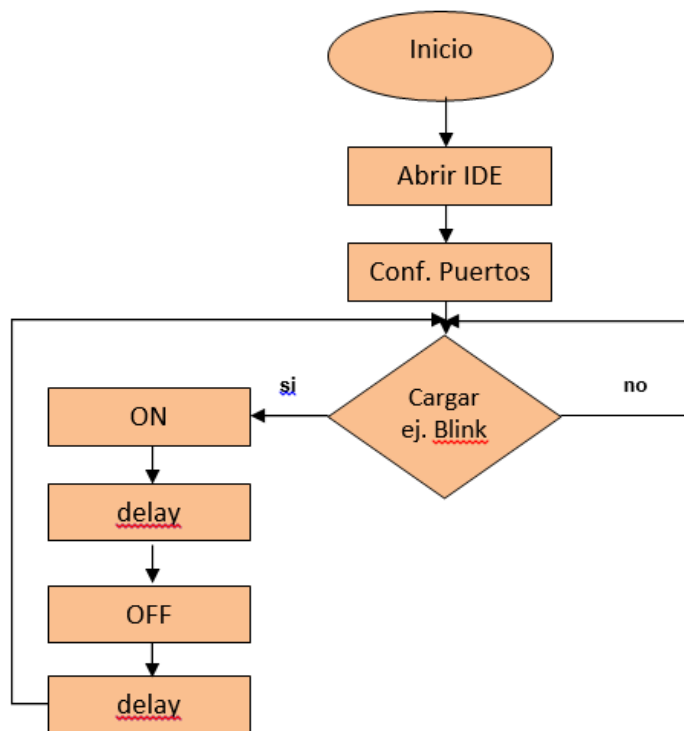


Figura 3.12. Diagrama ASM

3.2.4. Desarrollo del algoritmo

- Configurar el pin 13 como una constante LED
- Definir como pin de salida el pin 13
- Crear un lazo infinito con la siguiente subrutina:
 - Encender el led
 - Establecer un retardo
 - Apagar el led
 - Establecer un retardo

3.2.5. Código fuente

```
/* Blink
int led = 13;
void setup() {
  pinMode(led, OUTPUT);
}
voidloop() {
  digitalWrite(led, HIGH);    //enciende el LED
  delay(1000);                // esperar un segundo
  digitalWrite(led, LOW);    //apagar el LED
  delay(1000);                // esperar un segundo
}
```

3.3.Desarrollo de ejemplos y ejecución del proyecto

A continuación se explican modelos de ejemplos de programación que servirán como una introducción para la ejecución del proyecto final, estos ejemplos nos ayudan a familiarizarnos con la programación de Arduino y el manejo de puertos.

3.3.1. Luces de auto fantástico

En este ejercicio se emulará la secuencia de encendido de luces del auto fantástico, donde 6 leds conectados desde el pin 8 al pin13 (pines de entrada/salida digitales de Arduino) se encenderán secuencialmente, con un retardo de medio segundo entre cada led.

3.3.1.1. Diagramas

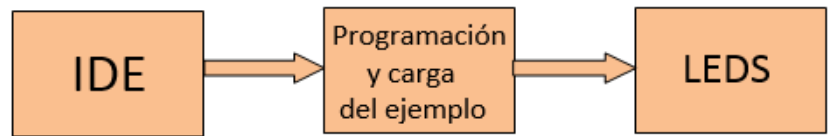


Figura 3.13. Diagrama de bloques

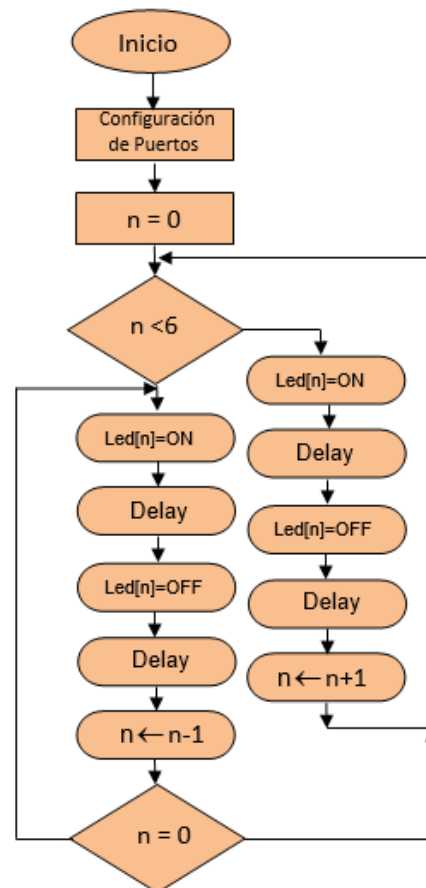


Figura 3.14. Diagrama ASM

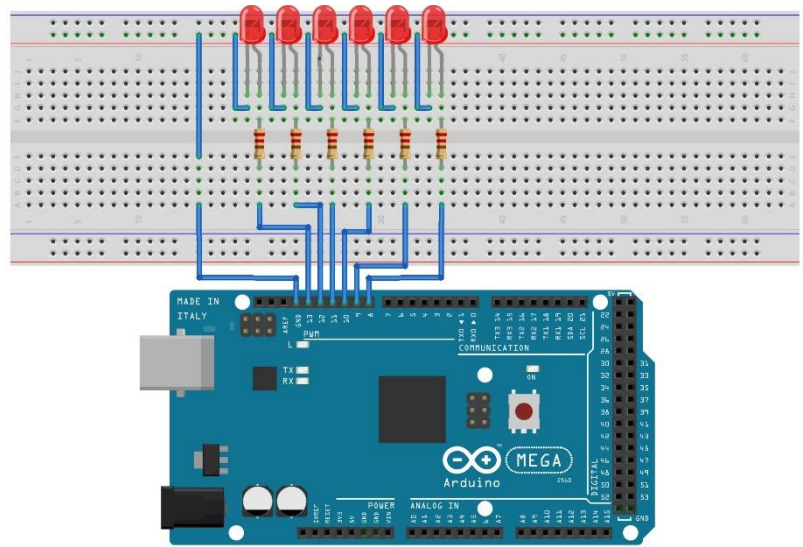


Figura 3.15. Diagrama de conexiones

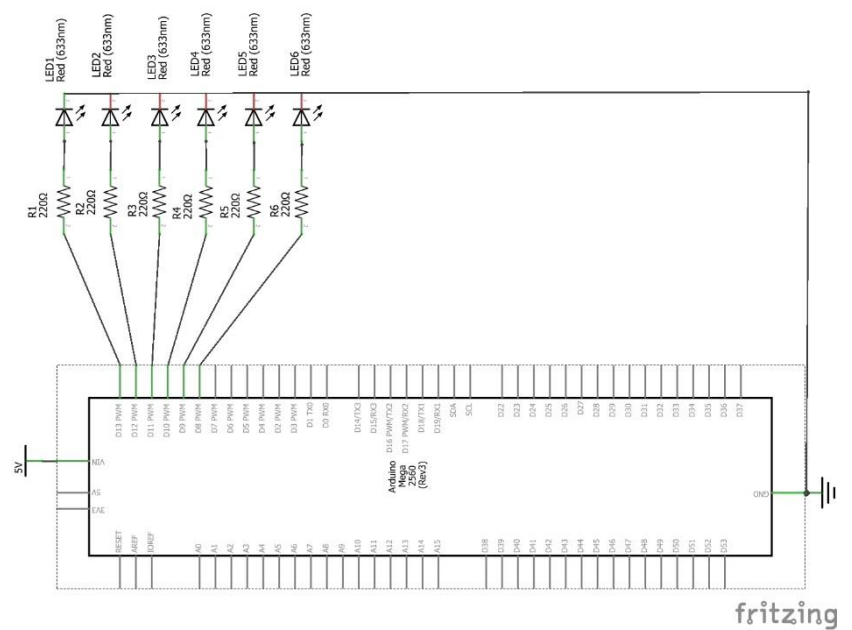


Figura 3.16. Diagrama esquemático

3.3.1.2. Desarrollo de algoritmos

- Declara los pines del 8 al 13 como un arreglo Led[n]
- Inicializa variable contador n=0
- Declara los pines del 8 al 13 como salidas
- Se declara un inicio
- Si $n < 6$ crear un lazo infinito con la siguiente subrutina:
 - Enciende Led[n]
 - Establece un retardo
 - Apaga Led[n]
 - Establece un retardo
 - Aumenta contador $n=n+1$
- Si no, si $n > 0$ crear un lazo infinito con la siguiente subrutina
 - Enciende Led[n]
 - Establece un retardo
 - Apaga Led[n]
 - Establece un retardo
 - Disminuir contador $n=n-1$
 - Si no, regresa a inicio

3.3.1.3. Código fuente

```
int leds[]={8,9,10,11, 12,13}; //declara leds como un
arreglo
int n=0;                        //Inicia el contador en 0
```

```

int tiempo=50;           //Inicia tiempo=50
void setup() {           //comienza la configuración
  for (n=0;n<6;n++) {
    pinMode(leds[n],OUTPUT); //Declara led[n] como salida
  }
}
void loop() {
  for (n=0;n<6;n++) {     //Empieza a contar
    digitalWrite (leds[n],HIGH); //Pone led[n] en alto
    delay(tiempo);        //Espera 50 milisegundos
    digitalWrite (leds[n],LOW); //Pone led[n] en bajo
    delay(tiempo);        //Espera 50 milisegundos
  }
  for (n=5;n>=0;n--) {   //Empieza a contar
    digitalWrite (leds[n],HIGH); //Pone led[n] en alto
    delay(tiempo);        //Espera 50 milisegundos
    digitalWrite (leds[n],LOW); //Pone led[n] en bajo
    delay(tiempo);        //Espera 50 milisegundos
  }
}

```

3.3.2. Control de movimiento de un motor de paso

En este ejercicio se programará la velocidad de giro, el número de pasos recorrido y el sentido de giro de un motor de paso bipolar. De este modo el motor avanzará una cantidad determinada de pasos en sentido horario con una cierta velocidad angular y luego cambiara el sentido de rotación avanzando una cierta cantidad de pasos con otra velocidad angular.

3.3.2.1. Diagramas

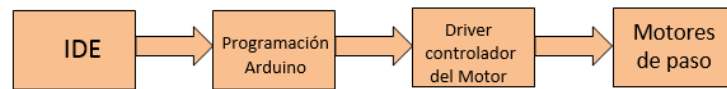


Figura 3.17. Diagrama de bloques

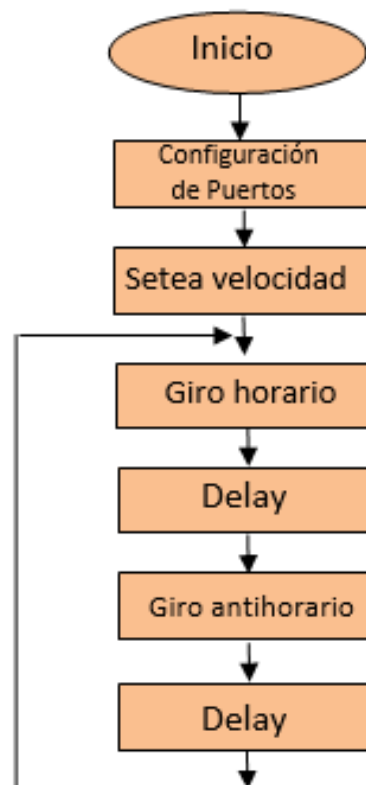
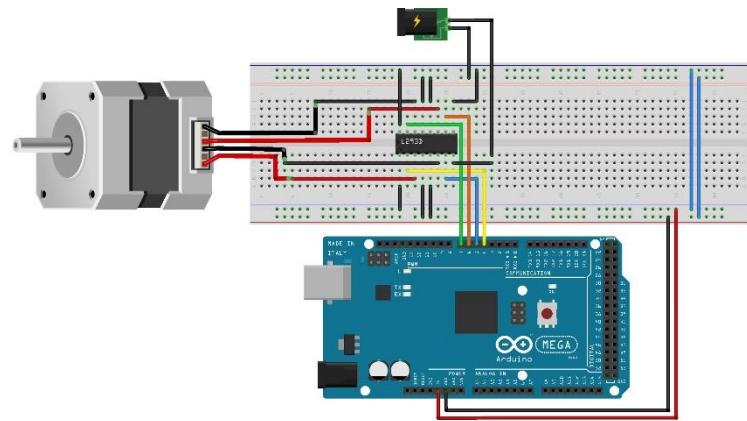


Figura 3.18. Diagrama ASM



fi

Figura 3.19. Diagrama de conexiones con Arduino.

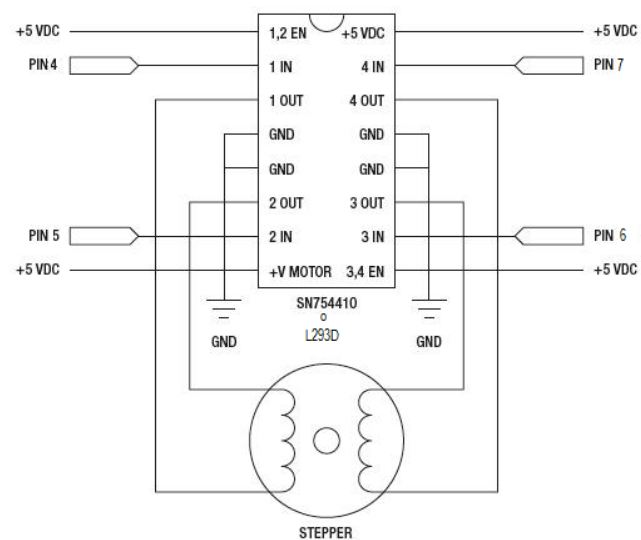


Figura 3.20. Esquemático de la conexión del driver SN754410 con motor de paso bipolar.

3.3.2.2. Desarrollo del algoritmo

- Declara el número de pasos por revolución,
- Declara la velocidad angular del motor
- Se crea un bucle infinito con la siguiente subrutina:
 - Se realiza giro horario del motor (200 pasos)
 - Se espera 1 segundo
 - Se realiza giro anti-horario del motor (200 pasos)
 - Se espera 1 segundo

3.3.2.3. Código fuente

```
#include <Stepper.h>

const int PasosPorRevo= 200;
// Numero de pasos por revolución.

const int Velocidad = 60;
// Velocidad de 60 revoluciones por segundo

const int Pasos = 200;
// Número de pasos que avanza

Stepper myStepper(PasosPorRevo, 4,5,6,7);
// inicializamos la libreria "Stepper"

void setup() {

    myStepper.setSpeed(Velocidad);
    //Seteamos la velocidad de giro
}

void loop() {
```

```
    myStepper.step(Pasos);  
    // Avanza 200 pasos  
  
    delay(1000);  
    // Esperamos 1000 milisegundos  
  
    myStepper.step(-Pasos);  
    // Avanza en sentido contrario  
  
    delay(1000);  
    // Esperamos 1000 milisegundos  
}
```

3.3.3. Sistema de Control de Impresora 3D mediante la Plataforma Arduino

Previo a la realización de este proyecto hemos desarrollado pequeñas prácticas para demostrar el proceso de trabajo de la impresora, incluyendo la verificación de la conexión Pc-Arduino, además del control de movimiento de motores de paso utilizando la plataforma de Arduino con el firmware Pronterface.

En la implementación final del proyecto de impresión 3D, se realizará la impresión física de un modelo tridimensional utilizando como materia prima plástico de tipo PLA o ABS, el proceso se lo llevará a cabo bajo la utilización de la plataforma Arduino (procesador y placa) como cerebro controlador del

proceso de impresión y de programas específicos (Pronterface, Slic3r) de control que funcionan en conjunto con Arduino.

Estos programas específicos de los que hacemos mención anteriormente pueden ser administrados a través de plataformas gráficas, de forma que, con estas interfaces, podemos cargar los archivos de los modelos 3D a imprimirse.

La impresora utilizada en nuestro proyecto es la Prusa DIY de la línea de impresoras Prusas, pertenecientes a la compañía 3DSTUFFMAKER y sucesoras de las impresoras iniciales del proyecto RepRap, consta de 4 motores de paso (3 para los ejes y 1 para la extrusora), además de 4 drivers controladores, sensores de fin de carrera, la cama caliente, la extrusora etc.

3.3.3.1. Descripción del proceso

En la primera fase del proyecto fue necesario realizar ajuste en el ensamblaje de la impresora para cambiar elementos de modo que se pueda adaptar la tarjeta Arduino para utilizarla en lugar de la que nos proporciona el fabricante.

En la computadora se deben instalar los programas Pronterface el cual ya viene con el subprograma Slic3r embebido en el mismo, aunque podría también instalarse por separado. Adicionalmente se debe instalar el programa de Interfaz de usuario de Arduino IDE, y descargarse el firmware Marlin que luego será cargado en el IDE.

Una de los primeros parámetros a configurar en el MARLIN es la tasa velocidad de transmisión de datos las cuales están predeterminadas en 115200 o en 25000, en la pestaña “Configuration.h” del IDE, seleccionaremos la de nuestro interés, esta tiene que ser la misma que definimos en el Pronterface.

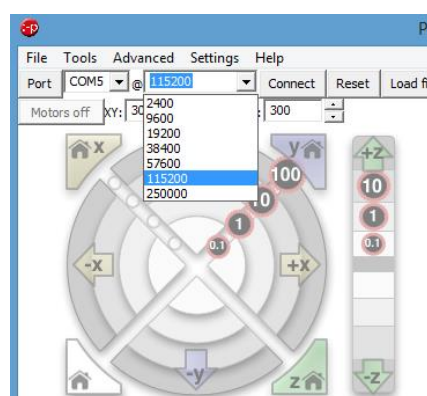


Figura 3.21. Elección del Baudrate en Pronterface

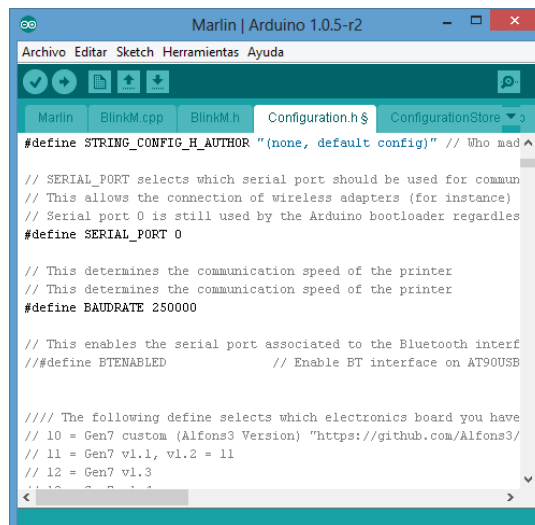


Figura 3.22. Elección del Baudrate en Marlin

Los sensores de fines de carrera que vienen incluidos en la impresora en su versión original son ópticos y funcionan con 3 terminales, su lógica de funcionamiento es negativa, en nuestro proyecto con la RAMPS 1.4 utilizamos fines de carrera mecánicos los cuales utilizan solo dos terminales de control, debido al cambio de lógica que se produce en esa transición, se deben configurar en FALSE los siguientes parámetros para los fines de carrera.

```

const bool X_MIN_ENDSTOP_INVERTING = false;

const bool Y_MIN_ENDSTOP_INVERTING = false;

```

```
const bool Z_MIN_ENDSTOP_INVERTING = false;  
const bool X_MAX_ENDSTOP_INVERTING = false;  
const bool Y_MAX_ENDSTOP_INVERTING = false;  
const bool Z_MAX_ENDSTOP_INVERTING = false;
```

Luego de esto realizamos todas las conexiones de la impresora al acople RAMPS-Arduino, verificando siempre la posición correcta de los cables que manejan cada motor de paso, debido a que si se los conecta en la posición contraria se produciría un cambio de giro.



Figura 3.23. Conexiones de los motores a la RAMPS

Para revisar que todo esté funcionando correctamente debemos hacer homing en el Pronterface y verificar si el nivel de los sensores es el correcto de modo que la

extrusora se encuentre siempre al nivel ideal, y la cama caliente no se salga del área de extrusión. El modelo a imprimir siempre está centrado por default, sin embargo, el centro de impresión se lo puede ubicar en cualquier lugar del plato de impresión modificando los parámetros del Slic3r al cual se accede desde la pestaña settings del Pronterface.

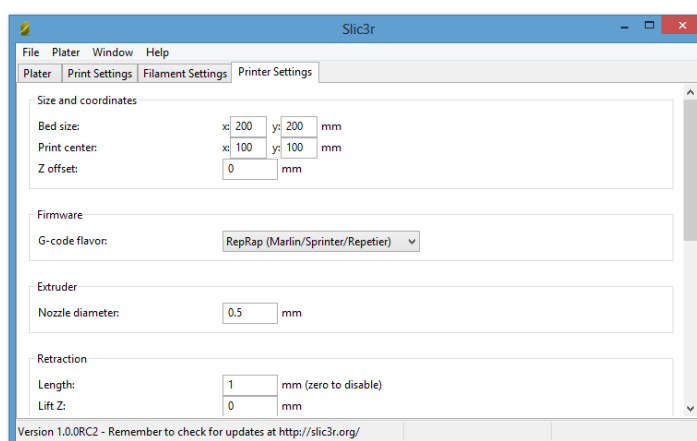


Figura 3.24. Configuración del centro de Impresión en Slic3r

3.3.3.2. Diagramas

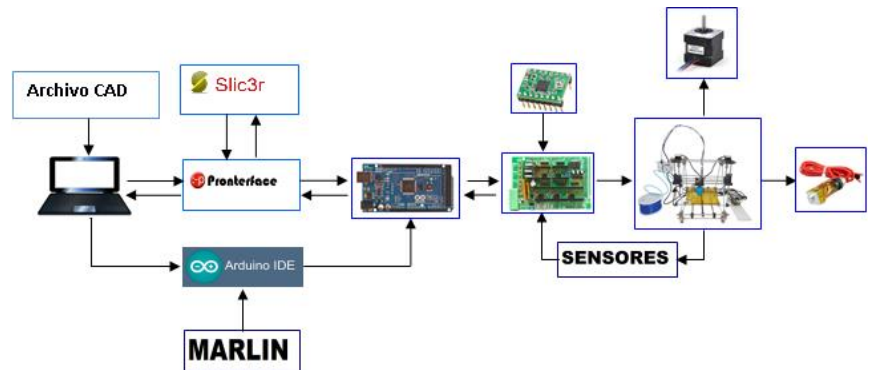


Figura 3.25. Diagrama de bloques del proceso de impresión

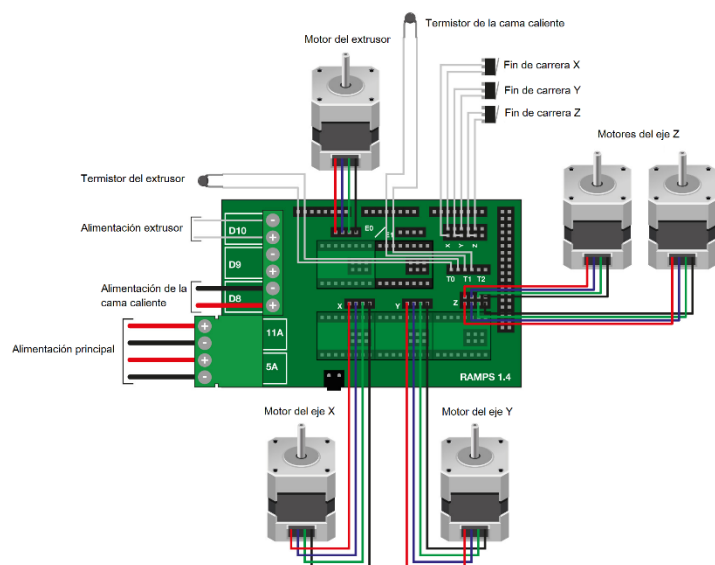


Figura 3.26. Diagrama de conexiones de RAMPS [31]



Figura 3.27. Diagrama ASM

3.3.3.3. Desarrollo del algoritmo

- Se ingresa el archivo del modelo a imprimir en formato *.CAD o *.STL a la computadora.
- El capeado y transformación del modelo a código G, se realiza en el programa Slic3r, el cual es un subprograma del Pronterface.

- Una vez generadas las instrucciones en código G, estas son enviadas al Arduino.
- Dentro del IDE se utiliza el Marlin, para configurar las especificaciones de trabajo de la impresora y esta información también es enviada al Arduino.
- La información del Arduino es transmitida a la RAMPS, la cual realiza la ejecución del trabajo de los motores de paso.
- Los sensores de la impresora realimentan a la RAMPS.

CAPÍTULO 4

4. IMPLEMENTACIÓN, EJERCICIOS DE PRUEBA Y SIMULACIÓN DEL PROYECTO

En el presente capítulo se mostrarán todos los diagramas de conexiones y se detallará gráficamente la manera física de conectar los elementos, además de presentar un listado de los componentes necesarios para la realización de cada uno de los ejercicios mostrados en el capítulo anterior.

Se realizaran observaciones acerca de los detalles importantes a tener en cuenta durante la realización de cada uno de los ejercicios, sustentando con fotos la ejecución de cada uno de los ejemplos.

En la mayoría de los casos los diagramas de conexiones y diagramas esquemáticos fueron realizados con el programa fritzing. En todos los ejercicios se utilizó la placa Arduino MEGA que contiene el microcontrolador ATmega2560 cuyas salidas programables fueron estudiadas previamente con el uso del mapa de pines del microcontrolador y del diagrama esquemático de la placa. [32]

4.1. Ejecución ejemplo Blink

Este ejercicio es básico pero fundamental para la ejecución de los demás ejemplos, pues se desarrolla la comunicación entre la placa Arduino MEGA 2560 y la computadora.

Para este ejercicio utilizamos la versión 1.0.5 del IDE; la placa Arduino MEGA 2560 tiene un diodo led que viene embebido en la placa con salida en el pin 13, el cual será utilizado en este ejemplo como se observó en el capítulo 3, también es posible utilizar dicho pin como una salida digital de modo que se pueda colocar un diodo led externo si se requiere.

Se conectó la placa Arduino Mega a la computadora por medio del cable de datos, el puerto COM4 quedó definido por default como puerto de comunicación entre la placa Arduino y la PC.

4.1.1. Componentes necesarios

- Tarjeta Arduino MEGA 2560
- Cable de datos USB

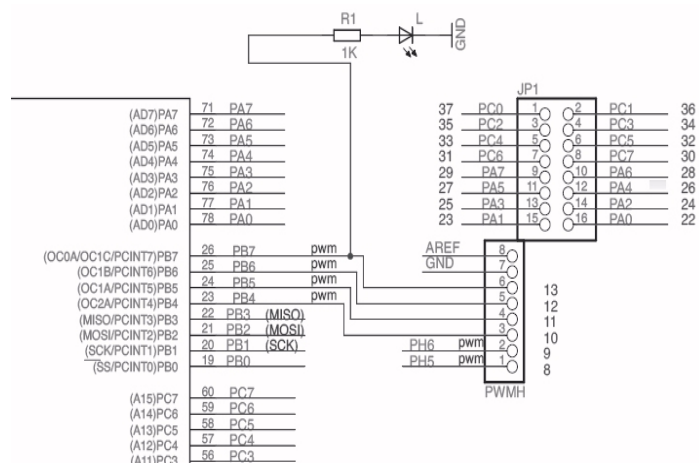


Figura 4.1. Diagrama interno de la tarjeta led pin 13.

4.1.2. Implementación

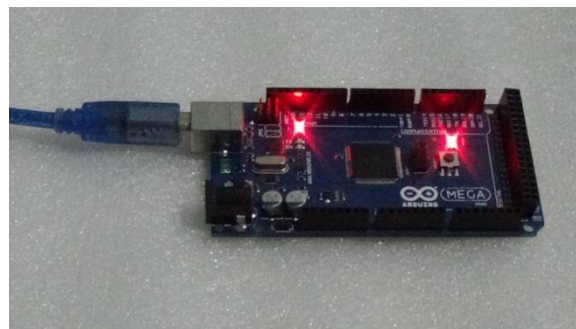


Figura 4.2. Ejemplo Blink en ejecución

4.1.3. Discusión del ejemplo Blink

Durante la conexión de la placa Arduino con la Pc, se deben observar el puerto con el que estamos trabajando y elegirlo en el IDE, para lograr una correcta comunicación. El diodo del pin 13 parpadea con la frecuencia asignada en la programación que es de 1 segundo, se puede también colocar un led en la salida del pin 13.

4.2. Luces del auto fantástico

Este ejercicio tiene como finalidad el reconocimiento de los puertos de entrada y salida digital del controlador, además de la configuración como entradas o salidas de dichos puertos y la asignación de tiempos de retardo programados.

Durante la ejecución del experimento, las luces van encendiéndose secuencialmente con un retardo de medio segundo entre cada led, llegando a sus extremos y cambiando de dirección el encendido, de modo que se crea un ciclo infinito.

4.2.1. Componentes necesarios

- Tarjeta Arduino MEGA 2560

- Cable de datos USB
- 6 Diodos LED
- 6 Resistencias de 330 Ohmios.
- Cable UTP

4.2.2. Implementación

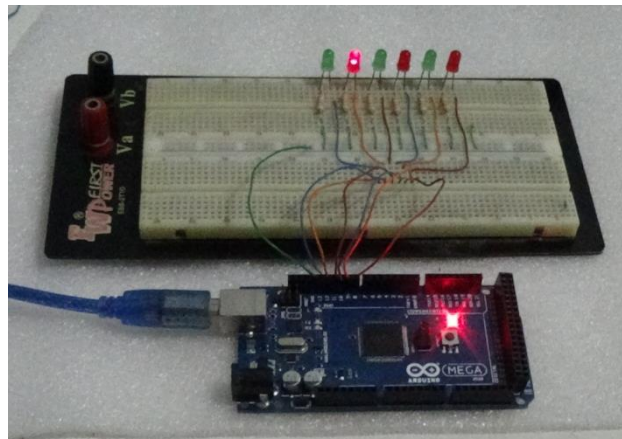


Figura 4.3. Luces del auto fantástico

4.2.3. Discusión del ejemplo del auto fantástico

Durante la ejecución del programa desde el controlador de Arduino, se puede observar el movimiento secuenciado de las luces de los diodos led conectados a los puertos configurados como salidas.

En la programación se debe tener en cuenta el correcto manejo del contador, pues uno de los errores que hubo durante la realización de la práctica es que el led conectado al último pin encendía muy tenuemente con poco brillo, cuando se verifico que el contador no incluía al último led y se corrigió el problema, todos los led encendieron de manera adecuada.

4.3. Control de movimiento de un motor de paso

En ejercicio verificamos practicamos el control sobre el giro de un motor de paso, verificaremos las librerías que son necesarias y las funciones que nos permitirán avanzar en un sentido u otro, además de poder seleccionar la velocidad angular de giro y el número de pasos de avance.

Para la realización de este proyecto se utiliza como driver de potencia como el controlador L293D (puente H), que permite precisamente el control simultaneo de la velocidad y dirección de giro de los motores de paso.

En el experimento el motor gira en sentido horario y avanza 200 pasos con una velocidad angular de 60 revoluciones por segundo.

4.3.1. Componentes necesarios

- Tarjeta Arduino MEGA 2560
- Cable de datos USB
- 1 motor de paso bipolar
- Driver de potencia L293D
- Cable UTP

4.3.2. Implementación

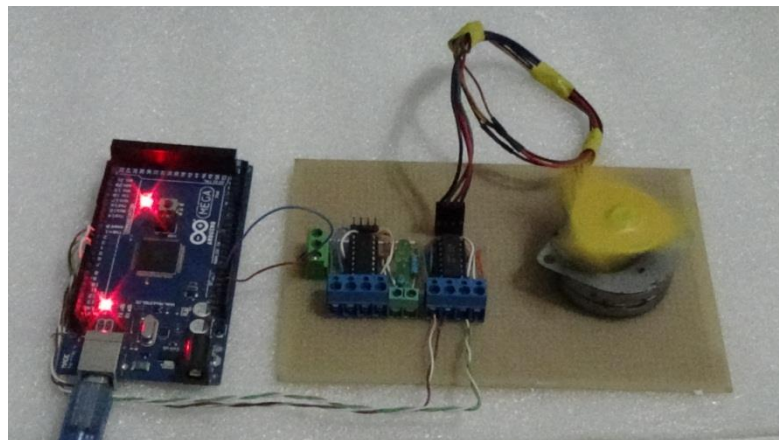


Figura 4.4. Control de movimiento de motor de paso

4.3.3. Discusión del ejemplo de control de movimiento de un motor de paso.

Durante la ejecución del ejemplo se deben tener en cuenta el orden de los cables que forman los bobinados de los motores de paso bipolares, pues se evitaría conectarlos de manera incorrecta. Una de las formas de verificar el orden de los cables es utilizando un multímetro y midiendo continuidad o una resistencia pequeña, aquellas que presenten resistencia pequeña son las terminales de una bobina.

4.4. Sistema de control de impresora 3D mediante la plataforma Arduino.

En este experimento se busca poner de manifiesto todos los procesos realizados durante la impresión de un modelo físico real por parte de una impresora 3D. Es importante declarar la importancia de los ejercicios anteriores para poder llegar a este último proyecto sin problemas.

En este proyecto se analiza el sistema de control basado en Arduino que se empleó para la impresión 3D de modelos reales utilizando una impresora 3D de la familia RepRap.

4.4.1. Implementación

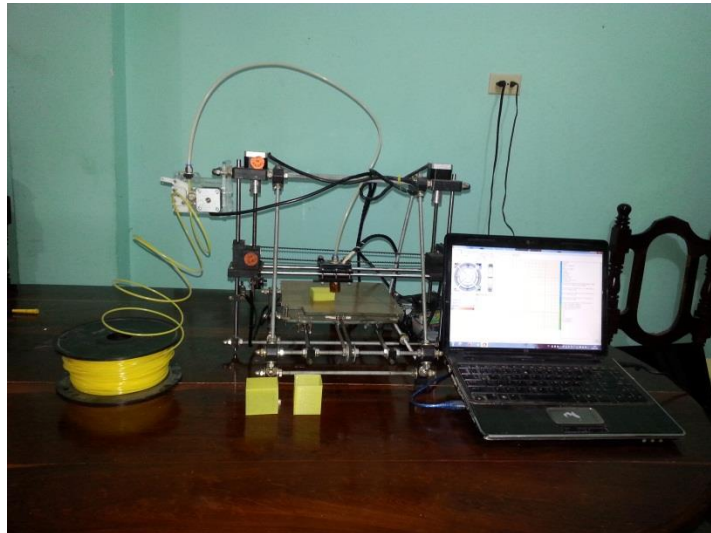


Figura 4.5. Sistema completo de impresión 3D

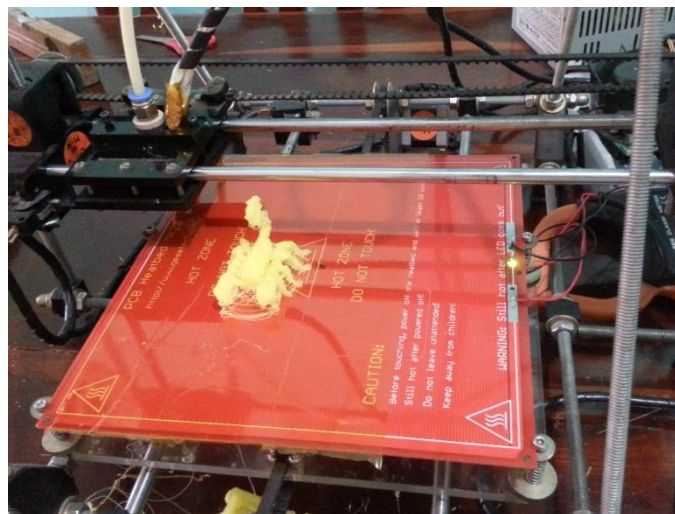


Figura 4.6. Impresión 3D de un escorpión

4.4.2. Discusión sobre el proyecto de impresión 3D controlado por plataforma Arduino.

Nuestro proyecto de impresión es un sistema realimentado basado en un controlador de tipo PID, esto implica que tenemos una muy buena flexibilidad en el control de los parámetros que involucran los movimientos de los motores y las temperaturas de extrusión y de la cama caliente, estos parámetros además pueden ser ajustados en el Marlin de modo que podamos modificarlos a las necesidades de impresión establecidas.

La posición de los sensores es fundamental antes de cada impresión pues se deben ajustar de tal modo que haciendo homing, el plato de impresión quede justamente al ras del plato de impresión, esta revisión puede verificarse haciendo pasar una hoja de papel entre el plato de impresión y la extrusora y observando que pase con facilidad.

Los parámetros de impresión determinan el tiempo que tarde en imprimirse una figura, estos parámetros se pueden calibrar en Slicer.

El acabado de la figura depende de estos parámetros mencionados y son proporcionales al tiempo de impresión, es decir, si se quiere imprimir un volumen sin tantas especificación en su acabado, este tendrá un tiempo de impresión menor que el tiempo empleado en imprimir la misma figura ajustando los parámetros para que la elabore con más detalles.

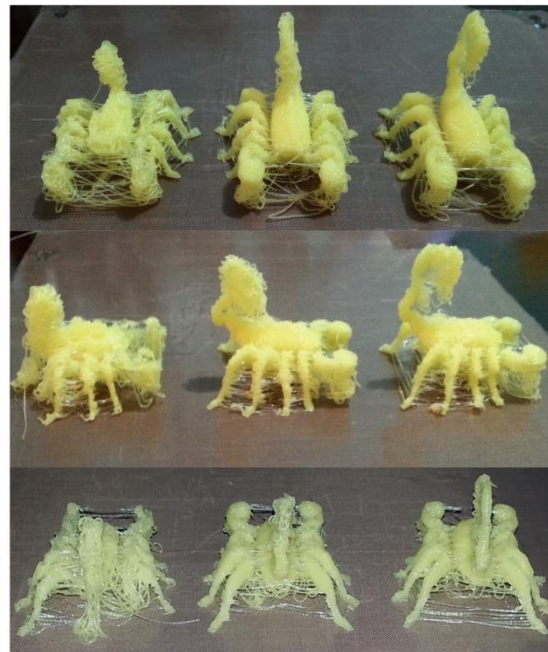


Figura 4.7. Tres modelos de impresión A (izquierda), B(central) y C(derecha) con distintas configuraciones de impresión.

Print speed	
Perimeters (mm/s):	30
Small perimeters (mm/s):	30
Infill (mm/s):	60
Solid infill (mm/s):	60
Bridges (mm/s):	60

Figura 4.8. Parámetros para las velocidades de impresión de los modelos A, B y C.

Para la elaboración del modelo A, B y C se utilizaron las configuraciones mostradas en la siguiente tabla.

	A	B	C
Tiempo de impresión (min)	15	28	33
Perimeters (mm/s)	30	25	20
Small perimeters (mm/s)	30	25	20
Infill (mm/s)	40	35	30
Solid Infill (mm/s)	40	35	30
Bridges (mm/s)	65	60	55

Figura 4.9. Tabla de los tiempos para los distintos parámetros de impresión del escorpión.

Como se puede apreciar en la tabla adjunta y en la figura anterior, la calidad de la figura B fue superior a la de la figura A y del mismo modo la calidad de la figura C fue superior a las dos anteriores debido a que en cada ocasión se disminuyeron los parámetros de velocidad en el rellenado de los bordes internos y externos, lo que ocasiono que realice las impresiones con más detalle pero en un mayor tiempo.

CONCLUSIONES

Basado en las experiencias obtenidas durante el desarrollo de esta tesina podemos concluir lo siguiente:

1. La tecnología de impresión de modelos en 3D se encuentra en un constante desarrollo y crecimiento tanto en su software como en su hardware a la vez que cada día encuentra más aplicaciones en el mundo de la industria y en las ciencias.
2. El control del proceso de impresión 3D se realizó mediante la tarjeta Arduino Mega, que contiene al microcontrolador ATmega 2560, el cual nos permitió trabajar en un entorno de fácil programación y dada la versatilidad de la tarjeta se pudo realizar las adaptaciones necesarias a la impresora de manera que funcione con Arduino como cerebro del proceso.

3. El acople de los drivers de potencia de los motores de paso a la tarjeta Arduino se lo realizó mediante la tarjeta RAMPS 1.4 la cual permite la conexión de los sensores de posicionamiento de los ejes y de la extrusora, esto forma parte de la realimentación, del sistema.
4. Por medio de la elaboración de ejemplos antes de la implementación del proyecto final, se logró mejorar nuestros conocimientos sobre programación del microcontrolador ATmega 2560, también se logró un mayor conocimiento acerca de los puertos de entrada-salida y la selección del puerto de comunicación.
5. La comunicación con la PC y la impresora se realiza mediante la plataforma Pronterface y el firmware Marlin, este último fue el seleccionado para proporcionar junto con Arduino el control del sistema de impresión que es de tipo PID.
6. La calidad de los modelos finales impresos depende de las características de velocidades de impresión seleccionadas en programa Slic3r de modo que cuando la velocidad de impresión es más lenta el acabado final de la figura tiene mejores detalles y acabados.
7. La ventaja que proporcionan el uso de microcontroladores para el desarrollo de la impresión en 3D, es que son de fácil programación y

con una adaptabilidad a diversas plataformas muy grande, las placas Arduino tiene un bajo costo relativamente bajo en comparación a la gran cantidad de aplicaciones que tiene, nos permite mejorar los procesos y tener cada vez resultados más precisos y trabajos más eficientes, estas tecnologías al ser de licencia libre se encuentran en constante evolución, lo que permite que en el futuro se apliquen estos principios para poder resolver problemas de mayor importancia.

RECOMENDACIONES

Las recomendaciones que logramos obtener del proyecto son las expuestas a continuación:

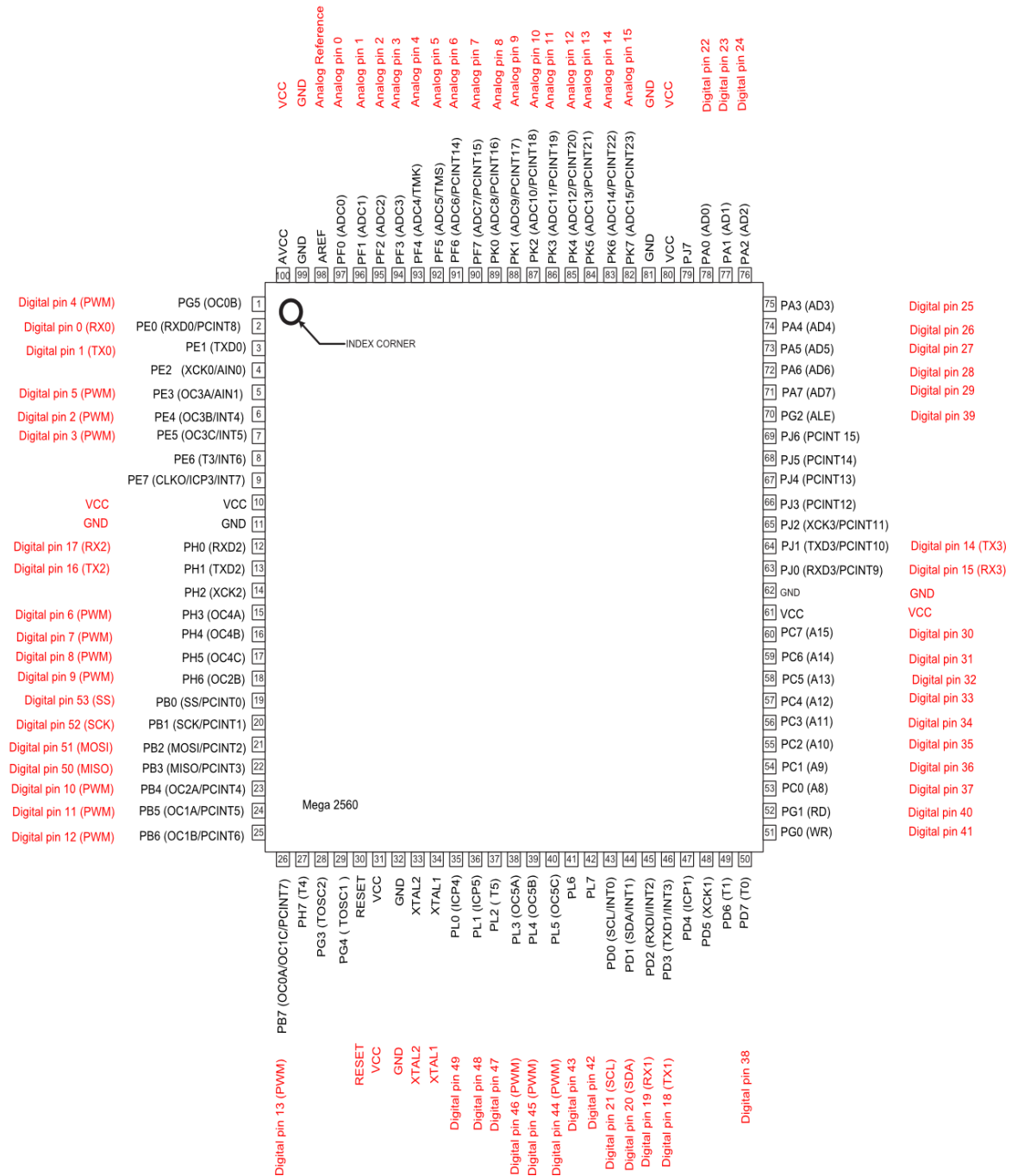
1. La posición de los sensores de fines de carrera para los ejes X,Y y Z es fundamental ya que permite calibrar la posición de origen de la impresora (homing) además de que permite adecuar la distancia que debe existir entre el plato de impresión y la extrusora.
2. Se recomienda revisar dos veces la posición en la que se conectan los motores a la RAMPS puesto que si un motor está mal ubicado girará en sentido contrario y podría dañar o descuadrar el plato de impresión o alguno de los ejes.

3. Es necesario ir evaluando la temperatura de los drivers y limitar la corriente de cada uno utilizando el potenciómetro que viene incluido en ellos, pues si la corriente no es suficiente, no habrá movimiento del motor y si hay exceso de corriente la corriente se elevará y se corre riesgo de quemar el driver.
4. La extrusora debe estar correctamente conectada y siempre evaluar si extruye la cantidad de material que se le envía pues puede haber problemas en el termistor de manera que la lectura de temperatura sea la incorrecta.
5. En algunas impresiones es importante notar el material que se se va colocando en el plato de extrusión se enfría rápidamente y en algunas ocasiones es barrido por la misma extrusora, esto se puede solucionar colocando una cama caliente y cubriéndola con cinta capton, esto proporcionará más estabilidad y firmeza en las primeras líneas de impresión de manera que no serán barridas por la extrusora.
6. Antes de realizar cualquier impresión hay que dejar calentar el tiempo suficiente a la extrusora de manera que alcance la temperatura indicada previamente para este fin, así mismo se debe verificar con un nivel si la posición del plato de impresión es completamente horizontal de forma

que no existan lugares de depresión o que puedan empujar la punta de la extrusora.

ANEXOS

ANEXO 1.- Arduino Mega 2560 PIN diagram



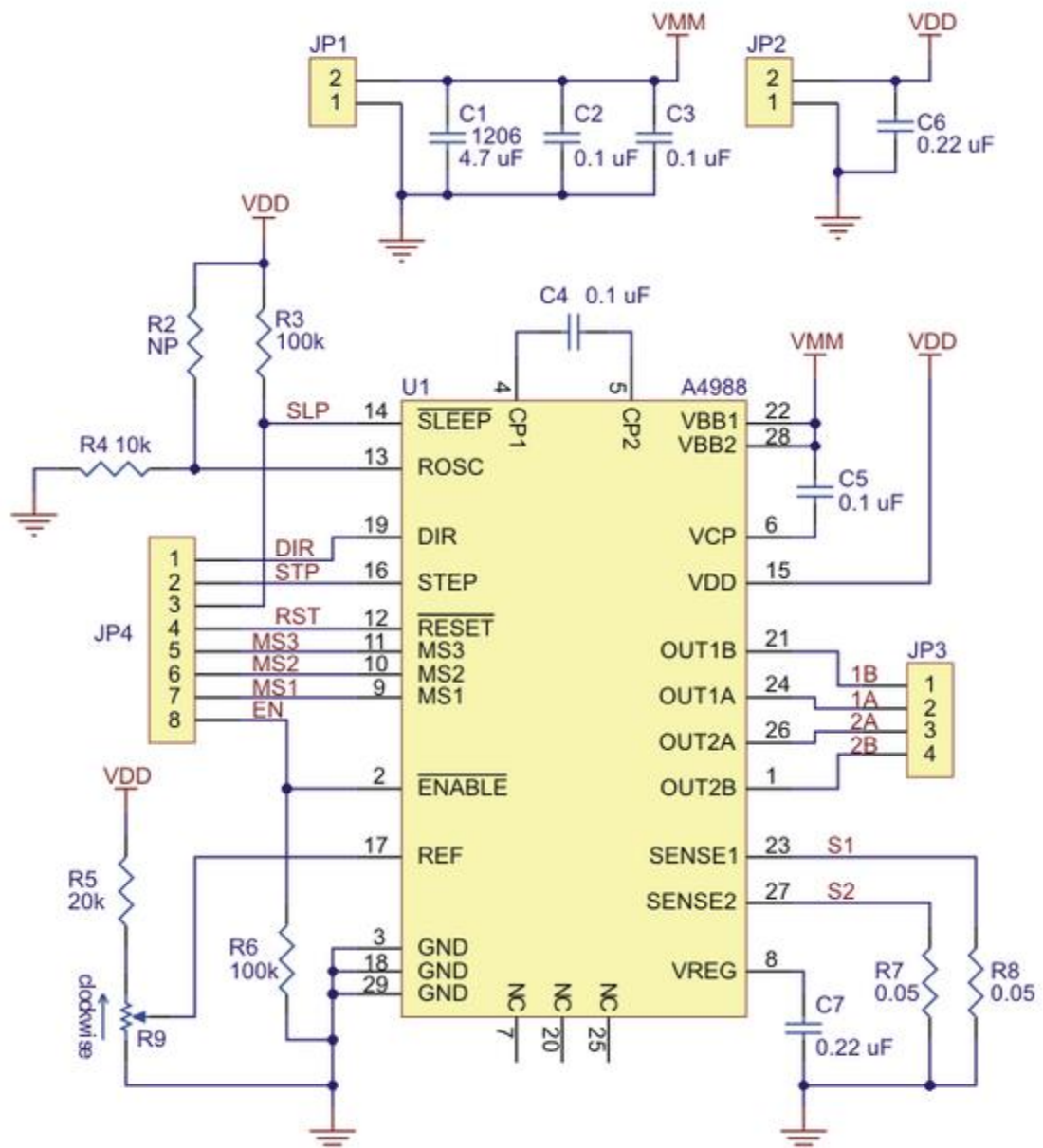
ANEXO 2.- Arduino Mega 2560 PIN mapping table

Pin Number	Pin Name	Mapped Pin Name	Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)	26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)	27	PH7 (T4)	
3	PE1 (TXD0)	Digital pin 1 (TX0)	28	PG3 (TOSC2)	
4	PE2 (XCK0/AIN0)		29	PG4 (TOSC1)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)	30	RESET	RESET
6	PE4 (OC3B/TNT4)	Digital pin 2 (PWM)	31	VCC	VCC
7	PE5 (OC3C/TNT5)	Digital pin 3 (PWM)	32	GND	GND
8	PE6 (T3/TNT6)		33	XTAL2	XTAL2
9	PE7 (CLK0/ICP3/TNT7)		34	XTAL1	XTAL1
10	VCC	VCC	35	PL0 (ICP4)	Digital pin 49
11	GND	GND	36	PL1 (ICP5)	Digital pin 48
12	PH0 (RXD2)	Digital pin 17 (RX2)	37	PL2 (T5)	Digital pin 47
13	PH1 (TXD2)	Digital pin 16 (TX2)	38	PL3 (OC5A)	Digital pin 46 (PWM)
14	PH2 (XCK2)		39	PL4 (OC5B)	Digital pin 45 (PWM)
15	PH3 (OC4A)	Digital pin 6 (PWM)	40	PL5 (OC5C)	Digital pin 44 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)	41	PL6	Digital pin 43
17	PH5 (OC4C)	Digital pin 8 (PWM)	42	PL7	Digital pin 42
18	PH6 (OC2B)	Digital pin 9 (PWM)	43	PD0 (SCL/TNT0)	Digital pin 21 (SCL)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)	44	PD1 (SDA/TNT1)	Digital pin 20 (SDA)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)	45	PD2 (RXD1/TNT2)	Digital pin 19 (RX1)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)	46	PD3 (TXD1/TNT3)	Digital pin 18 (TX1)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)	47	PD4 (ICP1)	
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)	48	PD5 (XCK1)	
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)	49	PD6 (T1)	
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)	50	PD7 (T0)	Digital pin 38

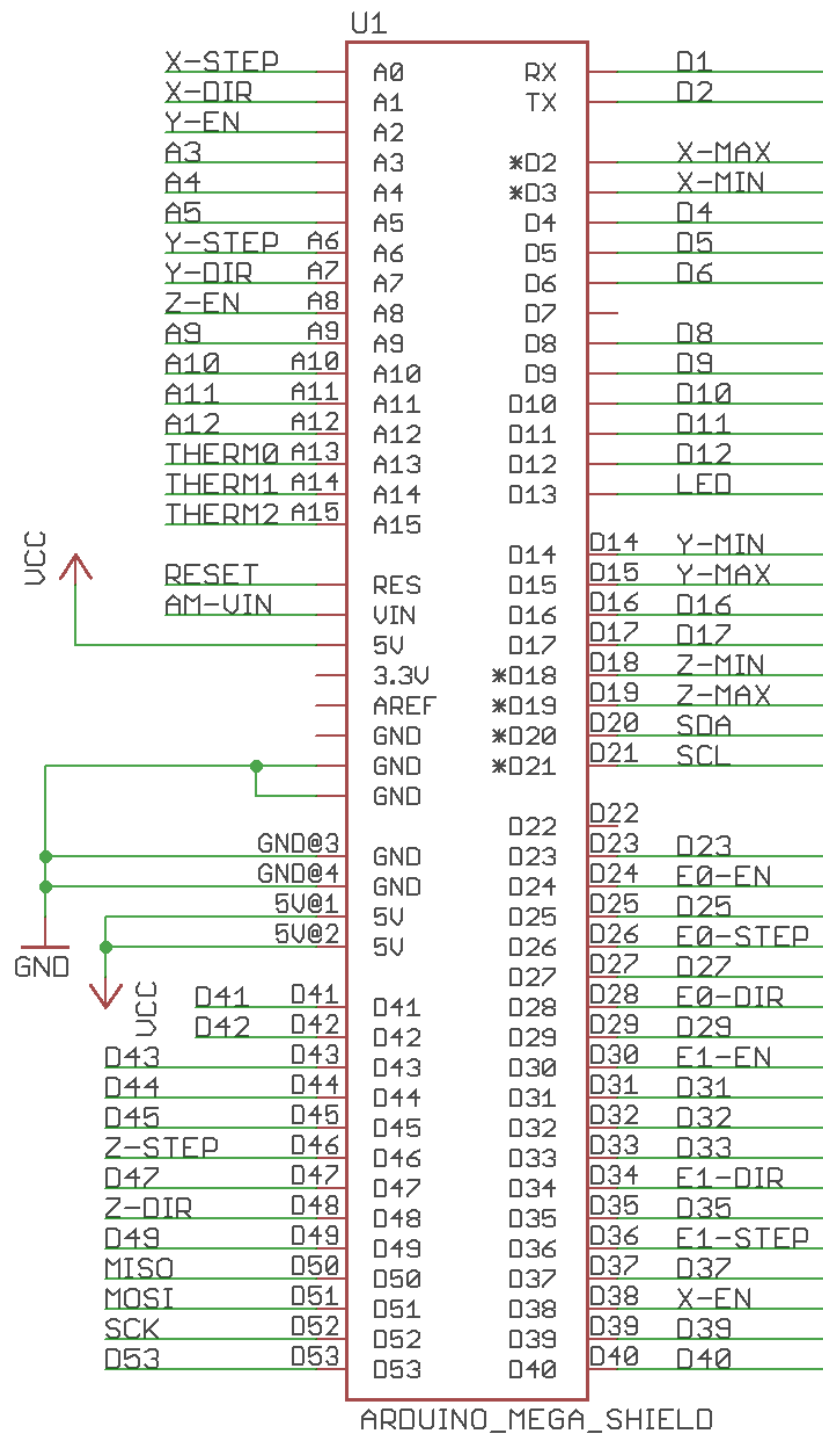
Pin Number	Pin Name	Mapped Pin Name
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25

Pin Number	Pin Name	Mapped Pin Name
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

ANEXO 3. Diagrama esquemático del pololu A4988

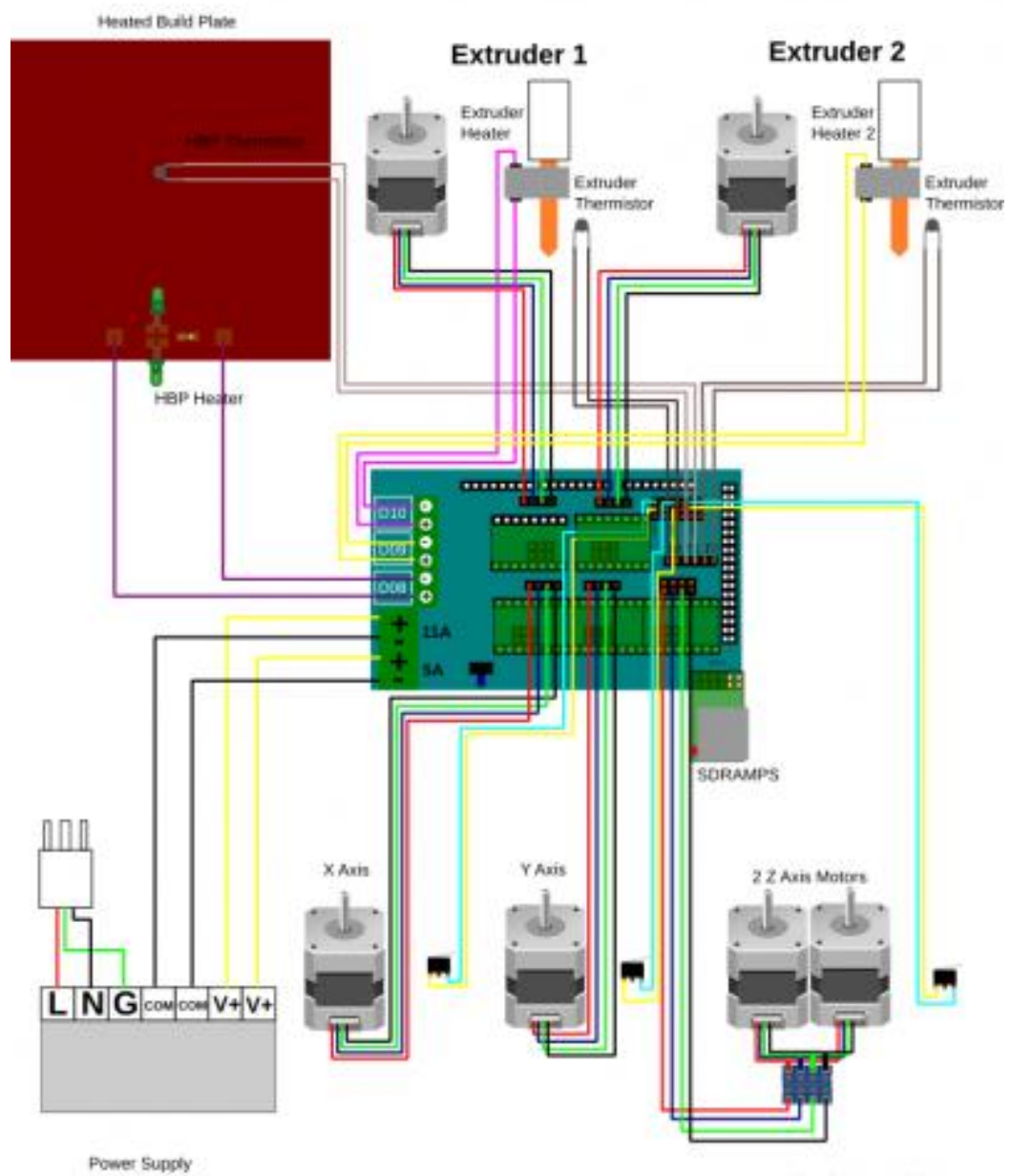


ANEXO 4.- Diagrama de pines RAMPS 1.4



ANEXO 5.- Diagrama de conexiones de la RAMPS

RepRap Arduino Mega Pololu Shield 1.4



ANEXO 6.- Especificaciones Técnicas L293D



L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

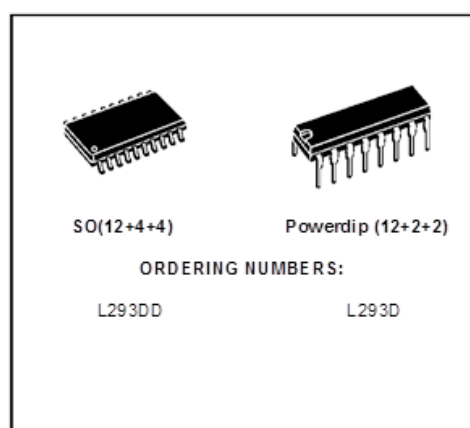
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY OVERTEMPERATURE PROTECTION LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

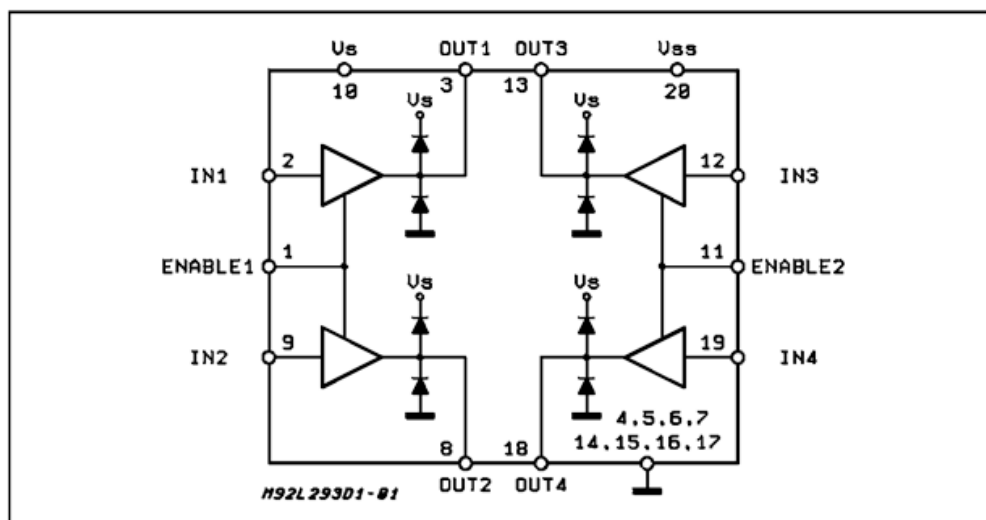
This device is suitable for use in switching applications at frequencies up to 5 kHz.



The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount package which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM

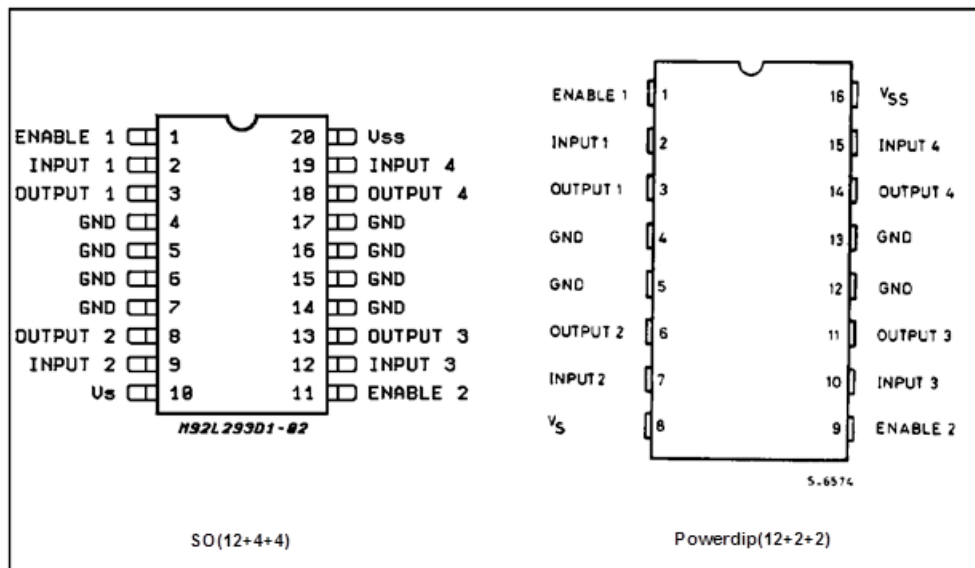


L293D - L293DD

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Supply Voltage	36	V
V_{SS}	Logic Supply Voltage	36	V
V_I	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_O	Peak Output Current (100 μ s non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at $T_{Diss} = 90^\circ\text{C}$	4	W
T_{stg}, T_J	Storage and Junction Temperature	- 40 to 150	$^\circ\text{C}$

PIN CONNECTIONS (Top view)



THERMAL DATA

Symbol	Description	DIP	SO	Unit
$R_{th-j-pins}$	Thermal Resistance Junction-pins	max.	14	$^\circ\text{C/W}$
$R_{th-j-amb}$	Thermal Resistance junction-ambient	max.	80	$^\circ\text{C/W}$
$R_{th-j-case}$	Thermal Resistance Junction-case	max.	14	$^\circ\text{C/W}$

(*) With 6sq. cm on board heatsink.

L293D - L293DD

ELECTRICAL CHARACTERISTICS (for each channel, $V_S = 24\text{ V}$, $V_{SS} = 5\text{ V}$, $T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	$V_I = L$; $I_O = 0$; $V_{en} = H$		2	6	mA
		$V_I = H$; $I_O = 0$; $V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_I = L$; $I_O = 0$; $V_{en} = H$		44	60	mA
		$V_I = H$; $I_O = 0$; $V_{en} = H$		16	22	mA
		$V_{en} = L$		16	24	mA
V_{IL}	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
V_{IH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_L	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_L = 1.5\text{ V}$			-10	A
I_H	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_H \leq V_{SS} - 0.6\text{ V}$		30	100	A
$V_{en L}$	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
$V_{en H}$	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
$I_{en L}$	Low Voltage Enable Current (pin 1, 11)	$V_{en L} = 1.5\text{ V}$		-30	-100	A
$I_{en H}$	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{en H} \leq V_{SS} - 0.6\text{ V}$			± 10	A
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = -0.6\text{ A}$		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = +0.6\text{ A}$		1.2	1.8	V
V_F	Clamp Diode Forward Voltage	$I_O = 600\text{ nA}$		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_O		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_O		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_I to 0.5 V_O		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_I to 0.5 V_O		200		ns

(*) See fig. 1.

BIBLIOGRAFÍA

[1] Evans, Brian, *Practical 3D printers*, 1ra Edición, Editorial Technology in action, 2012.

[2] Monk, Simon, *Programing Arduino-Getting Started with Sketches*, 1ra Edición, McGraw Hill, 2012.

[3] McRoberts, Michael, *Beginning Arduino*, 1ra Edición, Editorial Technology in action, 2010.

[4] Aguilar, Alejandro, *Diseño, impresión, montaje y control de un manipulador robotico*, [en línea], disponible en:

http://iearobotics.com/alberto/lib/exe/fetch.php?media=theses:memoria_alexaguiro.pdf.

Fecha de consulta 15/03/2013

[5] Illescas, Marco Esteban, *Construcción de una impresora 3D open Source*, [en línea], disponible en:

<http://iearobotics.com/alberto/lib/exe/fetch.php?media=theses:marcoestebanillescas.pdf>

Fecha de consulta 20/03/2013.

[6] Vera García, Arturo, *Optimización mecánica de una impresora 3D open Source*, [en línea], disponible en:

<http://iearobotics.com/alberto/lib/exe/fetch.php?media=theses:arturovera.pdf>,

Fecha de consulta 20/03/2013.

[7] Arduino CC, Hardware Arduino, [en línea], disponible en:

<http://arduino.cc/es/Main/> Fecha de consulta 20/03/2013.

[8] Kikai Labs, *El mercado global de impresión en 3D espera que llegue a EE.UU. \$ 2,99 mil millones en 2018*, [en línea], disponible en:

http://www.kikailabs.com.ar/articulos/-/asset_publisher/geXBssdlL5J1/blog/id/15537,

Fecha de consulta 1/04/2013.

[9] Wikipedia, Impresora 3D, [en línea], disponible en:

http://es.wikipedia.org/wiki/Impresora_3D

Fecha de consulta 20/03/2013.

[10] S3DJ, Soluciones 3D para joyería, [en línea], disponible en:

<http://www.s3dj.com/38801.html>

Fecha de consulta 20/03/2013.

[11] Caddyspain, Impresoras 3D, [en línea], disponible en:

<http://www.caddyspain.com/I3Dindex.htm>,

Fecha de consulta 20/03/2013.

[12] Terra Media, Introducción a las impresoras 3D, [en línea], disponible en:

<http://bloc.meddia.net/es/introduccion-a-las-impresoras-3d>

Fecha de consulta 10/03/2013.

[13] Revista Líderes, La producción parte de una impresora 3D, [en línea], disponible en:

http://www.revistalideres.ec/tecnologia/produccion-parte-impresora_0_757124301.html

Fecha de consulta 10/03/2013.

[14] Tecnofans, Una impresora 3D elabora un exoesqueleto para una niña con artrogriposis, tecnofans, [en línea], disponible en:

<http://www.tecnofans.es/hi-tech/articulo/una-impresora-3d-elabora-un-exoesqueleto-para-una-nina-con-artrogriposis/51355/>

Fecha de consulta 10/03/2013.

[15] Shdesigns, Ventajas y desventajas de la impresión 3D, [en línea], disponible en: <http://shdesigns.com.mx/will-3d-printing-change-the-world/>

Fecha de consulta 10/03/2013.

[16] RepRap Wiki, Building Your Own 3D Printer, [en línea], disponible en:

<http://reprapbook.appspot.com/>

Fecha de consulta 10/03/2013.

[17] Evans, Brian, *Beginning Arduino Programming*, 1ra Edición, Editorial Technology in action, 2012.

[18] Impresoras3D, ReplicatorG - Software de impresión 3D open source, [en línea], disponible en:

<http://www.impresoras3d.com/replicatorg-software-de-impresion-3d-open-source>

Fecha de consulta 15/03/2013.

[19] Wikipedia, G-code, [en línea], disponible en:

<http://es.wikipedia.org/wiki/G-code>

Fecha de consulta 15/03/2013.

[20] IMAC, EMC para el control de manipuladores, [en línea], disponible en:

<http://www.imac.unavarra.es/~cnc/gcode.php>

Fecha de consulta 15/03/2013.

[21] Luis XBOT, Códigos para CNC - Funciones Preparatorias (G), [en línea], disponible en:

<http://r-luis.xbot.es/cnc/codes03.html>

Fecha de consulta 15/03/2013.

[22] Kliment, Sprinter, [en línea], disponible en:

<https://github.com/kliment/Sprinter>

Fecha de consulta 15/03/2013.

[23] RepRap Wiki, Sprinter, [en línea], disponible en:

<http://reprap.org/wiki/Sprinter>

Fecha de consulta 15/03/2013.

[24] CURA – Software para Impresión 3D – Sprinter Vs Marlin, [en línea], disponible en:

<http://www.arduteka.com/2012/12/software-cura-para-impresion-3d-sprinter-vs-marlin/>

Fecha de consulta 10/11/2013

[25] Pololu.com, Pololus, [en línea], disponible en:

<http://www.pololu.com/product/1182/specs>

[26] Triquitraqueando, Impresora 3D, [en línea], disponible en:

<http://www.triquitraqueando.com/tiendita/image/cache/data/prusa-full-kit-500x500.jpg>

[27] NEMA, Motor de paso, [en linea], disponible en:

<http://www.seeedstudio.com/wish/nema-17-bipolar-stepper-motor-for-reprap-3d-printer-p824>

[28] RepRap Wiki, Tarjeta RAMPS, [en linea], disponible en:

http://reprap.org/wiki/File:RAMPS1-3_fin.JPG

[29] Pololu.com, Pololus, [en linea], disponible en:

<http://www.pololu.com/catalog/product/1182>

[30] Thingsandtrains, Firmware, [en linea], disponible en:

http://thingsandtrains.blogspot.com/2012_04_01_archive.html

[31] Wordpress, Arduino Parte II, [en linea], disponible en:

<http://extrusor3d.wordpress.com/2013/06/06/arduino-parte-ii/>

[32] Arduino CC, Arduino MEGA 2560, [en linea], disponible en:

<http://arduino.cc/en/Main/arduinoBoardMega2560>