



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería Eléctrica y Computación

“Diseño de un sistema de control mediante sistema Arduino utilizado para el manejo de bobinas de materiales en la impresión 3D y aplicación práctica utilizando prototipo”

TESINA DE SEMINARIO

Previo a la obtención del Título de:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL

Presentado por:

Leonidas Alberto Peralta Pérez

Eder Moisés Muñoz Zambrano

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTO

A Dios en primer lugar por darnos su bendición cada día de nuestras vidas, agradecemos a cada una de las personas que aportaron a nuestra formación humana, profesional y espiritual.

Ayudándonos así a sobre llevar y pasar cualquier obstáculo que se nos ha presentado, durante todo este proceso.

A nuestros padres y familiares por su apoyo incondicional, a nuestros amigos por ser partícipes de nuestra trayectoria universitaria y de nuestra vida personal.

DEDICATORIA

A mis padres, maestros y amigos
quienes depositaron su confianza
en mí para mi auto crecimiento
profesional

Alberto Peralta Pérez

A mis padres por su apoyo
incondicional en la formación de
mi vida, por siempre guiarme y
aconsejarme en mis decisiones, a
mi esposa e hijos que han sido los
motores de mi vida.

Moisés Muñoz

TRIBUNAL DE SUSTENTACIÓN

Ing. Carlos Valdivieso A.

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. Hugo Villavicencio V.

PROFESOR DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

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

(Reglamento de Graduación de Pregrado de la ESPOL)

LEONIDAS PERALTA PEREZ

EDER MUNOZ ZAMBRANO

RESUMEN

Este proyecto forma parte de un grupo de proyectos desarrollados en la escuela superior politécnica del litoral del Ecuador corresponde al seminario de graduación “Microcontroladores Avanzados” cuyo objetivo es el aprendizaje sobre el funcionamiento del control de las bobinas de la impresora 3D de manera sencilla y práctica. Donde la principal herramienta utilizada es la placa basada en un microcontrolador (ATmega2560), llamada Arduino, la cual basa su programación en el lenguaje C/C++.

Diseñaremos un sistema de control para el movimiento de bobinas inyectoras de material en una impresora tridimensional utilizando el sistema Arduino como parte central del control de rotación de motores “paso a paso”.

Se pondrá especial atención a las posibles mejoras que se puedan incluir en los diferentes módulos y se asentarán las bases necesarias para que cualquier usuario pueda construirse un sistema de control.

Por último, se implementará un software de control para comprobar de manera real la efectividad del movimiento de las bobinas, que serán capaces

de realizar movimientos en los tres ejes del espacio a través de un ordenador que servirá como punto de control.

ÍNDICE GENERAL

RESUMEN	i
INDICE GENERAL.....	iii
ABREVIATURAS	vi
INDICE DE FIGURAS.....	viii
INTRODUCCION	x
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. Impresora 3D.....	8
2.1.1. Descripción	9
2.2. Motor de Paso	10
2.3. Sensor de Fin de Carrera	12
2.4. Extrusora	13
2.5. Controlador Pololu.....	15
2.5.1. Descripción	16
2.6. RAMPS.....	17
2.6.1. Descripción	19
2.7. Tarjeta Arduino Mega	20
2.7.1. Descripción	21

2.8. Software	21
2.8.1.1. Entorno Interactivo de Desarrollo (IDE)	21
2.8.1.2. Descripción	22
2.8.2. Pronterface	22
CAPITULO 3	25
3. EJERCICIOS PREVIOS Y REALIZACIÓN DEL PROYECTO	25
3.1. Comunicación del Arduino y el Pc	26
3.1.1. Descripción	26
3.1.2. Ejemplo BLINK	27
3.1.3. Diagrama de bloques	29
3.1.4. Diagrama ASM	30
3.1.5. Descripción del algoritmo	30
3.1.6. Código Fuente	31
3.2. Secuencia de encendido de led's mediante retardos de tiempo... ..	32
3.2.1. Descripción	32
3.2.2. Diagrama de bloques	33
3.2.3. Diagrama ASM	34
3.2.4. Descripción del algoritmo	35
3.2.5. Código Fuente	36
3.3. Control de Motor de Paso mediante el Driver L298N	38
3.3.1. Descripción	38
3.3.2. Diagrama de bloques	39
3.3.3. Diagrama ASM	40
3.3.4. Descripción del algoritmo	40
3.3.5. Código Fuente	41
3.4. Selector de bobina de material para la impresión 3D utilizando Arduino y el driver L298N	46
3.4.1. Descripción	47

3.4.2.	Diagrama de bloques	51
3.4.3.	Diagrama ASM.....	52
3.4.4.	Descripción del algoritmo.....	53
CAPÍTULO 4.....		54
4.	IMPLEMENTACIÓN EJERCICIOS DE PRUEBA Y SIMULACIÓN DEL PROYECTO	54
4.1.	Comunicación del Arduino y el Pc.....	55
4.1.1.	Lista de Componentes	55
4.1.2.	Diagrama de conexiones	56
4.1.3.	Implementación.....	57
4.1.4.	Observaciones	57
4.2.	Secuencia de encendido de led's mediante retardos de tiempo... 58	
4.2.1.	Lista de Componentes	58
4.2.2.	Diagrama de conexiones	59
4.2.3.	Implementación.....	60
4.2.4.	Observaciones	60
4.3.	Control de Motores de Paso mediante Driver L289N	61
4.3.1.	Lista de Componentes	61
4.3.2.	Diagrama de conexiones	62
4.3.3.	Implementación.....	63
4.3.4.	Observaciones	63
4.4.	Selector de bobina de material para la impresión 3D utilizando Arduino y el driver L298N.....	63
CONCLUSIONES		69
RECOMENDACIONES.....		71
ANEXO 1.- Arduino Mega 2560 PIN diagram		74
ANEXO 2.- Arduino Mega 2560 PIN mapping table.....		75
ANEXO 3.- Especificaciones Técnicas L293D.....		78
REFERENCIAS BIBLIOGRÁFICAS.....		91

ABREVIATURAS

Amp	Amperios
ABS	Acrlonitrilo butadieno estireno
CC	Corriente Continua
CA	Corriente Alterna
CAD	Diseño asistido por computadora
GND	Tierra
I2C	Protocolo de comunicación serial
IDE	Entorno de desarrollo integrado
I/O	Entrada/Salida
LED	Diodo emisor de luz
MHZ	Un millón de Hertz
PC	Computadora Personal
PLA	Polímero de ácido Láctico
PWM	Modulación de Ancho de Pulso
STL	STereo Lithography (formato de archivo informático)
TX / RX	Transmisión / recepción
USB	Bus Universal en Serie
UTP	Par trenzado no blindado

V	Voltios
Vdd	Voltaje de alimentación +5V del microcontrolador
Vref	Voltaje de referencia
Vss	Voltaje de alimentación +0V del microcontrolador
3D	Tridimensional
°C	Grados centígrados

ÍNDICE DE FIGURAS

Figura 2.1. Esquema sencillo de la impresora 3D.....	9
Figura 2.2. Motor de paso.....	10
Figura 2.3. Sensor de fin de carrera	13
Figura 2.4. Extrusor termoplástico	14
Figura 2.5. Driver con chip Allegro A4988	16
Figura 2.6. Tarjeta Modular RAMPS	18
Figura 2.7. Esquemático conexiones de la RAMPS.....	19
Figura 2.8. Tarjeta Arduino Mega	20
Figura 2.9. Software de comunicación.....	22
Figura 2.10. Software de comunicación.....	23
Figura 3.1. Puerto serial en Arduino	26
Figura 3.2. Puerto serial.....	27
Figura 3.3. Verificación	28
Figura 3.4. Botón Reset.....	28
Figura 3.5. Cargar.....	28
Figura 3.6. LEDs TX- RX	29
Figura 3.7. Diagrama de bloques del ejemplo "Blink.....	29
Figura 3.8. Diagrama de flujo para cargar ejemplo "Blink"	30
Figura 3.9. Esquema de conexiones.....	33
Figura 3.10. Diagrama de bloques del control de encendido secuencial de led's mediante retardo de tiempo.....	33
Figura 3.11. Diagrama de flujo del control de encendido secuencial de led's mediante retardo de tiempo	34
Figura 3.12. Esquema de implementación del control de motor de paso mediante el driver L298N	38
Figura 3.13. Diagrama de bloques del control de posición de motores	39

Figura 3.14. Diagrama de flujo del ejemplo rutina de giro.....	40
Figura 3.15. Partes de la impresora Reprap Prusa Mendel	47
Figura 3.16. Firmware cargado en el entorno de desarrollo	49
Figura 3.17. Sistema completo	50
Figura 3.18. Diagrama de conexiones con la tarjeta Ramps	50
Figura 3.19. Diagrama de la implementación de la impresora.....	51
Figura 3.20. Diagrama bloques para la generación de los comandos de control.....	52
Figura 4.1. Diagrama interno de la tarjeta led pin 13.	56
Figura 4.2. Comunicación del Arduino y el Pc.	57
Figura 4.3. Diagrama de conexiones para el control de intensidad de leds usando la tarjeta ATmega2560.	59
Figura 4.4. Secuencia de encendido de led's.	60
Figura 4.5. Diagrama de conexiones para el control de posición de motores de paso usando la tarjeta ATmega2560.	62
Figura 4.6. Control de motor de paso mediante driver L298N	63
Figura 4.7. Maqueta para la simulación	65
Figura 4.8. Bobina 1 seleccionada.....	66
Figura 4.9. Bobina 2 seleccionada.....	66
Figura 4.10. Bobina 3 seleccionada.....	66
Figura 4.11. Sistema imprimiendo	67
Figura 4.12. Impresión terminada "leyenda"	68

INTRODUCCION

La impresora 3D es utilizada en el diseño industrial para materializar diseños virtuales creados en un software en formato CAD. Las Impresoras 3D es además una herramienta fundamental en el diseño y desarrollo de productos puesto que permiten realizar todo tipo de testeo del mismo antes de lanzarlo al mercado.

La denominada impresora 3D, está dirigida a asociar su simpleza de operación a la de una impresora de papel. “se materializa la pieza con sólo oprimir un botón” y se configuran los parámetros desde un “driver” similar al de cualquier impresora de chorro de tinta.

Para construcción de una impresora 3D se requiere de conocimientos técnicos y práctico, el hecho de poder obtener de manera física aquellos objetos que antes solo se podían imaginar, abre un mundo de posibilidades. Se puede añadir que el uso de impresoras 3D no solo mejora la creatividad y la innovación del usuario, sino que además supone una gran herramienta para la docencia [1].

El principio de funcionamiento es bastante parecido al de una impresora de cartuchos de tinta clásica, estas impresoras 3D pueden moldear formas tridimensionales depositando capas de un material plástico. Esto es, por

ejemplo, la forma en que en los laboratorios de Microsoft se realizan los prototipos de sus accesorios. Pero hasta ahora, estas “impresoras” eran enormes y su precio totalmente coherente con el tamaño de la máquina. Sin embargo, en los últimos meses, varios fabricantes se han embarcado en el desafío de fabricar impresoras 3D para ofrecerlas al público en general a un precio razonable [2].

CAPÍTULO 1

1. DESCRIPCIÓN GENERAL DEL PROYECTO

1.1. Descripción de la Planta

La impresora 3D que vamos a utilizar es de modelo Reprap Prusa Mendel, la cual viene con una boquilla de extrusión lo que nos va a permitir extrudir un filamento del material conocido como PLA y 5 motores los cuales están distribuidos de la siguiente manera: 1 motor para el eje x, 1 motor para el eje y, 2 motores para el eje z, 1 motor para la extrusora.

Utilizando Arduino como parte central de nuestro proyecto vamos a implementar un circuito electrónico, que con el uso de sensores mecánicos representados por led's, simularemos en una maqueta 3 bobinas, de las cuales se deberá utilizar solamente una para crear el

objeto deseado mediante un switch de 3 posiciones, es decir la maquina utilizara una de las tres bobinas.

El alcance de este diseño llega solamente a la selección de una de las bobinas mediante una lógica de programación C++ instalada en el Arduino que controla el movimiento de un motor de paso y analiza el estado de 3 sensores mecánicos.

La impresora se puede mover en 3 direcciones lineales, a través de los ejes X, Y y Z, también conocidas como coordenadas cartesianas. Para este proceso la impresora utiliza unos pequeños motores de paso que se pueden mover con bastante precisión y exactitud, usualmente 1.8° por paso, lo cual se traduce a un rango de resolución de fracciones de milímetros y que es la única forma a través de la cual estos motores de paso pueden ser controlados [3].

El interfaz de la impresora, es donde toda la cadena de herramienta se une. Desde esta aplicación podemos conectarnos con la impresora y hablar con su firmware, mover los tres diferentes ejes, leer y configurar la temperatura para la boquilla de la extrusora, ejecutar nuestro programa cortador, e imprimir nuestros modelos 3D.

1.2. Motivación

El poder aplicar el control de giro de un motor DC en una aplicación práctica que ya está brindando muchos beneficios a nuestra sociedad y simplificar tareas a futuras generaciones aplicando los conocimientos adquiridos previamente sobre “microcontroladores”, así como el auto beneficio de aprender el funcionamiento de lo que hace unos años atrás era simplemente producto de la imaginación.

La tecnología va en constante desarrollo por esto resulta motivante hacer uso de ella y aportar en algo en la investigación para próximos autómatas.

1.3. Identificación del Problema

Dado que el método de impresión 3D se basa en el control de posicionamiento mediante software. Es indispensable tener el conocimiento técnico de las funciones de las diferentes gamas de software que podemos encontrar para el control de la impresora, la configuración y la programación, es una de las dificultades que encontraríamos en la implementación del sistema. Siendo un punto de gran interés familiarizarse con el software implementado en este proyecto.

Otro problema específico, es la del control de los motores para la selección de la bobina a elegirse. Por lo cual es importante realizar varias pruebas con el propósito de familiarizarnos con la impresora y lograr la calibración del sistema.

1.4. Objetivos Principales

El objetivo principal para este proyecto de tesina Implementar un prototipo que nos permita seleccionar una bobina de material de impresión con el fin de seleccionar un color o simplemente reponer una bobina, de manera que el funcionamiento de nuestro sistema no sufra de interrupciones prolongadas, esta aplicación está basada en el microcontrolador ATmega2560 de Arduino y Prototipo de impresora Reprap Prusa Mendel.

Entre los objetivos específicos de nuestro proyecto tenemos:

- Reconocer y analizar el sistema.
- Mostrar el funcionamiento de la plataforma Arduino.
- Mostrar el manejo del interface para el control de la impresora 3D.
- Programar el Arduino.

- Simular el sistema diseñando de control para una respuesta específica.

1.5. Limitaciones

Puesto que para nuestro sistema de control, el elemento central es el Arduino deberemos familiarizarnos e informarnos correctamente sobre él, ya que presenta limitaciones como las de las salidas digitales que tiene Arduino y que debemos tener muy en cuenta para que todo funcione a la perfección.

Las salidas digitales del Arduino nos pueden ofrecer un máximo de 50mA (miliamperios) por lo que no debemos conectar directamente a su salida nada que tenga un consumo mayor que esos 50 mA, como por ejemplo los LED's de alta luminancia 750mA o 1025mA, motores de peristálticas con 500mA, etc.... Si conectamos directamente estos PERIFÉRICOS obligamos al Arduino a superar esos 50mA y quemaremos la placa o directamente los motores o led's no funcionarán.

En el caso de que queramos conectar un periférico que consuma más de lo que nuestro Arduino puede ofrecer, lo que tenemos que

hacer es incluir un pequeño circuito de potencia entre el Arduino y el periférico.

Como los driver de potencia que son circuitos integrados que nosotros alimentamos desde la fuente de alimentación y controlada desde el Arduino.

CAPÍTULO 2

2. FUNDAMENTO TEÓRICO

En este capítulo, se detalla las herramientas de hardware y software que se utilizará para el control de los motores de paso y su adecuado posicionamiento y describirá los elementos y componentes que dispone la impresora, debido que la impresión 3D se está convirtiendo en una revolución tecnológica y académica ayudando a la cristalización de diseño a escala de productos. Lo que en principio comenzó como una tecnología para imprimir pequeños objetos a partir de un archivo (.CAD), se ha convertido en un verdadero sistema de elaboración de maquetas en sectores como la arquitectura y en el diseño industrial, estos avances han mejorado el tiempo de elaboración de diseños de muestras de productos o implementaciones. [6].

Las impresoras 3D, son máquinas que pueden crear objetos tridimensionales a través de diferentes procesos y tecnologías, estos dispositivos ya no son de uso exclusivo de grandes empresas, sino que también existen versiones disponibles para aficionados, ya que cuenta con herramientas de fácil uso [7]. Esta tecnología incursiona en varias industrias como la armamentista, fotográfica y de telefonía celular; en la medicina también se está empezando a utilizar en la fabricación de prótesis, su campo de desarrollo es bastante elevado ya que el material con el que estas impresoras fabrican los objetos es muy variado. Uno de los materiales que puede ser impreso en tercera dimensión es el plástico.

2.1. Impresora 3D

La impresora 3D es una máquina que puede realizar movimientos en direcciones lineales, a través de los ejes X, Y y Z, también conocidas como coordenadas cartesianas. Para hacer esto la impresora utiliza unos pequeños motores de paso que se pueden mover con bastante precisión y exactitud, usualmente 1.8° por paso, lo cual se traduce a un rango de resolución de fracciones de milímetros y que es la única forma a través de la cual estos motores de paso pueden ser controlados.

La máquina es un explorador de código abierto diseñada para el aprendizaje. Sólo se necesita tiempo, recursos de ingeniería y paciencia para implementar una máquina. Las versiones se vuelven cada vez más fáciles, rápidas de construir y utilizar. El modelo implementado nos ofrecen la base que permite adquirir conocimientos de robótica, electrónica, programación, mecánica e interfaces [8].

2.1.1. Descripción

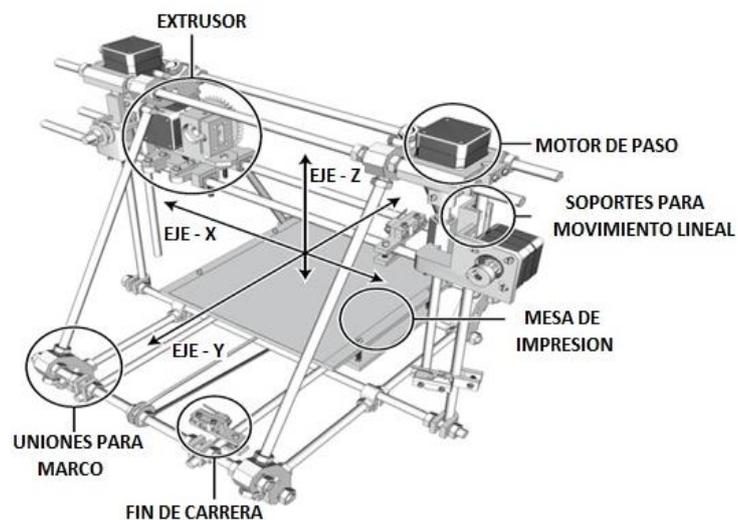


Figura 2.1. Esquema sencillo de la impresora 3D

➤ La impresora consta de 5 motores:

- 1 motor para el eje x
- 1 motor para el eje y

- 2 motores para el eje z
 - 1 para la extrusora
- Sensores de fin de carrera
 - Mesa (area de impresión)
 - Bobina (con plastico PLA)

2.2. Motor de Paso

El motor paso a paso es un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa es que es capaz de avanzar una serie de grados (paso) dependiendo de sus entradas de control.



Figura 2.2. Motor de paso

La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de tan solo 1.8° , es decir, que se necesitarán 4 pasos en el primer caso (90°) y 200 para el segundo caso (1.8°), para completar un giro completo de 360° [8].

Estos motores poseen la habilidad de poder quedar enclavados en una posición o bien totalmente libres. Si una o más de sus bobinas están energizada, el motor estará enclavado en la posición correspondiente y por el contrario quedará completamente libre si no circula corriente por ninguna de sus bobinas [9].

Este motor presenta las ventajas de tener alta precisión y repetitividad en cuanto al posicionamiento. Entre sus principales aplicaciones destacan como motor de frecuencia variable, motor de corriente continua sin escobillas, servomotores y motores controlados digitalmente.

Básicamente estos motores están constituidos normalmente por un rotor sobre el que van aplicados distintos imanes permanentes y por un cierto número de bobinas excitadoras bobinadas en su estator.

Las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) deber ser externamente manejada por un controlador.

2.3. Sensor de Fin de Carrera

Los sensores de fin de carrera son dispositivos ópticos o mecánicos que limitan el área de trabajo de la impresora [Figura 2.3]. Básicamente, son interruptores que indican al controlador de la impresora cuando se ha alcanzado un límite en una dirección de movimiento con el fin de evitar que el eje se mueva más allá de su límite [14].

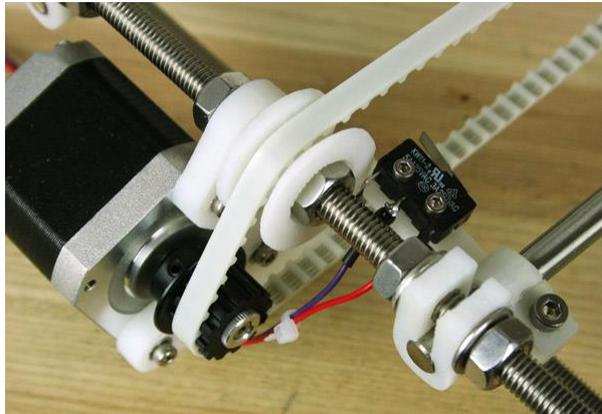


Figura 2.3. Sensor de fin de carrera

Los sensores de final de carrera son necesarios para el funcionamiento de la máquina, el objetivo de los sensores es obtener un tope de máxima posición en cada eje permitiendo que la impresora tenga un tipo de protección y previniendo constantes calibraciones y a su vez nos ayudaran a definir un punto de referencia [15].

2.4. Extrusora

La extrusora es el dispositivo encargado de dar forma a los diseños tridimensionales, cuando el plástico se somete a calentamiento el material toma estado semilíquido estando en la capacidad de establecer líneas finas de material termoplástico en el la trayectoria que toma su movimiento [10]. La extrusora Fig.2.4, sin duda la parte

más compleja de una impresora 3D que continúa disfrutando de un intenso desarrollo.

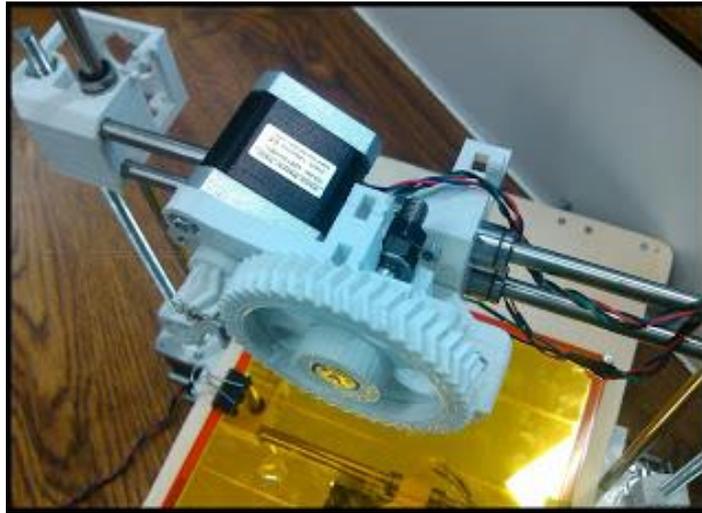


Figura 2.4. Extrusor termoplástico

La extrusora está constituida por una resistencia para el calentamiento, una termocupla para el control de la temperatura y un motor paso a paso para controlar mejor el flujo de plástico en el extremo caliente. Estos motores están a menudo orientados hacia abajo con los engranajes impresos o una caja de cambios integral y cuando el plástico alcanza el extremo caliente, tal calentamiento estará alrededor de 170°C a 220°C , dependiendo del plástico a extruir. Una vez en un estado semilíquido, el plástico es forzado a través de una boquilla del extrusor, con una abertura alrededor de

0,35 milímetros a 0,5 milímetros de diámetro, para colocar este material en caliente sobre delgada líneas del dibujo en el contorno de la capa o llenar esa capa utilizando algún tipo de patrón de relleno que dan forma a la impresión [11].

2.5. Controlador Pololu

El driver Pololu es una interface entre el microcontrolador y el motor de paso está constituido por un chip Allegro A4988 [12] que es utilizado como controlador de motor de paso el cual nos permite controlar un motor de paso bipolar y cuenta con limitación de corriente regulable y protección de sobre corriente.

El driver Pololu tiene la capacidad de controlar un dispositivo a la vez teniendo la necesidad de utilizar un driver por cada eje de la impresora.

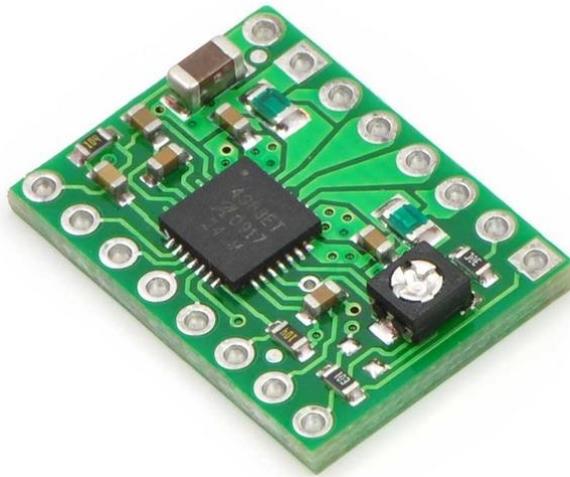


Figura 2.5. Driver con chip Allegro A4988

2.5.1. Descripción

- Paso simple y la interfaz de control de dirección.
- Control de corriente ajustable le permite ajustar la intensidad de corriente de salida máxima con un potenciómetro que permite utilizar tensiones superiores a su tensión nominal de corriente del motor paso a paso para lograr mayores tasas de paso.
- Control de picar inteligente que selecciona automáticamente el modo correcto de decaimiento actual (decaimiento rápido o lento decaimiento). Exceso de temperatura, apagado térmico, bloqueo de baja tensión y la corriente de cruce de protección.

2.6. RAMPS

La RepRap Arduino Mega Pololu Shield (RAMPS) está diseñada para adaptarse a la electrónica de la tarjeta Arduino [13]. La RAMPS consta de un diseño modular con gran margen para posibles expansiones, sirviendo como interfaz entre la Arduino y la impresora [Fig. 2.6].

Las conexiones electrónicas de la impresora con la tarjeta Ramps son más sencillas por su diseño modular y el fácil montaje de los drivers. Todos los elementos que necesitan algún tipo de conexión (motores, sensores) se conectan a la placa RAMPS en el lugar específicos designado en los pines de salida [10].

El controlador RAMPS representa una reducción de tamaño global y del costo en estos dispositivos de control, por lo que es una elección recomendada. De esta forma, los componentes necesarios para la electrónica de una impresora 3D son: Arduino Mega + RAMPS + 4 drivers de motores, estos controladores incluyen:

- Estándar de la electrónica Arduino, haciendo de la plataforma más fácil de usar.

- Salida de potencia adicional para ventiladores intercambiables de calor
- El apoyo futuro para extrusoras de doble cabezal de impresión.



Figura 2.6. Tarjeta Modular RAMPS

2.6.1. Descripción

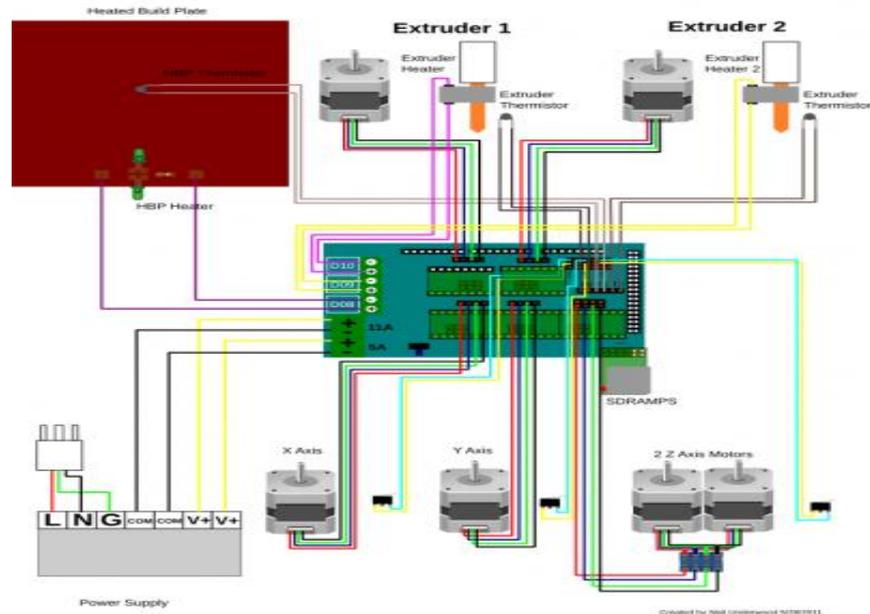


Figura 2.7. Esquemático conexiones de la RAMPS

- Hasta cinco tarjetas driver (A4988).
- Tres cargas de alta potencia de conmutación con un fusible de 5 amperios para la utilización de la extrusora y la mesa de impresión.
- Seis conexiones para los sensores de fin de carrera.
- Tres conexiones para la utilización de termistores.
- Doble entrada de alimentación de 12 a 35 voltios de hasta 16 amperios.
- Pins I2C y SPI queda disponible para una futura expansión.

2.7. Tarjeta Arduino Mega

La tarjeta Arduino mega [Fig. 2.8] es una plataforma open-hardware basada en una sencilla placa con entradas y salidas (I/O), analógicas y digitales y con un microcontrolador ATmega2560 y un entorno de desarrollo integrado (IDE) [16].

La tarjeta está diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios, por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader) que corre en la placa [17].

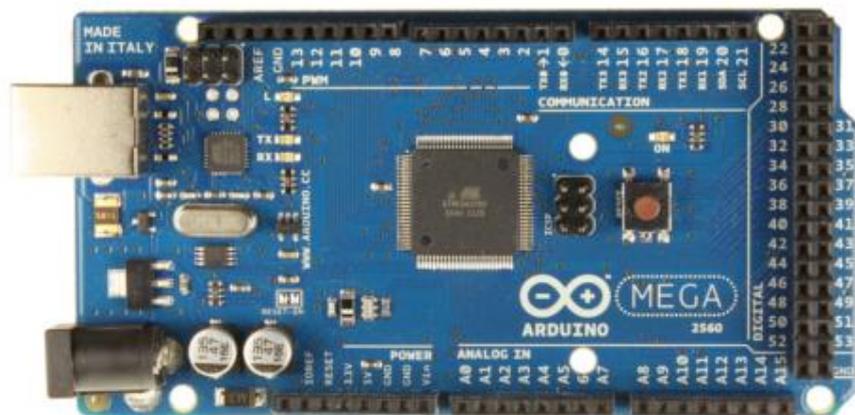


Figura 2.8. Tarjeta Arduino Mega

2.7.1. Descripción

- Microcontrolador ATmega2560
- Voltaje de entrada (recomendado): 7 a 12V
- Voltaje de entrada (limite): 6 a 20V
- 54 pines digitales I/O (14 salidas PWM)
- 16 entradas analógicas
- 256k de memoria flash
- Velocidad de reloj 16MHz

2.8. Software

2.8.1.1. Entorno Interactivo de Desarrollo (IDE)

El IDE de Arduino permite escribir códigos y cargarlos en la placa Arduino, diseñado específicamente para familiarizarnos con el desarrollo de software. Es una aplicación derivada de la programación IDE del lenguaje Processing el cual incluye un procesador de texto especializado en escribir programas [18].

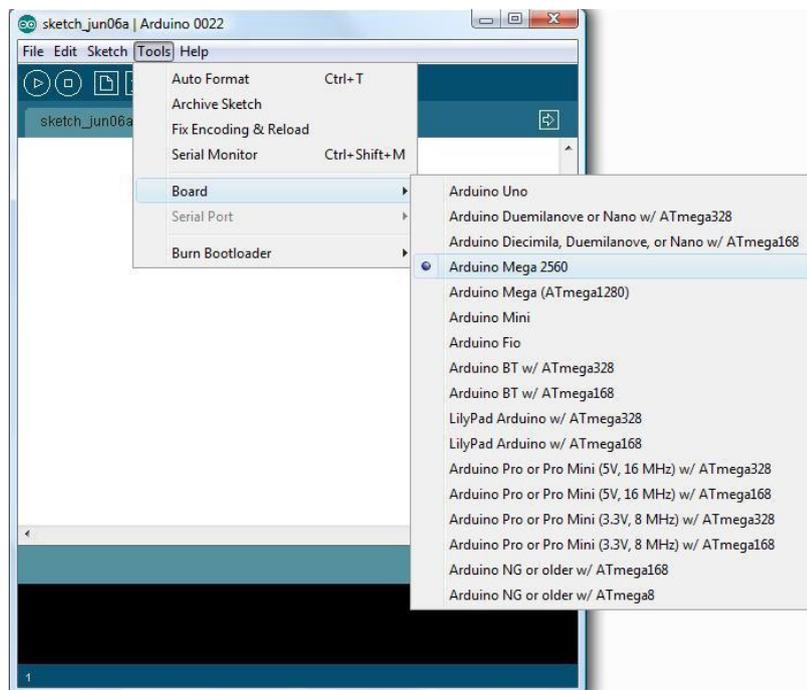


Figura 2.9. Software de comunicación

2.8.1.2. Descripción

- Editor de código.
- Resaltado de sintaxis.
- Compila y crea el programa máquina.
- Carga el programa al procesador de la tarjeta.
- Biblioteca escrita en C++

2.8.2. Pronterface

El Pronterface es un software de interface gráfico que nos permite acceder directamente a la configuración del programa

[Fig. 2.10], para editar y calibrar las impresiones. Tiene un entorno gráfico que nos permitirá manipular los ejes de nuestra impresora, calentar el extrusor y monitorizará la temperatura permanentemente, cambiar la velocidad de transmisión de datos.



Figura 2.10. Software de comunicación

Pronterface permite cargar archivos Gcode, que se podrán imprimir directamente, pero también permite cargar archivos en formato STL; en ese caso, se conecta con el software de creación de archivos Gcode que tiene integrado Pronterface,

generando y guardándolo automáticamente, en el propio Pronterface para que podamos imprimirlo cuando queramos.

Tiene una ventana en la que podemos ver las diferentes capas que tiene de la figura a imprimir, y otra ventana que nos informa de las dimensiones de la pieza y nos da una estimación del tiempo que tardará en imprimir el objeto 3D.

CAPÍTULO 3

3. EJERCICIOS PREVIOS Y REALIZACIÓN DEL PROYECTO

En el siguiente capítulo se desarrollaron un conjunto de ejercicios fundamentales para relacionarnos con el microcontrolador ATmega2560 de la plataforma Arduino; los cuales nos permitieron familiarizarnos con el microcontrolador y poder realizar avances en nuestro proyecto final.

Se realizó la verificación de la comunicación de la tarjeta Arduino con el Pc y dos ejemplos con el propósito de capacitarnos en el manejo de estas nuevas herramientas entre los que se encuentran:

- Comunicación del Arduino y el Pc mediante un ejemplo de la librería del entorno de desarrollo (IDE).
- Secuencia de encendido de led's mediante retardos de tiempo.
- Control de motor de paso mediante driver I298.

Con la finalidad de analizar y comprender el funcionamiento de cada uno de los ejercicios, realizamos las respectivas descripciones, diagrama de bloques, análisis de los algoritmos y códigos fuentes de la programación en C.

3.1. Comunicación del Arduino y el Pc

3.1.1. Descripción

En este ejemplo se explicara el procedimiento para configurar la comunicación entre el Arduino y el pc mediante el entorno de desarrollo (IDE). Para ello deberemos abrir el menú “Herramienta” la opción “Puerto Serial” (Fig. 3.1). En esta opción deberemos seleccionar el puerto serie al que está conectada nuestra placa [19].

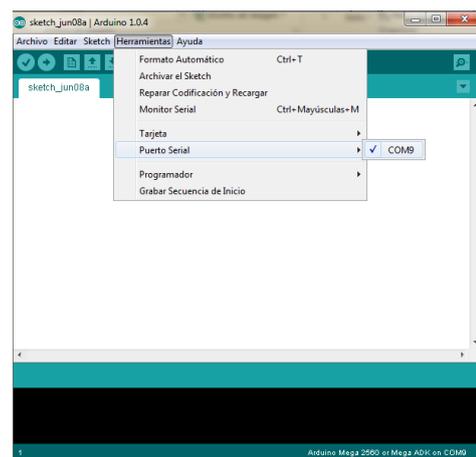


Figura 3.1. Puerto serial en Arduino

Para comprobar la comunicación, se debe abrir un ejemplo. Se recomienda abrir el ejemplo “Blink”. Para ello debemos acceder a través del menú Archivo → Ejemplos → 0.1basic → Blink (Fig. 3.2).

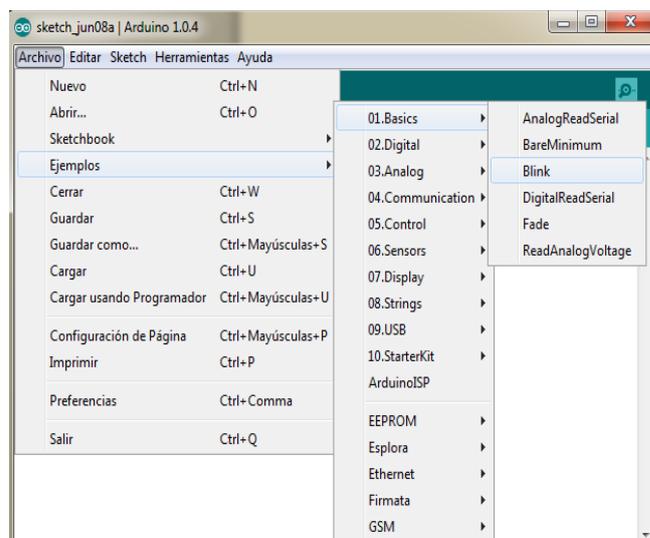


Figura 3.2. Puerto serial

3.1.2. Ejemplo BLINK

El procedimiento básico para comprobar la comunicación del Arduino con el Pc es la implementación del ejemplo “Blink” del entorno de desarrollo (IDE), el cual hará parpadear un LED que está colocado en el pin número 13 de la placa. Luego de analizar y comprobar el código fuente, pulsamos el botón de verificación indicado en la (Fig. 3.3). Aparecerá un mensaje en

la parte inferior de la interfaz indicando “Done compiling”, el cual nos indica que código ha sido verificado y procederemos a cargarlo en la placa. Para ello tenemos que pulsar el botón de reset de la placa (Fig. 3.4). E inmediatamente después pulsar el botón que comienza la carga (Fig. 3.5).

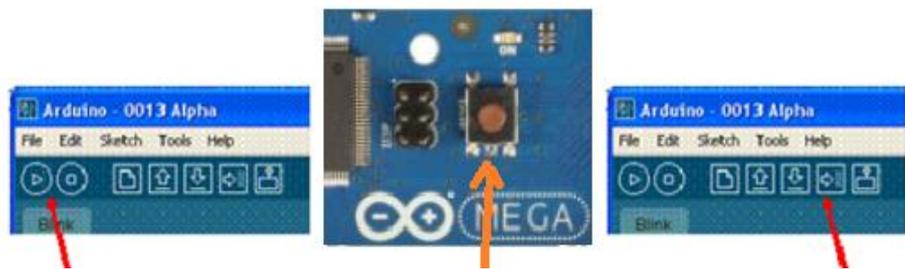


Figura 3.3. Verificación Figura 3.4. Botón Figura 3.5. Cargar
Reset

Durante el proceso de transmisión de datos, se encenderán los Led's TX / RX (Fig 3.6) los cuales indican que los datos han sido enviados y recibido por el puerto serie. Si el programa a cargado correctamente debe aparecer el mensaje “done uploading”. Después de unos 8 segundos aproximadamente el LED del pin 13 de la placa se enciende y se apaga a razón de cada segundo.

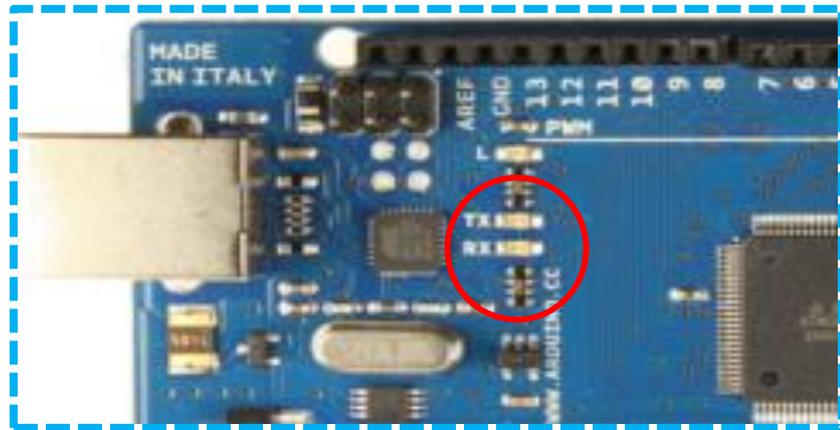


Figura 3.6. LEDs TX- RX

3.1.3. Diagrama de bloques

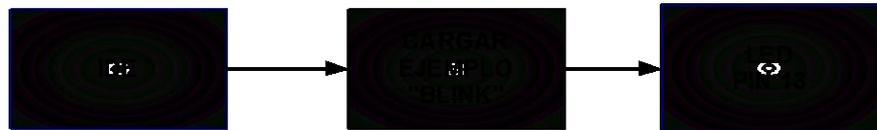


Figura 3.7. Diagrama de bloques del ejemplo "Blink"

3.1.4. Diagrama ASM

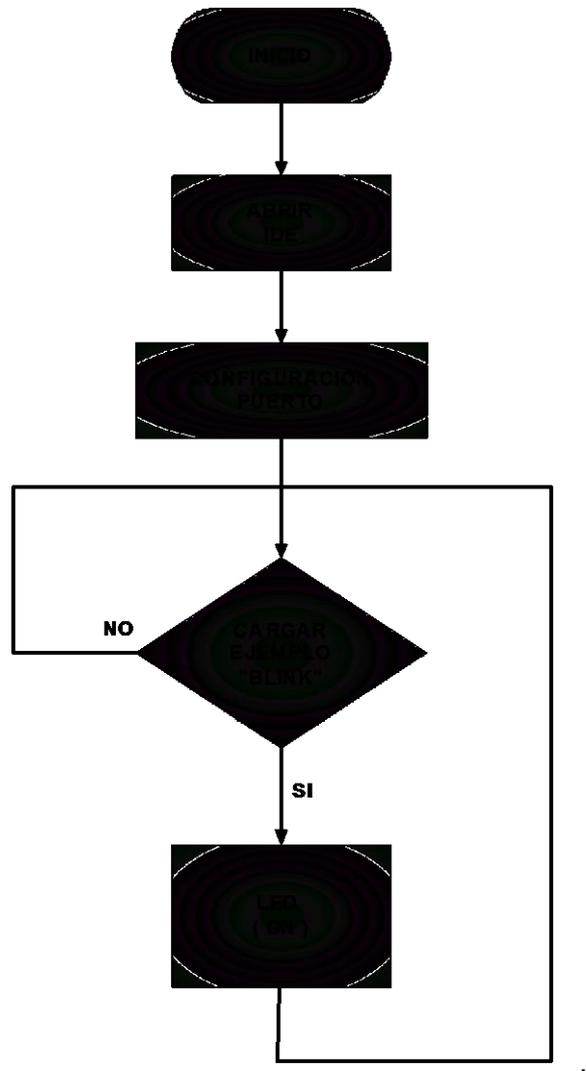


Figura 3.8. Diagrama de flujo para cargar ejemplo "Blink"

3.1.5. Descripción del algoritmo

- Abrir el entorno de desarrollo (IDE).

- Conectar la tarjeta al PC.
- Configurar IDE (seleccionar puerto).
- Cargar ejemplo (BLINK).
- Comprobar parpadeo del LED (pin 13).

3.1.6. Código Fuente

```
/* Blink

int led = 13;

void setup() {

    pinMode(led, OUTPUT);

}

void loop() {

    digitalWrite(led, HIGH);    // enciende el LED ( nivel alto
de tensión)

    delay(1000// esperar un segundo

    digitalWrite(led, LOW);    // apagar el LED ( nivel bajo
de tensión)
```

```
delay(1000);           // esperar un segundo  
  
}
```

3.2. Secuencia de encendido de led's mediante retardos de tiempo

3.2.1. Descripción

En este ejemplo se desarrollara el control de encendido de led's mediante retardo de tiempo, la simulación se la realizara mediante la representación de 3 led's de diferentes colores, los cuales cambiarán su estado on/off en forma secuencial en un ciclo infinito [20].

Los ánodos de los Led's están conectados a los pines digitales 5, 6 y 7 y mediante un cable de puente desde GND de la Arduino para el carril de tierra hasta el cátodo de los Led's a través de una resistencia de 330Ω .

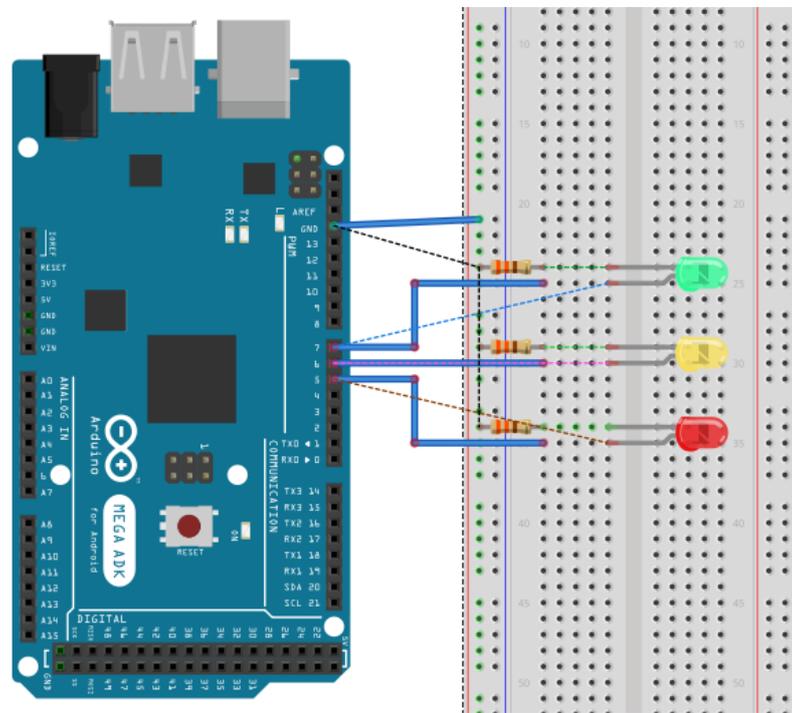


Figura 3.9. Esquema de conexiones

3.2.2. Diagrama de bloques



Figura 3.10. Diagrama de bloques del control de encendido secuencial de led's mediante retardo de tiempo

3.2.3. Diagrama ASM

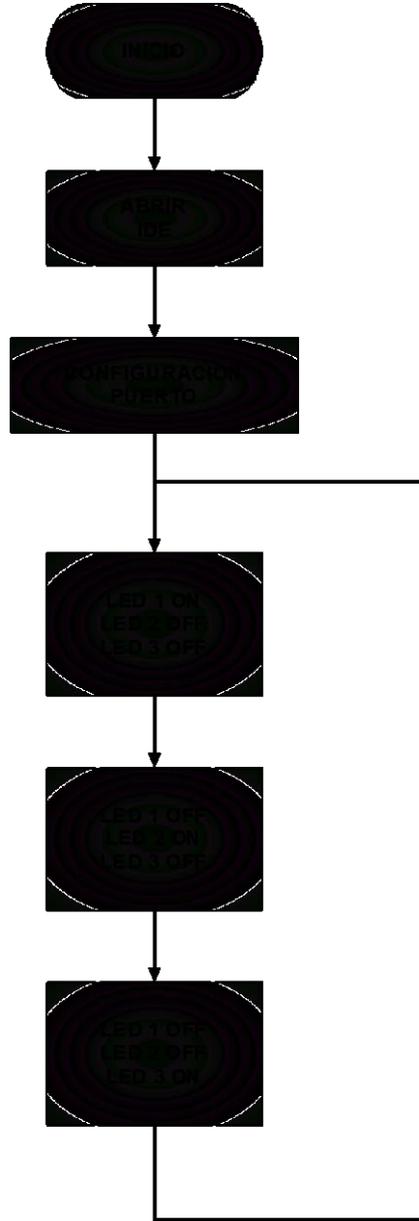


Figura 3.11. Diagrama de flujo del control de encendido secuencial de led's mediante retardo de tiempo

3.2.4. Descripción del algoritmo

- Se inicializan las variables.
- Se configuran los puertos referentes a las variables que se están utilizando.
- Se ejecuta el programa:
 - Se habilita el puerto 5, enviando una señal de nivel de voltaje alto a led verde encendiéndolo y manteniendo en estado ON por un segundo mientras están deshabilitados los puertos 6 y 7.
 - Se deshabilita el puerto 5, enviando una señal de nivel de voltaje bajo a led verde apagándolo y manteniendo en estado OFF hasta completar el primer ciclo de encendido.
 - Se habilita el puerto 6, enviando una señal de nivel de voltaje alto a led amarillo encendiéndolo y manteniendo en estado ON por un segundo mientras están deshabilitados los puertos 5 y 7.

- Se deshabilita el puerto 6, enviando una señal de nivel de voltaje bajo a led amarillo apagándolo y manteniendo en estado OFF hasta completar el primer ciclo de encendido.
- Se habilita el puerto 7, enviando una señal de nivel de voltaje alto a led rojo encendiéndolo y manteniendo en estado ON por un segundo mientras están deshabilitados los puertos 5 y 6.
- Se deshabilita el puerto 7, enviando una señal de nivel de voltaje bajo a led rojo apagándolo y manteniendo en estado OFF completando el primer ciclo.
- Después que todos los leds hayan prendido y apagado respectivamente, el procedimiento se volverá a repetir en el lazo infinito.

3.2.5. Código Fuente

```
//      REALIZACIÓN DE EJERCICIO  
  
// EJEMPLO #1  
  
//LEONIDAS ALBERTO PERALTA PEREZ
```

```
//EDER MOISES MUÑOZ ZAMBRANO

//    DECLARACIONES DE VARIABLES QUE USAREMOS

int LED1 = 5;

int LED2 = 6;

int LED3 = 7;

void setup()

{

pinMode(LED1, OUTPUT);

pinMode(LED2, OUTPUT);

pinMode(LED3, OUTPUT);

}

void loop()

{

digitalWrite(LED1, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000);           // wait for a second

digitalWrite(LED1, LOW); // turn the LED off by making the voltage LOW

delay(1000);

digitalWrite(LED2, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000);           // wait for a second

digitalWrite(LED2, LOW); // turn the LED off by making the voltage LOW

delay(1000);

digitalWrite(LED3, HIGH); // turn the LED on (HIGH is the voltage level)
```

3.3. Control de Motor de Paso mediante el Driver L298N

3.3.1. Descripción

En este ejemplo vamos a crear una rutina de giros para un motor de paso, el cual podremos controlar el sentido de giro y velocidad en una secuencia controlada, para lo cual utilizaremos el driver de potencia basado en el integrado L298. El funcionamiento es sencillo al iniciar la secuencia, el motor girará hacia un sentido por 5 segundos por una velocidad constante, luego cambiará su sentido a la mitad de la velocidad, el motor para, luego seguirá una secuencia de simulando el movimiento de un reloj [21].

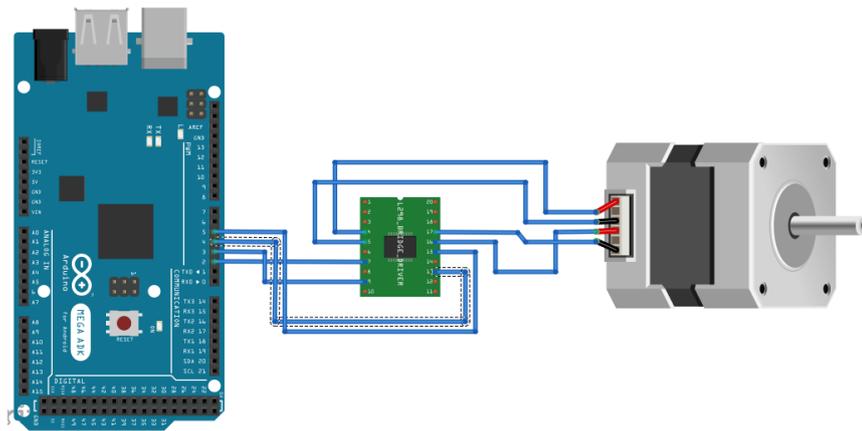


Figura 3.12. Esquema de implementación del control de motor de paso mediante el driver L298N

3.3.2. Diagrama de bloques

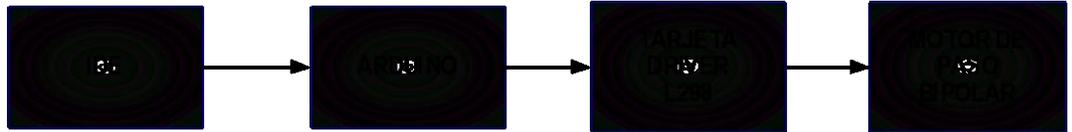


Figura 3.13. Diagrama de bloques del control de posición de motores

3.3.3. Diagrama ASM

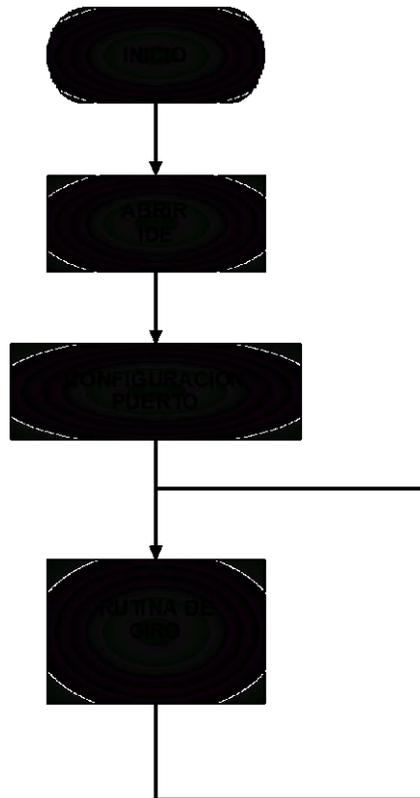


Figura 3.14. Diagrama de flujo del ejemplo rutina de giro

3.3.4. Descripción del algoritmo

- Se inicializan las variables.
- Se configuran los puertos referentes a las variables que se están utilizando.
- Se ejecuta el programa.

- El motor de paso se moverá hacia el lado derecho durante 5 segundos,
- Después que el motor se haya detenido se moverá al sentido contrario durante 5 segundos a la mitad de la velocidad.
- Después que el motor se haya detenido se moverá tal manera que simule el movimiento de un reloj.
- El proceso se repetirá continuamente en lazo infinito

3.3.5. Código Fuente

```
// PERALTA PEREZ LEONIDAS
// MUNOZ ZAMBRANO EDER
// REALIZACIÓN DE EJERCICIO
// CONTROL DE MOTOR DE PASO

// DECLARACIONES DE VARIABLES QUE USAREMOS
// EN EL TERCER EJEMPLO DE PRÁCTICA

int switch1=42; int boton1=38;

int led1=34; int led2=30;

int A=2; int B=3; int C=4; int D=5; // VARIABLES PARA LA BOBINAS DEL
PRIMER MOTOR DE PASO

int mSecPaso=1500; // VARIABLE DE TIEMPO DE CONMUTACIÓN

//
```

```
//      EN VOID SETUP SE DECLARARÁN LAS ENTRADAS
//      Y SALIDAS DE LOS PUERTOS
//

void setup()
{
//      VARIABLES DE ENTRADAS Y SALIDAS
//      PARA CONMUTADORES Y INDICADORES
pinMode(switch1, INPUT);
pinMode(boton1, INPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);

// VARIABLES DE BOBINAS DE MOTOR 1
pinMode(A, OUTPUT);
pinMode(B, OUTPUT);
pinMode(C, OUTPUT);
pinMode(D, OUTPUT);

//      INICIALIZACIÓN DE BOBINAS
//      DE MOTOR 1 EN BAJO
digitalWrite(A, LOW);
digitalWrite(B, LOW);
digitalWrite(C, LOW);
digitalWrite(D, LOW);
```

```
}  
  
//    PROGRAMA DE GIRO DE MOTOR 1  
  
//    HACIA LA IZQUIERDA  
  
void izquierdamotor1(){  
  
    //paso 1  
  
    digitalWrite(A, LOW);  
    digitalWrite(B, HIGH);  
    digitalWrite(C, HIGH);  
    digitalWrite(D, LOW);  
  
    delayMicroseconds(mSecPaso);  
  
    //paso 2  
  
    digitalWrite(C, LOW);  
    digitalWrite(D, HIGH);  
  
    delayMicroseconds(mSecPaso);  
  
    //paso 3  
  
    digitalWrite(A, HIGH);  
    digitalWrite(B, LOW);  
  
    delayMicroseconds(mSecPaso);  
  
    //paso 4  
  
    digitalWrite(C, HIGH);  
    digitalWrite(D, LOW);  
  
    delayMicroseconds(mSecPaso);  
  
} //FIN izquierda
```

```
//      PROGRAMA DE GIRO DE MOTOR 1
//      HACIA LA DERECHA
void derechamotor1(){
//paso 1
digitalWrite(A, HIGH);
digitalWrite(B, LOW);
digitalWrite(C, HIGH);
digitalWrite(D, LOW);
delayMicroseconds(mSecPaso);
//paso 2
digitalWrite(C, LOW);
digitalWrite(D, HIGH);
delayMicroseconds(mSecPaso);
//paso 3
digitalWrite(A, LOW);
digitalWrite(B, HIGH);
delayMicroseconds(mSecPaso);
//paso 4
digitalWrite(C, HIGH);
digitalWrite(D, LOW);
delayMicroseconds(mSecPaso);
} //FIN derecha
//
```

```
//    PROGRAMA PRINCIPAL Y EJECUCION DE LAS
//    RUTINAS DEL EJERCICIO
void loop()
{
    //    CONDICIÓN PARA LA INICIALIZACION
    //    DEL PROGRAMA
    if (digitalRead(switch1)==LOW)
    {
        for(int i=0; i<1000; i++)
        {
            digitalWrite(led1,HIGH); // ENCIENDE EL LED 1
            digitalWrite(led2,LOW); // MANTIENE APAGADO EL LED 2
            derechamotor1();    // PROGRAMA DEL MOTOR 1 GIRA HACIA LA
DERECHA
        }
    }
    else
    {
        //SWITCH NO ES ACTIVADO
        digitalWrite(led1,LOW); //APAGA EL LED 1
        digitalWrite(led2,LOW); //APAGA EL LED 2
        //SALIDAS DEL MOTOR DE PASO 1 EN BAJO
        digitalWrite(A, LOW);
        digitalWrite(B, LOW);
    }
}
```

```
digitalWrite(C, LOW);  
digitalWrite(D, LOW);  
}  
}
```

3.4. Selector de bobina de material para la impresión 3D utilizando Arduino y el driver L298N

El objetivo de este proyecto se basa en el control y manejo de motores de paso, para la selección de bobinas de material de impresión 3D con el propósito de seleccionar el color o simplemente reponer material en el proceso de impresión, con el fin de crear objetos tridimensionales, mediante la superposición de capas de material PLA o ABS empleando un sistema con Arduino y Pronterface que es un interfaz de visualización e iteración con el usuario [10].

Para la implementación del proyecto se hará uso de un programa que se cargara en el Arduino, el prototipo de impresora Reprap Prusa Mendel y una interfaz que permite la transmisión y recepción de datos para el control de los diferentes motores de la impresora, es necesario también conocer el funcionamiento del sistema de

impresión 3D, por lo cual se indicara las partes que conforman la impresora 3D desde un punto de vista práctico [12].

3.4.1. Descripción

Haciendo uso del prototipo Reprap Prusa Mendel DIY se procedió al ensamblado de la impresora 3D, se realizaron los cambios necesarios en la impresora adaptando de la tarjeta Arduino en la implementación del proyecto, en el diagrama de bloques figura 3.14 se observa las partes en que está conformada la impresora tales como: motores, sensores, tarjetas, drivers, extrusor y bobinas con material de impresión [22].

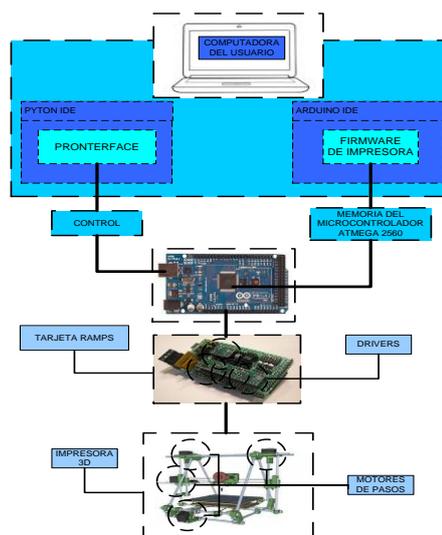


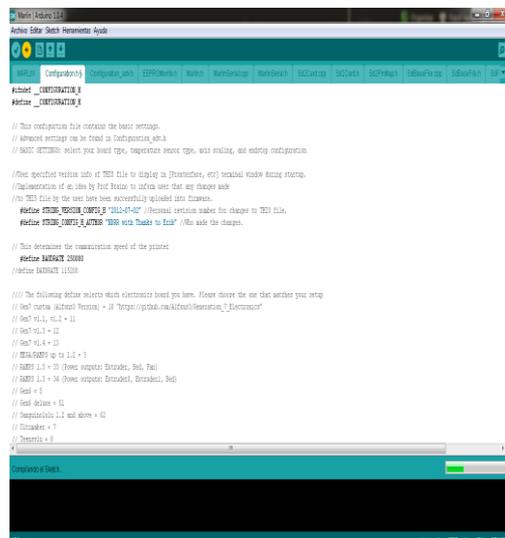
Figura 3.15. Partes de la impresora Reprap Prusa Mendel

El hardware está estructurada de la siguiente manera, La computadora donde se instalara el entorno de desarrollo en el cual se podrá configurar y modificar el programa antes de la instalación en la tarjeta, también se instalara pronterface, que es el software de interacción con el usuario y controlador de nuestro sistema. La tarjeta arduino con el controlador atmega2560, el cual hará lectura de los diferentes sensores y activará los dispositivos necesarios para la impresión. Las board ramps es un tarjeta modular donde se encuentra la parte de fuerza y adaptación de las drivers y finalmente tenemos la impresora que esta constituida por el motor extrusor que es el encargado de proveer de material al cabezal de impresion, los motores de los ejes (x,y,z) los cuales daran los movimiento del cabezal en el transcurso de impresión y los sensores fin de carrera.

Los primeros pasos en el desarrollo del proyecto es la instalación del entorno de desarrollo integrado (IDE) de Arduino en la PC, luego cargamos el firmware (Marlín) y se procede hacer los cambios en la configuración para adaptar la tarjeta

Arduino al sistemas, unos de los cambio se basó en la velocidad de comunicación que se tiene en el Pronterface y el firmware, por las pruebas realizadas de comunicación e impresión, se aconseja utilizar la máxima velocidad de comunicación de transmisión de datos de 250000 BAUDRATE.

Con los cambios realizados procedemos a cargar el firmware en el Arduino, luego comunicamos el Pronterface con el Arduino y accediendo a las aplicaciones de Pronterface verificamos el funcionamiento mediante lo movimientos del cabezal en los ejes (x, y, z) y el calentamiento del extrusor.



```

// This configuration file contains the basic settings.
// Advanced settings can be found in Configuration_adv.h
// BASIC SETTINGS: select your board type, temperature sensor type, noise filtering, and endstop configuration

// Other specified settings info of TMC files to display in [Information, etc] terminal window during startup.
// Description of an idea by JoulBrook to inform user that any changes made.
// To TMC file by the user have been successfully uploaded into firmware.
#define TMC_CURRENT_SENSOR_CONF 0 "0111-01-01" //Personal version enables the changes to TMC files.
#define TMC_CURRENT_SENSOR_CONF "0111-01-01" //Personal version enables the changes.

// This determines the communication speed of the printer
#define BAUDRATE 250000
//Serial baudrate 115200

// The following defines select which electronics board you have. Please choose the one that matches your setup
// Don't forget to update the board ID in Configuration_adv.h
#define BOARD_ID 10 // (https://github.com/klipper/klipper/blob/master/src/gcode/printer_common/printer_common.h)
// Board ID 10
// Board ID 11
// Board ID 12
// Board ID 13
// Board ID 14
// Board ID 15
// Board ID 16
// Board ID 17
// Board ID 18
// Board ID 19
// Board ID 20
// Board ID 21
// Board ID 22
// Board ID 23
// Board ID 24
// Board ID 25
// Board ID 26
// Board ID 27
// Board ID 28
// Board ID 29
// Board ID 30
// Board ID 31
// Board ID 32
// Board ID 33
// Board ID 34
// Board ID 35
// Board ID 36
// Board ID 37
// Board ID 38
// Board ID 39
// Board ID 40
// Board ID 41
// Board ID 42
// Board ID 43
// Board ID 44
// Board ID 45
// Board ID 46
// Board ID 47
// Board ID 48
// Board ID 49
// Board ID 50
// Board ID 51
// Board ID 52
// Board ID 53
// Board ID 54
// Board ID 55
// Board ID 56
// Board ID 57
// Board ID 58
// Board ID 59
// Board ID 60
// Board ID 61
// Board ID 62
// Board ID 63
// Board ID 64
// Board ID 65
// Board ID 66
// Board ID 67
// Board ID 68
// Board ID 69
// Board ID 70
// Board ID 71
// Board ID 72
// Board ID 73
// Board ID 74
// Board ID 75
// Board ID 76
// Board ID 77
// Board ID 78
// Board ID 79
// Board ID 80
// Board ID 81
// Board ID 82
// Board ID 83
// Board ID 84
// Board ID 85
// Board ID 86
// Board ID 87
// Board ID 88
// Board ID 89
// Board ID 90
// Board ID 91
// Board ID 92
// Board ID 93
// Board ID 94
// Board ID 95
// Board ID 96
// Board ID 97
// Board ID 98
// Board ID 99
// Board ID 100

```

Figura 3.16. Firmware cargado en el entorno de desarrollo

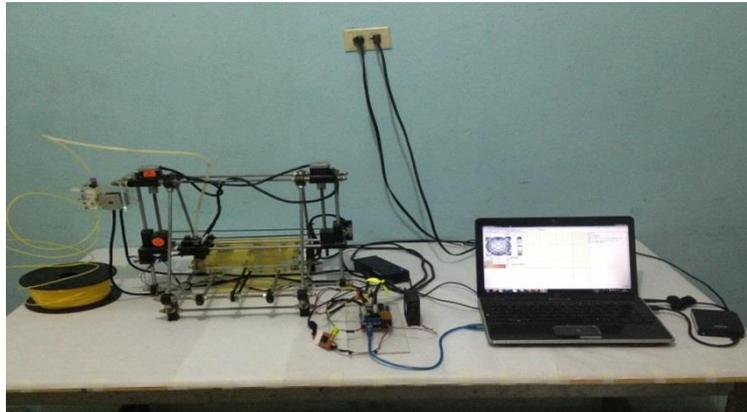


Figura 3.17. Sistema completo

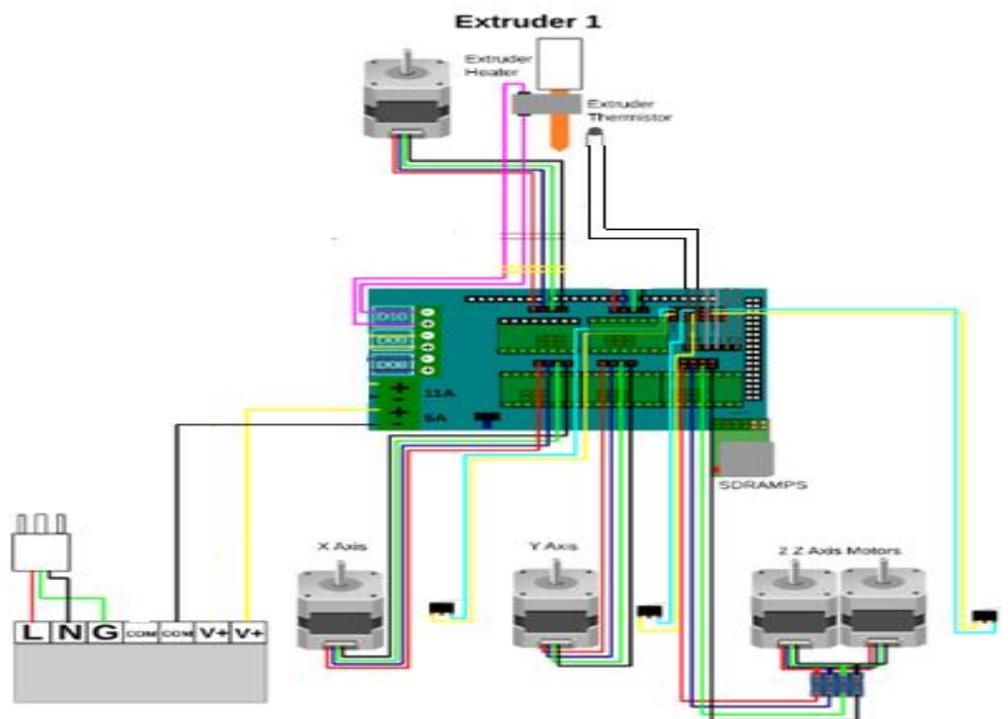


Figura 3.18. Diagrama de conexiones con la tarjeta Ramps

3.4.2. Diagrama de bloques

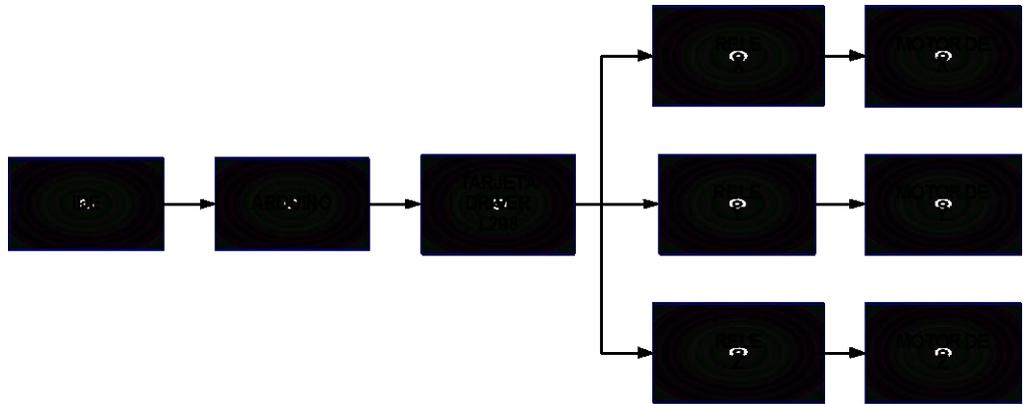


Figura 3.19. Diagrama de la implementación de la impresora

3.4.3. Diagrama ASM

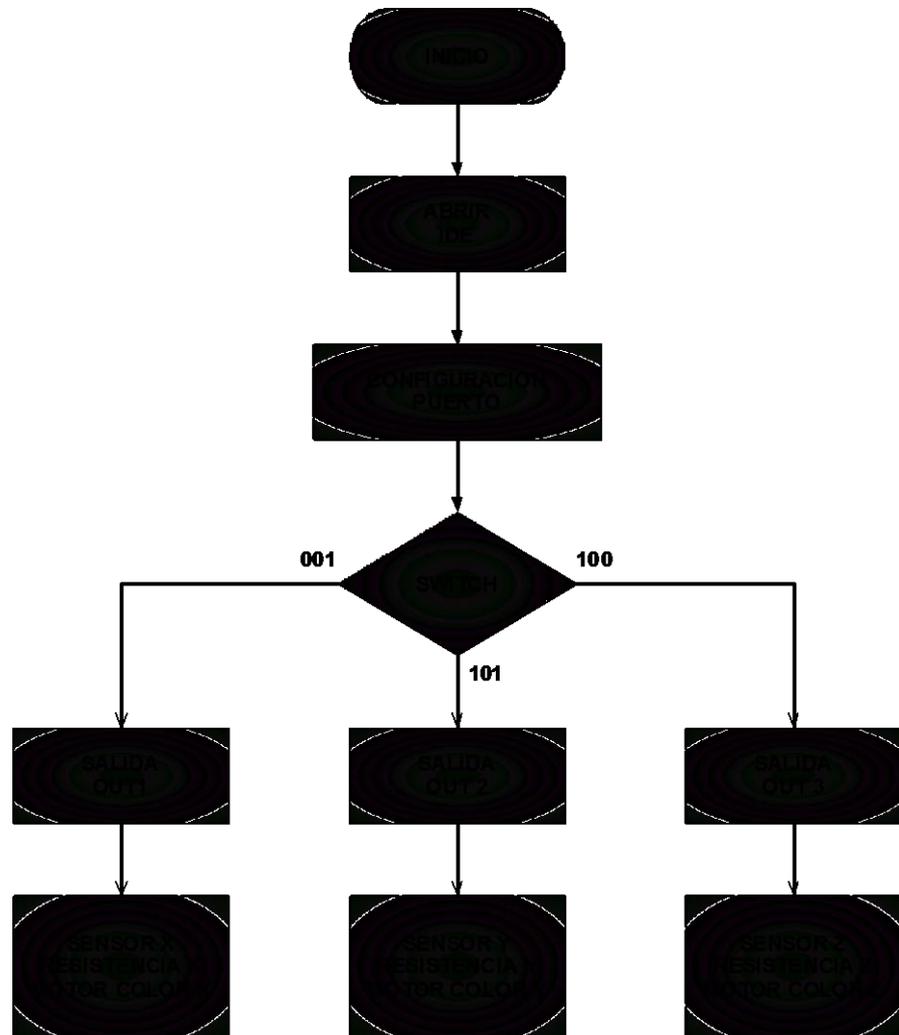


Figura 3.20. Diagrama bloques para la generación de los comandos de control.

3.4.4. Descripción del algoritmo

- Se inicializan las variables.
- Se configuran los puertos referentes a las variables que se están utilizando.
- Se ejecuta el programa
 - Cargamos el archivo a imprimir.
 - Seleccionamos la bobina (color).
 - 001: bobina No. 1 (color Amarillo).
 - 100: bobina No. 2 (color Azul).
 - 101: bobina No.3 (color rojo).
 - Comprobamos el funcionamiento de la impresora.
 - Movimientos de los ejes (x, y, z).
 - Calentamiento del extrusor.
 - Movimiento de la alimentación del extrusor.

CAPÍTULO 4

4. IMPLEMENTACIÓN EJERCICIOS DE PRUEBA Y SIMULACIÓN DEL PROYECTO

En este capítulo se muestran las simulaciones correspondientes de los ejercicios realizados y detallados en el capítulo anterior con el propósito de fortalecer y afianzar los conocimientos en la programación de microcontroladores y encaminarnos a la realización de nuestro proyecto, también se presenta el listado de materiales, diagramas de conexiones e ilustración gráfica de cada uno de los ejercicios.

Los diagramas de conexiones están basados en las hojas de datos esquemáticos del microcontrolador ATmega2560 de la plataforma Arduino y del integrado L298, luego de estudiar e identificar las entradas y salida que cuentan dichas tarjetas, se diseñaron y elaboraron los diagramas de los ejercicios planteados [10].

4.1. Comunicación del Arduino y el Pc

En este ejercicio nuestro objetivo es realizar la comunicación entre la tarjeta Arduino Atme2560 y el Pc, utilizando el entorno de desarrollo (IDE) versión 1.0.4, para lo cual aprovechamos el LED que está colocado en el pin número 13 de la placa [20].

Mediante el uso del cable USB para conectar el Arduino y el Pc, se realiza la transmisión de datos con el propósito de cargar el ejemplo “blink”, este ejemplo está diseñado para enviar una señal al pin 13 y gracias al encendido del led de la tarjeta podemos observar que se realiza la comunicación de manera óptima, este procedimiento básico es esencial es para involucrarnos en el entorno de Arduino y afianzarnos antes de la implementación del proyecto.

4.1.1. Lista de Componentes

- Tarjeta Arduino ATmega2560.
- Cable Adaptador USB.
- Diodo Led integrado en la tarjeta.

4.1.2. Diagrama de conexiones

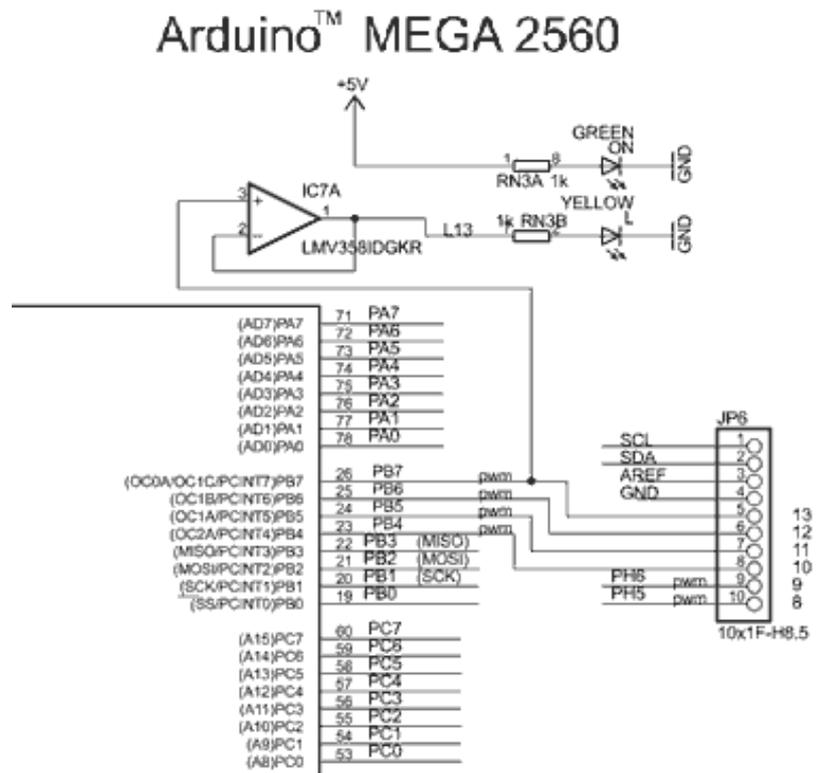


Figura 4.1. Diagrama interno de la tarjeta led pin 13.

4.1.3. Implementación

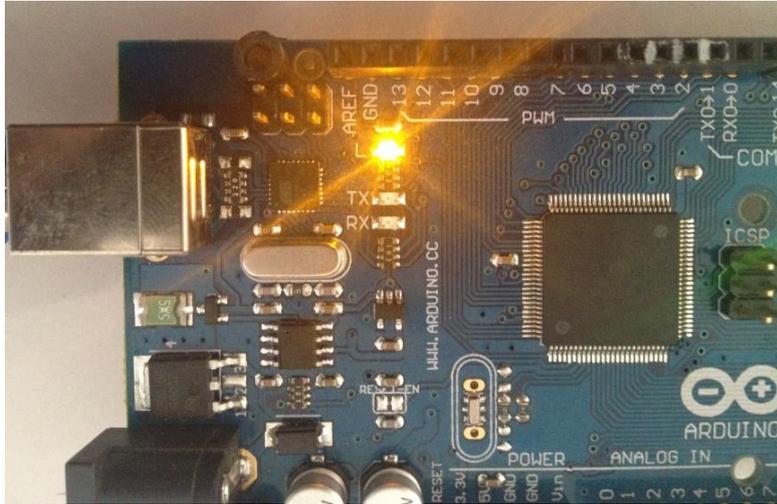


Figura 4.2. Comunicación del Arduino y el Pc.

4.1.4. Observaciones

Al cargar el ejemplo “blink” del entorno de desarrollo (IDE) en nuestra tarjeta Arduino, se pudo observar que el led L del pin 13 de la tarjeta, parpadea a razón de un segundo, comprobándose la correcta comunicación entre la tarjeta Arduino y el pc.

4.2. Secuencia de encendido de led's mediante retardos de tiempo

La implementación de este ejercicio tiene como objetivo familiarizarnos con las entradas/salidas digitales que cuenta el chip Atmega2560 de las cuales emplearemos las salidas digitales e interactuar con las diferentes funciones que esta nos ofrece.

El funcionamiento consiste en el control de encendido de 3 led's de forma secuencial manteniéndose un led encendido a la vez, esta secuencia se la realizo con led's de distintos colores simulando el funcionamiento de un semáforo.

4.2.1. Lista de Componentes

- Tarjeta Arduino ATmega2560
- Cable Adaptador USB.
- 3 Diodos LED's.
- 3 Resistencias de 330 Ω .
- Cable UTP.

4.2.2. Diagrama de conexiones

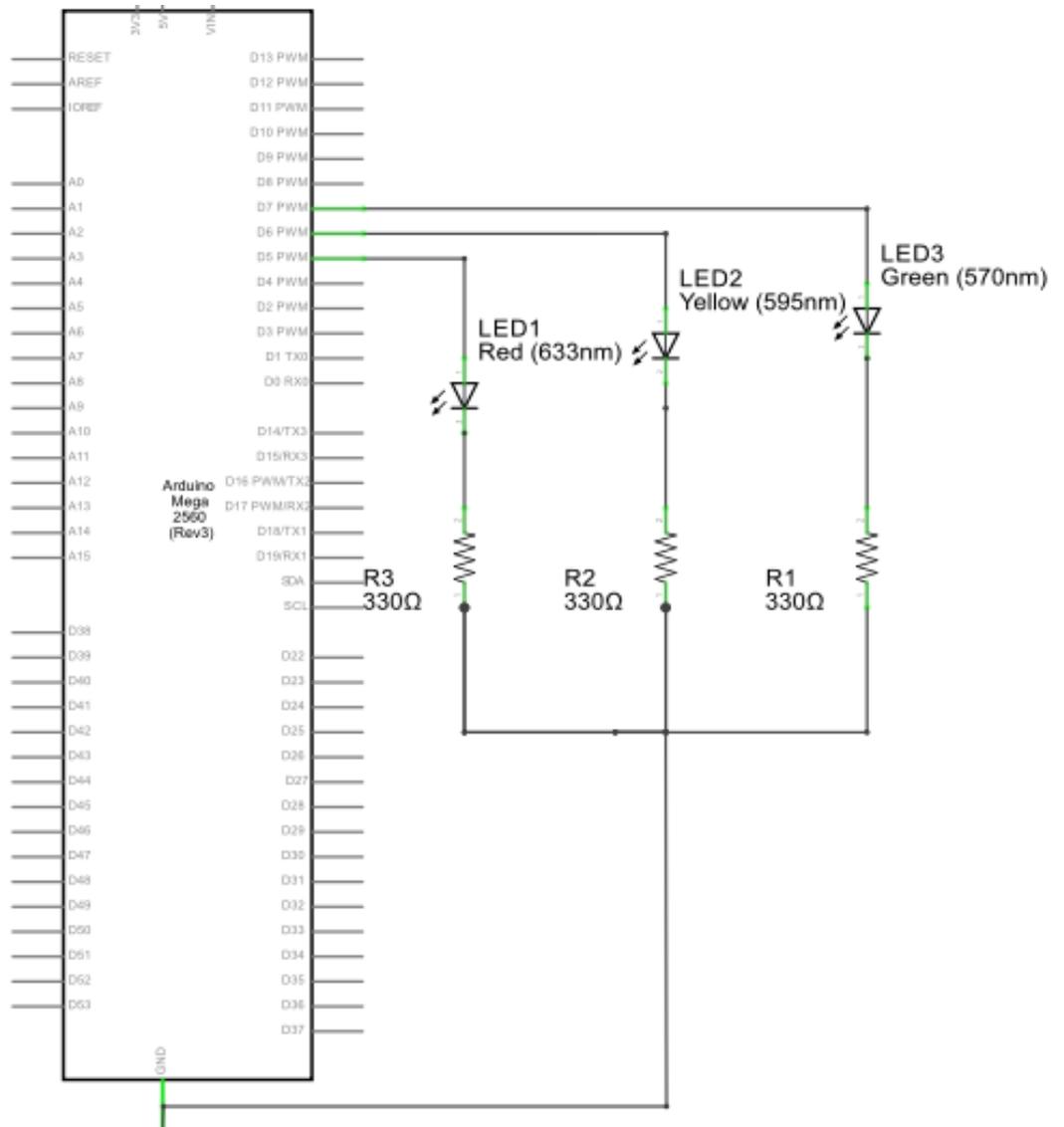


Figura 4.3. Diagrama de conexiones para el control de intensidad de leds usando la tarjeta ATmega2560.

4.2.3. Implementación

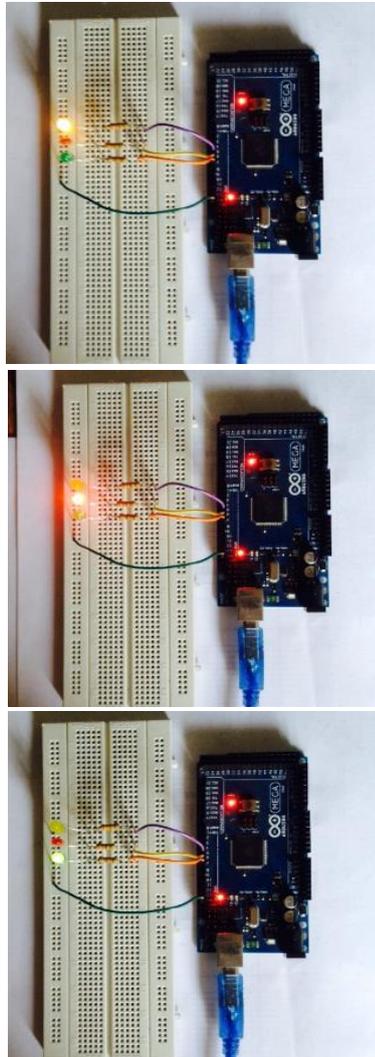


Figura 4.4. Secuencia de encendido de led's.

4.2.4. Observaciones

Al ejecutar el programa cargado en el microcontrolador, se observa en la secuencia de encendidos de los led's simulando

correctamente el funcionamiento de un semáforo convencional como se muestra en la figura 4.4

4.3. Control de Motores de Paso mediante Driver L289N

En este ejercicio el objetivo principal es el control de motores de paso mediante la implementación de un motor de paso bipolar y una interface de potencia, para la respectiva simulación hacemos uso de la tarjeta ATMEGA2560 y la tarjeta de fuerza con integrado L298N encargado de controlar motores de paso [24].

El ejercicio se basa fundamentalmente en la realización de un rutina movimientos simulando el control de giro de un motor y el movimiento de un reloj; el motor empezara su movimiento hacia un sentido a una velocidad determinada velocidad e invertirá el giro a la mitad de su velocidad en periodos de 5 segundos, luego simulara el movimiento de un reloj.

4.3.1. Lista de Componentes

- Tarjeta Arduino ATmega2560
- Cable Adaptador USB.
- Cable UTP.

- Fuente de 12V.
- 1 Motor de paso.
- Tarjeta de interfaz de potencia I298

4.3.2. Diagrama de conexiones

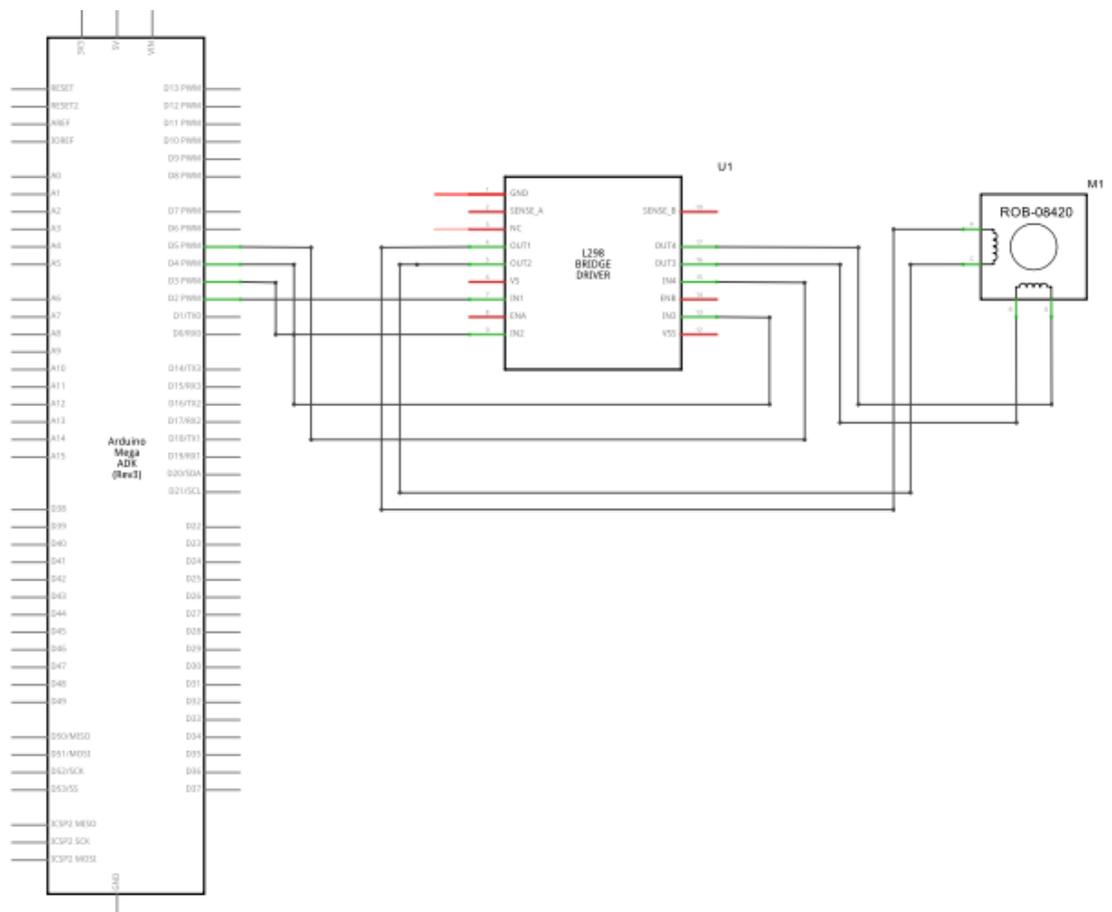


Figura 4.5. Diagrama de conexiones para el control de posición de motores de paso usando la tarjeta ATmega2560.

4.3.3. Implementación

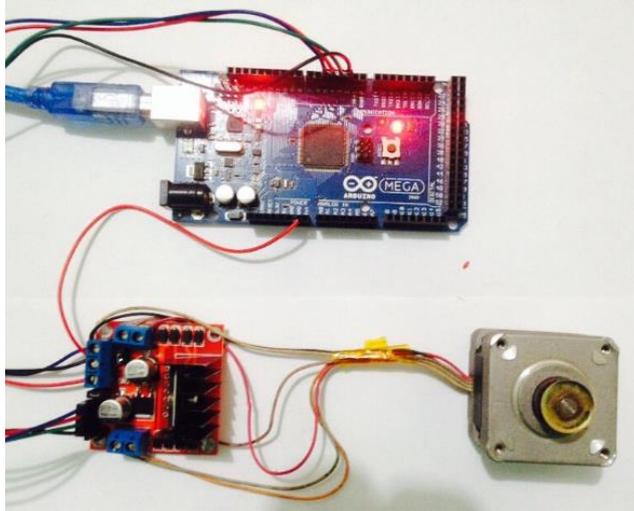


Figura 4.6. Control de motor de paso mediante driver L298N

4.3.4. Observaciones

Este tipo de control se basa en la activación de bobinas para el control de los cambio de giro del motor de paso, motivo por el cual es necesario saber la distribución de los cables a los bobinados establecidos en la hoja de datos de los motores o averiguarlo mediante inspección.

4.4. Selector de bobina de material para la impresión 3D utilizando Arduino y el driver L298N

El objetivo principal para este proyecto de tesina Implementar un prototipo que nos permita seleccionar una bobina de material de

impresión con el fin de seleccionar un color o simplemente reponer una bobina de material de impresión, de manera que el funcionamiento de nuestro sistema no sufra de interrupciones prolongadas, esta aplicación está basada en el microcontrolador ATmega2560 de Arduino y una maqueta crea para la aplicación.

Para el desarrollo del ejercicio se implemento una maqueta de simulación la cual está compuesta por los siguientes elementos: motores de pasos para el movimiento de las bobinas que alimenta el extrusor, un led que simulara el funcionamiento de la resistencia dentro del extrusor, un led que simulará la termocupla para el control de la temperatura del extrusor, un selector para la eleccion de la bobina a utilizar, el driver L298N para el control de los motores y la tarjeta arduino como se muestra en la figura 4.7

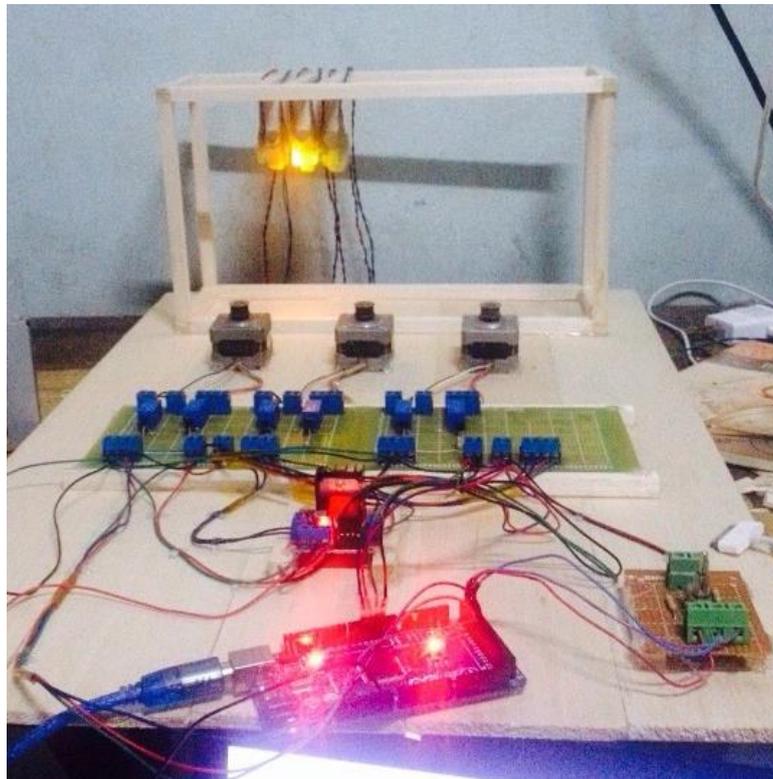


Figura 4.7. Maqueta para la simulación

Para la selección de la bobina se debería tomar en cuenta los siguientes criterios como la cantidad de material en la bobina o simplemente la elección por el color, en las siguientes imágenes se desarrolla el selección de la bobinas

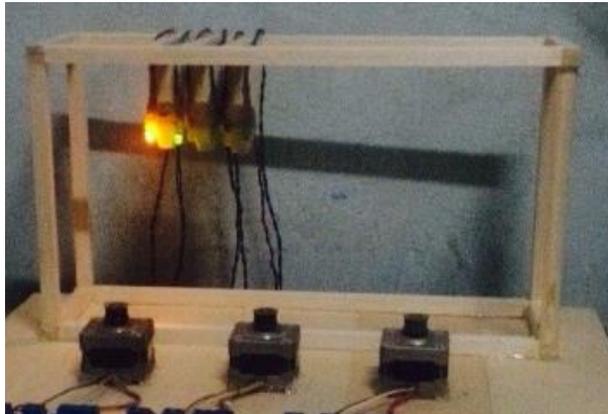


Figura 4.8. Bobina 1 seleccionada

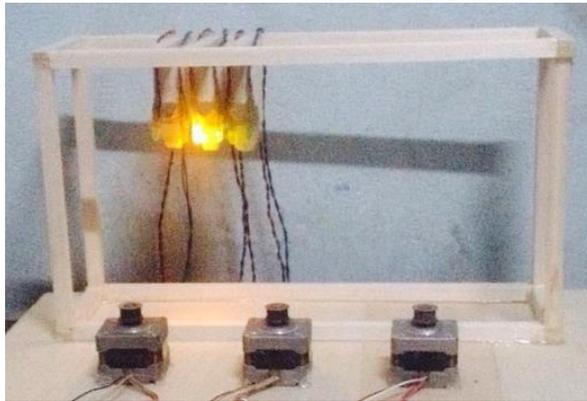


Figura 4.9. Bobina 2 seleccionada

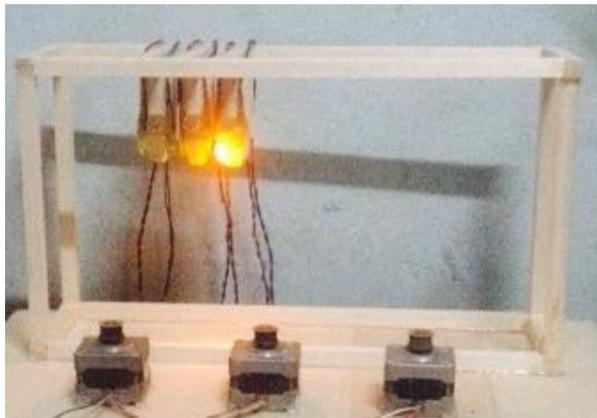


Figura 4.10. Bobina 3 seleccionada

En el proceso de creación del objeto como se muestra en la figura 4.11 el programa crea un código mediante capas. Y también calcula el tiempo que demorará en hacer la impresión, eso dependerá del acabado que se desee por ejemplo, el relleno, detalle de los bordes internos y externos. Con la finalidad de tener una buena resolución del objeto.

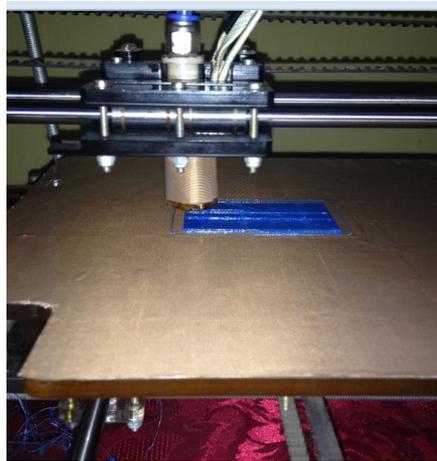


Figura 4.11. Sistema imprimiendo

La elaboración de la figura se llevó alrededor en treinta minutos en culminar su impresión, por los detalles que tiene en los bordes internos y externos, hace que reduzca su velocidad y tome un poco más tiempo. Pero tendremos una impresión de gran calidad como se muestra en la figura 4.12.



Figura 4.12. Impresión terminada "leyenda"

CONCLUSIONES

Basado en lo expuesto anteriormente en esta tesina podemos obtener las siguientes conclusiones:

1. Los consejos y mejoras que hemos propuesto a lo largo de este proyecto han sido fruto de la investigación y experiencia obtenida en el desarrollo de nuestro proyecto, experiencia que hemos obtenido gracias al trabajo de montaje, impresión, mantenimiento y reparación de las impresoras 3D Reprap Prusa Mendel que disponemos en la Universidad.
2. Los microcontroladores ATMega son de una máxima utilidad para las aplicaciones de control debido a que tiene una elevada capacidad de almacenamiento, y también en la transmisión y recepción de datos y su sencilla programación, teniendo una gama de utilidades en diferentes campos de estudios.

3. Con la implementación de los ejercicios y la simulación del proyecto se obtuvo una visión más amplia acerca del funcionamiento de los microcontroladores ATmega2560 y de los motores de paso, así como la forma en que se realiza la comunicación entre la tarjeta y Pronterface.

4. En general, la aplicación de microcontroladores en la impresión 3D se encuentra innovando y cambiando constantemente, y cada vez podemos observar nuevos modelos con mayores capacidades y prestaciones. Podemos concluir que la plataforma Arduino que ha sido objeto de estudio en esta tesina es una herramienta con gran ventaja para desarrollar objetos autónomos e interactivos, capaz de comunicarse con software instalado en un computador como Pronterface, Dada su sencillo aprendizaje y su bajo precio constituye una herramienta ideal para estudiantes, maestros, diseñadores y cualquier interesado en electrónica y robótica.

RECOMENDACIONES

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

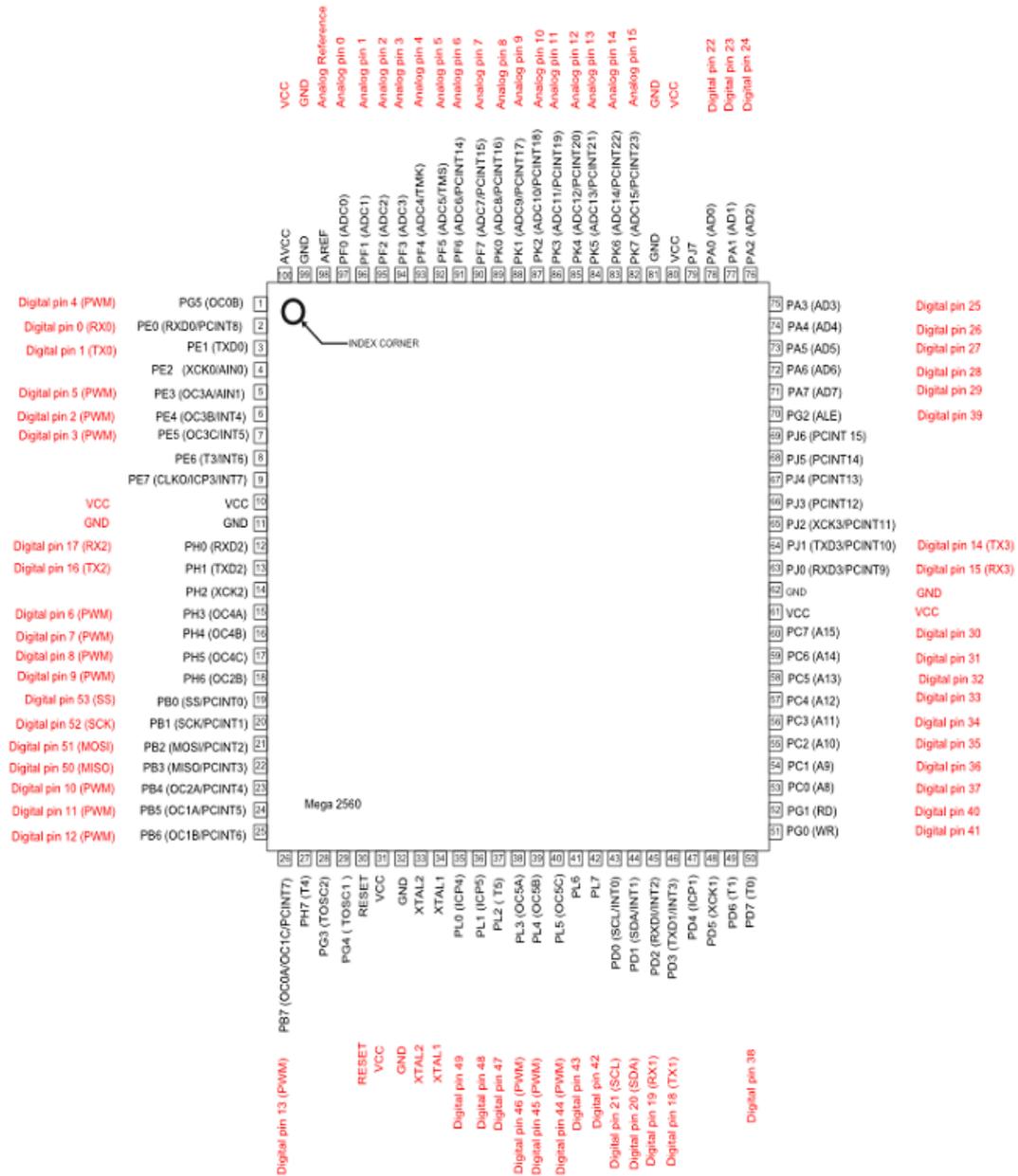
1. Para la conexión y uso de los pines de la tarjeta Arduino es necesario identificar cada uno de ellos y verificar cuales se encuentran reservados por el programa, y cuáles pueden ser usados como I/O y qué pines sirven para funciones específicas, además es necesario verificar que se encuentren correctamente conectado el puerto de la tarjeta para asegurar el correcto funcionamiento del sistema.
2. Se recomienda chequear las tarjetas (Ramps y drivers) antes de su instalación para verificar que se encuentren en un buen estado para su funcionamiento. Después realizar su respectiva calibración para limitar su corriente y frecuencia para los motores.
3. Después de ensamblar la impresora se requiere una calibración en el eje que mantiene al motor x, y también en el plato base de la

impresora, es necesario realizar este procedimiento por las dificultades que se podría obtener al crear un objeto.

4. Se recomienda verificar la temperatura de los motores, en las pruebas los driver llegaron un nivel de temperatura elevada, fue necesario adaptar un ventilador de mayor fuerza y fue necesario chequear los driver cada media hora para ver si requieren un segundo ventilador adicional.

ANEXOS

ANEXO 1.- Arduino Mega 2560 PIN diagram



ANEXO 2.- Arduino Mega 2560 PIN mapping table

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLK0/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)

41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
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
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.- Especificaciones Técnicas L293D



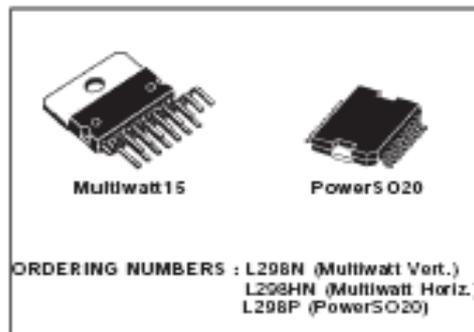
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

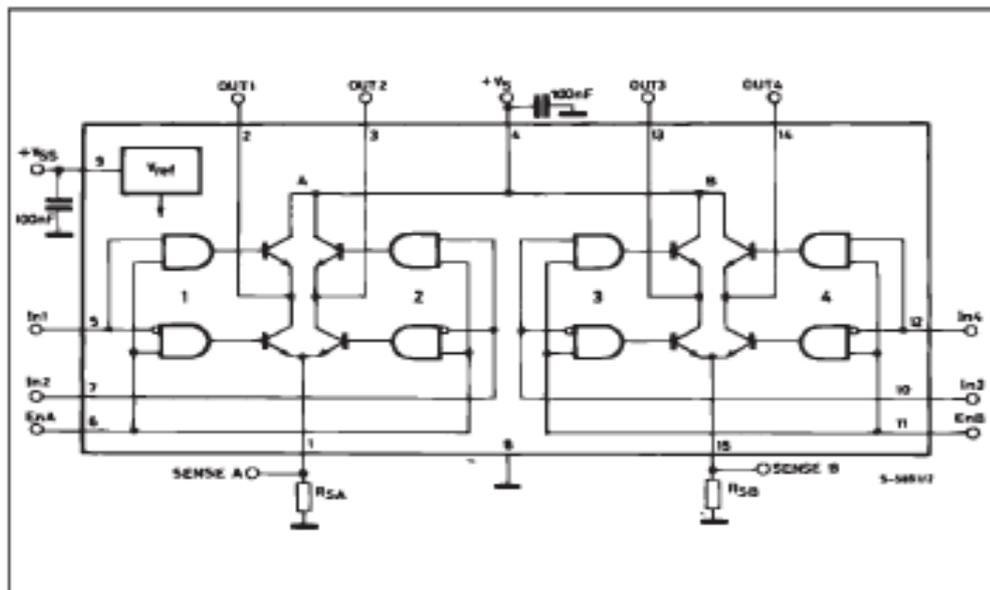
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

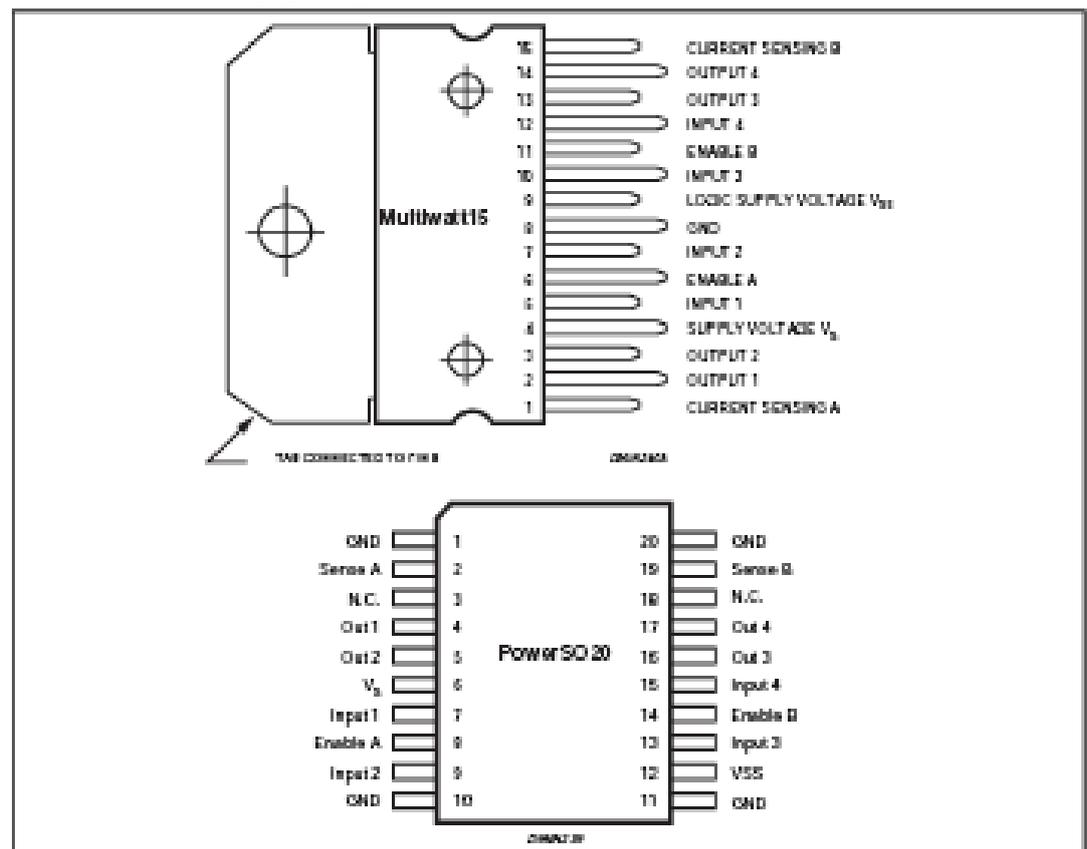


L 298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{DD}	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu s$) - Repetitive (80% on -20% off; $t_{on} = 10ms$) - DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th(j-c)}$	Thermal Resistance Junction-case	Max.	3	$^\circ C/W$
$R_{th(j-a)}$	Thermal Resistance Junction-ambient	Max.	35	$^\circ C/W$

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW. 15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IL} +2.5		48	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0		13	22	mA
		V _I = L		50	70	mA
		V _{en} = L			4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0		24	38	mA
		V _I = L		7	12	mA
		V _{en} = L			6	mA
V _{IL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _I = L			-10	μA
I _{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _I = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CE(sat)} (H)	Source Saturation Voltage	I _L = 1A I _L = 2A	0.85	1.35 2	1.7 2.7	V
V _{CE(sat)} (L)	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V
V _{CEtot}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.0	V
V _{SEN}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

L298

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$T_1 (V)$	Source Current Turn-off Delay	$0.5 V_1$ to $0.9 I_L$ (2); (4)		1.5		μs
$T_2 (V)$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		0.2		μs
$T_3 (V)$	Source Current Turn-on Delay	$0.5 V_1$ to $0.1 I_L$ (2); (4)		2		μs
$T_4 (V)$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.7		μs
$T_5 (V)$	Sink Current Turn-off Delay	$0.5 V_1$ to $0.9 I_L$ (3); (4)		0.7		μs
$T_6 (V)$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.25		μs
$T_7 (V)$	Sink Current Turn-on Delay	$0.5 V_1$ to $0.9 I_L$ (3); (4)		1.6		μs
$T_8 (V)$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.2		μs
$f_c (V)$	Commutation Frequency	$I_L = 2A$		25	40	KHz
$T_1 (V_{ce})$	Source Current Turn-off Delay	$0.5 V_{ce}$ to $0.9 I_L$ (2); (4)		3		μs
$T_2 (V_{ce})$	Source Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (2); (4)		1		μs
$T_3 (V_{ce})$	Source Current Turn-on Delay	$0.5 V_{ce}$ to $0.1 I_L$ (2); (4)		0.3		μs
$T_4 (V_{ce})$	Source Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (2); (4)		0.4		μs
$T_5 (V_{ce})$	Sink Current Turn-off Delay	$0.5 V_{ce}$ to $0.9 I_L$ (3); (4)		2.2		μs
$T_6 (V_{ce})$	Sink Current Fall Time	$0.9 I_L$ to $0.1 I_L$ (3); (4)		0.35		μs
$T_7 (V_{ce})$	Sink Current Turn-on Delay	$0.5 V_{ce}$ to $0.9 I_L$ (3); (4)		0.25		μs
$T_8 (V_{ce})$	Sink Current Rise Time	$0.1 I_L$ to $0.9 I_L$ (3); (4)		0.1		μs

1) Sensing voltage can be $-1V$ for $t \leq 50 \mu s$; in steady state $V_{sens, min} \geq -0.5V$.

2) See Fig. 2.

3) See Fig. 4.

4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

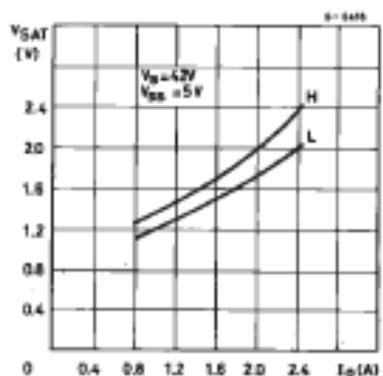
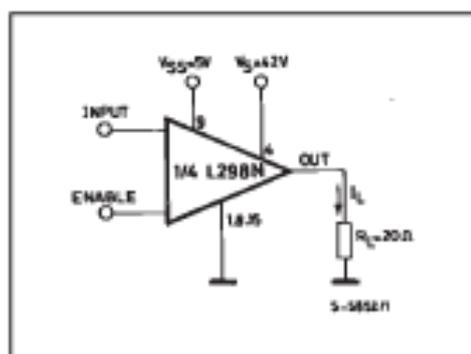


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching set EN - H
For ENABLE Switching set IN - H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

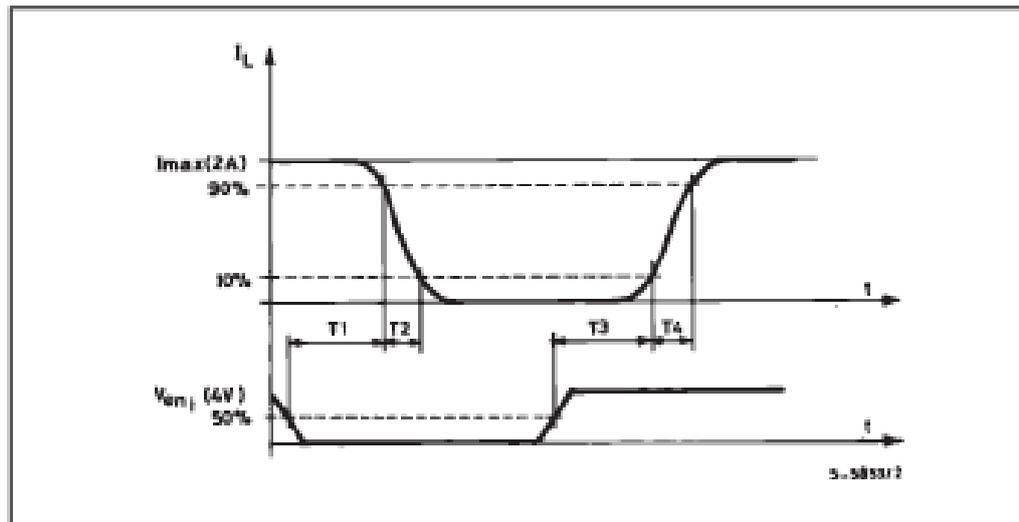
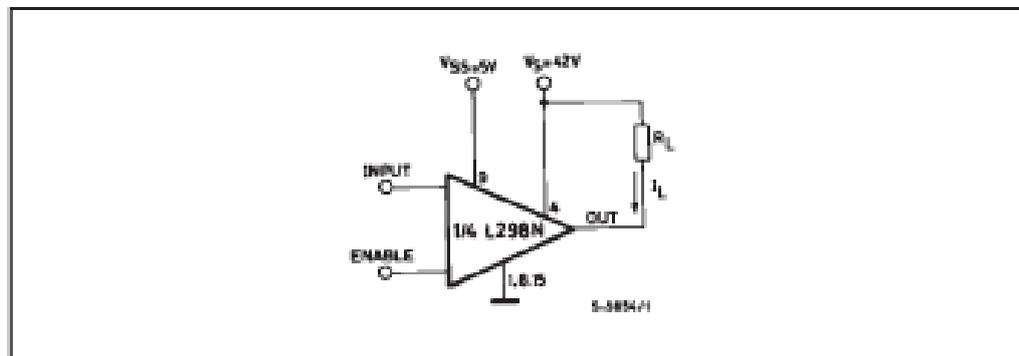


Figure 4 : Switching Times Test Circuits.



Note: For INPUT Switching use EN - H
 For ENABLE Switching use IN - L

L298

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

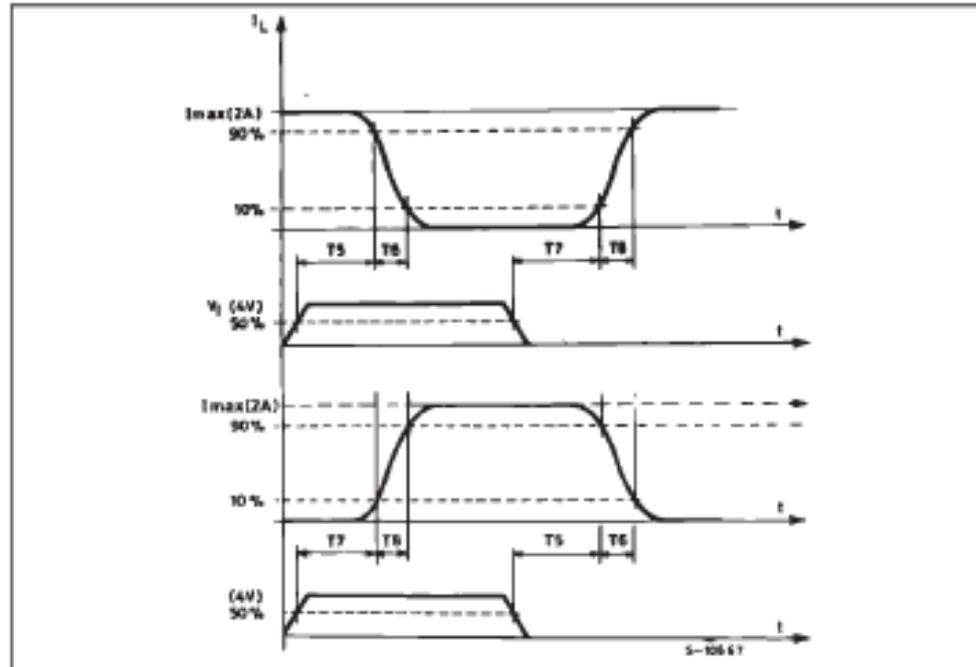


Figure 6 : Bidirectional DC Motor Control.

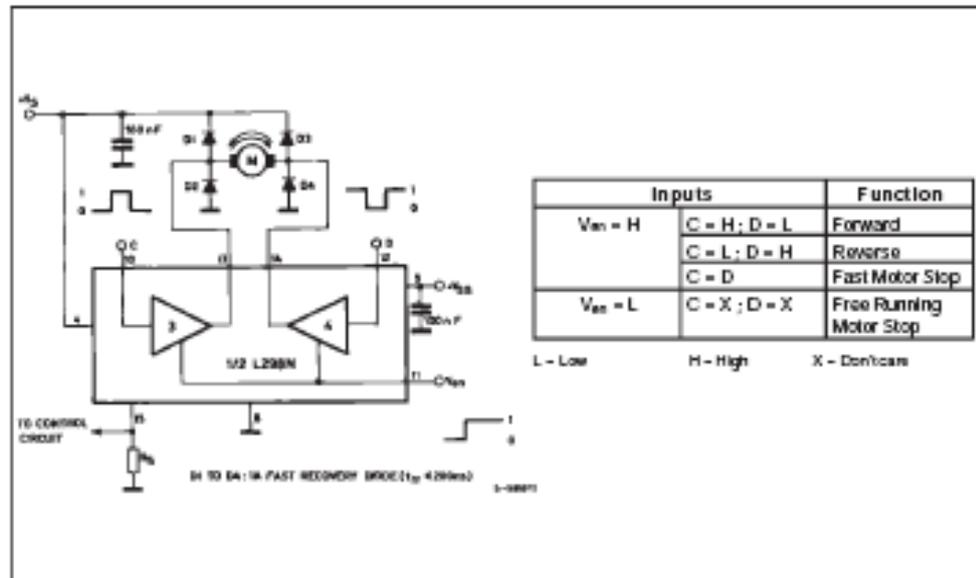
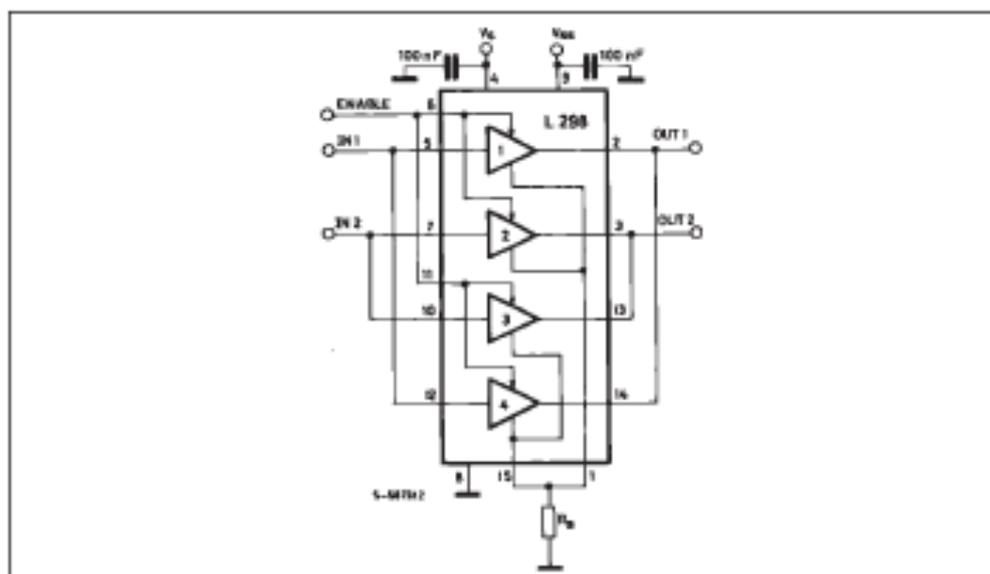


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are $In1$; $In2$; EnA and $In3$; $In4$; EnB . The In inputs set the bridge state when The En input is high; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off: Before to Turn-ON the Supply Voltage and before to Turn OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig. 7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Shock diodes would be preferred.

L298

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

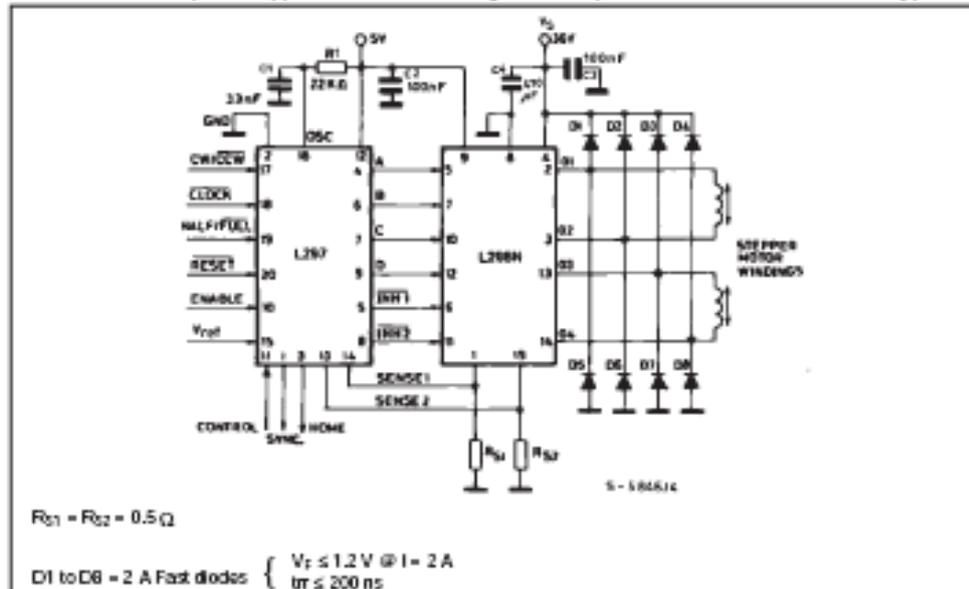
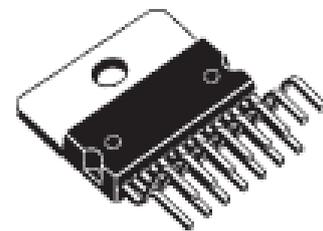
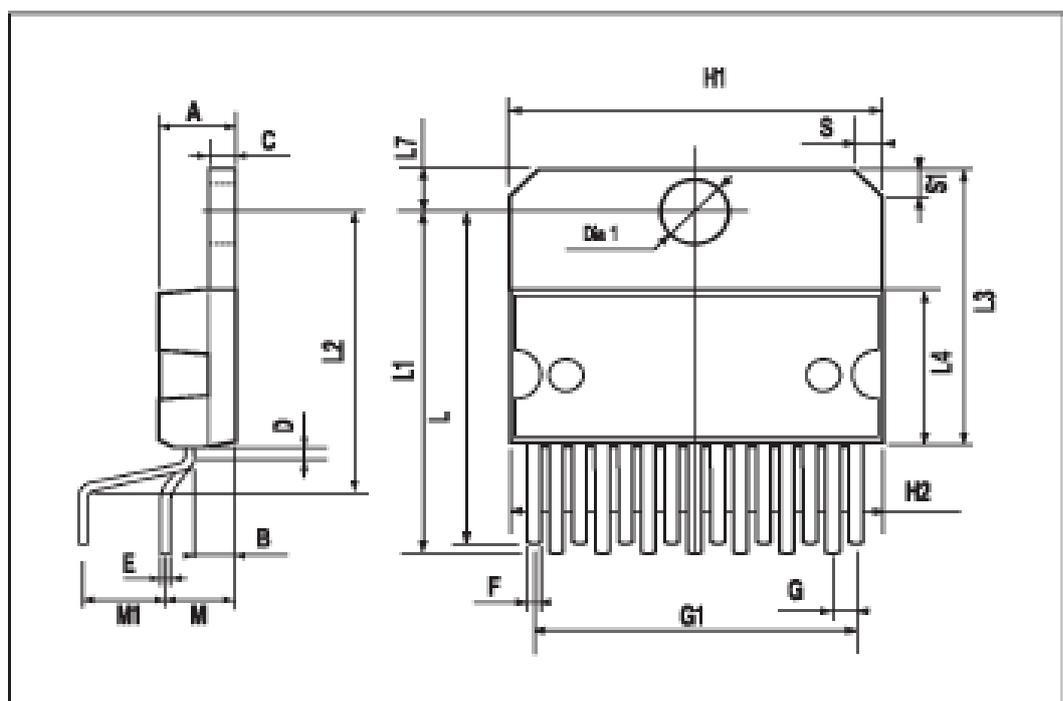


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

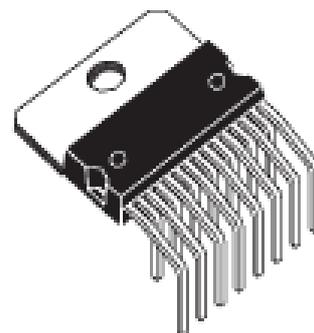
L 298

DIM.	mm			Inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.65		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.796
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Da1	3.65		3.85	0.144		0.152

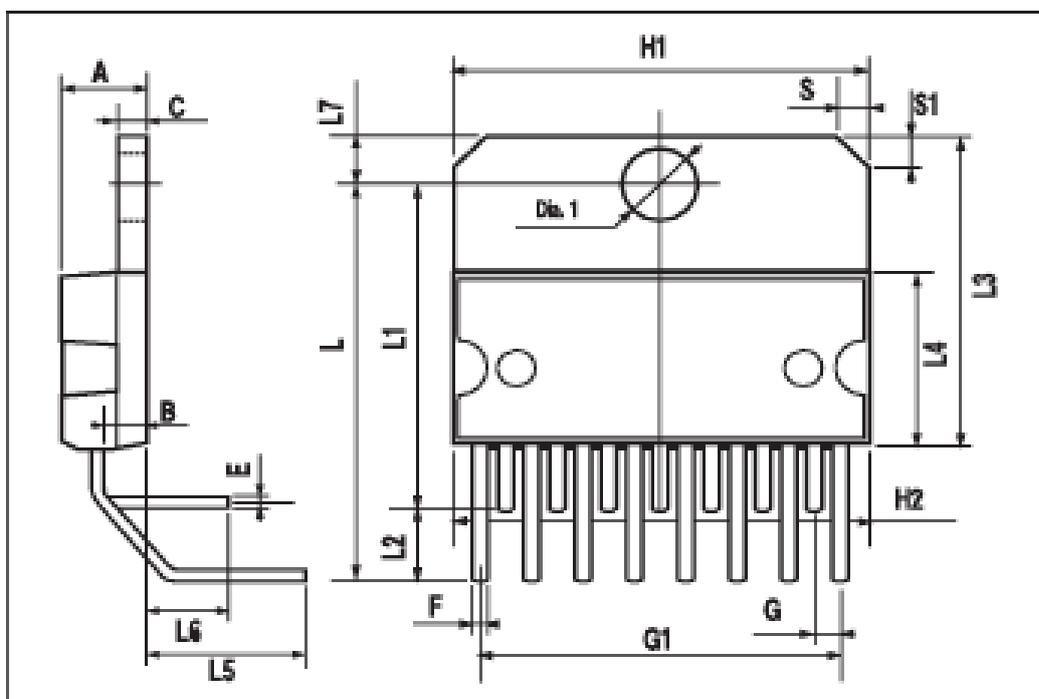
**OUTLINE AND
MECHANICAL DATA**

Multiwatt15 V


DIM.	mm			Inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.95	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 H



L 298

DIM.	mm			Inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

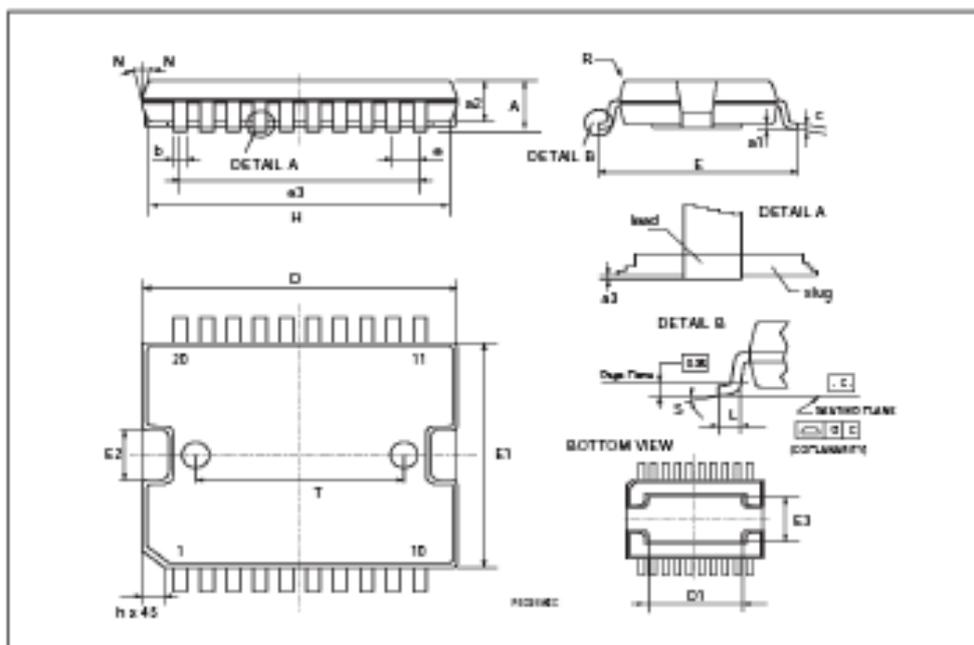
(1) "D" and "F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

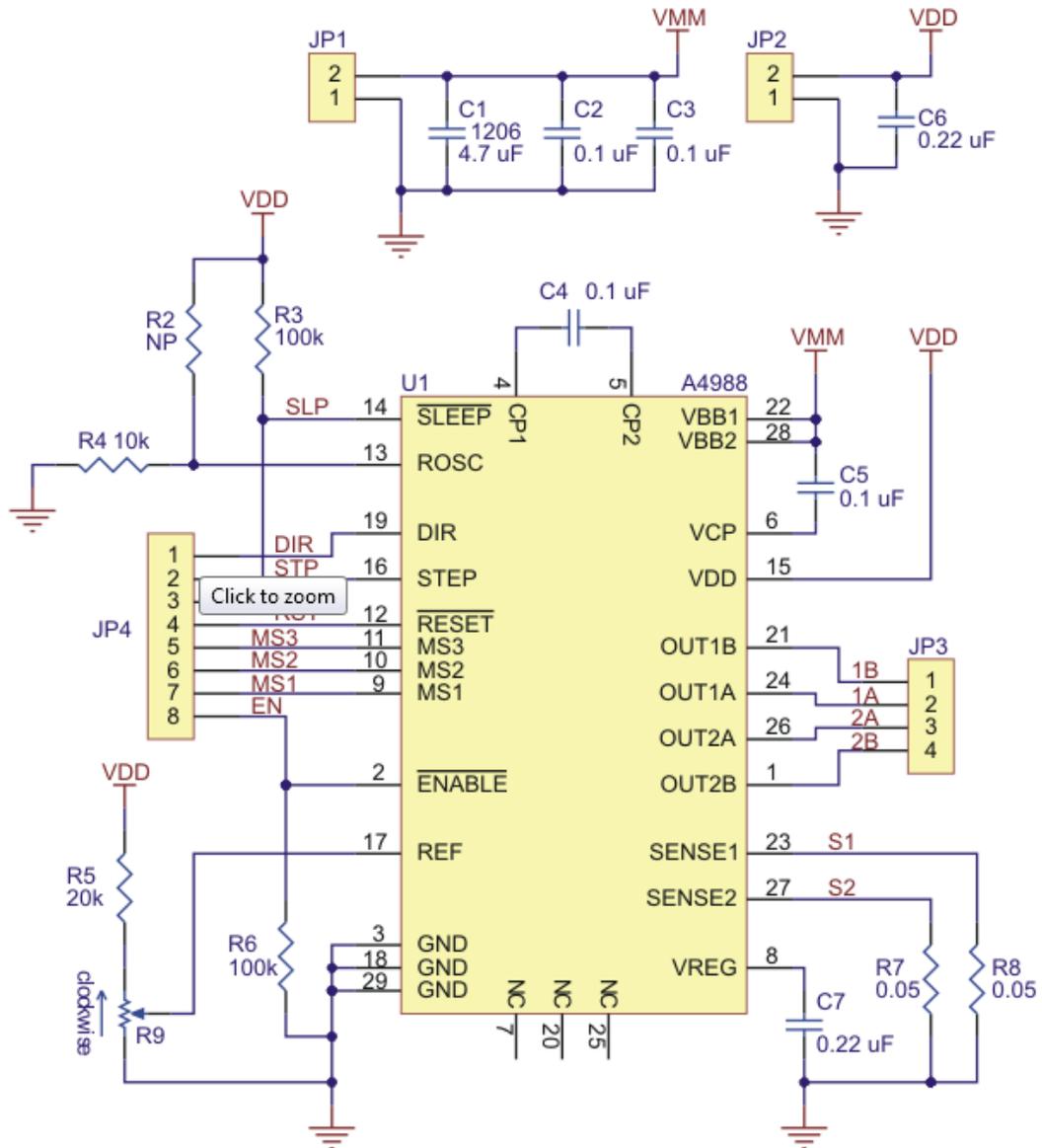
OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20





Schematic diagram of the md09b A4988 stepper motor driver carrier.

BIBLIOGRAFÍA

- [1] M. McRoberts, *Beginning Arduino*, Primera Edición, 2010.
- [2] T. D. Gauray Tyagi, «NIC-Muzaffamagar, UP, 3D Printing Technology,» [En línea]. Available: , <http://123seminarsonly.com/Seminar-Reports/2013-02/83204431-3D-Printing-Technology.pdf>. [Último acceso: marzo 2013].
- [3] D. Girardo B. y I. Tabares G., *Teoría de Control*, Primera Edición, 1997.
- [4] «Revista Letreros N°92 - Impresoras 3D, tecnología de avanzada al alcance de la industria nacional,» [En línea]. Available: <http://www.revistaletreros.com/pdf/92-052a056.pdf>. [Último acceso: abril 2013].
- [5] M. P. Groover, *Fundamentos de Manufactura Moderna*, McGraw Hill - Primera Edición, 2007.
- [6] R. V. 29, *Impresora de tipo RepRap*, Cambridge University Press 2011, 2013.
- [7] J. Floyd Kelly y P. Hood-Daniel, *Printing in Plastic: Build Your Own 3D Printer*, 2011.
- [8] J. C. Rico, F. Ros y A. Serna Ruiz, *Guía práctica de sensores*, Primera Edición, 2010.
- [9] J.-D. Warren, J. Adams y H. Molle, *Arduino Robotics*, Primera Edición, 2011.
- [10] Hernández Bello y Ochoa Luna, «Sección de motores de paso,» de *Motor de Paso*, Primera Edición, 2004.
- [11] «Impresoras 3D,» [En línea]. Available: <http://www.impresoras3d.com/la-cabeza-de-impresion-tf3d>. [Último acceso: abril 2013].

- [12] Allegro, «Data sheet description,» Microsystem, Inc, [En línea]. Available: <http://www.electronicaembajadores.com/datos/pdf1/sm/smci/a4988.pdf>. [Último acceso: 26 marzo 2013].
- [13] [En línea]. Available: rerap.org, <http://rerap.org/wiki/RAMPS>. [Último acceso: 27 marzo 2013].
- [14] J. Pomares, Manual de programación de Arduino, Universidad de Alicante, 2013.
- [15] S. Monk, Programming arduino, getting started with sketches, Mcgraw hill, 2012.
- [16] T. Olsson, Arduino Wearables, Primera Edición, 2009.
- [17] J. Purdum, Beginning C for Arduino: Learn C Programming for the Arduino, Primera Edición, 2012.
- [18] D. Wheat, Arduino Internals, Primera Edición, 2011.
- [19] B. Evans, Practical 3D Printer, Primera Edición, 2011.
- [20] G. Borenstein, Making Things See: 3D Vision with Kinect, Processing, Arduino, and MakerBot, Primera Edición, 2012.
- [21] E. Ramos Melgar y C. Castro Diez, Arduino and Kinect Projects: Design, Build, Blow Their Minds, Primera Edición, 2012.
- [22] T. Olsson, D. Gaetano, S. Wiklund y J. Odhner, Open Softwear, Segunda Edición, 2011.
- [23] S. Mateo, «Impresoras 3D, la nueva revolución tecnológica,» [En línea]. Available: <http://sergimateo.com/impresoras-3d-la-nueva-revolucion-tecnologica/>. [Último acceso: marzo 2013].
- [24] Hernandez Bello y Ochoa Luna, Motor de Paso, Primera Edición, 2004.
- [25] S. F. Barrett, Arduino Microcontroller: Processing for Everyone, Primera

Edición, 2012.

[26] M. Böhmer, Beginning Android ADK with Arduino, Primera Edición, 2012.