

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



## **FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**“DISEÑO DE INTERFAZ USANDO MINICOMPUTADOR RASPBERRY PI A  
SER USADA EN LA IMPRESIÓN EN 3 DIMENSIONES, CON  
DEMOSTRACIÓN PRÁCTICA DE FUNCIONAMIENTO EN PROTOTIPO DE  
IMPRESORA REPRAP”**

### **TESINA DE SEMINARIO**

Previa la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

Jorge Enrique Ordoñez García.

Jazmín Nataly Fiallos Barragán.

GUAYAQUIL – ECUADOR

AÑO 2014

## AGRADECIMIENTO

En primer lugar a Dios, ya que él me ha ayudado siempre, me ha dado fuerzas para seguir adelante con perseverancia, para lograr mis metas. A mi familia quienes siempre han estado presente. A los profesores, debido que gracias a ellos, he adquirido muchos conocimientos en el largo de mi carrera profesional. Especialmente al Ing. Carlos Valdivieso quien nos guiado en el transcurso de nuestro proyecto.

*Jorge Enrique Ordoñez García.*

Agradecimiento a Dios, sobre todo, a mis familiares, profesores, amigos ya que ellos siempre han estado presente en todo momento durante el transcurso de mi carrera dándome

fuerzas para seguir adelante. También al Ing. Valdivieso quien nos ayudado en nuestro proyecto.

*Jazmín Nataly Fiallos Barragán.*

## DEDICATORIA

A mi madre Irene Maura García Quimis y mi padre Jorge Ordoñez quienes se han sacrificado para que no me falte nada en la vida, ellos siempre con preocupación, me han guiado por el camino del bien, llegando a ser una persona fuerte y con valores.

A mi hermana y sobrinos quienes también siempre me han apoyado en todo momento.

*Jorge Enrique Ordoñez García*

Le dedico este trabajo a toda mi familia, a mis padres Luis Fiallos Moreno y Margarita Barragán Piza que aunque ya no se encuentren con vida, siempre me dieron todo su apoyo

incondicional mientras los tuve conmigo, a mi abuelita Luz Amada Piza Coello quien me ayudo en todos los sentidos, siempre creyó en mi y siempre me amó y trato de acompañarme en el camino de la vida lo más que pudo, a estos 3 seres les dedico esta meta cumplida y yo sé que hubieran querido estar a mi lado en este momento.

También a mis 3 hermanos que están junto a mí, y a toda mi familia por acompañarme.

*Jazmín Nataly Fiallos Barragán*

**TRIBUNAL DE SUSTENTACIÓN**

---

Ing. Carlos Valdivieso A.

**Profesor del Seminario de Graduación**

---

Ing. Hugo Villavicencio V.

**Profesor Delegado por la Unidad Académica**

## DECLARACIÓN EXPRESA

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

(Reglamento de exámenes y títulos profesionales de la ESPOL)

---

Jorge Enrique Ordoñez García

---

Jazmín Nataly Fiallos Barragán

## RESUMEN

El presente proyecto tiene como fin el diseño y construcción de una interfaz entre un minicomputador Raspberry Pi y una impresora RepRap, la cual es usada en la impresión 3d.

La implementación se realizará, mediante un circuito que permita interconectar un minicomputador mediante los puertos GPIO y la Tarjeta Arduino, también puede ser mediante conexión serial pero igual hay que hacer una configuración mediante ciertos comandos en las Raspberry para comunicar al Arduino, desde ahí mismo hacia a la impresora RepRap.

La interfaz, en si está controlando la interconexión de una placa Raspberry Pi y una placa Arduino Mega, ya que estas placas tienen diferentes características, el objetivo será poder lograr comunicación entre ellas, además de ensamblar la impresora y de instalar todos los programas requeridos para que exista comunicación entre ambas de manera exitosa, ya que la instalación y configuración de la Raspberry no es fácil debido a que trabaja mediante sistema operativo Linux toca investigar los comandos necesarios para lograr nuestro propósito.



## ÍNDICE GENERAL

AGRADECIMIENTO .....	II
DEDICATORIA .....	IV
ÍNDICE GENERAL.....	IX
ÍNDICE DE TABLAS .....	XVII
INTRODUCCIÓN .....	XVIII
CAPITULO 1.....	1
1.1 ANTECEDENTES.....	1
1.2 MOTIVACIÓN.....	6
1.3 DESCRIPCIÓN DEL PROYECTO.....	7
1.4 OBJETIVOS PRINCIPALES .....	9
1.5LIMITACIONES.....	10
CAPÍTULO 2.....	12
2. FUNDAMENTO TEÓRICO .....	12
2.1 HERRAMIENTAS DE SOFTWARE PARA EL DESARROLLO DEL PROYECTO. ....	13

2.1.1 ¿QUÉ ES ARDUINO?.....	13
2.1.1.1 ¿PORQUE EL USO DE ARDUINO?.....	14
2.1.1.2 VENTAJAS DEL SISTEMA ARDUINO SOBRE OTROS.....	15
2.1.2 SOTFWARE CONTROLADRES.....	16
2.1.2.1 FIRMWARE SPRINTER.....	16
2.1.2.2 PRONTERFACE.....	20
2.1.2.3 SLICER.....	21
2.2 HERRAMIENTAS DE HARDWARE PARA LA IMPLEMENTACIÓN DEL PROYECTO. ....	22
2.2.1 PLACAS E/S .....	22
2.3 ARDUINO MEGA.....	23
2.3.1 VISIÓN GENERAL DEL ARDUINO MEGA .....	23
2.3.2 RESUMEN DE CARACTERÍSTICAS Y DESCRIPCIONES .....	23
2.3.3 ALIMENTACIÓN.....	24
2.3.4 MEMORIA .....	26
2.3.5 ENTRADAS Y SALIDAS .....	26
2.3.6 COMUNICACIÓN.....	28
2.3.7 PROGRAMACIÓN.....	29
2.3.8 INSTALAR EL SOFTWARE ARDUINO.....	29

2.4 QUE ES RASPBERRY PI?.....	31
2.5 OTRAS COMPONENTES ELECTRÓNICAS.....	34
2.5.1. MOTORES PASO A PASO.....	34
2.5.2. MOSFET .....	36
2.5.3 DRIVER CONTROLADOR DE MOTORES.....	37
CAPÍTULO 3.....	40
3 DISEÑO E IMPLEMENTACIÓN DEL PROYECTO.....	40
3.1. INTRODUCCIÓN.....	40
3.1.1. INTERFAZ DE CONEXIÓN ENTRE LA RASPBERRY PI AL ARDUINO VÍA SERIAL .....	41
3.1.2. DIAGRAMA DE BLOQUES.....	44
3.1.3. DIAGRAMA DE FLUJO.....	45
3.1.4. DESCRIPCIÓN DEL ALGORITMO .....	46
3.1.5. CÓDIGO FUENTE.....	46
3.2. EJERCICIO 2: .....	47
3.2.1.INTRODUCCIÓN.....	47
3.2.2. DIAGRAMA DE BLOQUE .....	48
3.2.3. DIAGRAMA DE FLUJO .....	49
3.2.4. DESCRIPCIÓN DEL ALGORITMO .....	50
3.2.5. CÓDIGO FUENTE.....	50

3.3. EJERCICIO 3: .....	52
3.3.1. INTRODUCCIÓN.....	52
3.3.2. DIAGRAMA DE BLOQUES .....	53
3.3.3. DIAGRAMA DE FLUJO .....	54
3.3.4. DESCRIPCIÓN DEL ALGORITMO .....	55
3.3.5. CÓDIGO FUENTE.....	56
3.4. EJERCICIO 4: .....	57
3.4.1. INTRODUCCIÓN .....	57
3.4.2. DIAGRAMA DE BLOQUES .....	58
3.4.3. DIAGRAMA DE FLUJO .....	59
3.4.4. DESCRIPCIÓN DEL ALGORITMO .....	60
3.4.5. CÓDIGO FUENTE.....	60
CAPÍTULO 4.....	65
4. PRUEBAS Y SIMULACIÓN DEL PROYECTO .....	65
4.1. INTRODUCCIÓN .....	65
4.2. ELEMENTOS FÍSICOS .....	65
4.2.1. PLATAFORMA DEL PROYECTO .....	65
4.2.1.1. PROTOBOARD.....	66
4.2.1.2. RASPBERRY PI.....	68
4.2.1.3. ARDUINO.....	69

4.2.1.4. KIT DE DRIVER DE MOTORES (POLOLUS) Y RAMPS 1.4.71	
4.3.IMÁGENES Y SIMULACIONES .....	72
4.3.1. INTERFAZ DE COMUNICACIÓN DE ARDUINO CON RASPBERRY PI.....	72
4.3.1.1 CONCLUSIONES.....	74
4.3.2. EJERCICIO 2: INTERFAZ DE ARDUINO CON RASPBERRY PI PARA UN EFECTO INTERACTIVO LED .....	75
4.3.2.1 CONCLUSIONES.....	76
4.3.3. EJERCICIO 3: BÁSICO CONTROL DE MOTORES PASO A PASO CON LOS DRIVER L293D .....	77
4.3.3.1 CONCLUSIONES.....	79
4.3.4. PROYECTO DE GRADUACIÓN: .....	79
CONCLUSIONES.....	82
RECOMENDACIONES.....	83
BIBLIOGRAFÍA.....	85

## ÍNDICE DE FIGURAS

FIGURA 1.1 MODELO DE UN CRÁNEO HECHO EN IMPRESORA 3D.....	3
FIGURA 1.2 IMPRESORA 3D PRUSA MENDEL.....	5
FIGURA 1.3 RASPBERRY PI, PLACA ARDUINO, IMPRESORA 3D.....	8
FIGURA 2.1 PLACA ARDUINO MEGA.....	16
FIGURA 2.2 RASPBERRY PI .....	25
FIGURA 2.3 MOTOR PASO A PASO .....	29
FIGURA 2.4 MOSFET CANAL N .....	31
FIGURA 2.5 POLOLU .....	31
FIGURA 2.6 CONEXIÓN DE LA POLOLUS EN LA RAMPS .....	32
FIGURA 3.1 COMPILACIÓN DELPROGRAMA ARDUINO.....	37
FIGURA 3.2 DIAGRAMA DE BLOQUES DEL EJERCICIO 1 .....	38
FIGURA 3.3 DIAGRAMA DE FLUJO DEL EJERCICIO 1 .....	39
FIGURA 3.4 CONEXIÓN DE INTERFAZ DE EFECTO INTEREACTIVO EDS .....	42
FIGURA 3.5 DIAGRAMA DE BLOQUES EJERCICIO2 .....	42
FIGURA 3.6 DIAGRAMA DE FLUJO EJERCICIO 2 .....	43
FIGURA 3.7 CONEXIÓN DE MOTORES CON EL DRIVER L293 .....	47
FIGURA 3.8 DIAGRAMA DE BLOQUES EJERCICIO 3 .....	48
FIGURA 3.9 DIAGRAMA DE FLUJO DEL EJERCICIO 3 .....	49

FIGURA 3.10 CONEXIÓN DE LAS TARJETAS RAMPS 1.4 Y POLOLU .....	52
FIGURA 3.11 PANTALLA DE COMPILACIÓN DEL SPRINTER .....	53
FIGURA 3.12 DIAGRAMA DE BLOQUES DEL EJERCICIO 4 .....	53
FIGURA 3.13 DIAGRAMA DE FLUJO DEL EJERCICIO 4 .....	54
FIGURA 4.1 PROTOBOARD .....	62
FIGURA 4.2 PROTOBOARD CON EL PROYECTO FINAL .....	62
FIGURA 4.3 RASPBERRY PI IMPLEMENTADO EJERCICIOS Y PROYECTO.....	64
FIGURA 4.4 ARDUINO MEGA IMPLEMENTADO EN PROTOBOARD .....	65
FIGURA 4.5 CONEXIÓN DE DRIVERS DE MOTORES.....	66
FIGURA 4.6 POLOLUS + ARDUINO + RAMPS.....	67
FIGURA 4.7 CONEXIONES DE LA TARJETA ARDUINO CON RASPBERRY.....	68
FIGURA 4.8 CICUITO PARA IGUALACIÓN DE VOLTAJES.....	68
FIGURA 4.9 PANTALLA QUE MUESTRA LA INTERCONEXIÓN.....	69
FIGURA 4.10 EFECTO INTERACTIVO LEDS.....	71
FIGURA 4.11 PINES DEL DRIVER L293.....	73
FIGURA 4.12 ESQUEMÁTICO DEL CIRCUITO DE CONTROL DE MOTORES	73
FIGURA 4.13 CIRCUITO PROTOTIPO DE CONTROL DE MOTORES.....	74
FIGURA 4.14 CONTROL DE LOS MOTORES CON LA TARJETA RAMPS .....	75
FIGURA 4.15 EJECUCIÓN DEL PROGRAMA PRONTERFACE.....	76

FIGURA 4.16 INTERFAZ FINAL DEL PROYECTO..... 76



## ÍNDICE DE TABLAS

TABLA 1.1 CARACTERÍSTICAS TÉCNICAS DE ARDUINO MEGA.....	17
---	----

## INTRODUCCIÓN

El objetivo principal es diseñar una interfaz, que permita realizar una interconexión entre un minicomputador Raspberry Pi y una impresora 3D, para lo cual se usara un circuito físico para acoplar los voltajes, además diversos programas que instalamos en el minicomputador los cuales permitirán la comunicación de estas 2 tarjetas y cables, para lo cual se explicará el proceso de diseño en los siguientes cuatro capítulos.

En el Capítulo I, se detalla una descripción general del proyecto y elementos que se usaron.

En el Capítulo II, fundamento teórico de herramientas de hardware y software que fueron de gran utilidad para la realización del proyecto.

En el Capítulo III, la descripción e implementación del proyecto, se especifica claramente el diseño del proyecto en su totalidad, variables, funciones y demás códigos que se usaron en la programación y al final las limitaciones que se produjeron en el equipo.

En el Capítulo IV, se amplía información acerca de la simulación y pruebas experimentales, conexiones de hardware y además el diseño final de la PCB

Y finalmente, conclusiones sobre lo aprendido y recomendaciones para futuros proyectos y bibliografía.

# **CAPÍTULO 1**

## **1 Descripción general del proyecto**

### **1.1 Antecedentes**

El presente trabajo describe la implementación de una interfaz usando un minicomputador Raspberry Pi, el cual trabaja con el sistema operativo Linux que es un software de libre distribución, la interfaz que estamos realizando será usada en la impresión en 3 dimensiones, en este caso usaremos la impresora RepRap para la demostración de nuestra interfaz.

La implementación de esta aplicación, está basada en el minicomputador Raspberry Pi, el cual trabaja con un sistema operativo llamado Raspbian, el cual no es el único sistema operativo pero si el mas recomendado, el objetivo

principal de este proyecto es poder controlar una impresora mediante una interfaz creada en el sistema operativo Raspbian, la cual sea capaz de controlar a una impresora 3D a través de una interfaz que logre comunicar el minicomputador Raspberry Pi con la placa Arduino Mega, el cual es la base de la impresora Reprap Prusa.

Con el deseo de realizar una buena exportación e importación de datos nos hemos basado en el Raspberry Pi y en la placa Arduino, los cuales han sido interconectados mediante una interfaz, para que así nuestro minicomputador envíe los diseños que han sido generados mediante una interfaz gráfica y estos puedan convertirse de un objeto que vive en la mente del usuario a un diseño real y contundente.

### **Antecedentes de la impresora Reprap 3D**

Las impresoras 3D nacen del deseo de convertir diseños que solo existen en la mente de los usuarios en objetos reales, los cuales son diseñados con programas de tipo CAD, con fines para crear prototipos en el área de la arquitectura, para crear piezas y modelos de maquetas arquitectónicas, su aplicación no es solo en estos campos sino también en el diseño de prótesis médicas, las cuales se van a adaptar a las necesidades de cada paciente.

En estos momentos es cuando nos damos cuenta de lo que será la medicina del futuro, en la cual ya no son suficiente las imágenes en dos dimensiones, ahora con esta nueva tecnología se podrá mejorar en un gran porcentaje la planificación de las operaciones, el objetivo de una impresora 3D es transformar una idea en un algo real y tangible.



**FIGURA 1. 1: Modelo 3D de un cráneo hecho en impresora 3D [1]**

Se inició en la elaboración de piezas individuales de plásticos e incluso de metal. Tantos arquitectos, ingenieros, doctores, etc. elaboran los dibujos en su computador en un software de CAD para transformar estos archivos en tercera dimensión.

Las versiones comerciales construyen piezas a partir de los datos de un archivo CAD en formato **.STL** (monocromo) o **.VRML** (color). Esta tecnología se ve aplicado actualmente en campos específicos como: Automatización, Diseño de calzados, Industrias, Arquitecturas, en

Medicina, en Educación o topografías, entre otras aplicaciones.

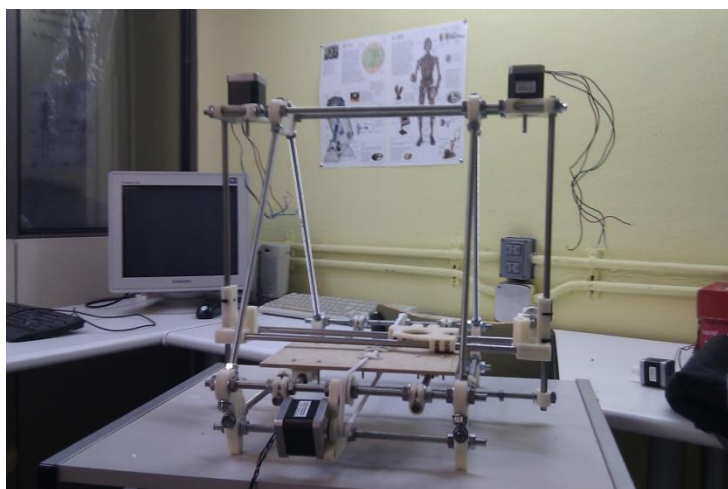
Los modelos comerciales de las impresoras 3D se han fabricado de dos tipos:

- De adición, o de inyección de polímeros
- De compactación

**De adición o inyección de polímeros:** En las cuales el material se añade por capas. Se Inyecta resina cuando el material está en estado líquido mientras una luz ultravioleta lo va solidificando. Se caracteriza por gran la precisión de los acabados de la impresión. En cuanto finaliza la impresión las piezas están preparadas para la manipulación.

**De compactación:** La masa de polvo se compacta por estratos, existen dos tipos de impresoras: impresoras de laser (el resultado de la impresión es de gran calidad y en el momento que el material ha secado se puede manipular sin problemas) y de tinta (los materiales que se utilizan para la impresión son económicos, pero el resultado es más frágil, normalmente hay que fortalecerlo con otros materiales para darle la dureza necesaria).

La naturaleza en autorreplicación de RepRap también puede facilitar una expansión de manera muy acelerada y una mayor ciencia revolucionaria en el diseño y fabricación de productos, desde la producción comercial de artículos bajo patente hasta una de escala personal, basada en especificaciones libres. Dando apertura al diseño de productos y oportunidades de manufactura a cualquier persona, debería en gran medida reducir el ciclo de trabajo para las mejoras a productos y soporte de una mayor diversidad de materiales, mayor que la capacidad actual de producción de las fábricas.



**FIGURA 1.2: Impresora 3D Prusa Mendel [2]**

Como en el caso de una impresora láser, los datos provienen del computador, o más exacto de un programa CAD, con el cual tanto



arquitectos como ingenieros elaboran los dibujos en su computador. Un software divide estos modelos de CAD en capas virtuales, que luego se va a plasmar con muy alta precisión con la impresora de 3 Dimensiones. Por ejemplo el milagro de los datos digitales, en todo el mundo se puede imprimir el mismo boleto en papel, y ahora también se puede imprimir en todas partes el mismo abrebotellas en plástico – sin pedido, envío o fabricación laboriosa. El trabajo lo hace su pequeña fábrica en el escritorio.

**[3]**

## **1.2 Motivación**

Crear un sistema en el cual podamos poner en práctica los conocimientos adquiridos en la materia de graduación y durante todo el transcurso de nuestras vidas en la universidad los cuales nos fortalecen cada día más en nuestra vida profesional y conocer mucho más sobre el funcionamiento de otros dispositivos electrónicos.

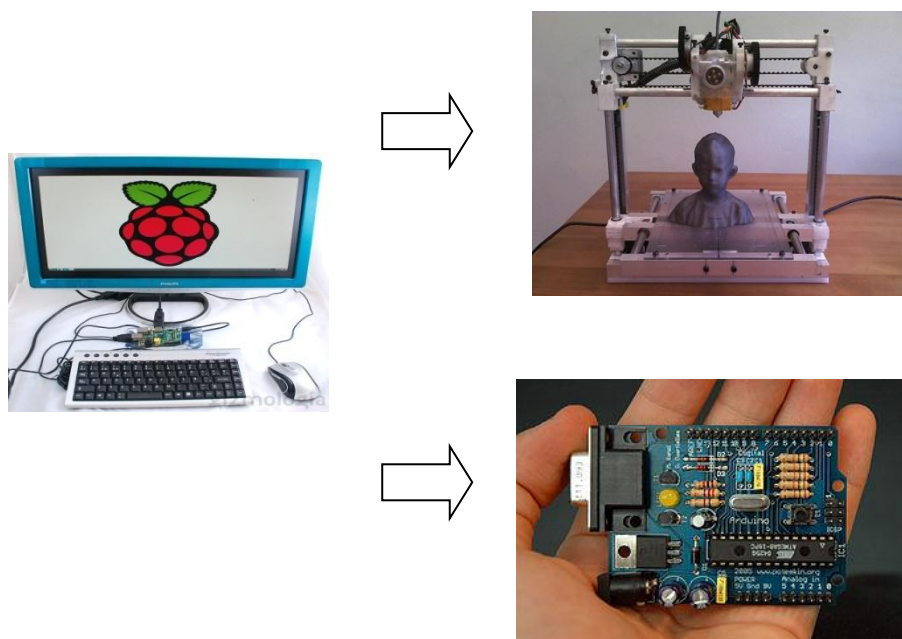
Además quisimos indagar y experimentar con las nuevas formas de la electrónica moderna basada en micro-controladores y ponerlas en práctica mostrando a los demás todo lo que se puede realizar con lo nuevo de la tecnología.

Diseñar una interfaz para que el usuario se pueda comunicar con el mini computador Raspberry Pi. Se adquirió conocimientos del hardware y software para así poder lograr el objetivo en el proyecto, con el cual se demostrará que con pequeñas inversiones se logra grandes cosas en la vida.

La motivación al realizar este proyecto fue la idea de poder construir piezas propias , tanto las que pueda aprender a diseñar en un futuro, como los modelos que pueda conseguir por internet.

### **1.3 Descripción del Proyecto**

Diseño de Interfaz usando minicomputador Raspberry Pi a ser usada en la impresión en 3 Dimensiones con demostración práctica de funcionamiento en prototipo de impresora RepRap, vamos controlar una impresora 3DReprap, la cual está basada en la tarjeta Arduino Mega, para lo cual nos veremos en obligación de estudiar, analizar y dominar a la perfección estas dos nuevas tarjetas, las cuales poseen diferentes características, lenguajes y sistemas operativos, además se debe tener en cuenta que para armar la impresora 3d, debemos poner en práctica, no solo nuevos conocimientos, sino también conocimientos de electrónica que se aprendió al cursar la carrera. [4]



**FIGURA 1.3 Raspberry Pi, impresora 3d y placa Arduino [5]**

Para empezar a ensamblar nuestra impresora 3d usamos la tarjeta Arduino Mega como base de la placa ramps 1.2 o la 1.4 o la 1.3, y se le añadirá los elementos que se necesita como disipadores para los drivers de potencia, esta placa se alimentará a 5v, los elementos extras que necesitemos, los soldaremos fácilmente a nuestra placa, luego de esto descargaremos e instalaremos el software, antes de esto verificamos que funcione nuestra tarjeta arduino, Luego se descarga los firmware que se necesitan para que funcione nuestra impresora, lo recomendado es que se use el más estable, en este caso se usara el Sprinter y se verificará que funcione bien el programa que se ha descargado, luego hay que verificar que este operando

correctamente el Arduino, se trabajará en la parte electrónica y la parte de ensamblado de la máquina, se comprobará que los motores paso a paso que van a ser controlados por software funcionen, luego se ajusta la corriente que va a recibir cada uno de los motores mediante un multímetro, luego de esto se procede a ensamblar la impresora, con el kit de piezas que se ha recibido, se empotra tuercas, las poleas, las correas, aplanar los ejes de los motores, después de esto calibraremos los ejes x, y y los 2 motores del eje z con el extrusor, después de esto se empieza a montar las partes del chasis de la impresora, los de la parte delantera, trasera ,etc. Luego se conoce el extrusor y todas las partes electrónicas de la impresora.

#### **1.4 Objetivos principales**

- Diseñar una interfaz amigable y de fácil comprensión para poder usar la impresora 3D, dicha interfaz será utilizada por usuarios generales, los cuales mediante esta, podrán diseñar modelos a su elección.
- Demostrar dominio y comprensión de los dispositivos electrónicos y el medio programable.
- Aprender sobre otros sistemas operativos y software para el manejo de las tarjetas Raspberry Pi y arduino, las cuales usaremos para realizar nuestro proyecto con éxito.

- Manipular y controlar la impresora 3D a través de las tarjetas arduino y Raspberry Pi.
- Introducir al lector en el mundo de la impresión 3D y, fundamentalmente, de las impresoras 3D código abierto.
- Aplicar conceptos básicos sobre electrónica, buscando nuevas posibilidades para cada uno de los elementos; y aconsejar qué hacer para evitar errores en el ensamblado, tanto de la estructura como de la electrónica.

### **1.5 Limitaciones**

- En el país el desarrollo y la divulgación de las nuevas tecnologías es prácticamente nulo, por lo tanto este proyecto ha sido muy complicado conseguir los componentes, como por ejemplo las tarjetas Arduinos, Raspberry y Ramps, partes para la Impresora que se necesita, debido a eso se ha procedido importar de países desarrollados por que los proveedores no se sienten tentados a comercializarla, debido a que las personas que no posee altos recursos económicos no se verían en la posibilidad de adquirirlo, es decir es complicado obtenerlo en nuestro país debido a la poca demanda.

- El RaspBerry Pi como su aplicación es en Linux se complicó un poco, ya que al tener conocimientos de los algoritmos y su aplicación, pero se buscó información de tutoriales y programas básicos para poder entender sus librerías, para así poder construir el software necesario para el proyecto.
- Pronterface: Este programa es uno de los software más usados para la impresión en 3D, tiene sus versiones para Windows, Mac y Linux. El cual lo debemos instalar en la Raspberry, el cual no es tan fácil porque su sistema operativo es Linux, debido a que hay q investigar los comandos para poder ejecutar e instalar los driver para que funcione con normalidad el programa.

## **CAPÍTULO 2**

### **2. FUNDAMENTO TEÓRICO**

Describiremos Arduino como el principal software y hardware de uso para controlar la impresora 3D RepRap, entre otros componentes los cuales nos ayudaran para implementar y diseñar el interfaz para usarlo mediante el Rapsberry Pi.

También se describirá las ventajas de uso de dichas placas y componentes para lograr el proyecto.

## **2.1 HERRAMIENTAS DE SOFTWARE PARA EL DESARROLLO DEL PROYECTO.**

Para el desarrollo del Proyecto se utilizará las siguientes herramientas de Software:

Arduino, Entorno de programación para Micro-controladores

Arduino es una tarjeta de interfaz que proporciona un bajo costo, fácil de utilizar la tecnología de la que esta creado se apoya en un micro-controlador de proyectos con una electrónica en la que tu Arduino realice tipo de cosas de las luces de control en instalación de arte del poder en el sistema de energía solar.

### **2.1.1 ¿QUÉ ES ARDUINO?**

Arduino es una plataforma de prototipos electrónicos de código abiertos, basado en un hardware y un software flexibles y fáciles de usar, con un bus serie universal (USB) para conectar a su ordenador y una serie de conexiones de cableados electrónicos externos, puede afectar a su alrededor mediante el control tales como motores, relés, sensores de luz, diodos láser, altavoces, micrófonos y mucho más o bien puede ser alimentado a través de la conexión USB del ordenador, con una batería de 9V o fuente de alimentación. Se pueden controlar desde el ordenador o programado por el equipo y luego se desconecta y se deja de trabajar independientemente.



El microcontrolador de la placa se programa usando el ArduinoProgrammingLanguage (basado en Wiring) Y el ArduinoDevelopmentEnvironment (basado en processing).

**Processing** es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.

## **[6] Programing Arduino – getting started and sketch**

### **2.1.1.1 ¿Por qué el uso de Arduino?**

Hay muchos otros micro-controladores y plataformas micro-controladoras disponibles para computación física. Parallax Basic Stamp, Netmedia's Bx-24, Phdgets, MIT'sHandyboard y muchas otras ofertas de funcionalidad similar. Todas estas herramientas toman los desordenados detalles de la programación de microcontrolador y la encierra en un paquete más fácil de usar.

Arduino también simplifica el proceso de trabajo con micro-controlares, pero ofrece algunas ventajas para quienes lo usen sobre otros sistemas

El sistema Arduino sirve para:

- Desarrollar programas que interactúan con el mundo, Redes de sensores, Robótica sencilla, Alarma, Sistema de comunicaciones

### 2.1.1.2 Ventajas del sistema Arduino sobre otros.

**Es económico,** las placas de Arduino son relativamente baratas con otras plataformas micro-controladores. La versión menos cara del módulo Arduino puede ser ensamblada a mano.

**Multiplataforma:** el software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/ Linux

**Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar, pero suficientemente flexible para que usuarios puedan aprovecharlos también. Basado en el entorno de programación Processing de manera que aprenderán a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino

**Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++ y los detalles técnicos puede hacer el salto desde Arduino a la programación e lenguaje AVR C.

**Código abierto y hardware extensible:** Arduino está basado en micro-controladores ATMEGA8 Y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo.

## **2.1.2 Software Controladores**

### **2.1.2.1 Firmware “Sprinter”**

Los principales desarrolladores de Sprinter Actualmente Kliment y la CARU, aunque muchos otros contribuyen con sus parches. Se trata de un firmware para las rampas y otras configuraciones de un solo procesador de electrónica de RepRap. Es compatible con la impresión de la tarjeta SD, control heatbed activo y ATmega dominada internos. Este trabajo está licenciado bajo la GNU GPL v3 o (a elección del usuario) cualquier versión posterior. Se basa en el firmware de Tonokips, que fue licenciado bajo GPL v2 o posterior.

Los ajustes a la calibración se realizan en el firmware porque aquí es donde los motores se les dice lo mucho que se mueva cada vez que se les dice que mover una unidad. Usted tendrá que saber cómo compilar y subido firmware a su Arduino para encontrar la siguiente información útil.

El firmware velocista se puede encontrar en la siguiente pagina  
<https://github.com/kliment/Sprinter> .

En primer lugar, usted tendrá que elegir sus aparatos electrónicos en el archivo configuration.h:

```
// MEGA / RAMPAS hasta 1,2 = 3,  
  
// RAMPAS 1,3 = 33  
  
// Gen6 = 5,  
  
// Sanguinololu hasta 1,1 = 6  
  
// Sanguinololu 1,2 y por encima de = 62  
  
// Teensylu (AT90USB) = 8  
  
// Gen 3 Plus = 21  
  
// Gen 3 monolítico Electrónica = 22  
  
# Define MOTHERBOARD 3
```

Lo siguiente define selecciona qué tarjeta de la electrónica que tiene.

Por favor, elegir el que corresponda a su configuración

Ajuste de termistor.

```
// 1 es termistor 100k

// 2 es termistor 200k

// 3 es mendel-parts termistor

// 4 es termistor 10k

// 5 se suministra Parcan 104GT-2 100 K

// 6 es 100k EPCOS

// 7 es 100k Honeywell termistor 135-104LAG-J01

# Define THERMISTORHEATER 1

# Define THERMISTORBED 1
```

Ajustes de calibración son el interior del archivo configuration.h (tome nota de los comentarios en torno a este código si utiliza Metric Prusa Mendel con Makergear extrusora de pasos orientados o MakerGear híbrido Prusa Mendel):

Las variables //// calibración

// X, Y, Z, E pasos por unidad - Metric Prusa Mendel con Wade extrusora:

```
flotar axis_steps_per_unit [] = {40, 80, 3200/1.25, 700};
```

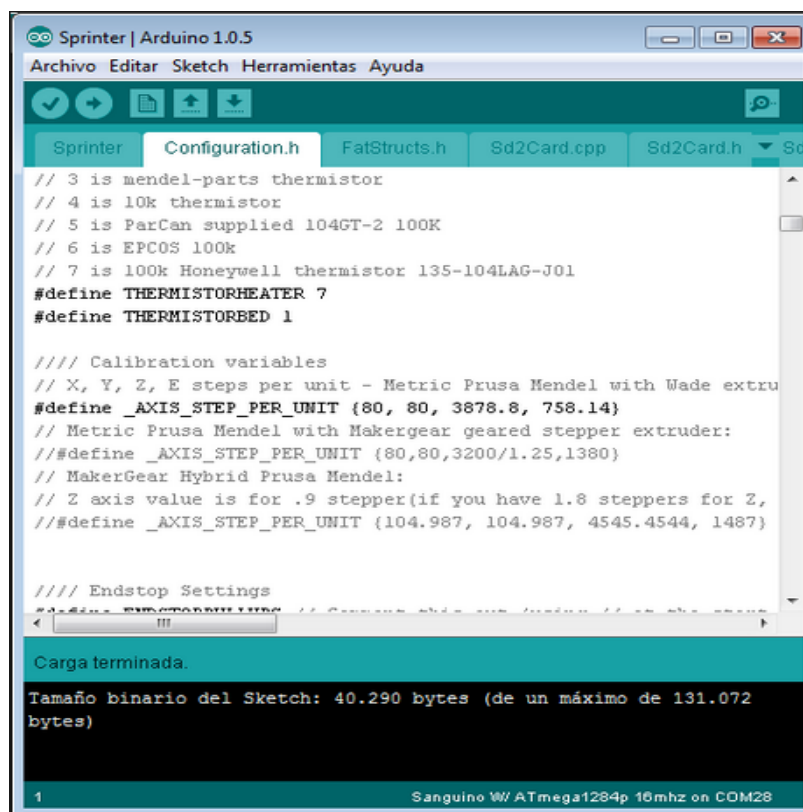
Si está utilizando RAMPAS, es importante encontrar las siguientes líneas en el archivo pins.h y hacer lo que los comentarios que dicen:

// // Descomentar alguna de las siguientes líneas para v1.3 o v1.0 RAMPAS, comentan tanto para v1.2 o 1.1

# Define RAMPS\_V\_1\_3

# Define RAMPS\_V\_1\_0

## [7] Sprinter



The screenshot shows the Sprinter Arduino IDE interface. The main window displays the Configuration.h file with the following code:

```
// 3 is mendel-parts thermistor
// 4 is 10k thermistor
// 5 is ParCan supplied 104GT-2 100K
// 6 is EPCOS 100k
// 7 is 100k Honeywell thermistor 135-104LAG-J01
#define THERMISTORHEATER 7
#define THERMISTORBED 1

//// Calibration variables
// X, Y, Z, E steps per unit - Metric Prusa Mendel with Wade extruder
#define _AXIS_STEP_PER_UNIT {80, 80, 3878.8, 758.14}
// Metric Prusa Mendel with MakerGear geared stepper extruder:
//#define _AXIS_STEP_PER_UNIT {80,80,3200/1.25,1380}
// MakerGear Hybrid Prusa Mendel:
// Z axis value is for .9 stepper(if you have 1.8 steppers for Z,
//#define _AXIS_STEP_PER_UNIT {104.987, 104.987, 4545.4544, 1487}

//// Endstop Settings
// #define ENDSTOPDRUINS // Comment this out (unless // on the board
```

The terminal window at the bottom shows the following output:

```
Carga terminada.
Tamaño binario del Sketch: 40.290 bytes (de un máximo de 131.072 bytes)
```

The status bar at the bottom indicates: Sanguino W/ ATmega1284p 16mhz on COM28

Figura 2.2 Sprinter versión estable [8]

### 2.1.2.2 Pronterface

El pronterface es una interfaz gráfica que nos permite comunicarnos con la impresora, y tanto hacer movimientos manuales, como enviarle los datos generados por el Slic3r. Existen otros software pero no los he probado. En Windows ya viene integrado el slic3r en el pronterface.

Es muy sencillo de utilizar, utiliza el Slic3r para convertir los sólidos (.stl) en láminas o capas de plástico, y posteriormente genera los movimientos para imprimir cada capa. En Mac hay que instalarlos por separado y luego configurar la ruta de acceso al slic3r dentro del Pronterface.

### [9] Pronterface

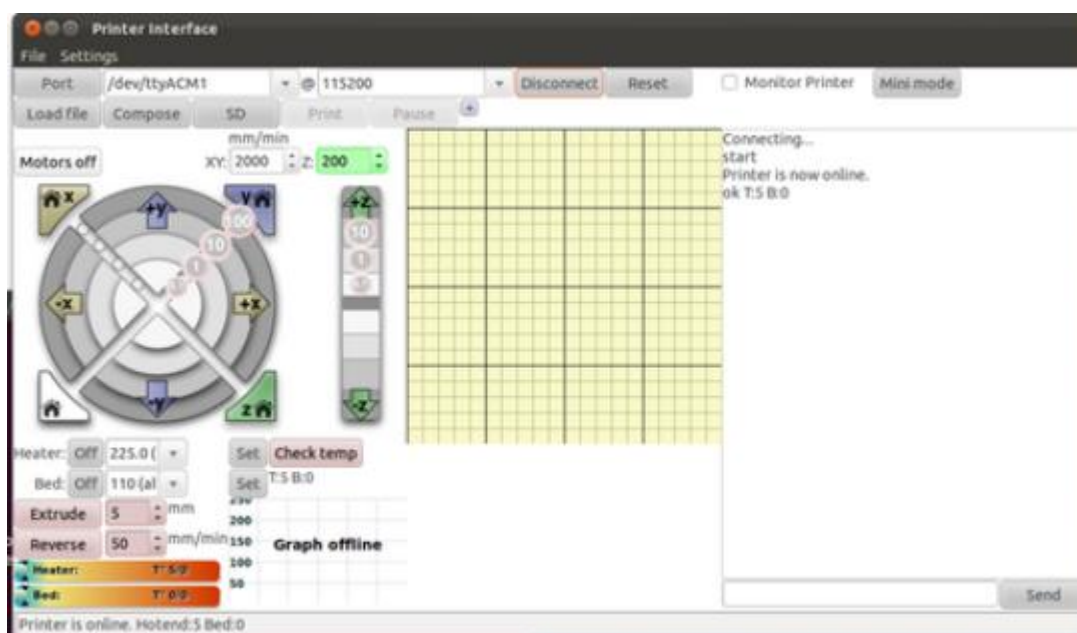


Figura 2.3 Pronterface Interfaz [10]

### 2.1.2.3 Slic3r (Gcode)

Slic3r es la herramienta que necesita para convertir un modelo en 3D digital en las instrucciones de impresión para la impresora 3D. Se corta el modelo en rebanadas horizontales (capas), genera las trayectorias de herramientas para llenarlos y calcula la cantidad de material a extruir. El proyecto se inició en 2011 a partir de cero: el código y los algoritmos no se basan en ninguna otra obra anterior. La legibilidad y mantenibilidad del código son algunos de los objetivos de diseño. Slic3r se lía con los más importantes paquetes de software de host: MatterControl , Pronterface , Repetier en host , ReplicatorG. Ha sido apoyado y financiado por casi todas las principales empresas de impresión en 3D en el mundo. [11] Slicer

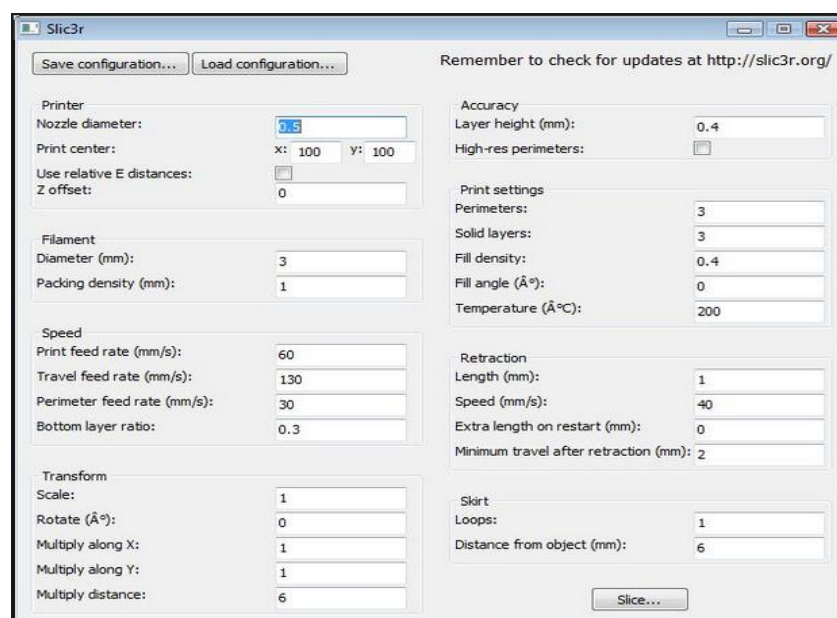


Figura 2.4 Calibración de Slic3r [12]



## 2.2 HERRAMIENTAS DE HARDWARE PARA LA IMPLEMENTACIÓN DEL PROYECTO.

Hay múltiples versiones de placa Arduino. La mayoría usan el ATmega168 de Atmel, mientras que las placas más antiguas usan el ATmega8

### 2.2.1 Placas E/S

**Diecimila:** Esta es la placa Arduino más popular. Se conecta al ordenador con un cable estándar USB y contiene todo lo que necesitas para programar y usar la placa.

**Nano:** Una placa compacta diseñada para uso como tabla de pruebas, el Nano se conecta al ordenador usando un cable USB Mini B

**Bluetooth:** El Arduino BT contiene un módulo Bluetooth que permite comunicación y programación sin cables. Es compatible con los dispositivos del sistema Arduino.

**LilyPad:** Diseñada para aplicaciones listas para llevar esta placa puede ser conectada en fábrica y un estilo sublime.

**Mini:** esta es la placa más pequeña de Arduino. Trabaja bien en tabla de pruebas o para aplicaciones en las que prima el espacio. Se conecta al ordenador usando el cable Mini USB.

**Serial:** es una placa básica que usa RS232 como un interfaz con el ordenador para programación y comunicación. Esta placa es fácil de ensamblar incluso como ejercicio de aprendizaje.

## 2.3 Arduino Mega

### 2.3.1 Visión General del Arduino Mega

El Arduino Mega es una placa micro-controladora basada en el Atmega168. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usados como salidas PWM), 6 entradas analógicas, un oscilador de cuarzo a 16Mhz, Una conexión USB, un conector para alimentación, una cabecera ICSP, y un botón de reseteo.

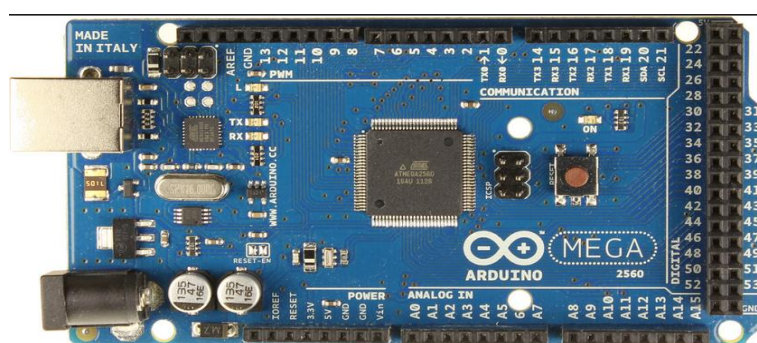


Figura 2.5 Placa de Arduino Mega (USB) [13]

### 2.3.2 Resumen de características y descripciones

Características	Descripción
Microcontrolador	ATmega168
Voltaje de operación	5V
Tensión de entrada	7-12 V
Tensión de entrada (límite)	6-20V
Pines digitales de E/S	14 pines de los cuales 6 proveen salidas PWM
Pines de entrada analógicas	6 pines
Corriente DC por pin E/S	40mA
Corriente DC para pon 3.3 V	50mA
Memoria Flash	16KB (de los cuales 2KB usados para Bootloader)
SRAM	1KB
EEPROM	512 bytes
Frecuencia de reloj	16MHz

**Tabla 1. Características técnicas de Arduino Mega**

### 2.3.3 Alimentación

El Arduino puede ser alimentado a través de conexión USB o con un suministro de energía externo. La fuente de energía se selecciona mediante el jumper PWMR SEL. Para alimentar la placa desde la conexión USB, colocarlo en los dos pines más cercanos al conector USB, para un suministro de energía externo, en los dos pines más cercanos al conector de alimentación externa.

La alimentación externa (no USB) puede venir o desde un adaptador AC a DC o desde una batería. La placa puede operar con un suministro externo de 6 a 20 V. Si es suministrada con menos de 7V, sin embargo el pin de 5V, puede suministra menos de cinco voltios y la podría ser inestable. Si usa más de 12V el regulador de tensión podrá sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 Voltios.

Los pines de alimentación son los siguientes:

**VIN.** La entrada de tensión a la placa Arduino está usando una fuente de alimentación externa (al contrario de los 5 voltios de la conexión USB u otra fuente de realimentación regulada. Puedes suministrar tensión a través de este pin, o si suministra tensión a través del conector de alimentación, acceder a él a través de este pin.

**5V.** El suministro regulado de energía usando para alimentar al microcontrolador y otros componentes de la placa. Este puede venir o desde VIN a través de un regulador en la placa o ser suministrado por USB u otro suministro regulado de 5V.

**3V3.** Un suministro de 3.3V generado por el chip FTDI de la placa. La corriente máxima es de 50mA

**GND.** Pines de tierra.

### **2.3.4 Memoria**

El ATmega168 tiene 16KB de memoria Flash para almacenar código (de los cuales 2KB se usa para el bootloader). Tiene un 1KB DE SRAM Y 512 bytes de EEPROM (que puede ser leída y escrita con la librería EEPROM)

### **2.3.5 Entradas y Salidas**

Cada uno de los 14 pines digitales del Arduino puede ser usado como entrada o salida usando aplicaciones pinMode(), digitalWrite() y digitalRead(), operan a 5 Voltios.

Cada pin puede proporcionar o recibir un máximo de 40mA y tiene una resistencia interna pull up (desconectado por defecto) de 20 a 50Kohms.

Además hay algunos pines que tienen funciones especiales.

**Serial: 0 (RX) y 1 (TX).** Usador para recibir RX y transmitir TX datos TTL en serie. Estos pines están conectados a los pines correspondientes del chip FTDI USB a TTL Serie.

Interruptores externos: 2 y 3. Estos pines pueden ser configurados para disparar un interruptor en un valor bajo, un margen crecente o decreciente, un cambio de valor. Usan la función `attachInterrupt ()`

**PWM:** 3, 5, 6, 9 ,10 y 11. Proporción salida PWM de 8 bits con la función `analogWrite ()`.

SPI: 10(SS), 11(MOSI), 12 (MISO), 13 (SCK). Estos pines soportan comunicación SPI, la cual aunque proporcionada por el hardware subyacente, no está actualmente incluida en el lenguaje Arduino.

**LED: 13.** Hay un LED conectado al pin digital 13. Cundo el pin esta a valor HIGH, el LED se enciende, cuando pin esta LOW, está apagado.

I2C: 4 (SDA) Y 5 (SCL). Soportan comunicación I2C (Twi) usando la librería Wire.

Hay otro par de pines en la placa

**AREF.** Voltaje de referencia para las entradas analógicas. Usando con `analogReference ()`.

**RESET.** Pone esta línea a LOW para resetear el microcontrolador. Típicamente usada para añadir un botón de reseteo a dispositivos que bloquean la placa principal.

### 2.3.6 Comunicación

El Arduino tiene un número de infraestructuras para comunicarse con un ordenador, otro Arduino u otros microcontroladores. El ATmega168 provee comunicación serie UART TTL (5V), La cual está disponible en los pines digitales 0 (RX) Y 1 (TX). Un FTDI FT232RL en la placa canaliza esta comunicación serie al USB y a los drivers FTDI (incluidos con el software Arduino) proporcionan un puerto de comunicación virtual al software del ordenador.

Una librería Software Serial permite comunicación serie en cualquier de los pines digitales del Arduino.

El ATmega168 también soporta comunicación 12C (TWI) Y SPI. El software Arduino incluye una librería Wire para simplificar el uso del bus 12C. Para usar la comunicación SPI, consultar con el esquema del ATmeg168.

### **2.3.7 PROGRAMACIÓN**

El Arduino puede ser programada con el software Arduino Ide

El ATmega168 de Arduino viene con un bootloader pregrabado que te permite subirle nuevo código sin usar programador hardware externo. Se comunica usando el protocolo original STK500.

También puedes saltar el bootloader y programar el ATmega16 a través de la cabecera ICSP (In-circuit serial programming)

### **2.3.8 Instalar el Software Arduino**

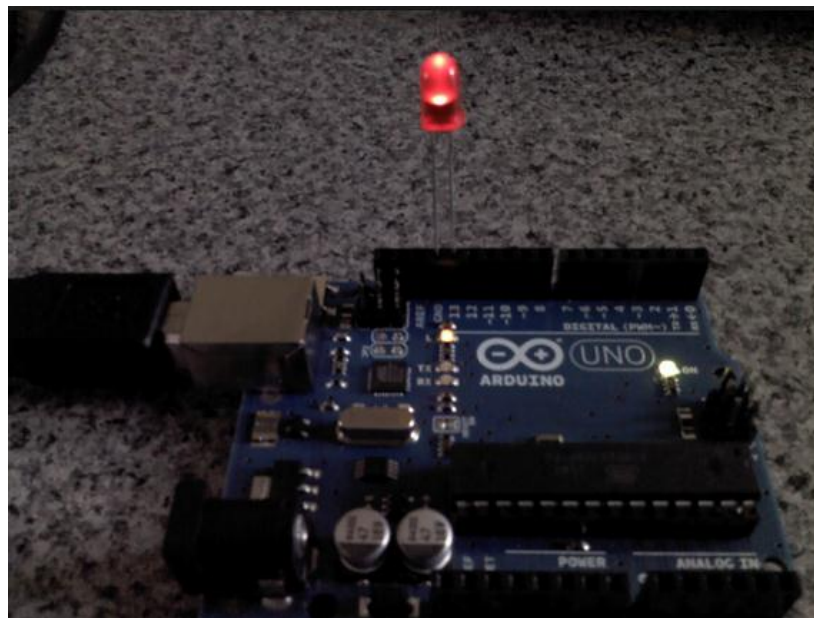
Esta sección explica como instalar el software Arduino en un ordenador que ejecuten cualquiera de los siguientes Sistemas Operativos: Windows, Mac OS X, GNU/Linux.



Este documento explica como conectar tu placa Arduino al ordenador y cargar tu primera rutina.

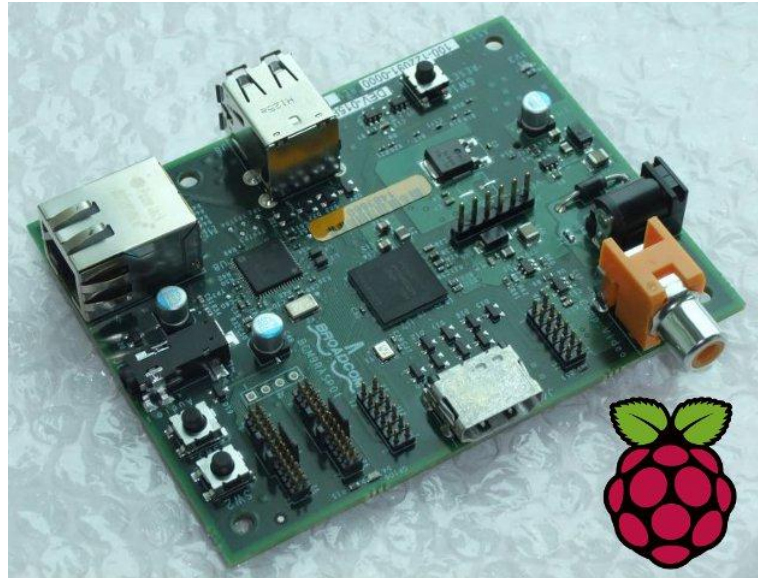
Estos son los pasos que se seguirá:

1. Obtener una placa Arduino y un cable.
2. Descargar el entorno Arduino.
3. Instalar los drivers USB.
4. Conectar la placa.
5. Ejecutar el entorno Arduino.
6. Subir un programa.
7. Buscar el Led que parpadea. **[14]Arduino tutorial**



**Figura 2.6 Funcionamiento de Arduino [15]**

## 2.4 Que es Raspberry Pi?



**Figura 2.4 Raspberry Pi [16]**

El Raspberry Pi es que es tan sólo un ordenador de propósito general muy pequeño (que puede ser un poco más lento de lo que estamos acostumbrados para algunas aplicaciones de escritorio, pero mucho mejor en algunas cosas que no sea un PC normal) para que pueda hacer lo que podría hacer en un ordenador normal con él. Además, el Raspberry Pi tiene multimedia de gran alcance y capacidades de gráficos en 3D, por lo que tiene el potencial para ser utilizado como juego de plataforma.

El Raspberry Pi es una maravilla en miniatura, el embalaje de alimentación del equipo considerado en un tamaño no más grande que una tarjeta de crédito, es capaz de algunas cosas asombrosas.

El Raspberry Pi es barato para comprar y es probable que tenga todo el equipo necesario para hacerlo funcionar: un TV y una tarjeta SD que puede venir de la cámara vieja, un cargador de teléfono móvil, un teclado y ratón, no se comparte poco con la familia, sino que pertenece al equipo y es lo suficientemente pequeño como para poner en un bolsillo y llevarlo a cualquier parte que uno desee.

Gracias al análisis se pudo tomar en cuenta que el Raspberry Pi podría funcionar de manera eficaz en modernos centro de medios de comunicación muy baratos, por lo que hizo hincapié en que la capacidad para absorber a los incautos - con las personas que iban a recoger un poco de programación mientras están en ello.

El procesador en el corazón del sistema de Raspberry Pi es un **BCM2835**, esto significa que la gran mayoría de los componentes del sistema, incluyendo su central y las unidades de procesamiento gráfico junto con el audio y el hardware de comunicación que están contruidos

en un solo componente oculto bajo 256 MB de chip de memoria en el centro del tablero.

Otra diferencia importante entre el Raspberry Pi y su computadora de escritorio o portátil, con excepción de tamaño y precio, es el sistema operativo - el software que le permite controlar el ordenador.

La mayoría de las computadoras de escritorio y portátiles disponibles en la actualidad ejecuta uno de los dos sistemas operativos: Microsoft Windows o Apple OS X. PLATAFORMAS ambos son de código cerrado, creado en un ambiente reservado con técnicas patentadas.

El Raspberry Pi, por el contrario, está diseñado para ejecutar un sistema operativo llamado GNU Linux en lo sucesivo como Linux. A diferencia de Windows u OS X, Linux de código abierto, es posible descargar el código fuente del sistema operativo y hacer los cambios que desee. No hay nada oculto y todos los cambios se hacen a la vista del público. Este desarrollo de código abierto ha permitido Linux para ser rápidamente modificado para correr en el Raspberry pi algunas versiones de Linux conocida como distribuciones han sido portados a la pi frambuesa es de chip BCM2835 incluyendo Debian, Remix y Fedora Linux arch.

### **[17] Raspberry pi Guide**

## **2.5 OTRAS COMPONENTES ELECTRÓNICAS**

### **2.5.1 Motores paso a paso.**

A continuación, se describirá otros dispositivos electrónicos cuyos parámetros también son críticos para el funcionamiento correcto de la impresora 3D, y que no están integrados en la placa electrónica; uno de ellos son los motores 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. El motor paso a paso se comporta de la misma manera que un conversor digital-analógico (D/A) y puede ser gobernado por impulsos procedentes de sistemas lógicos

Por otro lado está la alimentación de los motores. Los controladores Pololus, drivers de los motores, utilizan un modo de control a corriente constante, de modo 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.

Los 5 motores deben de ser motores paso a paso. Asegúrese de obtener los al menos 20 mm de longitud del eje. Hacerlo más corto, requerirá de modificaciones. Uno de los cinco motores, para su uso en el extrusor, debe ser capaz de crear un par de mantenimiento de por lo menos 40Ncm.

La cifra que tienes que mirar es el holding torque que sea entorno a los 4kg-cm y la corriente menor de 2A y la tensión menor de 12V (algunos son de 24V y no sirven). Normalmente los motores los vamos a usar por debajo de sus especificaciones máximas (como a 1A) por lo que el par disponible será menor. Por eso no conviene partir de un motor con mucho menos par. El par (torque) se expresa también en Ncm o Nm y puedes encontrar por ahí convertidores de unidades.



**Figura 2.5 Motor paso a paso Nema 17 de RepRap World [18]**

### 2.5.2. Mosfet

El último elemento característicos de las placas electrónicas, escojamos la opción que escojamos, son los Mosfet. Las siglas MOSFET significan Metal Oxide Semiconductor Field Effect Transistor, es decir, Transistor de Efecto de Campo Metal Óxido Semiconductor. Se llaman así porque en la patilla gate o puerta, a través de la cual controlamos el paso de corriente de la patilla source o fuente a la patilla drain o drenador (también podemos verlo como “sumidero”), está conectada a un metal, el mismo que está separado del semiconductor por un óxido.

Funcionando de forma similar a un condensador, de tal forma que permite polarizar el transistor, hacer que funcione, pero no consume energía por la puerta; es decir, que no hay corriente pero si tensión. Esto quiere decir que se controlará el paso de corriente mediante aplicaciones de tensión; de esta forma, en modo estático, o reposo, el transistor apenas consumirá energía pues el valor de la corriente será del orden de nano amperios

Es un transistor utilizado para amplificar o conmutar señales electrónicas. Es el transistor más utilizado en la industria microelectrónica, ya sea en circuitos analógicos o digitales, aunque el transistor de unión bipolar fue mucho más popular en otro tiempo. Prácticamente la totalidad de los microprocesadores comerciales están basados en transistores MOSFET

**[19] Mosfet**



**Figura 2.6 Mosfet canal N, STP55NF06L [20]**

### **2.5.3 Driver controlador de motores**

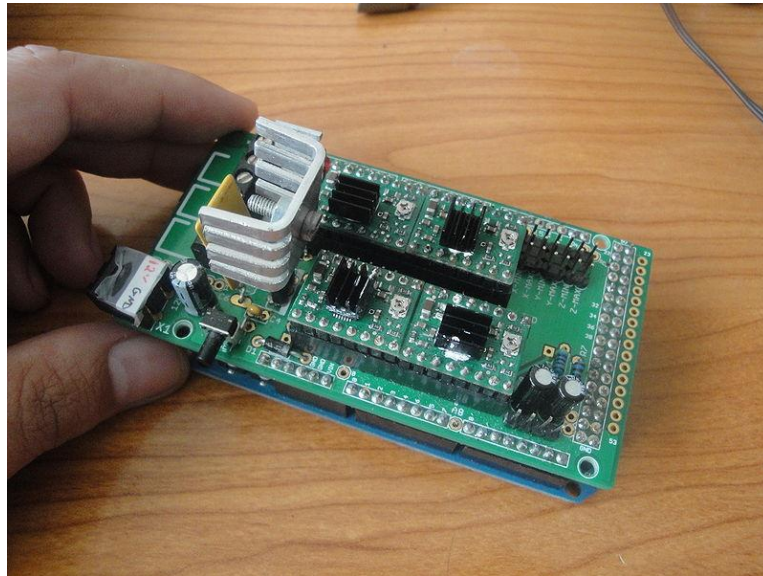
Los drivers que utilizamos para controlar los motores paso a paso son unos micro-controladores, más pequeños, que se conectan a la placa electrónica, tanto a la Sanguinololu, como a la Ramps (en el caso de Arduino + Ramps). En concreto, se usan los del modelo Pololus A4988, que es el modelo de la figura aunque algunos proveedores como Reprap World venden otro modelo que imita al Pololus A4988 y que es de menor costo que el primero, se refiere el modelo Stepstick. También se usa el modelo Pololus A4983, que es exactamente igual que el Pololus. **[21] Pololus driver de motores**



**Figura 2.7 Pololus A4988 [22]**



Estos elementos se instalan en los pines destinados para ello de la placa electrónica, quedando como vemos en la imagen de la figura



**Figura 2.8 Conexión de Pololus en la Ramps 1.2 [23]**

En la figura 122 se ha colocado los cuatro Pololus en sus respectivas posiciones dentro de un circuito integrado modelo Ramps 1.2.

La función de los Pololus es ordenar el movimiento de los motores paso a paso, a través de impulsos eléctricos. La magnitud del movimiento la determinará el usuario a través del ordenador, mientras no estemos imprimiendo; o, el propio Software de control mientras se ejecuta una impresión.

Los Pololus, así como los Stepstick, llevan integrados un potenciómetro para regular la corriente que le suministraremos al motor, que estará en función de la carga a mover, y que debemos calibrar como se verá en el apartado

## **CAPÍTULO 3**

### **3 Diseño e implementación del proyecto**

#### **3.1. Introducción**

A continuación se detalla el proceso de diseño e implementación del proyecto que consiste en el envío de instrucciones desde la Raspberry Pi a una tarjeta Arduino Mega la cual controlará una impresora 3D, la tarjeta Raspberry Pi usará el programa pronterface como programa CAD, para diseñar los objetos a imprimir.

Para explicar cada ejercicio, se hará una pequeña introducción seguida de un diagrama de bloques explicativo. Después se explicará gráficamente el

algoritmo utilizado en un diagrama de flujo, y por último se agregará los códigos de programación utilizados.

### **3.1.1. Interfaz de conexión entre la Raspberry PI al Arduino vía Serial**

#### **Introducción**

En este ejercicio se dará a conocer al Raspberry PI, luego al Arduino y luego se procederá a conectarlos vía puerto serial. Al final, se podrá usar esta configuración para conectar sus Arduinos (o cualquier dispositivo que se comunique via Serial) al Raspberry PI, usando ciertos códigos en Python para lograr con éxito dicha interfaz.

Para enviar y recibir datos por el puerto se tiene que cambiar un par de líneas de configuración.

Primero en la RasPI tenemos que modificar el archivo ***/boot/cmdline.txt*** y eliminar la parte que dice:

```
console=ttyAMA0,115200 kgdboc=ttyAMA0,115200
```

Lo siguiente que hay que hacer es comentar la siguiente línea en el archivo ***/etc/inittab***:

```
#Spawn a getty on Raspberry Pi serial line  
  
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Una vez realizadas estas modificaciones se reinici nuestra RasPI. Estas modificaciones evitarán que el Kernel envíe mensajes a la línea serial.

Dentro de la consola se ejecuta el siguiente comando:

```
$ stty -F /dev/ttyAMA0 9600
```

El comando anterior se encarga de establecer la velocidad del puerto serial de la tarjeta Raspberry a 9600bps, algunas personas han reportado problemas de comunicación con arduinos a velocidades mayores, así que nos brindara seguridad y eficacia, esto conlleva a que sea posible que se conecte a mayores velocidades.

Para leer los datos de la consola ejecutamos el siguiente comando:

```
$ cat/dev/ttyAMA0
```

Deberá de mostrarse una pantalla como la mostrada en la figura 3.1:

### [24] Interfaz de conexión entre la Raspberry PI al Arduino vía Serial

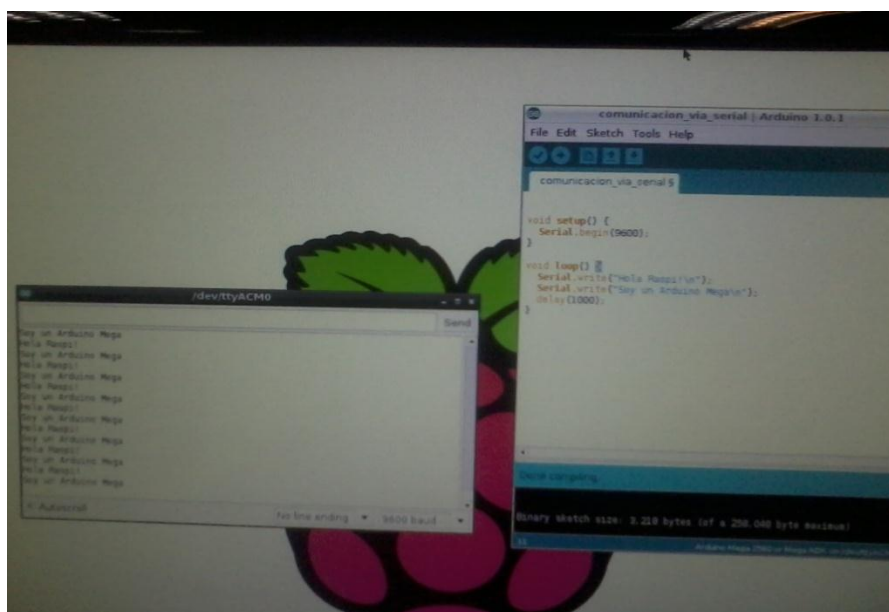


Figura 3.1: Muestra la compilación del Programa Arduino en la ventana /dev/ttyACM0 entre la comunicación de las 2 Tarjetas

### 3.1.2. Diagrama de bloques

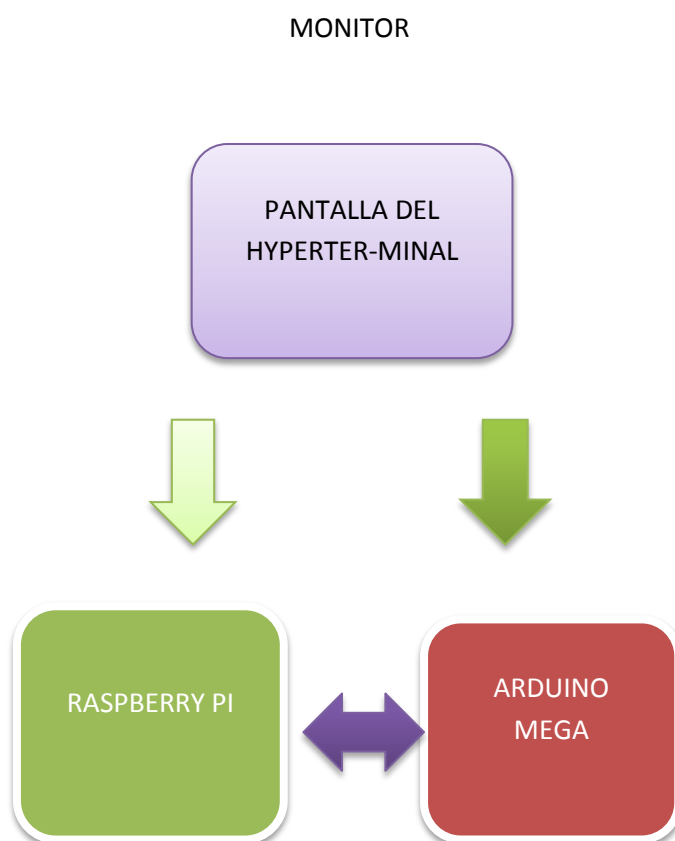


Figura 3.2: Diagrama de bloques del ejercicio 1

### 3.1.3. Diagrama de flujo

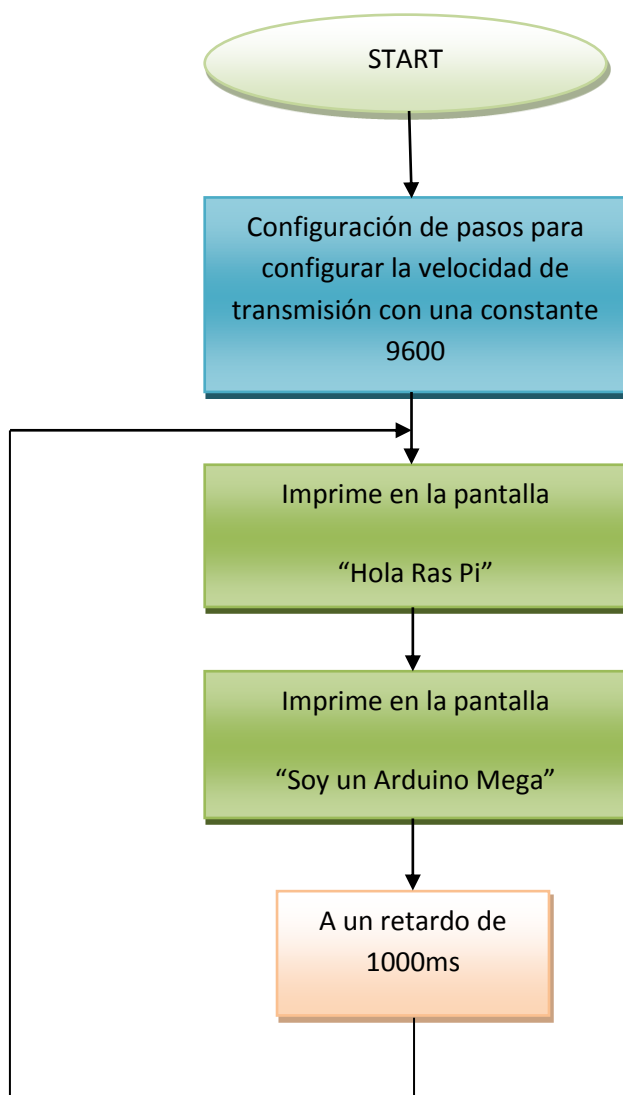


Figura 3.3: Diagrama de Flujo ejercicio 1



### 3.1.4. Descripción del algoritmo

1. Se inicia la función ***void setup***
2. Se inicializa `Serial.begin(9600)`
3. Se inicia la función de lazo `voidloop`
4. Imprime mediante estas dos comandos del código `Serial.write` la comunicación entre la Raspberry y el Arduino  
  
"Hola Raspi"  
  
"Soy un Arduino Mega"
5. Se ejecuta un `Delay` obligatorio después de escribir.

### 3.1.5. Código fuente

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.write("HolaRaspi!\n");  
  Serial.write("Soy un Arduino Mega\n");  
  delay(1000);  
}
```

## 3.2. Ejercicio 2:

### Efecto interactivo LED

#### 3.2.1.Introducción

En este ejercicio se usará una serie de LEDs (10 en total) para hacer un efecto de persecución LED, este proyecto introducirá el concepto de las matrices.

#### Piezas necesarias

10 LEDs rojos de 5 mm



10 Resistencias 3.3k



En primer lugar, nos debemos asegurar de que la placa Arduino este apagada desconectando el cable USB. Ahora, se usa la placa, LED, resistencias, y cables para armar el circuito. Revise su circuito completamente antes de conectar de nuevo la alimentación a la placa Arduino.

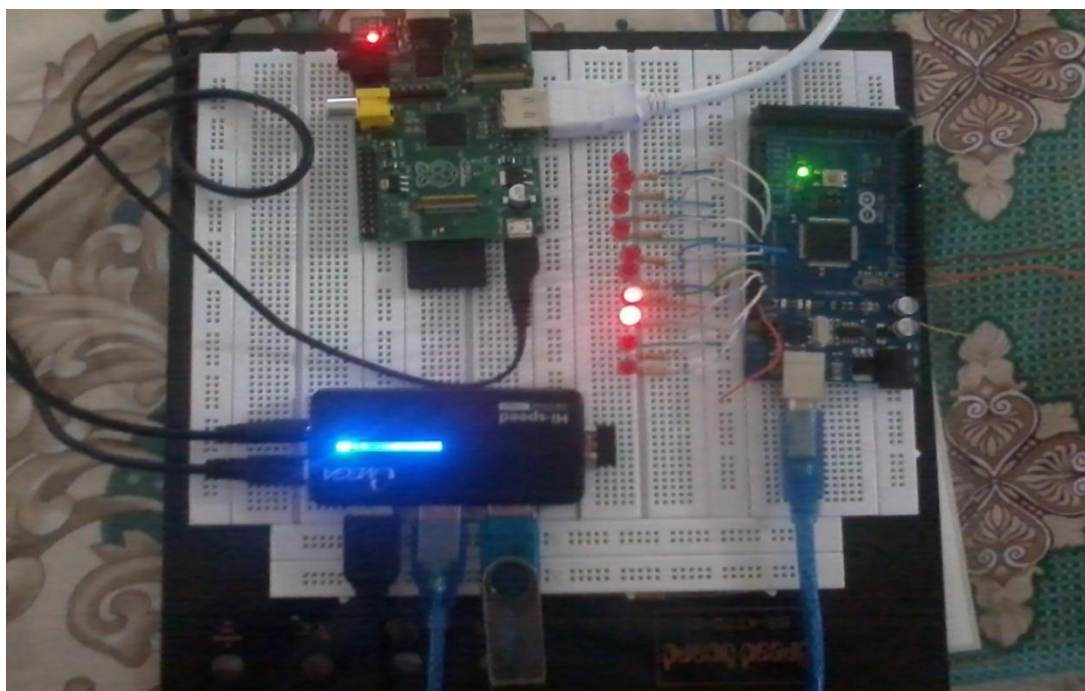


Figura 3.4: Conexión de Interfaz de efecto interactivo de LED'S

### 3.2.2. Diagrama de bloque

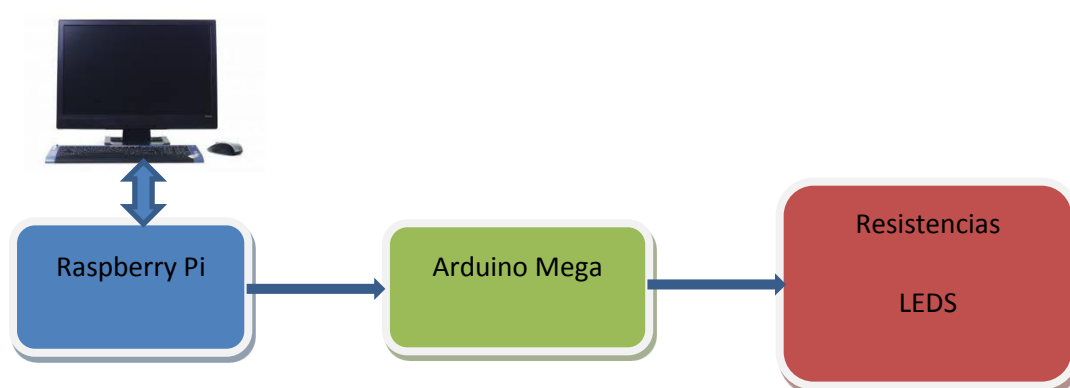


Figura 3.5 Diagrama de bloques ejercicio 2

## 3.2.3. Diagrama de flujo

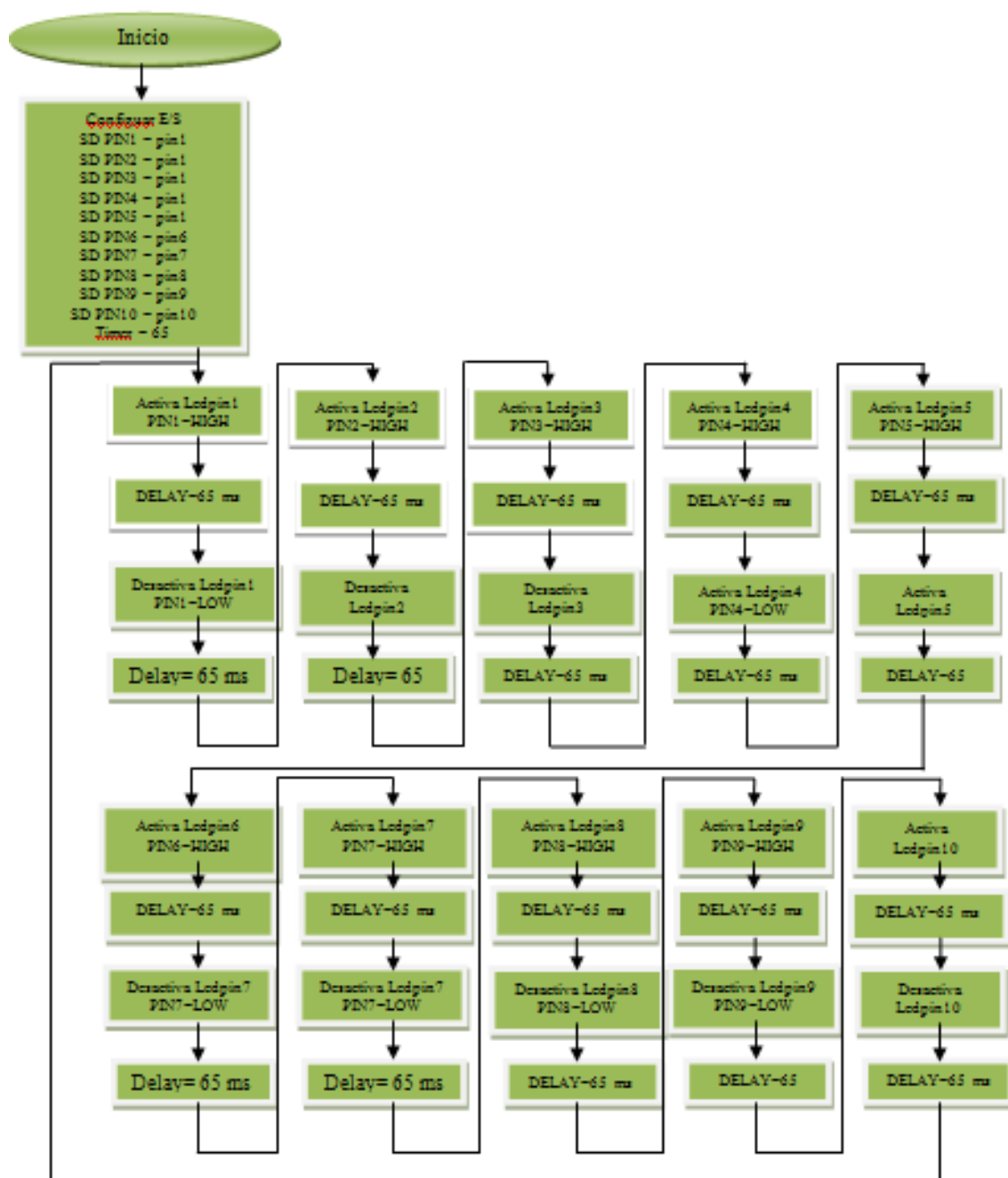


Figura 3.6 diagrama de flujo ejercicio 2

### 3.2.4. Descripción del algoritmo

1. Se declaran las variables la matriz LedPin, las variables, las constantes, etc.
2. Inicializamos la matriz LedPin con valores del 1 al 10.
3. Se inicia la función voidsetup()
4. Se realiza un for para declarar los puertos del arduino como puertos de salida.
5. Se desactiva s todo los LEDS mediante una función for.
6. Si se llega al final de los LEDS se cambia de dirección, se ubica en la primera posición a la vez.

### 3.2.5. Código fuente

```
byteledPin[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; // Crear un arreglo para los LEDS

intledDelay(65); // retardo entre los cambios

int direction = 1;

intcurrentLED = 0;

unsigned long changeTime;

voidsetup()

{
```

```

    for (int x=0; x<10; x++) { // configurar todos los pines como pines de
    salida

    pinMode(ledPin[x], OUTPUT); }

    changeTime = millis();

}

voidloop()

{

    if ((millis() - changeTime) >ledDelay) // si se hallevadomsRetardodesde el
    últimocambio

    {

        changeLED();

        changeTime = millis();

    }}

voidchangeLED ()

{

    for (int x=0; x<10; x++) { //Desactivar todos los LED

        digitalWrite(ledPin[x], LOW);}

    digitalWrite(ledPin[currentLED], HIGH); // encienda eILEDactual

    currentLED += direction; // Incrementaelvalor de la dirección

    // cambiar de dirección, si se llega al final

    if (currentLED == 9) {direction = -1;}

    if (currentLED == 0) {direction = 1;}}

```

### 3.3. Ejercicio 3:

#### Básico control de motores paso a paso

##### 3.3.1. Introducción

En este proyecto muy sencillo, se conectará un motor de paso a paso y luego se obtendrá el Arduino para controlar en diferentes direcciones y diferentes velocidades para un número determinado de pasos.

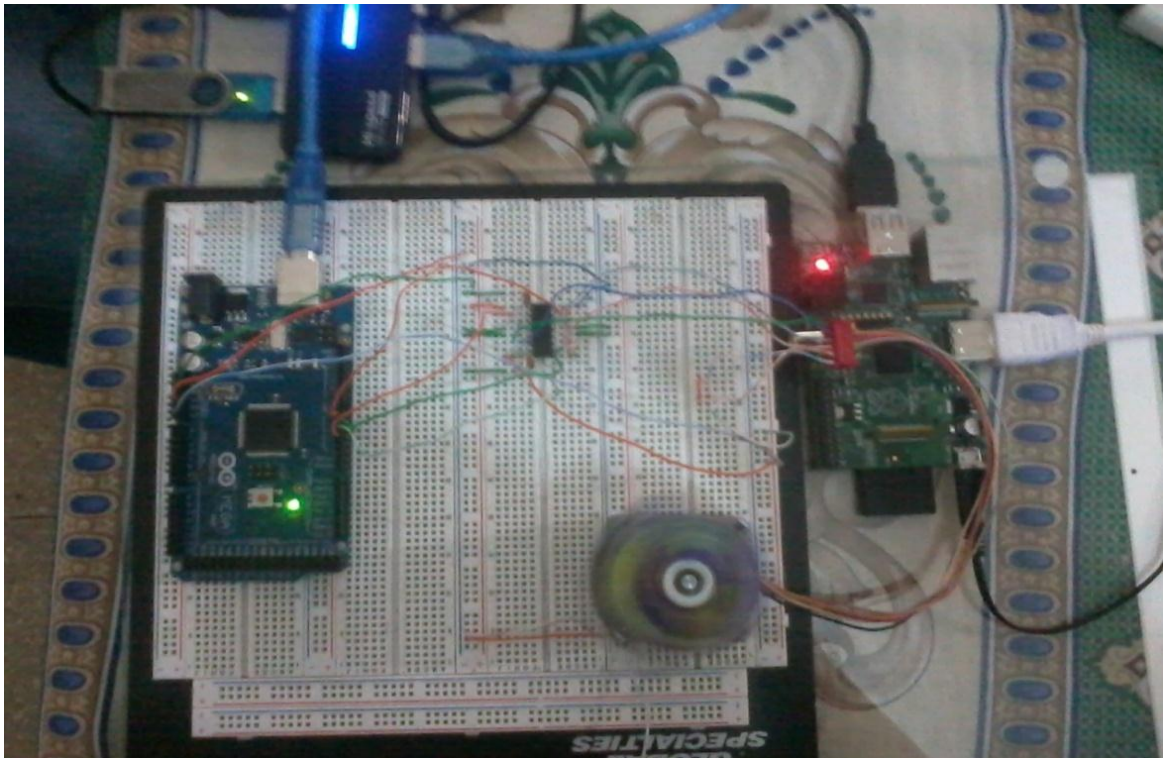


Figura 3.7: conexión de Motores con el Driver L293

### 3.3.2. Diagrama de bloques

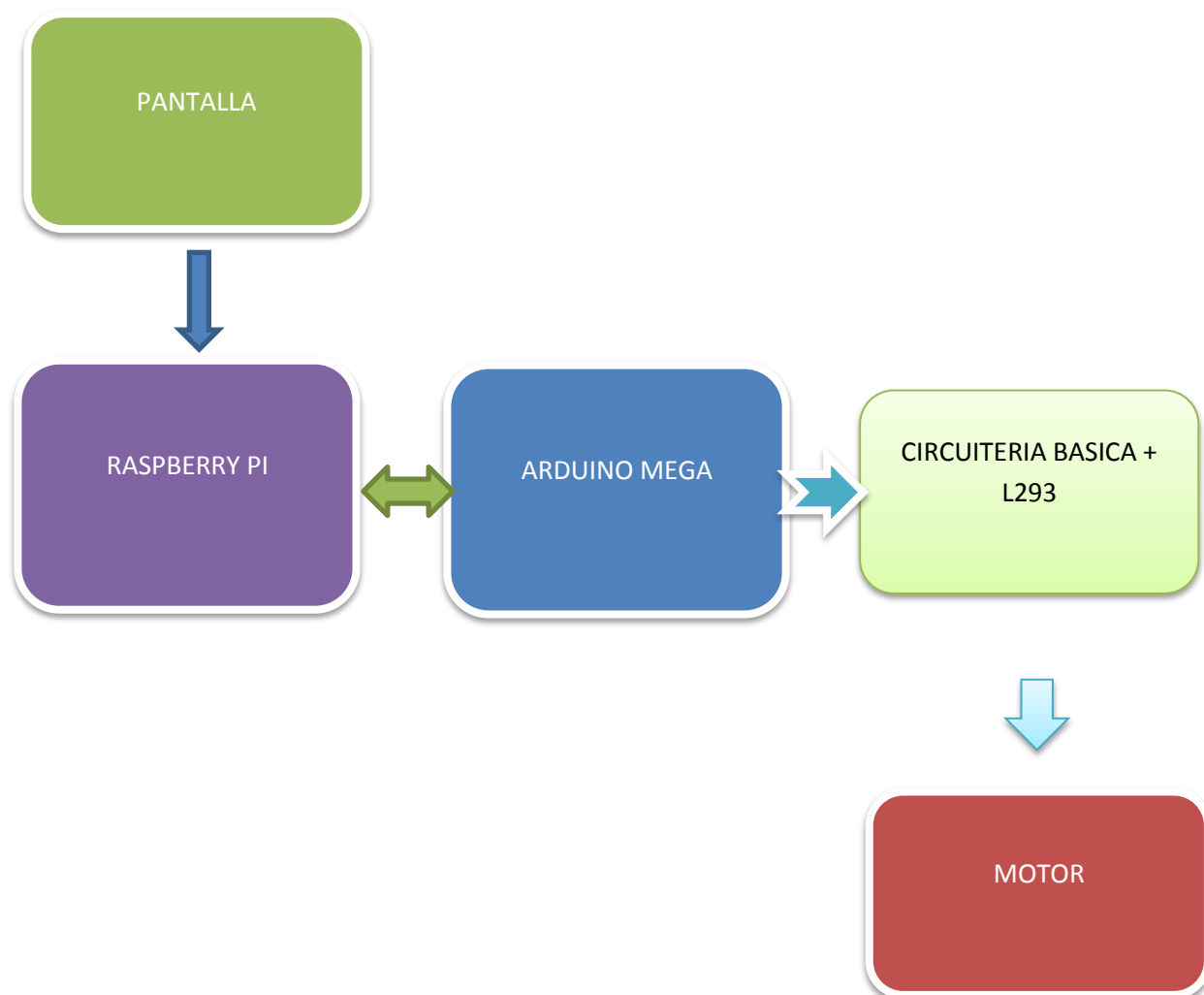


Figura 3.8: Diagrama de Bloques ejercicio 3



### 3.3.3. Diagrama de Flujo

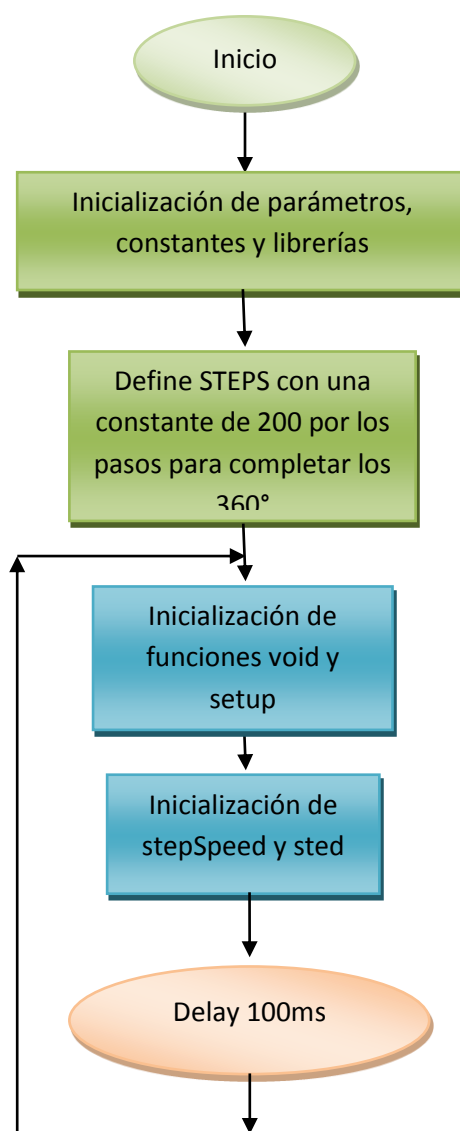


Figura 3.9: Diagrama de Flujo ejercicio 3

### 3.3.4. Descripción del algoritmo

1. Se declaran librerías como stepper.h, variables y constantes para iniciar la programación y pines de elegir en el Arduino
2. Define STEP 200, ya que le motor es de 1.8° por paso para que complete los 360° debe hacer 200 pasos
3. Declaración de pines mediante la función de la librería stepper.
4. Inicialización de la función void setup y void loop.
5. Dentro de la librería stepper.h ahí dos funciones setSpeed y step donde la una hace la cantidad de rpm es decir la velocidad del motor en revoluciones por minuto y la otra función la cantidad de pasos que hará el motor hasta completar los grados de giro requeridos, cuando la cantidad de pasos es positiva gira para un sentido y cantidad de pasos negativos hace que gire el motor lado contrario
6. Un delay de retardo para poder visualizar el cambio de giro al lado contrario

### 3.3.5. Código fuente

```
// Project 3

#include <Stepper.h>

// steps value is 360 / degree angle of motor

#define STEPS 200

// create a stepper object on pins 4, 5, 6 and 7

Stepper stepper (STEPS, 4, 5, 6, 7);

void setup ()

{

}

void loop()

{

stepper.setSpeed(60);

stepper.step(200);

delay(100);

stepper.setSpeed(20);

stepper.step(-50);

delay (100);

}
```

### 3.4. Ejercicio 4:

#### Interfaz final de Arduino con Raspberry

#### Control de motores usando las tarjetas RAMPS 1.4 Y Pololu + Arduino

##### 3.4.1. Introducción

En este ejercicio es parecido al anterior con la diferencia que en vez de usar el L293, se usa con la tarjeta RAMPS 1.4 y los driver Pololu, ejecutando en el Programa Arduino el SPRINTER, para poder manipular los motores mediante el programa instalado en la Raspberry pi el "Pronterface", observando los giros del motor en ambos sentidos.

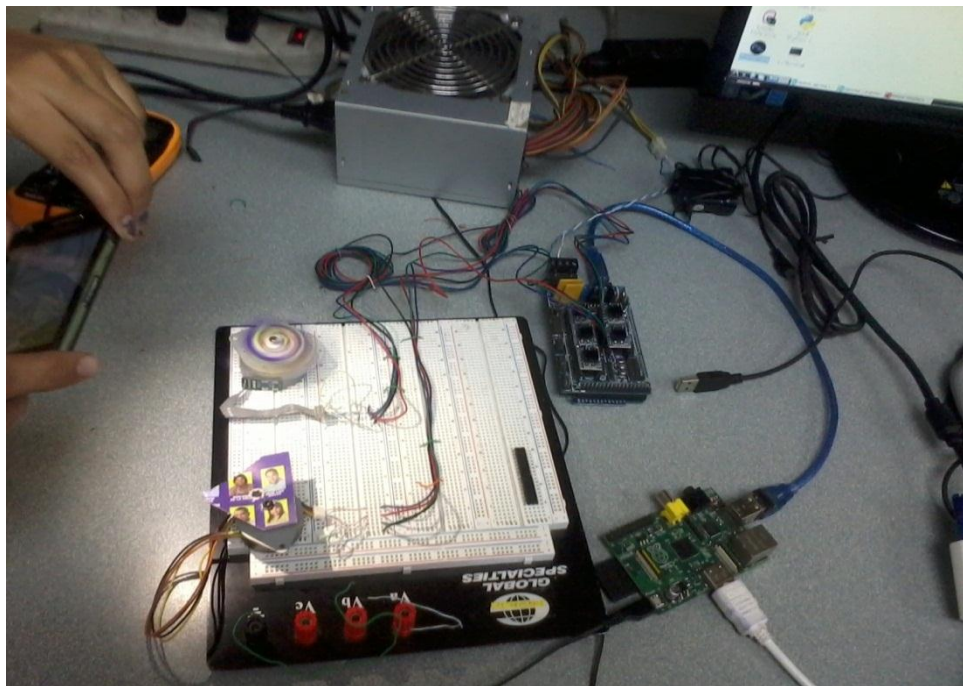


Figura 3.10: conexión con la tarjetas Ramps 1.4 y Pololu

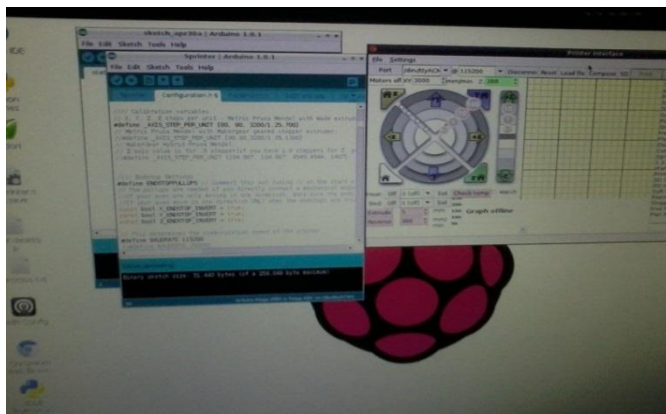


Figura 3.11: Pantalla de compilación del Sprinter y uso del Pronters

### 3.4.2. Diagrama de bloques

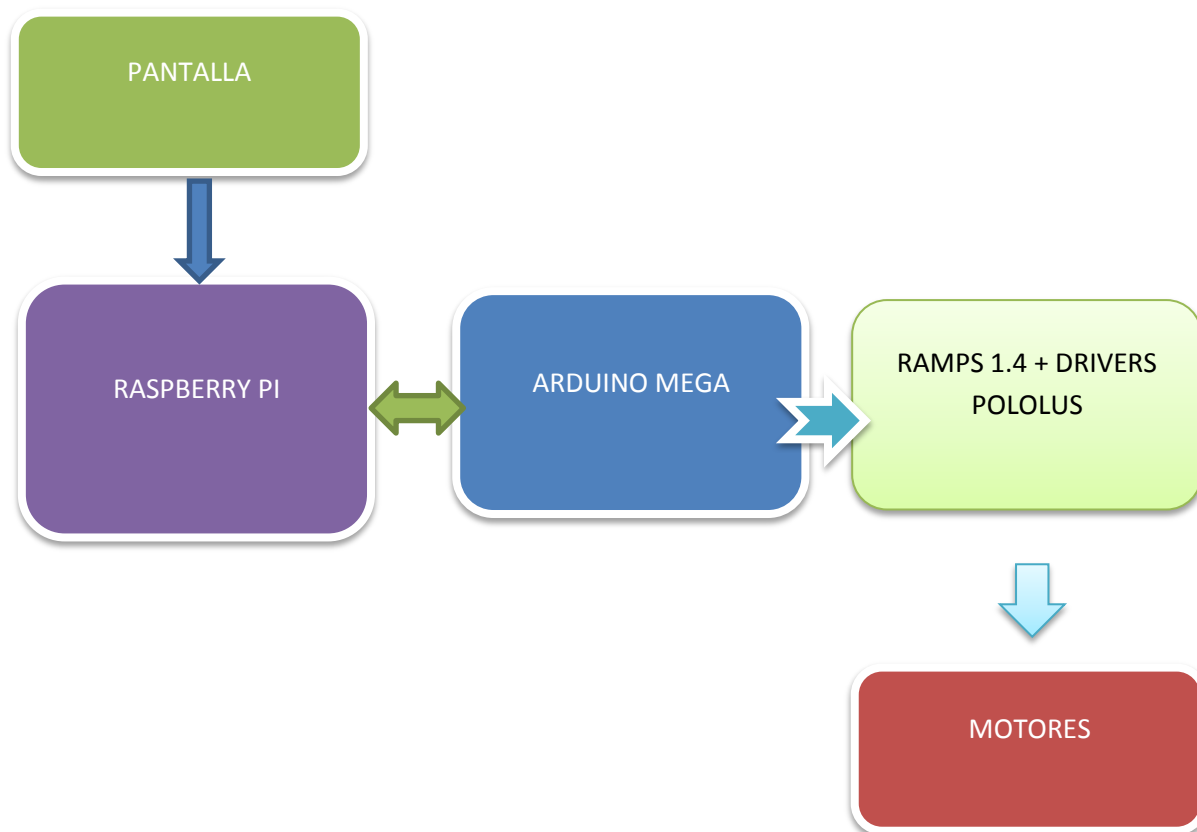


Figura 3.12 : Diagrama de Bloques ejercicio 4

### 3.4.3. Diagrama de Flujo

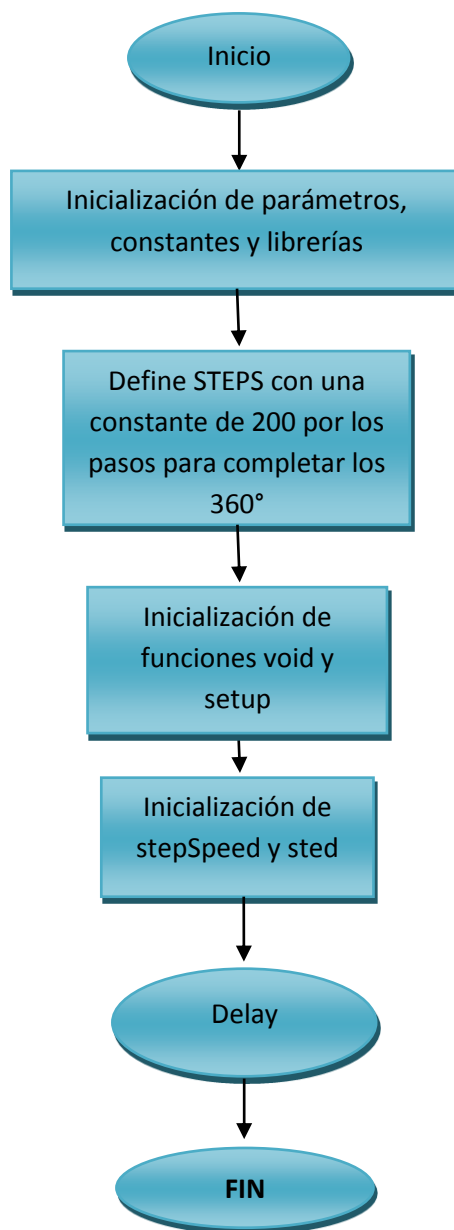


Figura 3.13: Diagrama de Flujo ejercicio

### 3.4.4. Descripción del algoritmo

1. Se declaran librerías, variables y constantes para iniciar la programación de pines de elegir en el Arduino y Ramps
2. Selección de Ramps **#define MOTHERBOARD 33, debido a que estamos usando una Ramps superior a 1.2**
3. Luego activamos la inversión de los motores para que puedan girar en ambos sentidos para lograr el objetivo de la impresión.
4. Activación de los ejes x, y, z para el giro de motores
5. Configuración de los movimientos de los motores y calibración

### 3.4.5. Código fuente

```
#ifndef CONFIGURATION_H

#define CONFIGURATION_H

//// The following define selects which electronics board you have. Please
choose the one that matches your setup

// MEGA/RAMPS up to 1.2 = 3,

// RAMPS 1.3/1.4 = 33

// Gen6 = 5,

// Gen6 deluxe = 51

// Sanguinololu up to 1.1 = 6
```

```
// Sanguinololu 1.2 and above = 62

// Gen 7 @ 16MHZ only= 7

// Gen 7 @ 20MHZ only= 71

// Teensylu (at90usb) = 8

// Printrboard Rev. B (ATMEGA90USB1286) = 9

// Gen 3 Plus = 21

// gen 3 Monolithic Electronics = 22

// Gen3 PLUS for TechZone Gen3 Remix Motherboard = 23

#define MOTHERBOARD 33

//// Calibration variables

// X, Y, Z, E steps per unit - Metric Prusa Mendel with Wade extruder:

#define _AXIS_STEP_PER_UNIT {80, 80, 3200/1.25,700}

// Metric Prusa Mendel with Makergear geared stepper extruder:

//#define _AXIS_STEP_PER_UNIT {80,80,3200/1.25,1380}

// MakerGear Hybrid Prusa Mendel:

// Z axis value is for .9 stepper(if you have 1.8 steppers for Z, you need to
use 2272.7272)

//#define _AXIS_STEP_PER_UNIT {104.987, 104.987, 4545.4544, 1487}

//// Endstop Settings

#define ENDSTOPPULLUPS // Comment this out (using // at the start of the
line) to disable the endstoppullup resistors
```



**// Thepullups are needed if you directly connect a mechanical endswitch between the signal and ground pins.**

**//If your axes are only moving in one direction, make sure the endstops are connected properly.**

**//If your axes move in one direction ONLY when the endstops are triggered, set [XYZ]\_ENDSTOP\_INVERT to true here:**

**constbool X\_ENDSTOP\_INVERT = true;**

**constbool Y\_ENDSTOP\_INVERT = true;**

**constbool Z\_ENDSTOP\_INVERT = true;**

**// This determines the communication speed of the printer**

**#define BAUDRATE 115200**

**##define BAUDRATE 250000**

**// Comment out (using // at the start of the line) to disable SD support:**

**#define SDSUPPORT**

**// Uncomment to make run init.g from SD on boot**

**##define SDINITFILE**

**//Only work with Atmega1284 you need +1 kb ram**

**##define SD\_FAST\_XFER\_AKTIV**

```
##define USE_EEPROM_SETTINGS

// to disable EEPROM Serial responses and decrease program space by
~1000 byte: comment this out:

// please keep turned on if you can.

##define PRINT_EEPROM_SETTING

//Uncomment to activate the arc (circle) function (G2/G3 Command)

//Without SD function an ARC function the used Flash is smaller 31 kb

#define USE_ARC_FUNCTION

#ifdef SDSUPPORT

    #ifdef SD_FAST_XFER_AKTIV

        //Fast transfer chunk size (> 1024 is unstable, change at your own
risk).

        #define SD_FAST_XFER_CHUNK_SIZE 1024

    #endif

#endif

// For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting
(Active High) use 1

#define X_ENABLE_ON 0

#define Y_ENABLE_ON 0

#define Z_ENABLE_ON 0
```

```
#define E_ENABLE_ON 0

##define DELAY_ENABLE 15

// Disables axis when it's not being used.

constbool DISABLE_X = false;

constbool DISABLE_Y = false;

constbool DISABLE_Z = true;

constbool DISABLE_E = false;

// Inverting axis direction

constbool INVERT_X_DIR = false;

constbool INVERT_Y_DIR = false;

constbool INVERT_Z_DIR = true;

constbool INVERT_E_DIR = false;

//// ENDSTOP SETTINGS:

// Sets direction of endstops when homing; 1=MAX, -1=MIN

#define X_HOME_DIR -1

#define Y_HOME_DIR -1

#define Z_HOME_DIR -1

#endif
```

## **CAPÍTULO 4**

### **4. Pruebas y Simulación del Proyecto**

#### **4.1. Introducción**

En el capítulo 4, se explicará la realización del proyecto y los elementos utilizados. Además se mostrarán imágenes de la implementación de los ejercicios realizados, anteriormente en el capítulo 3

#### **4.2. Elementos físicos**

##### **4.2.1. Plataforma del Proyecto**

El proyecto realizado está enfocado a realizarse en etapa de prototipo, es decir, que no se implementará en un circuito impreso. Para la implementación del prototipo se ha utilizado los siguientes elementos:

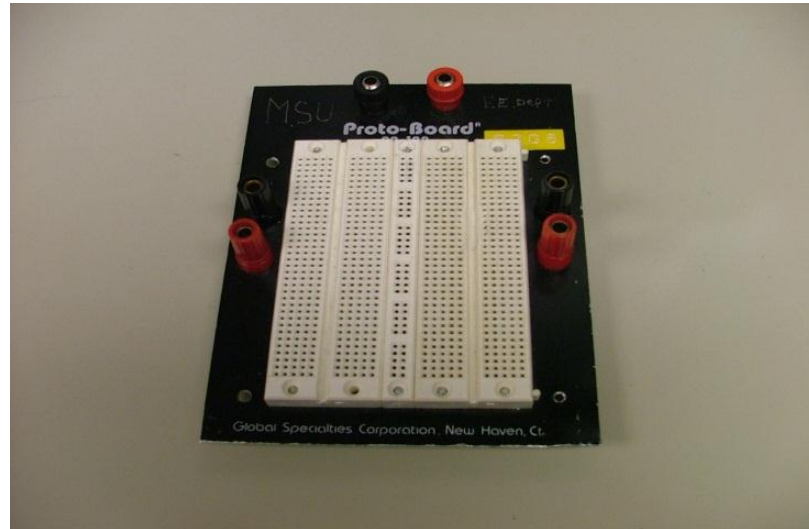
#### **4.2.1.1. Protoboard**

El Protoboard es un tablero con orificios conectados eléctricamente entre sí, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipos de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos. Normalmente se utilizan para la realización de pruebas experimentales, como en nuestro caso.

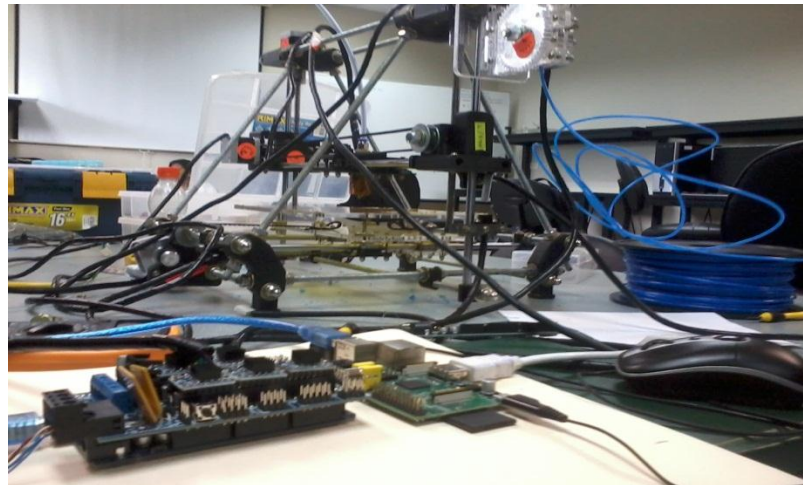
Las líneas se cortan en la parte central del bloque de plástico para garantizar que dispositivos, en circuitos integrados tipo DIP (Dual Inline Packages), puedan ser insertados perpendicularmente a las líneas de conductores.

Los demás componentes electrónicos pueden ser montados sobre perforaciones adyacentes que no compartan la tira o línea conductora e interconectados a otros dispositivos usando cables, usualmente unifilares. Uniendo dos o más protoboard es posible ensamblar complejos prototipos electrónicos que cuenten con decenas o cientos de componentes.

#### **[25]Placas de prueba**



**Figura 4.1: Protoboard [26]**



**Figura 4.2: Proyecto final.**

#### **4.2.1.2. Raspberry PI**

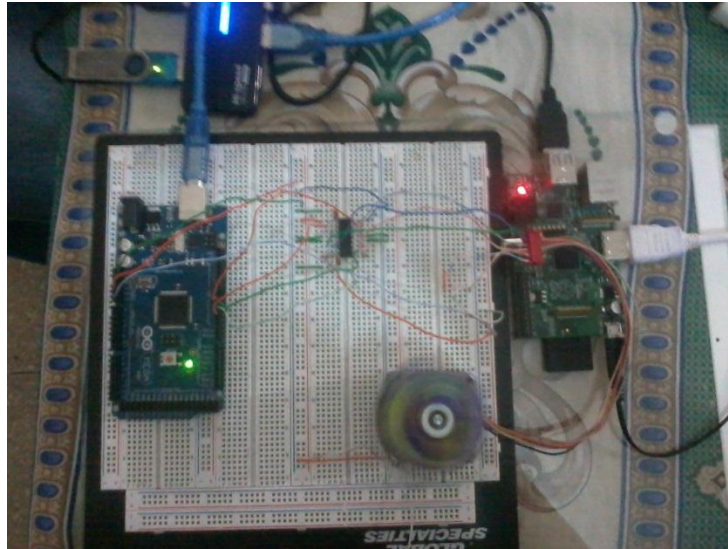
Para los que no conocen la "RaspPI" o "RPi", como le dicen, es una pequeña computadora completamente funcional del tamaño de una tarjeta de crédito, la cual se puede utilizar conectándole un teclado, un mouse y una TV.

Las RaspPI's son fabricadas por una "ONG" británica y su objetivo es promover un poco la informática a través de una pequeña computadora, barata y fácil de usar.

Muy fuera del contexto original, miles de personas alrededor del mundo han adquirido su RasPI para un sin fin de tareas, desde centros de entretenimiento hasta el control de robots.

En nuestro caso la adquirimos para lograr nuestro proyecto, haciendo una interfaz de conexión entre la Raspi y el Arduino Mega para el control de la impresora 3D del proyecto RepRap

**[27]Conociendo la Raspberry Pi**



**Figura 4.3: Raspberry PI implementado en ejercicios y proyecto conectado al Protoboard**

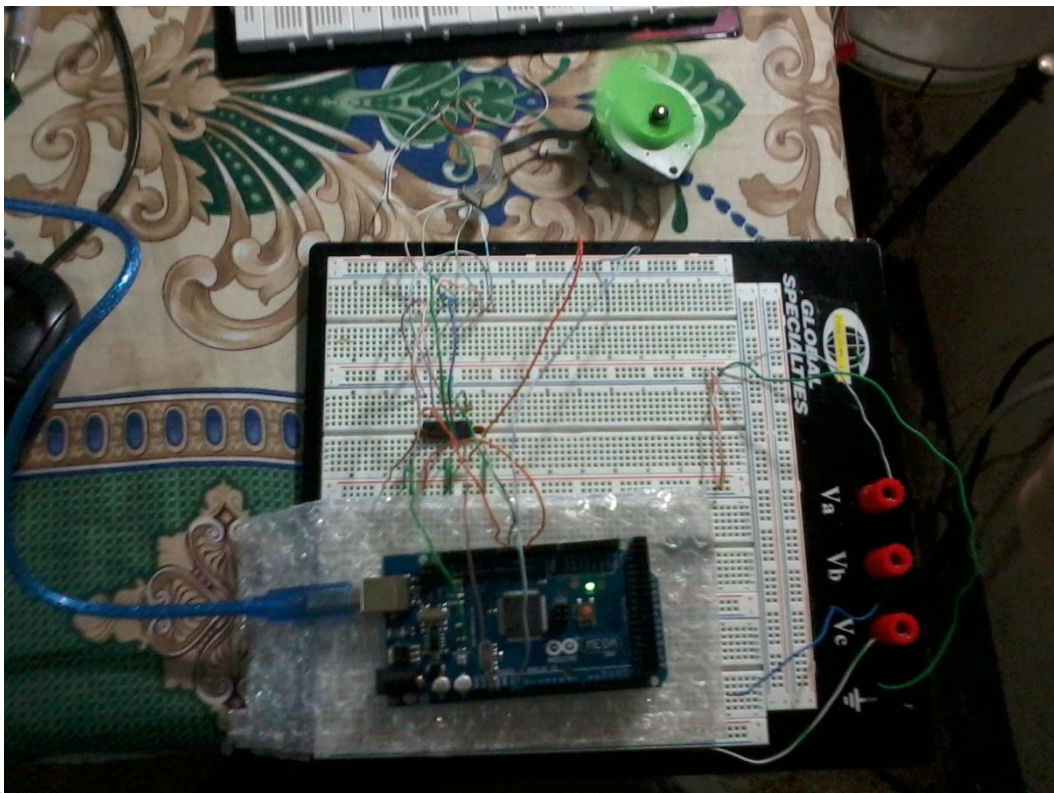
#### **4.2.1.3. Arduino Mega**

Por otra parte tenemos a los Arduinos, estos dispositivos son pequeñas tabletas para el montaje de prototipos electrónicos que incluyen un pequeño microprocesador que puede programarse muy fácilmente.

Realmente desde hace muchos años que se pueden hacer cosas como las que se hacen actualmente con los Arduinos. Sin embargo, la ventaja de estas pequeñas tabletitas es que ya incluyen todo lo necesario para



comenzar proyectos. Por una parte, no es necesario armar circuitería extra, solo se conectan a una fuente de poder y están funcionando. El otro punto a favor es incluyen un pequeño software que hace programar los Arduinos una "delicia". Simplemente uno escribe su programa y le da click al botón enviar al dispositivo y listo para hacer lo que deseemos. **[28] Conociendo Arduino**

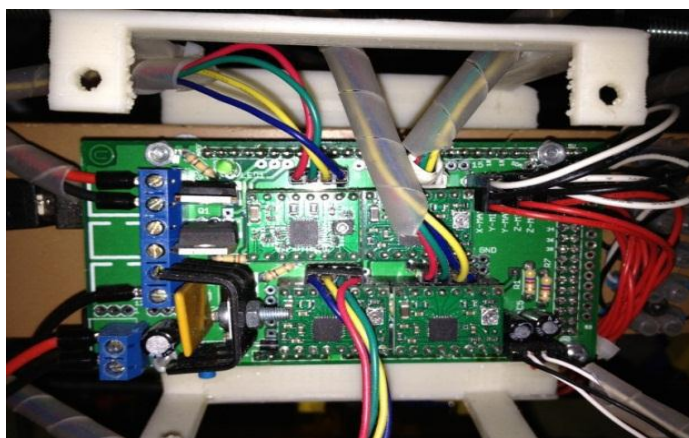


**Figura 4.4: Arduino Mega implementada en el Protoboard con los motores de paso.**

#### 4.2.1.4. Kit de Driver de Motores (Pololus) y RAMPS 1.4

Se describirán a través de sencillos esquemas y explicaciones de los mismos, cómo se conectan todos los dispositivos de la impresora a la placa electrónica de la misma.

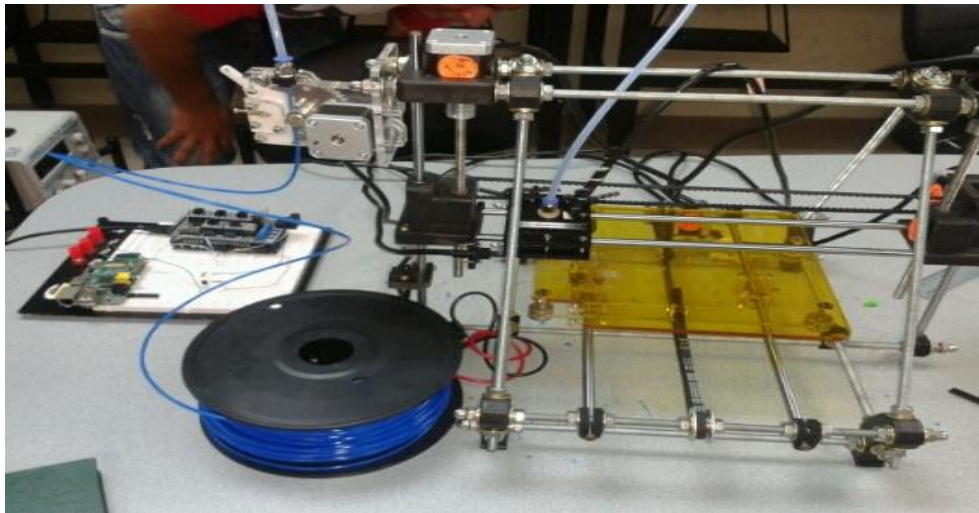
La placa Arduino se adapta a las Ramps, donde tenemos cuatro drivers para motores, cuatro Pololus, al lado de cada uno se encuentran cuatro pines donde se conectan los cuatro cables que salen de cada motor.



**Figura 4.5: Conexión de Drivers**

Los cuatro drivers nos permiten controlar el movimiento del motor del X, el del eje Y, los dos del eje Z y el del extrusor, que es el que impulsa el plástico para que sea extruido. Los motores del eje Z tienen que ir, a una clema, y de esta vamos ya a la placa electrónica. La conexión es la realizada en la clema situada aproximadamente en el centro de una de las varillas de la parte superior de la impresora que son las que se ven en la imagen, en la que se

ha puesto la clema, sujeta con una brida, en una de las varillas roscadas de la parte superior de la impresora. **[29] Pololus**



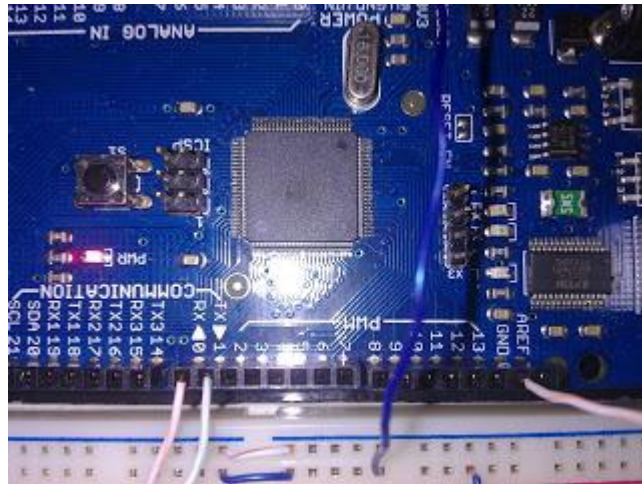
**Figura 4.6 Pololus + Arduino + RAMPS 1.4 en la impresora 3D**

### **4.3. Imágenes y simulaciones**

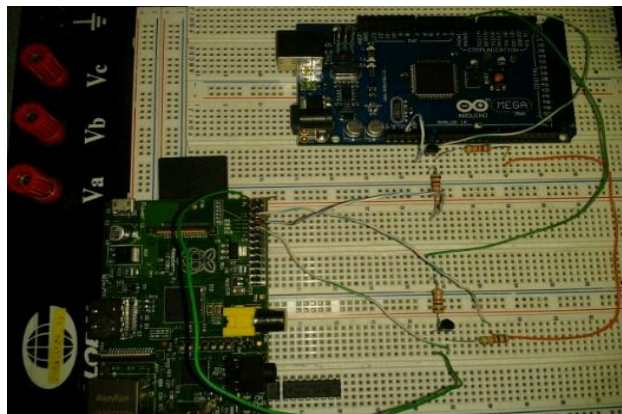
#### **4.3.1. Interfaz de Comunicación de Arduino con Raspberry Pi**

**El puerto GPIO de la Raspberry PI** Las Raspberry incluyen una serie de pines de propósito general que pueden ser utilizados como entradas o salidas digitales entre los cuales se incluye un pequeño puerto "serial". Este puerto puede ser utilizado para enviar o recibir datos desde y hacia otros dispositivos. De fábrica viene configurado como un puerto de "consola" para monitorear el funcionamiento del RasPi. Utilizamos un Protoboard para

realizar la conexión de los cables y la circuitería de para equilibrar los niveles, en la siguiente imagen pueden ver las líneas que vienen de la Protoboard al puerto serial del Arduino Mega: **[30] Conectando Arduino y Raspberry Pi**

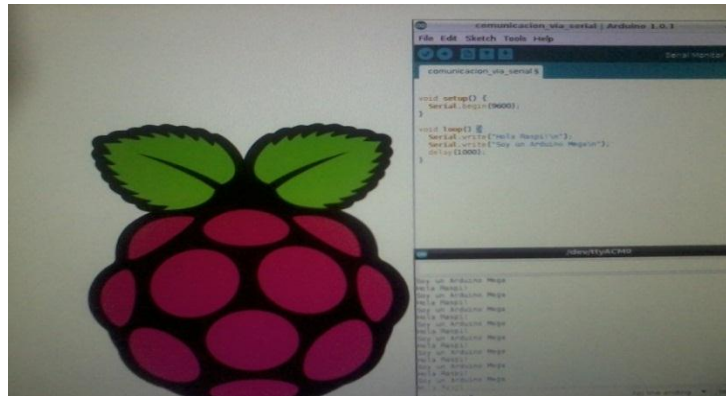


**Figura 4.7: Conexiones de tarjeta Arduino con la GPIO del Raspberry Pi**



**Figura 4.8: Circuito para igualación de voltajes en los puertos GPIO del Raspberry Pi**

Si la conexión mediante puerto GPIO es correcta y más el código cargado en el Arduino y la configuración de los puertos, deberá mostrarse una pantalla como la siguiente:



**Figura 4.9: Pantalla que muestra la conexión entre ambas tarjetas**

#### 4.3.1.1. Conclusiones

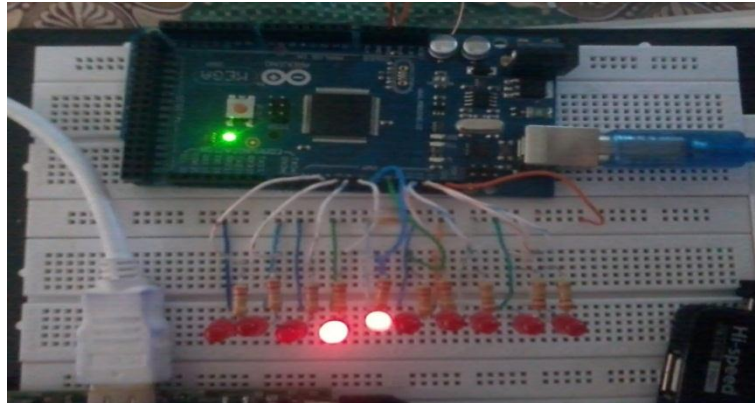
- En este ejercicio se ha logrado con éxito la interfaz de comunicación entre ambas tarjetas ya como habíamos mencionado ambos eran mundos distintos tanto la Raspberry Pi como la Tarjeta Arduino, todo se logró gracias a la configuración de los puertos GPIO y una circuitería sencilla para igualar los voltajes.
- También se demuestra que todos los bloques de un sistema electrónico se pueden acoplar entre sí, a la vez recordando y respetando las indicaciones de su fabricante, para no dañarlas.

- La Raspberry Pi reemplazó prácticamente la Pc común haciendo más fácil trasladar nuestro proyecto a cualquier parte, tan solo conectándole un mouse, teclado y un monitor o televisor y ya se puede usar de una manera sencilla, aunque su forma de configuración es muy distinta ya que trabaja con el sistema operativo de Linux, que es otro entorno, ya que para instalar un programa no era de la manera usual, buscar en internet, descargarlo e instalarlos, no en cambio en la Raspberry Pi para instalar un programa se debía usar códigos de comando en el programador de Python del LX terminal.

#### **4.3.2. Ejercicio 2: Interfaz de Arduino con Raspberry Pi para un Efecto interactivo LED**

Este ejercicio 2 Hay que conectar 11 LED-s a los pines digitales de la placa a través de resistencias de 220 Ohmios tal y como se muestra en la imagen superior.

El programa de Arduino comienza encendiendo LED-s hasta que llegue al número de LED-s establecido. En ese momento se seguirá encendiendo LED-s hacia la izquierda (si se monta tal y como se muestra en la fotografía inferior), para mantener el movimiento, al mismo tiempo que apaga LED-s por la derecha. De otra forma seguiría encendiendo LED-s hasta encenderlos todos.



**Figura 4.10: Efecto interactivo de Led's**

#### **4.3.2.1. Conclusiones**

- En este ejercicio podemos darnos cuenta, que con el Arduino se pudo hacer cualquier conexión de circuitería y programarlo para nuestro objetivo, en este caso se usa como un ejercicio de practica en encender Led's y apagarlos, como una especie de estrella fugaz en la línea de Led's.
- Antes de conectar no olvidar las recomendaciones de las Tarjetas, es decir la cantidad de Voltaje y Corriente que soportan las Raspberry y Arduino.
- Observar cuidadosamente las conexiones antes de prender el circuito armado, porque podría ocasionar daños no solo en la parte circuital, sino también en las tarjetas, tratemos de siempre asegurarse verificando las conexiones antes de conectar a la fuente.

### 4.3.3. Ejercicio 3: Básico control de motores paso a paso con los driver L293D

Con esta aplicación se va a mover un motor de cc haciendo uso de un CI de potencia que es específico para estas aplicaciones. El circuito podrá mover hasta dos motores, nosotros solo lo haremos con uno.

En este montaje podremos mover el motor en los dos sentido de giro. El funcionamiento será que vamos a crear una rampa de subida de tensión y una de bajada con el fin de que el motor modifique su velocidad de modo automático.

#### Chip L293D/B(puente H):

Es un circuito integrado o chip, que puede ser utilizado para controlar simultáneamente la velocidad y dirección de dos motores de continua (contiene dos puentes H). La diferencia entre el modelo L393D y L293B, es que el primero viene con diodos de protección que evita los daños producidos por los picos de voltaje que puede producir el motor [31] Manejo de motores y L293D

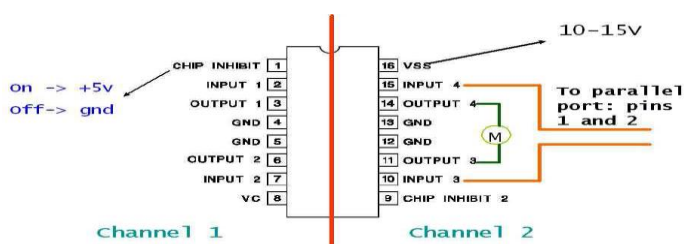


Figura 4.11: Pines del Driver L293D [32] Controlador de motores



Esquemático del circuito que debemos armar en el protoboard con el integrado L293D, usando las salidas PWM del Arduino y los motores paso a paso

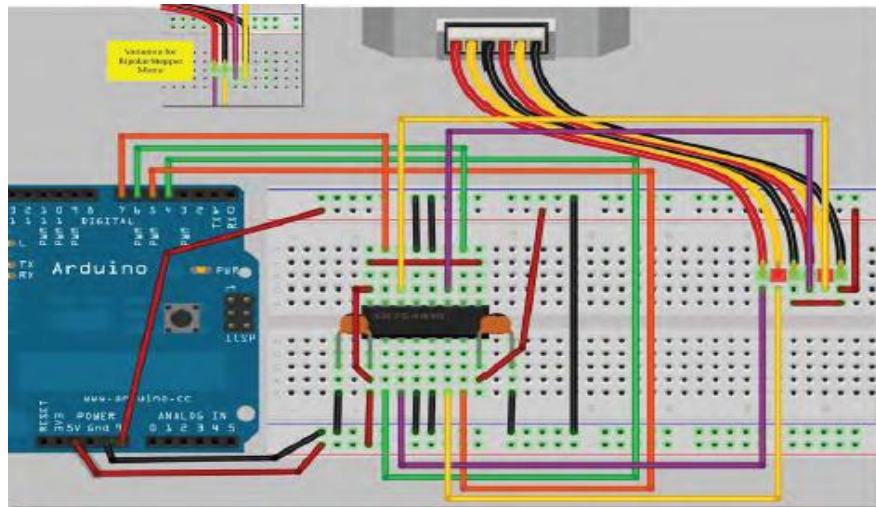


Figura 4.12: Esquemático del circuito para control de motores. [33]

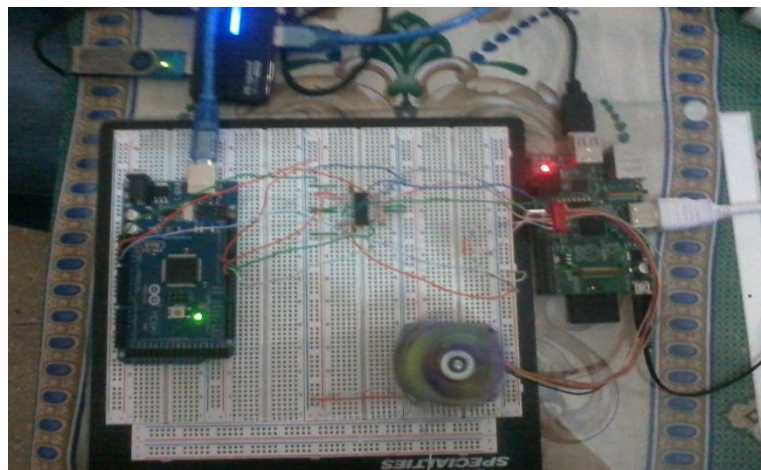


Figura 4.13: circuito prototipo de control de motores con el L293D

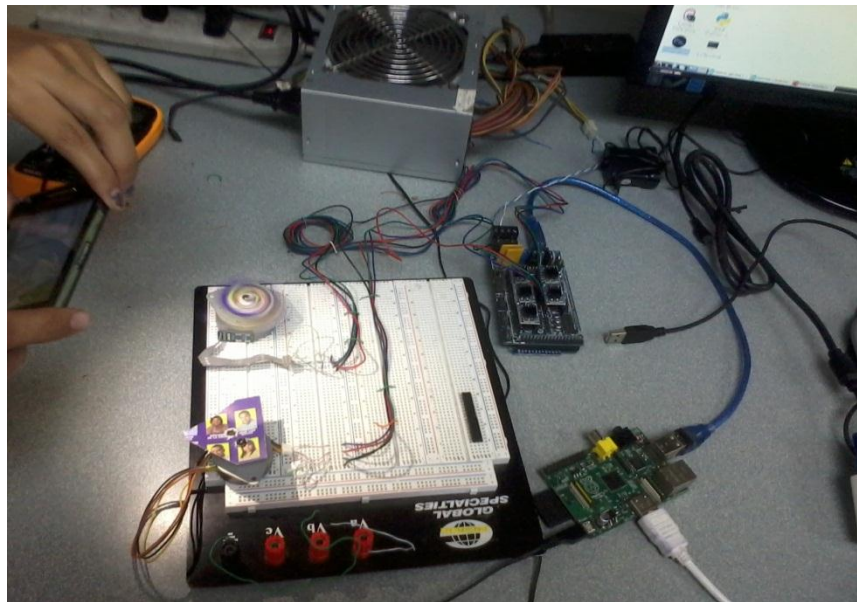
#### **4.3.3.1. Conclusiones**

- Este ejercicio fue implementado para tratar de lograr el objetivo de manipular los motores paso a paso que luego deberemos manipularlos con las tarjetas Ramps y los driver Pololus en el circuito final del proyecto para la impresora 3D.
- Si nos percatamos el circuito es muy básico y el que hace todo el trabajo es la tarjeta Arduino quien se va a programar y el integrado L293D, ellos son los responsables del giro de los motores y de que gire en ambos sentidos.

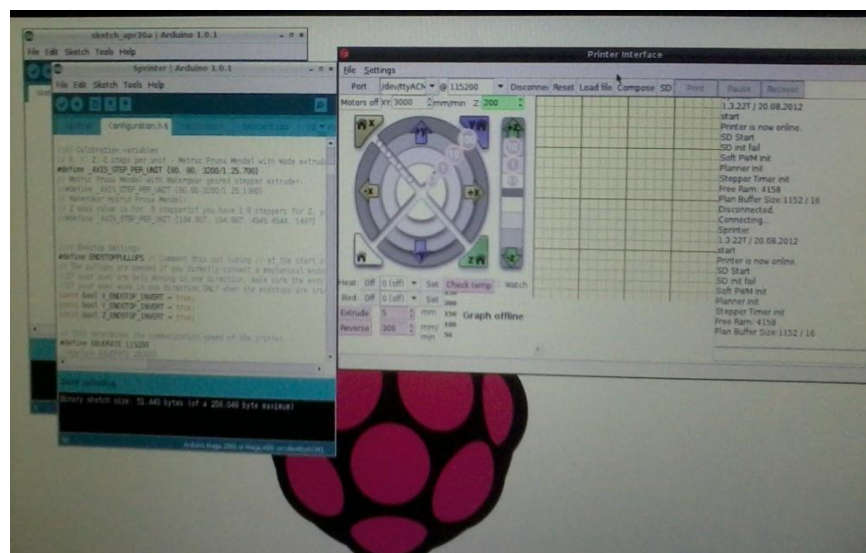
#### **4.3.4. Proyecto de Graduación:**

**Interfaz Final de conexión de tarjetas controlando los motores de la impresora 3D con tarjeta Ramps 1.4 y los driver Pololus para los motores.**

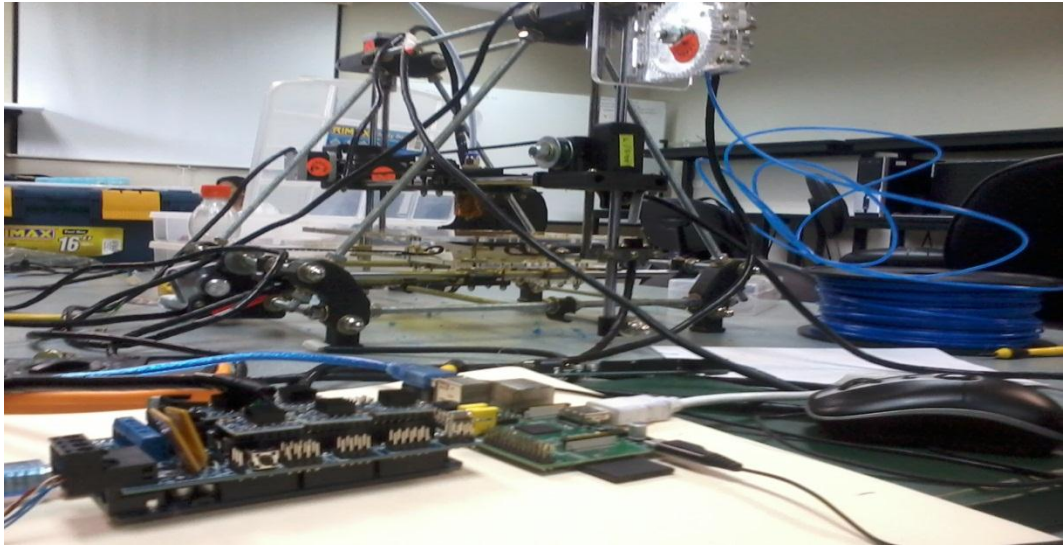
En el proyecto final, acoplamos parte de los ejercicios 1 y 4 para ya poder controlar los motores de la impresora 3D una vez ya acoplando las dos tarjetas Arduino y Raspberry con el uso de los puertos GPIO, a la vez con el uso de la tarjeta Ramps 1.4 y los driver Pololus. Usando el programa Pronterface que lo instalamos en el entorno Raspberry y cargando el programa Sprinter en el Arduino se lograra el objetivo.



**Figura 4.14: Control de los motores con las tarjeta Ramps 1.4 y Driver de motores"**



**Figura 4.15: Usando el Pronterface instalado en el entorno Raspberry Pi y cargando el Sprinter en el Arduino**



**Figura 4.16: Interfaz del proyecto final usando Arduino con Raspberry Pi, acoplado con las Tarjetas Ramps y Pololus en la impresora 3D**

## CONCLUSIONES

1. Con el desarrollo de los ejercicios y la implementación práctica de cada uno de ellos, se logra el objetivo del proyecto, primero acoplando la interfaz entre dos mundos distintos la Raspberry con el Arduino y demostramos que entre sistemas electrónicos siempre habrá un interfaz de conexión entre ambos, solo investigando y siendo perseverante se alcanzó el objetivo deseado con éxito.
2. La programación de los ejercicios nos ayudó a ver cómo interactúan la interfaz entre tarjetas y comprender mejor, especialmente el entorno de las Tarjetas Raspberry Pi, que usan el sistema operativo de Linux pero gracias a los conocimientos básicos de programación se logró con éxito la configuración sin ningún problema, también la programación en el Arduino fue sencilla ya que era programación en C en micro-controladores, las cuales ya se había adquirido en cursos anteriores.

3. El módulo de control de motores procesa instrucciones para el movimiento de los motores en ambos sentidos mediante la programación en el Arduino IDE y las tarjetas Ramps y los drivers Pololus se facilitó el uso con el programa de la impresora 3D Pronterface, para así controlar todo nuestro proyecto.

## RECOMENDACIONES

1. Se debe tener mucho cuidado al manipular las Tarjetas de trabajo, ya que cualquier contacto innecesario con los componentes electrónicos podría causar que se dañen por estática, dado que los elementos están diseñados para trabajar a corrientes muy bajas.
2. Cuando conectamos las Raspberry Pi con el Arduino Mega hacer bien la configuración y el circuito para igualar los voltajes y evitarnos algún daño de las tarjetas.
3. Para evitar confusiones al colocar los cables en los puertos correctos después de una desconexión accidental, se deben colocar etiquetas en el cableado, por sobre todo en los que comunican pines de comunicación y

de voltaje de alimentación, dado que por posible error se podrían dañar seriamente los elementos de las tarjetas de trabajo.

4. Verificar los programas sean bien compilados en el Arduino con el respectivo circuito de prueba que se esté simulando, porque podría causar un confusión en la parte circuital de lo que se esta armando como practica y podría afectar daños a las tarjetas.

## BIBLIOGRAFÍA

[1] Modelo 3D de un cráneo hecho en impresora 3D, <http://www.impresoras yescaneres3d.com/wp-content/uploads/2011/10/TMD1011pg52-11.jpg>,

Fecha de consulta: 16 de abril del 2013

[2] Impresora 3D Prusa Mendel, [http://www.triquitraqueando.com/tiendita/RepRap Prusa Mendel](http://www.triquitraqueando.com/tiendita/RepRap_Prusa_Mendel), Fecha de consulta: 17 de abril 2013

[3] impresora 3D, [http://corsaje902-2013.blogspot.com/2013\\_07\\_01\\_archive.html](http://corsaje902-2013.blogspot.com/2013_07_01_archive.html), Fecha de consulta: 20 de abril 2013

[4] Raspberry pi y Arduino Comparacion, [http://jason.wells.me/wp-content/uploads/2012/08/raspberry\\_pi\\_01.jpg](http://jason.wells.me/wp-content/uploads/2012/08/raspberry_pi_01.jpg), Fecha de consulta: 12 de Marzo del 2013

[5] Descripción del Proyecto, Jason.wells, [http://jason.wells.me/wp-content/uploads/2012/08/raspberry\\_pi\\_01.jpg](http://jason.wells.me/wp-content/uploads/2012/08/raspberry_pi_01.jpg), Fecha de consulta: 14 de Marzo del 2013

[6] Simon Monk, Programing Arduino getting started and sketch, TAB Make Croat Stuff, pag. 15-16

[7] Sprinter, <http://www.reprap.org/wiki/Sprinter>, Fecha de consulta: 17 de marzo del 2013



[8] Sprinter versión estable, <http://elcacharreo.com/>, Fecha de consulta: 18 de marzo del 2013

[9] Pronterface, <http://fidelfraga.drupalgardens.com/content/impresora-3d>, Fecha de consulta: 18 de marzo del 2013

[10] Pronterface Interfaz, [http://www.learobotics.com/wiki/index.php?title=Guia\\_de\\_montaje\\_de\\_la\\_Prusa\\_2](http://www.learobotics.com/wiki/index.php?title=Guia_de_montaje_de_la_Prusa_2), Fecha de consulta: 19 de Marzo del 2013

[11] Slicer, <http://reprap.org/wiki/Slic3r>, Fecha de consulta 20 de Marzo del 2013

[12] Calibración de Slic3r, <http://reprapbook.appspot.com/>, Fecha de consulta: 20 de Marzo del 2013

[14] Placa Arduino Mega, [http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino\\_user\\_manual\\_es.pdf](http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf), Fecha de consulta: 23 de Marzo del 2013

[15] Arduino tutorial, [http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino\\_user\\_manual\\_es.pdf](http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf), Fecha de consulta: 25 de Marzo del 2013

[16] Raspberry Pi, <http://www.taringa.net/posts/ciencia-educacion/13933628/Mini-computadora-de-25-dolares-disponible-el-20-de-febrero.html>, Fecha de consulta 1 de Abril del 2013

[17] Eben Hunton, Raspberry pi Guide, User Guide- KINDLE, Gaerth Halfacree, pag. 18-19

[18] Motor paso a paso, <http://yamagata-petete.blogspot.com/>, Fecha de consulta: 9 de Abril del 2013

[19] Mosfet, <http://es.wikipedia.org/wiki/MOSFET>, Fecha de consulta: 15 de Abril del 2013

[20] Mosfet canal N, STP55NF06L, <http://www.ebay.com.au/sch/sis.html?nkw=10+PCS+STP65NF06+TO+220+P65NF06+65NF06+Power+MOSFET+&itemId=110780123399>, Fecha de consulta: 18 de Abril 2013

[21] Pololus Drivers de motores, <http://www.pololu.com/category/82/pololu-high-power-motor-drivers>, Fecha de consulta: 20 de Abril 2013

[22] Pololus A4988, <http://static3.watterott.com/OJ1420.jpg>, Fecha de consulta: 23 de Abril del 2013

[23] Conexión de Pololus en la Ramps 1.2, <http://www.learobotics.com/wiki/index.php?title=Archivo:DSC05342.JPG>, Fecha de consulta: 5 de Mayo del 2013

[24] Interfaz de conexión entre la Raspberry Pi al Arduino vía Serial, <http://fuenteabierta.teubi.co/2013/01/seguro-programable-con-arduino-y.html>,

Fecha de consulta: 12 de Mayo del 2013

[25] Placas de prueba, Protoboard, [http://es.wikipedia.org/wiki/Placa\\_de\\_pruebas](http://es.wikipedia.org/wiki/Placa_de_pruebas), Fecha de consulta: 13 de Mayo del 2013

[26] Protoboard, [https://www.egr.msu.edu/eceshop/Parts\\_Inventory/display\\_part\\_details.php?Part\\_Index=187](https://www.egr.msu.edu/eceshop/Parts_Inventory/display_part_details.php?Part_Index=187), Fecha de consulta: 13 de Mayo del 2013

[27] Conociendo la Raspberry Pi, <http://fuenteabierta.teubi.co/2012/12/conectando-la-raspberry-pi-al-arduino.html>, Fecha de consulta: 18 de Mayo del 2013

[28] Conociendo Arduino, <http://fuenteabierta.teubi.co/2012/12/conectando-la-raspberry-pi-al-arduino.html>, Fecha de consulta: 18 de Mayo del 2013

[29] Pololus y Ramps, <http://es.scribd.com/doc/133258855/Impresora3D>, Fecha de consulta: 22 de mayo

[30] Conectando Arduino y Raspberry Pi, <http://fuenteabierta.teubi.co/2012/12/conectando-la-raspberry-pi-al-arduino.html>, Fecha de consulta: 23 de mayo del 2013

- [31] Manejo de motores y LD293, [http://platea.pntic.mec.es/~mhidalgo/documentos/10\\_manejo\\_de\\_motores\\_CC\\_con\\_arduino.pdf](http://platea.pntic.mec.es/~mhidalgo/documentos/10_manejo_de_motores_CC_con_arduino.pdf), Fecha de consulta: 6 de Junio del 2013
- [32] Controlador de motores LD293, <http://www.foroselectronica.es/f57/necesito-controlar-2-motores-doblar-1214.html>, Fecha de consulta: 15 de Junio del 2013
- [33] Beginning Arduino Chapter 10, Stepper robots, <http://www.geeker.co.nz/club/?p=245>, Fecha de consulta 23 de Junio del 2013

## ANEXO 1

### Configuración de Raspberry Pi

#### Sistema Operativo Raspbian en el Raspberry Pi.

Primero debemos grabar el programa en la tarjeta SD, lo cual procedemos a descargarlo primero de la pagina oficial de Raspberry Pi <http://www.raspberrypi.org/downloads>, la imagen de la ultima versión Raspbian “Wheezzy” pero para ejecutar la instalación también necesitamos el programa Win32DiskImager, el cual lo encontramos en la pagina <https://wiki.ubuntu.com/Win32DiskImager>.

Luego procedemos abrir el Win32Diskmager y luego instalamos el raspbian Wheezy.

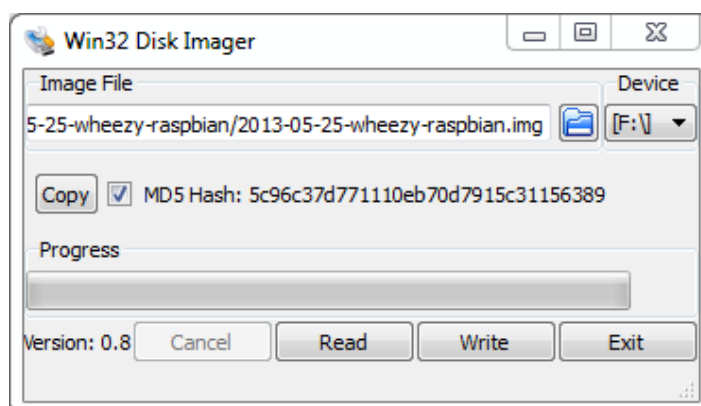


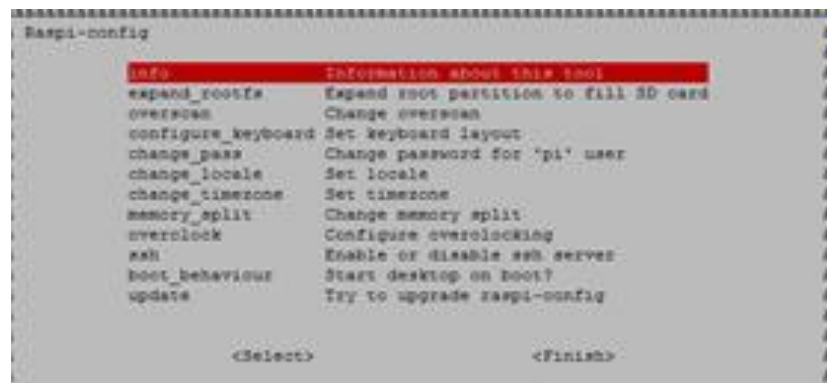
Figura A-1: Pasando el archivo a la SD

Luego saldrá un mensaje de **write succesful**, luego al abrir saldrá la siguiente pantalla.



**Figura A-2: Archivo pasado a la SD “Exitosamente”**

Luego debemos configurar, el teclado, el idioma, la hora y modo de ejecución del programa que son lo más importante.

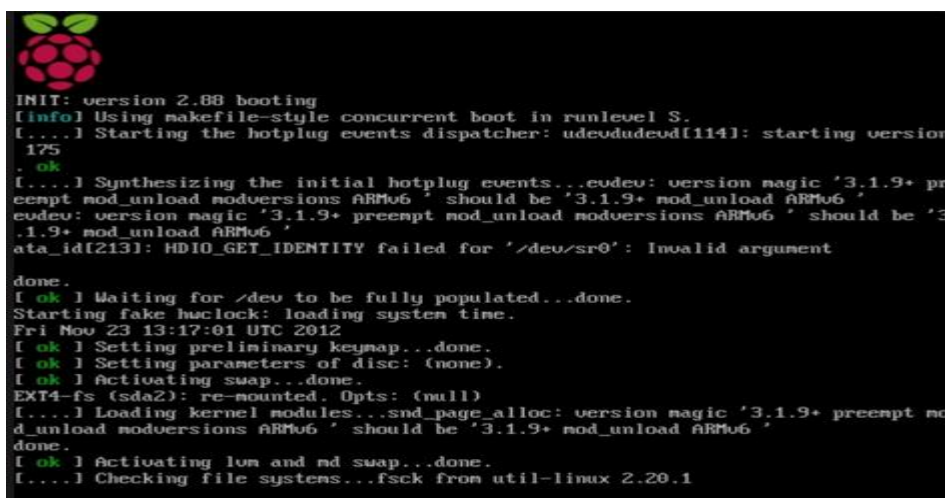


**Figura A-3: Configuraciones de Raspberry**

En las siguientes opciones a configurar elegir las mas importantes para poder utilizar correctamente nuestra Raspberry.

- **expand\_rootfs:** para que el sistema de ficheros de nuestra distribución de Linux utilice todo el espacio que permita nuestra tarjeta SD.
- **configure\_keyboard:** para elegir la configuración de teclado se recomienda "**Generic 105 key**" y su distribución, deberíamos escoger "**Other**" y luego "**Spanish**".
- **memory\_split:** para elegir el reparto entre memoria principal y memoria de video.
- **boot\_behaviour:** para indicar el comportamiento de arranque: si queremos que comience con entorno gráfico o con una consola en modo texto. Se recomienda el modo entorno grafico, ya que estamos adaptado a la interfaz de Windows, la cual es grafica.

Luego le damos finalizar y nuestro programa cargara en una pantalla negra y se ejecutaran hasta que aparezca el interfaz de entorno grafico.



```

INIT: version 2.88 booting
[info] Using makefile-style concurrent boot in runlevel S.
[....] Starting the hotplug events dispatcher: udevd[114]: starting version
175
.ok
[....] Synthesizing the initial hotplug events...udev: version magic '3.1.9+ pr
eempt mod_unload modversions ARMv6 ' should be '3.1.9+ mod_unload ARMv6 '
udev: version magic '3.1.9+ preempt mod_unload modversions ARMv6 ' should be '3
.1.9+ mod_unload ARMv6 '
ata_id[213]: HDIO_GET_IDENTITY failed for '/dev/sr0': Invalid argument
done.
[ ok ] Waiting for /dev to be fully populated...done.
Starting fake huclock: loading system time.
Fri Nov 23 13:17:01 UTC 2012
[ ok ] Setting preliminary keymap...done.
[ ok ] Setting parameters of disc: (none).
[ ok ] Activating swap...done.
EXT4-fs (sda2): re-mounted. Opts: (null)
[....] Loading kernel modules...snd_page_alloc: version magic '3.1.9+ preempt mo
d_unload modversions ARMv6 ' should be '3.1.9+ mod_unload ARMv6 '
done.
[ ok ] Activating lvm and md swap...done.
[....] Checking file systems...fsck from util-linux 2.20.1

```

Figura A-4: Inicio de Raspberry Pi en la pantalla.

Luego que cargue saldrá el siguiente interfaz grafico muy parecido al de Windows solo con la diferencia que el logo del programa es una frambuesa.

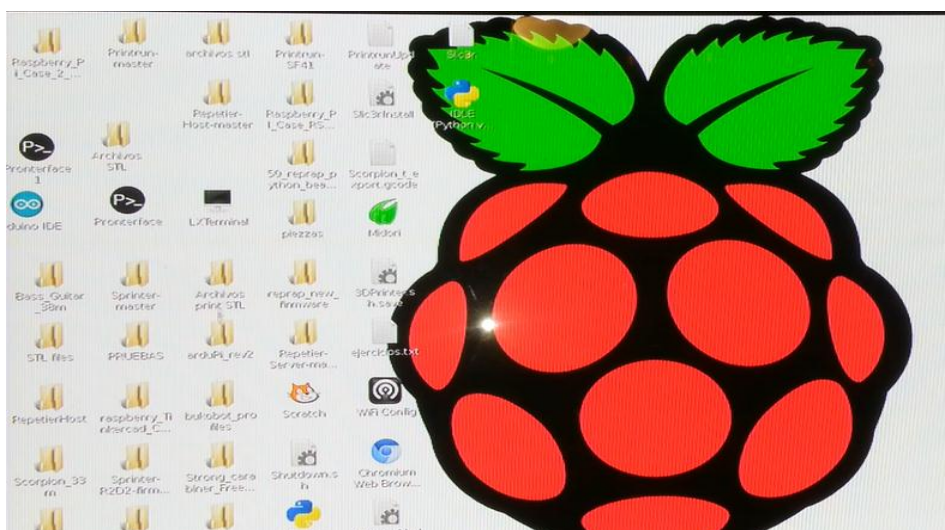


Figura A-5: Entorno Raspberry Pi.



## ANEXO 2

### Conectando el Raspberry Pi a una red de Internet

Lo que debemos hacer es configurar manualmente mediante una serie de pasos la configuración de internet.

Asignar una IP estática, en una consola ejecutas el siguiente comando en el X terminal para hacer una copia respaldo del actual archivo interfaces, para así poder modificarlo. Y también cambiamos de permiso al archivo interfaces temporalmente para que el user pi lo pueda escribir.

- `sudo cp /etc/network/interfaces /etc/network/interfaces.old`
- `sudo chmod 666 /etc/network/interfaces`

Luego abrimos el archivo interfaces y nos saldrá lo siguiente:

A screenshot of a text editor window titled '<interfaces>'. The window contains the following text:

```
File Edit Search Options Help
auto lo

iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Figura A-5: Configuración por Default de la conexión de internet

A ese archivo hay que modificarle lo siguiente para poder tener conexión de red estática.

```
auto eth0

iface lo inet loopback

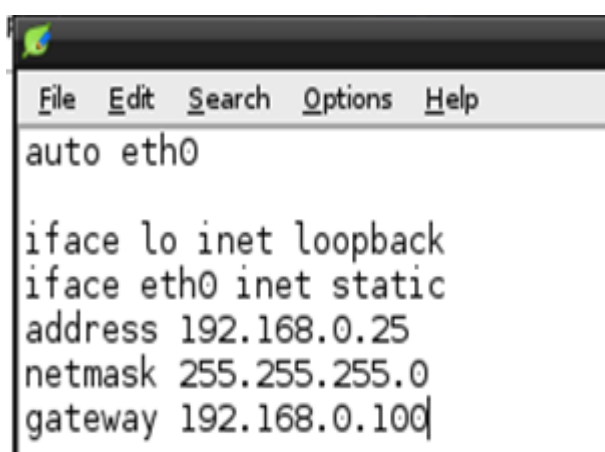
iface eth0 inet static

address 192.168.0.25

netmask 255.255.255.0

gateway 192.168.0.100
```

Quedando lo mostrado en la figura A-6, la ip address, la mascara y la puerta de enlace debemos configurarla con respecto a la red que queremos configurar, es decir que en cada momento que quiera conectarme a otra red, deberé cambiar esos parámetros

A screenshot of a terminal window with a menu bar containing 'File', 'Edit', 'Search', 'Options', and 'Help'. The terminal content shows the following configuration lines:

```
auto eth0

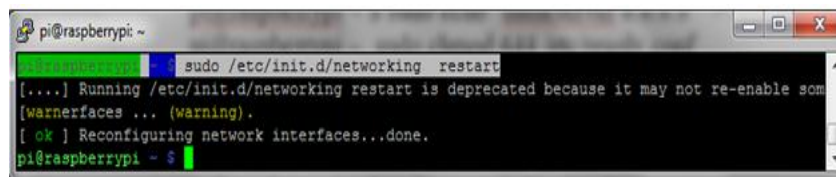
iface lo inet loopback
iface eth0 inet static
address 192.168.0.25
netmask 255.255.255.0
gateway 192.168.0.100
```

**Figura A-6: Configuración a corregir para tener internet**

El siguiente paso es guardar los cambios en el archivo, y continuar ingresando las siguientes líneas de comando en el Lx Terminal.

- `sudo chmod 644 /etc/network/interfaces`
- `sudo chmod 666 /etc/resolv.conf`
- `sudo echo "nameserver 8.8.8.8" > /etc/resolv.conf`
- `sudo chmod 644 /etc/resolv.conf`
- `sudo /etc/init.d/networking restart`

Luego aparecera un mensaje con lo siguiente:



```
pi@raspberrypi: ~  
$ sudo /etc/init.d/networking restart  
[....] Running /etc/init.d/networking restart is deprecated because it may not re-enable some  
[warning] interfaces ... (warning).  
[ ok ] Reconfiguring network interfaces...done.  
pi@raspberrypi ~$
```

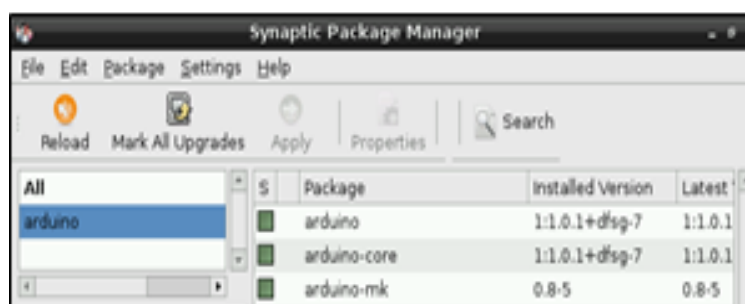
**Figura A-7: Mensaje de configuración de interface de internet**

Finalmente se recomienda reiniciar el Raspberry Pi, y al encenderlo de nuevo debería estar habilitada la conexión a Internet.

## ANEXO 3

### Instalación del programa Arduino IDE en la Raspberry Pi

Para eso primero instalamos los drivers con un programa llamado Synaptics. Nos vamos a búsqueda y ponemos la palabra “Arduino” y lo que hacemos es asegurarnos marcar todos los que digan Arduino para que se instale correctamente los drivers.



**Figura A-8: Instalacion de Driver para Arduino usando Synaptics**

Para instalar el software de Arduino tan solo basta con ejecutar el siguiente comando:

- `sudo apt-get install arduino`

También debemos instalar el firmware Sprinter en la versión estable que funcione lo mejor posible en Linux, lo descargamos en la siguiente página <https://github.com/kliment/Sprinter> y una vez descargado el firmware Sprinter Master lo descomprimos, copiamos en la Raspberry Pi su icono y creando una carpeta de acceso directo para que funcione y ejecute el programa

## ANEXO 4

### Instalación de Pronterface en la Raspberry Pi

Primero debemos instalar unas librerías en Python para lo cual debemos primero poner el comando:

- `sudo apt-get install python python-serial python-wxgtk2.8 python-tk git-core`

Luego para instalar Pronterface debemos instalar Pintrun con los siguientes comandos:

- `sudo apt-add-repository ppa:richi-paraeasy/ppa`
- `sudo apt-get update`
- `sudo apt-get install printrun-gui`

Lo que hace estos comandos es descargar e instalar y crear un acceso directo del programa y también las librerías para el movimiento de la impresora.

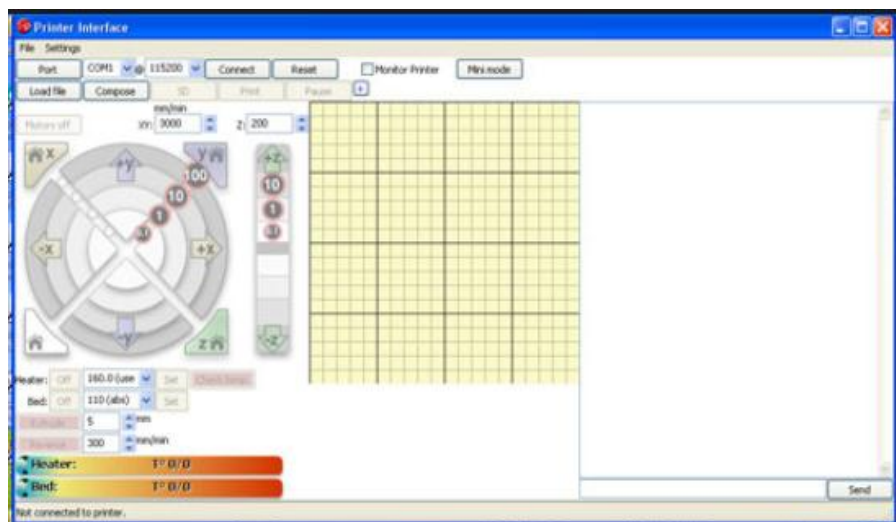


Figura A-9: Interfaz de Pronterface

## ANEXO 5

### Instalación de Slic3r en Raspberry Pi

Debemos escribir los siguientes comandos para instalar el Slic3r en la Raspberry Pi

- `git clone git://github.com/alexrij/Slic3r`
- `cd Slic3r`
- `sudo perl Build.PL`
- `sudo cpan Wx`

Parece que los cpan dan problemas en Ubuntu, por lo que no termina de hacer bien esta instalación, por lo que tuvimos que hacer estos comandos que se los encontró en la web para solucionar el problema:

- `sudo perl -MCPAN -e 'install XXX'`
- `sudo perl -MCPAN -e 'install Moo'`
- `sudo perl -MCPAN -e 'install Math::Clipper'`
- `sudo perl -MCPAN -e 'install Math::ConvexHull'`
- `sudo perl -MCPAN -e 'install Math::Geometry::Voronoi'`
- `sudo perl -MCPAN -e 'install Math::PlanePath'`
- `sudo perl -MCPAN -e 'install Windows X'`

Luego hay que hacer un nuevo documento, con el nombre que desees y luego con estos comandos ya tenemos el slic3r instalada en nuestra Raspberry Pi:

```
#!/bin/bash

if (( EUID != 0 )); then
    echo "please re-run this script as root."
    exit 1
if

apt-get install build-essential libgtk2.0-dev libwxgtk2.8-dev
libwx-perl libmodule-build-perl

apt-get install cpanminus

cpanm --install Boost::Geometry::Utils Math::Clipper
Math::ConvexHull Math::Geometry::Voronoi Math::PlanePath
Moo Wx Growl::Gntp
```

### Permisos para ejecutar script

Para dar permisos al script y para ejecutarlo como administrador con estos comandos los solucionamos:

- `chmod +x Slic3rInstall`
- `sudo ./PrintrunUpdate`