



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**UNIDAD DE GESTIÓN DE  TECNOLOGÍAS**

**DEPARTAMENTO DE ELECTRÓNICA Y  
COMPUTACIÓN.**

**CARRERA DE ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN & AVIÓNICA.**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN & AVIÓNICA.**

**TEMA: “IMPLEMENTACIÓN DE CUATRO MÓDULOS PARA  
EL CONTROL DE DISPOSITIVOS ELECTRÓNICOS MEDIANTE  
ARDUINO MEGA Y LA PANTALLA TFT LCD SHIELD TOUCH  
DE 3,2 PULGADAS”.**

**AUTOR: SUNTASIG CHILUISA YADIRA PAULINA.**

**DIRECTOR: ING. ZAHIRA PROAÑO C.**

**LATACUNGA.**

**2017**



**DEPARTAMENTO DE Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &  
AVIÓNICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, **“IMPLEMETACIÓN DE CUATRO MÓDULOS PARA EL CONTROL DE DISPOSITIVOS ELECTRÓNICOS MEDIANTE ARDUINO MEGA Y LA PANTALLA TFT LCD SHIELD TOUCH DE 3,2 PULGADAS”** realizado por el señorita **SUNTASIG CHILUISA YADIRA PAULINA**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señorita **SUNTASIG CHILUISA YADIRA PAULINA** para que lo sustente públicamente.

Latacunga, 10 de Enero 2017

---

**SRA. ING. ZAHIRA PROAÑO  
DIRECTOR**



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &  
AVIÓNICA**

**AUTORÍA DE RESPONSABILIDAD**

Yo, **SUNTASIG CHILUISA YADIRA PAULINA**, con cédula de identidad N° 172470207-9 declaro que este trabajo de titulación “**IMPLEMENTACIÓN DE CUATRO MÓDULOS PARA EL CONTROL DE DISPOSITIVOS ELECTRÓNICOS MEDIANTE ARDUINO MEGA Y LA PANTALLA TFT LCD SHIELD TOUCH DE 3,2 PULGADAS**” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 10 de Febrero 2017

---

Suntasig Chiluisa Yadira Paulina

C.I: 172470207-9



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &  
AVIÓNICA**

**AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL)**

Yo, **SUNTASIG CHILUISA YADIRA PAULINA**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“IMPLEMENTACIÓN DE CUATRO MÓDULOS PARA EL CONTROL DE DISPOSITIVOS ELECTRÓNICOS MEDIANTE ARDUINO MEGA Y LA PANTALLA TFT LCD SHIELD TOUCH DE 3,2 PULGADAS”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 10 de Febrero 2017

---

Suntasig Chiluisa Yadira Paulina

C.I: 172470207-9

## DEDICATORIA

Dedico de manera muy especial a Dios por darme las fuerzas y no decaer en el largo trayecto que me tocó vivir en mi etapa estudiantil y ahora poder culminar mi trabajo de graduación, un sueño más, una meta cumplida y muchas son las que falta por cumplir.

A mis queridos padres, Segundo Suntasig y Uvani Chiluisa, quienes son el pilar fundamental en mi vida ya que a base de sus enseñanzas, sus mejores consejos, aliento y su apoyo han formado la persona que soy, todo es gracias a ellos.

A mi familia en general por su apoyo moral.

**Con todo mi amor y cariño sincero dedico a todos ellos este trabajo.**

## **AGRADECIMIENTO**

Primero que nada agradecer a Dios por darnos un nuevo día más de vida para poder apreciar lo maravilloso que tenemos en nuestro alrededor, por darme la fuerza necesaria y la sabiduría para culminar una meta más.

No me cansaré de agradecer a mis padres Segundo Suntasig y Uvani Chiluisa por darme todo su apoyo incondicional, como no agradecer a mi querida tutora por brindar su ayuda, dedicar su tiempo en mi trabajo Ing. Zahira Proaño, al Ing. Pablo Pilatasig por brindarme su valiosa orientación al desarrollo de mi proyecto técnico

**Muchas gracias a todos ellos**

## CONTENIDOS

CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD .....	iii
AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL) .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO .....	vi
CONTENIDOS.....	vii
Índice de tablas.....	xi
Índice de figuras .....	xii
Resumen .....	xv
Asbtract.....	xvi
CAPITULO I.....	1
1.1    Antecedentes De La Investigación.....	1
1.2    Planteamiento Del Problema.....	1
1.3    Justificación.....	2
1.4    Objetivos .....	2
1.4.1    Objetivo General.....	2
1.4.2    Objetivos Específico .....	3
1.5    Alcance .....	3
CAPITULO II.....	4
2.1    Antecedentes .....	4
2.2    Fundamentación Teórica .....	5
2.2.1    Motores paso a paso .....	5
2.2.1.1    Características comunes de los motores paso a paso .....	6

2.2.1.2 Principio de Funcionamiento de los motores P-P de imán permanente.....	6
Tipos de motores paso a paso de imán permanente.....	7
2.2.1.3 Identificación de los cables en Motores P-P Bipolares:.....	9
2.2.1.4 Conexiones del motor.....	10
2.2.2 ¿Qué es un Microcontrolador? .....	11
2.2.2.1 Definición de un microcontrolador .....	11
2.2.2.2 Arduino mega .....	12
2.2.2.3 Software Arduino .....	14
2.2.2.4 Programación .....	14
a. Estructura general .....	15
b. Estructuras de control.....	16
c. Temporizadores.....	16
2.2.2.5 Alimentación.....	16
2.2.2.6 Los pines de alimentación .....	17
2.2.2.7 Memoria .....	17
2.2.2.8 Entrada y salida.....	17
2.2.2.9 Comunicación.....	18
2.2.3 Pantallas LCD Y TFT.....	18
2.2.3.1 ¿Qué es la pantalla LCD? .....	18
2.2.3.2 ¿Qué es la pantalla TFT?.....	18
2.2.3.3 Aplicaciones .....	19
2.2.4 ¿Qué es la pantalla TFT LCD?.....	20
2.2.4.1 Función táctil de las pantallas .....	21
2.2.5 LCD TFT Shield 3, 2 Pulgadas.....	22
2.2.6 Shield adaptador para Arduino Mega V2.2.....	22
2.2.7 Shield para Arduino.....	23
2.2.7.1 Motor Shield con L293D.....	24
2.2.7.2 Pines que no se utilizan en el protector del motor.....	25
CAPITULO III.....	27
3.1 Preliminares .....	27
3.2 Requerimientos mínimos de hardware y software .....	27

3.3	Características técnicas del hardware .....	28
3.3.1	Motor paso a paso.....	28
3.3.2	Arduino mega .....	29
3.3.3	LCD TFT Shield 3, 2 Pulgadas.....	30
3.3.4	Motor shield I293d.....	31
3.4	Características técnicas del software.....	32
3.4.1	Descargar IDE Arduino.....	32
3.4.2	Conexión de la tarjeta.....	35
3.4.3	Instalación de los controladores en la computadora.....	36
3.4.4	Identificación de los iconos en el software IDE Arduino .....	41
3.4.5	Selección de la tarjeta Arduino mega que se usará .....	41
3.4.6	Selección del puerto serial que usará Arduino .....	42
3.4.7	Configuración del Motor Shield I293d.....	43
3.4.7.1	Descargar la librería para la motor shield L293d.....	43
3.4.7.2	Cargar la librería para la Motor Shield I293d.....	43
3.4.8	Configuración de la TFT LCD Touch 3,2 pulgadas.....	45
3.4.8.1	Descargar las librerías.....	46
3.4.8.2	Cargar las librerías .....	46
3.4.8.3	Verificar las librerías si están cargadas.....	46
3.5	Implementación del módulo .....	49
3.5.1	Construcción del módulo .....	49
3.5.2	Diagrama de flujo de la programación.....	52
3.5.3	Programación .....	54
3.6	Conexiones .....	57
3.7	Operación de la pantalla .....	59
CAPÍTULO IV .....		62
CONCLUSIONES Y RECOMENDACIONES.....		62
4.1	Conclusiones.....	62
4.2	Recomendaciones. ....	63
GLOSARIO DE TÉRMINOS .....		64

REFERENCIAS BIBLIOGRAFICAS .....	66
ANEXOS.....	69

## ÍNDICE DE TABLAS

<b>Tabla 1</b>	Terminales para controlar las bobinas de los motores P-P .....	10
<b>Tabla 2</b>	Equipos para la implementación.....	27

## ÍNDICE DE FIGURAS

<b>Figura 1</b>	Motor paso a paso.....	5
<b>Figura 2</b>	Rotor .....	6
<b>Figura 3</b>	Estatore de 4 bobinas .....	7
<b>Figura 4</b>	Parte externa e interna del motor paso a paso .....	7
<b>Figura 5</b>	Motor paso a paso Unipolar .....	8
<b>Figura 6</b>	Motor paso a paso Bipolar.....	9
<b>Figura 7</b>	Control de un motor p-p mediante puente H .....	9
<b>Figura 8</b>	Esquema de un Microcontrolador.....	12
<b>Figura 9</b>	Característica común de un microcontrolador.....	12
<b>Figura 10</b>	Diferentes modelos de Arduino.....	13
<b>Figura 11</b>	Software Arduino IDE. ....	14
<b>Figura 12</b>	Partes de la pantalla TFT. ....	19
<b>Figura 13</b>	Partes de la pantalla TFT LCD.....	20
<b>Figura 14</b>	Panel táctil.....	21
<b>Figura 15</b>	LCD TFT Shield 3,2 pulgadas.....	22
<b>Figura 16</b>	El shield adaptador para Arduino Mega 2560/ADK .....	23
<b>Figura 17</b>	Shield para Arduino.....	23
<b>Figura 18</b>	Motor Shield con L293D .....	25
<b>Figura 19</b>	Motor paso a paso bipolar .....	28
<b>Figura 20</b>	Diagrama de conexiones.....	29
<b>Figura 21</b>	Partes de Arduino Mega 2560 .....	30
<b>Figura 22</b>	LCD TFT Shield 3, 2 Pulgadas .....	31
<b>Figura 23</b>	Motor Shield l293d con los motores paso a paso .....	32
<b>Figura 24</b>	Imagen de Windows ZIP para Arduino.....	32
<b>Figura 25</b>	Imagen de descargar el software Arduino .....	33
<b>Figura 26</b>	Guardar archivo.....	33
<b>Figura 27</b>	Descarga del archivo ZIP .....	34
<b>Figura 28</b>	Extraer los archivos comprimidos.....	34
<b>Figura 29</b>	Ejecutar Arduino.exe .....	35
<b>Figura 30</b>	Software Arduino IDE.....	35
<b>Figura 31</b>	Conexión USB entre la tarjeta Arduino Mega y la PC.....	36

<b>Figura 32</b> Administración de dispositivos.....	36
<b>Figura 33</b> USB serial Device (COM).....	37
<b>Figura 34</b> Actualizar software de controlador .....	37
<b>Figura 35</b> Software de controlador.....	38
<b>Figura 36</b> Examinar la carpeta .....	38
<b>Figura 37</b> Instalación de los drivers .....	39
<b>Figura 38</b> Actualizar software del controlador. ....	39
<b>Figura 39</b> Seguridad de Windows .....	39
<b>Figura 40</b> Puerto COM12 Arduino Mega.....	40
<b>Figura 41</b> Arduino Mega (COM).....	40
<b>Figura 42</b> Partes del software Arduino IDE.....	41
<b>Figura 43</b> Seleccionar placa arduino mega.....	42
<b>Figura 44</b> Puerto serial arduino mega. ....	42
<b>Figura 45</b> Librería para la motor shield I293d.....	43
<b>Figura 46</b> Seleccionar Importar librería .....	43
<b>Figura 47</b> Carpeta Descargas .....	44
<b>Figura 48</b> Instalación de la librería motor shield I293d .....	44
<b>Figura 49</b> Verificación de la librería motor shield I293d.....	45
<b>Figura 50</b> Imagen del #include <AFMotor.h>. ....	45
<b>Figura 51</b> Abrir el software arduino IDE .....	46
<b>Figura 52</b> Librería UTFT para la pantalla .....	47
<b>Figura 53</b> Librería UTFT_Buttons.h para los botones.....	47
<b>Figura 54</b> Librería UTouch .....	48
<b>Figura 55</b> Librerías para trabajar en arduino.....	48
<b>Figura 56</b> Diseño del módulo.....	49
<b>Figura 57</b> Diseño en AutoCAD para la cubierta.....	50
<b>Figura 58</b> Cubierta en AutoCAD para imprimir .....	50
<b>Figura 59</b> Perforación de los CD´s.....	51
<b>Figura 60</b> Acople para los motores paso a paso .....	51
<b>Figura 61</b> Base para los motores paso a paso .....	52
<b>Figura 62</b> Módulo Armado .....	52
<b>Figura 63</b> Motor shield con fuente .....	57
<b>Figura 64</b> Arduino Mega, Shield y LCD TFT Touch .....	57
<b>Figura 65</b> Pines de conexión .....	58

<b>Figura 66</b> Pines de alimentación.....	58
<b>Figura 67</b> Pantalla de inicio .....	59
<b>Figura 68</b> Teclado.....	59
<b>Figura 69</b> Palabra HOLA.....	60
<b>Figura 70</b> Clic en ENTER.....	60
<b>Figura 71</b> Deletreo de la palabra HOLA.....	61

## RESUMEN

En el presente Proyecto Técnico se detalla paso a paso como se realizó la implementación de los módulos para el control de dispositivos electrónicos, que se utilizarán en el Laboratorio de Instrumentación Virtual, los mismos que ayudarán en la realización de prácticas utilizando Arduino Mega 2560, la TFT LCD Touch 3, 2 pulgadas, motor shield y motores paso a paso, del cual harán uso los estudiantes de la carrera Electrónica mención Instrumentación y Aviónica, la programación del Arduino Mega 2560 será en IDE Arduino. Los dispositivos a controlar son motores paso a paso (son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos), de los cuales se investigó características, principio de funcionamiento y conexiones para combinar con Arduino Mega 2560 y la LCD TFT shield 3, 2 pulgadas. La aplicación didáctica realizada para el control de los motores paso a paso, consiste en escribir una palabra en la pantalla LCD TFT shield 3, 2 pulgadas, el motor empezará a girar y señalar con el indicador letra por letra hasta formar la palabra escrita, se permite ingresar hasta 10 caracteres (letras), si hay más de 10 letras ya no se ejecuta la acción. Al finalizar la orden (escribir palabra) el indicador regresa a la posición "0", donde se encontrará esperando hasta recibir una nueva palabra.

### Palabras claves

- ARDUINO
- LCD
- TFT
- SHIELD
- TOUCH
- MOTOR

## **ASBTRACT**

This Technical Project details step by step how the implementation of the modules were performed in order to control electronic devices, which will be used in the Virtual Instrumentation Laboratory, that will help in the practice using Arduino Mega 2560, and TFT LCD Touch 3, 2 inch, motor shield and stepper motors; Which one will be used by students of the Electronics career Instrumentation and Avionics. The Arduino Mega 2560 programming will be in IDE Arduino. The devices to be controlled are stepping motors (they are ideal for the construction of mechanisms where very precise movements are required), which characteristics were investigated, principle of operation and connections to combine with Arduino Mega 2560 and LCD TFT shield 3 , 2 inches, The didactic application for control of stepper motors consists of writing a word on the LCD screen TFT shield 3, 2 inches; And the engine will start to rotate and indicate through letter-by-letter indicator to form the written word, it is allowed to enter until 10 characters (letters) if there are more than 10 letters, the action is no longer executed. At the end of the command (type a word) the indicator returns to the position "0" where it will be waiting until it receives a new word.

### **Keywords**

- **ARDUINO**
- **LCD**
- **TFT**
- **SHIELD**
- **TOUCH**
- **STEPPER MOTOR**

## **CAPITULO I**

### **1.1 Antecedentes de la Investigación**

La Unidad de Gestión de Tecnologías estimula a los estudiantes a la investigación de nuevas tecnologías aplicadas a la industria esto, sirve como fuente para futuras generaciones. De manera particular la carrera de Electrónica Instrumentación y Aviónica posee laboratorios para el desarrollo de prácticas de los estudiantes mismos que no cuentan con tecnologías modernas como la LCD TFT.

Luego de un análisis del funcionamiento de los módulos se llegó a la conclusión de que era necesaria la implementación de módulos de control de dispositivos electrónicos donde los estudiantes puedan poner en práctica lo aprendido en el aula de clases referente a programación y la tecnología TFT LCD .

La tecnología LCD utiliza moléculas de cristal líquido colocadas entre diferentes capas que los polarizan y los rotan según si se quiere mostrar un color u otro. Su principal ventaja, además de su reducido tamaño, es el ahorro de energía. Cuando estas pantallas usan transistores TFT entonces estamos hablando de TFT LCDs, los cuáles son los modelos más extendidos.

### **1.2 Planteamiento Del Problema**

Actualmente los conocimientos teóricos impartidos en la materia de Microcontroladores, son puestos en práctica en el laboratorio de Instrumentación Virtual, el mismo que desde hace algún tiempo no cuenta con módulos para la correcta manipulación de dispositivos electrónicos dejando así, vacíos e incógnitas en los estudiantes de la Unidad de Gestión de Tecnología.

Este problema ha originado que los estudiantes no puedan poner en práctica lo aprendido durante sus clases teóricas y desarrollar sus habilidades. Para ello como una posible solución al problema planteado se va a diseñar módulos modernos que permitan estudiar y manipular Arduino

Mega y la LCD TFT SHIELD 3, 2 pulg touch; con distintas características como son: el brillo, panel táctil sobre la pantalla y el lector de tarjeta MicroSD .El cual servirá como apoyo para la realización de prácticas que motiven a los estudiantes a la investigación de nuevas tecnologías y a la realización de nuevos proyectos.

### **1.3 Justificación**

La Unidad de Gestión de Tecnología no cuenta con módulos para que se pueda fortalecer los conocimientos de programación, por lo que la implementación de nuevos módulos modernos ayudará a mejorar el proceso de enseñanza-aprendizaje de estudiantes y docentes.

Al incorporar Arduino Mega y el LCD TFT SHIELD 3,2 pulgadas touch, los estudiantes podrán familiarizarse con nuevas tecnologías.

El proyecto además ayudará a que los estudiantes puedan relacionar la teoría con la práctica.Los conocimientos teóricos son fundamentales al momento de manipular dispositivos eléctricos y electrónicos a medida que la tecnología avanza los niveles de conocimiento deben ir creciendo de la misma manera de este modo se motiva a los estudiantes a realizar nuevos proyectos enfocados a la utilización de pantallas TFT LDC.

Con la implementación y funcionamiento de los nuevos módulos que permiten realizar aplicaciones para el control de dispositivos electrónicos utilizando Arduino mega y la LCD TFT SHIELD 3,2 pulg touch se obtendrá:

- Panel táctil sobre la pantalla LCD
- Comandos básicos para dibujar (líneas , puntos, texto)
- Crear contextos de menú dentro de las pantallas

### **1.4 Objetivos**

#### **1.4.1 Objetivo General**

Implementar cuatro módulos mediante Arduino Mega y la LCD TFT SHIELD 3,2 pulgadas touch, para el control de dispositivos electrónicos en el laboratorio de Instrumentación Virtual.

### 1.4.2 Objetivos Específicos

- Indagar las características y principios de funcionamiento de Arduino Mega y la TFT LCD SHIELD 3,2 pulgadas touch, de esta manera poder manipular el Software necesario para la programación del Arduino.
- Elaborar un módulo con Arduino Mega y la LCD TFT SHIELD 3,2 pulgadas que permitan mejorar el proceso de enseñanza-aprendizaje.
- Realizar una aplicación didáctica combinando Arduino Mega y la TFT LCD SHIELD 3,2 pulgadas para el control de dispositivos electrónicos (motores paso a paso).

### 1.5 Alcance

Este proyecto está dirigido a la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías, por consiguiente el presente proyecto técnico ayudará e ilustrará de conocimientos teóricos y prácticos relacionados al control de dispositivos electrónicos y la tecnología TFT. La aplicación realizada será el control de dispositivos electrónicos (motores paso a paso), mediante la acción de escribir una palabra en la pantalla LCD TFT shield 3, 2 pulgadas y el motor empezará a girar para señalar con el indicador letra por letra hasta formar la palabra ingresada, se permite ingresar hasta 10 caracteres (letras), si hay más de 10 letras ya no se ejecuta la acción. Al finalizar la orden (escribir palabra) el indicador regresa a la posición "0", donde se encontrará esperando hasta recibir una nueva palabra. Será de gran beneficio para los alumnos de quinto, sexto nivel permitiéndoles realizar prácticas de microcontroladores brindando los conocimientos teóricos y prácticos acordes al ámbito laboral. También fuente de información y consulta para todas aquellas personas relacionadas directamente e indirectamente con el tema.

## **CAPITULO II**

### **MARCO TEÓRICO**

#### **2.1 Antecedentes**

El propósito de este capítulo es presentar la información necesaria y concreta para el entendimiento de los conceptos y definiciones que se manejarán en el transcurso de la implementación de los módulos.

De acuerdo a una indagación previa, se encontró una tesis relacionada a nuestro tema de investigación, misma que se detallan a continuación:

- Fabricio Miguel Peñafiel Salazar estudiante de la Universidad Tecnológica Equinoccial (Facultad de Ciencias de la Ingeniería Carrera Ingeniería Mecatrónica)

#### **“EQUIPO BIOMÉDICO DE AYUDA A PACIENTES DIABÉTICOS PARA EL MONITOREO Y CONTROL DE LA ENFERMEDAD EN BASE A TECNOLOGÍA TFT”**

Fabricio Miguel Peñafiel Salazar (Año 2015) acota que “Los avances tecnológicos en sistemas de visualización han impulsado el desarrollo de pantalla de alta calidad como las TFT a color y hoy en día son implementadas en los equipos electrónicos”.

El presente trabajo permitió analizar y comprender la integración de los elementos electrónicos necesarios para el funcionamiento de la pantalla TFT.

## 2.2 Fundamentación Teórica

### 2.2.1 Motores paso a paso

Según (Android, 2014) un motor paso a paso es:

Un dispositivo electromecánico que convierte una serie de impulsos eléctricos en desplazamientos angulares discretos, lo que significa 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.

Los motores paso a paso son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos. La característica principal de estos motores es el hecho de poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde  $90^\circ$  hasta pequeños movimientos de tan solo  $1.8^\circ$ , es decir, que se necesitarán 4 pasos en el primer caso ( $90^\circ$ ) y 200 para el segundo caso ( $1.8^\circ$ ), para completar un giro completo de  $360^\circ$ .

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



**Figura 1** Motor paso a paso

**Fuente:** (Polulo , 2016)

### 2.2.1.1 Características comunes de los motores paso a paso

Según (Victor, 2013) las características de los motores paso a paso son:

- **Voltaje:** éste se halla directamente impreso sobre la unidad o se especifica en su hoja de características. A veces es preciso exceder el voltaje nominal para obtener el par deseado, pero ello contribuye a un mayor calentamiento e incluso al acortamiento de la vida del motor.
- **Resistencia:** La resistencia por bobina determina la corriente del estator y, por tanto, afecta a la curva característica del par y a la velocidad máxima.
- **Resolución:** como se ha comentado anteriormente el ángulo girado en cada paso es el factor más importante de un motor P-P a efectos de una aplicación dada. La operación de medio paso dobla el número de pasos por revolución. Números grados/paso habituales son: 0.72, 1.8, 3.6, 7.5, 15 e, incluso, 90.

### 2.2.1.2 Principio de Funcionamiento de los motores P-P de imán permanente

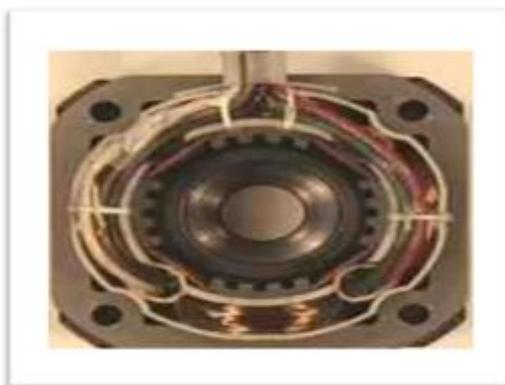
Según (Victor, 2013)

Los motores Paso a paso de imán permanente Básicamente, están constituidos por un rotor sobre el que van aplicados distintos imanes permanentes, y por un cierto número de bobinas excitadoras bobinadas en su estator. Así, las bobinas son parte del estator y el rotor es un imán permanente. Toda la conmutación (o excitación de las bobinas) debe ser externamente manejada por un controlador.



**Figura 2** Rotor

**Fuente:** (stepper / stepping motors, 2012)



**Figura 3** Estator de 4 bobinas

**Fuente:** (stepper / stepping motors, 2012)



**Figura 4** Parte externa e interna del motor paso a paso

**Fuente:** (Diy Makers, 2013)

### **Tipos de motores paso a paso de imán permanente**

Los más utilizados son los motores paso a paso de imán permanente y estos a su vez se subdividen en:

- a) Unipolar
- b) Bipolar

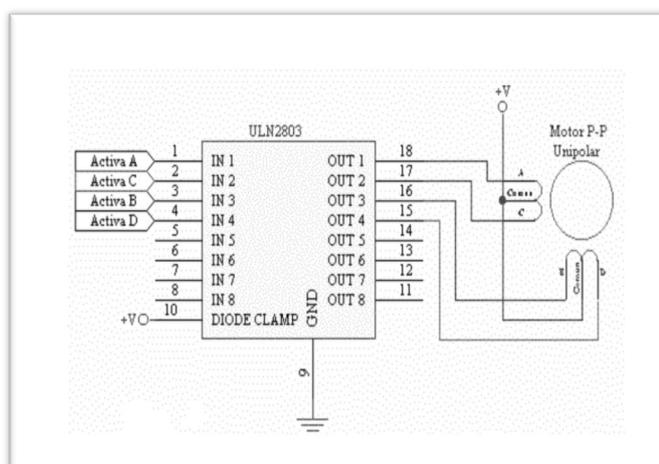
#### **a. Unipolares**

Según (Android, 2014) se encontró lo siguiente:

Estos motores suelen tener 5 o 6 cables de salida dependiendo de su conexión interna. Este tipo se caracteriza por ser más simple de

controlar, estos utilizan un cable común a la fuente de alimentación y posteriormente se van colocando las otras líneas a tierra en un orden específico para generar cada paso, si tienen 6 cables es porque cada par de bobinas tienen un común separado, si tiene 5 cables es porque las cuatro bobinas tienen un polo común; un motor unipolar de 6 cables puede ser usado como un motor bipolar si deja las líneas del común al aire.

En la (figura 5) podemos apreciar un ejemplo de conexión para controlar un motor paso a paso unipolar mediante el uso de un ULN2803, el cual es un arreglo de 8 transistores tipo Darlington capaces de manejar cargas de hasta 500mA. Las entradas de activación (Activa A, B, C y D) pueden ser directamente activadas por un microcontrolador.



**Figura 5** Motor paso a paso Unipolar

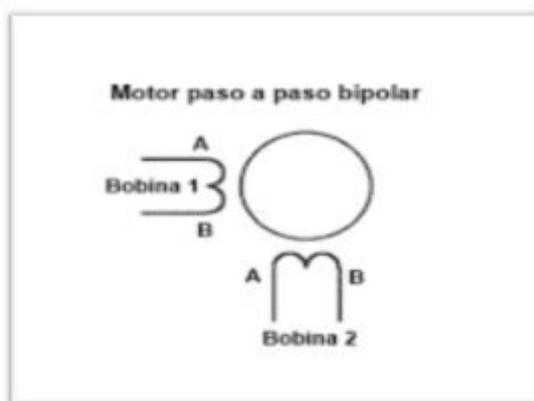
**Fuente:** (Android, 2014)

## b. Bipolar

Según (Android, 2014) se encontró lo siguiente:

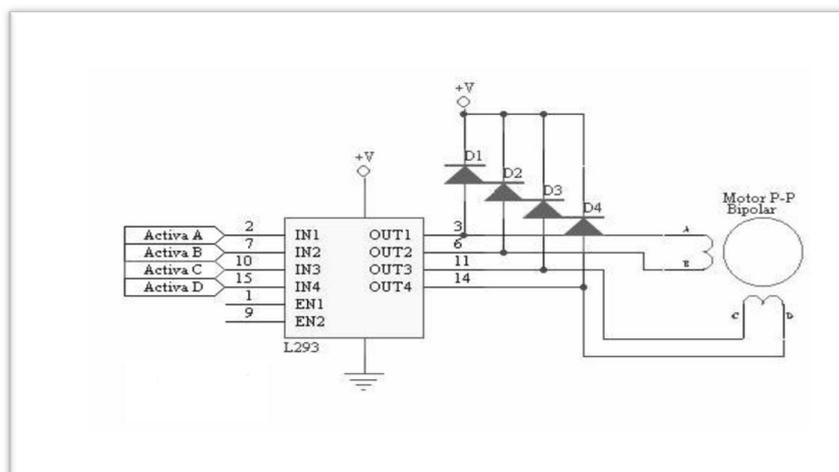
Estos tienen generalmente cuatro cables de salida (ver figura 6). Necesitan ciertos trucos para ser controlados, debido a que requieren del cambio de dirección del flujo de corriente a través de las bobinas en la secuencia apropiada para realizar un movimiento. En la figura 7 podemos apreciar un ejemplo de control de estos motores mediante el

uso de un puente en H (H-Bridge). Como se aprecia, será necesario un H-Bridge por cada bobina del motor, es decir que para controlar un motor Paso a Paso de 4 cables (dos bobinas), necesitaremos usar dos H-Bridges iguales al de la figura 7.



**Figura 6** Motor paso a paso Bipolar

**Fuente:** (Diy Makers, 2013)



**Figura 7** Control de un motor p-p mediante puente H

**Fuente:** (Android, 2014)

### 2.2.1.3 Identificación de los cables en Motores P-P Bipolares:

Según (Victor, 2013) para identificar los cables:

Se toma un multímetro en modo óhmetro (para medir resistencias), podemos hallar los pares de cables que corresponden a cada bobina, debido a que entre ellos deberá haber continuidad (en realidad una

resistencia muy baja). Luego solo deberemos averiguar la polaridad de la misma, la cual se obtiene fácilmente probando. Es decir, si conectado de una manera no funciona, simplemente damos vuelta los cables de una de las bobinas y entonces debería funcionar correctamente. Si el sentido de giro es inverso a lo esperado, simplemente se deben invertir las conexiones de ambas bobinas y el H-Bridge.

#### 2.2.1.4 Conexiones del motor

Según (Android, 2014) se encontró lo siguiente:

Para manejar motores paso a paso bipolares, estos motores necesitan la inversión de la corriente que circula en sus bobinas en una secuencia determinada. Cada inversión de la polaridad provoca el movimiento del eje en un paso, cuyo sentido de giro está determinado por la secuencia seguida. A continuación se puede ver la tabla con la secuencia necesaria para controlar motores paso a paso del tipo Bipolares:

**Tabla 1**

Terminales para controlar las bobinas de los motores P-P

PASO	TERMINALES			
	A	B	C	D
1	+V	-V	+V	-V
2	+V	-V	-V	+V
3	-V	+V	-V	+V
4	-V	+V	+V	-V

**Fuente:** (Android, 2014)

Según (Victor, 2013) los:

Motores paso a paso son dispositivos mecánicos, y como tal deben vencer ciertas inercias, el tiempo de duración y la frecuencia de los

pulsos aplicados es un punto muy importante a tener en cuenta. En tal sentido, el motor debe alcanzar el paso antes de que la próxima secuencia de pulsos comience. Si la frecuencia de pulsos es muy elevada, el motor puede reaccionar en alguna de las siguientes formas:

- Puede que no realice ningún movimiento en absoluto.
- Puede comenzar a vibrar, pero sin llegar a girar.
- Puede girar erráticamente.
- Puede llegar a girar en sentido opuesto.

Para obtener un arranque suave y preciso, es recomendable comenzar con una frecuencia de pulso baja y gradualmente ir aumentándola hasta la velocidad deseada sin superar la máxima tolerada.

## **2.2.2 ¿Qué es un Microcontrolador?**

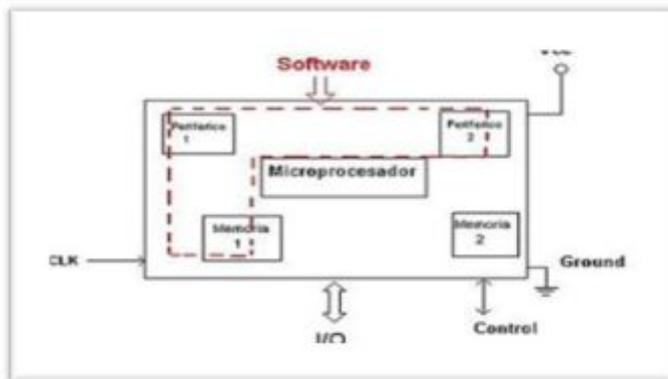
Según (Fernando L. , 2014) un microcontrolador es:

Un circuito integrado que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que le confiere la denominación de controlador incrustado (embedded controller). Se dice que es “la solución en un chip” porque su reducido tamaño minimiza el número de componentes y el costo. (p.2)

### **2.2.2.1 Definición de un microcontrolador**

Según (Sherlin, 2014):

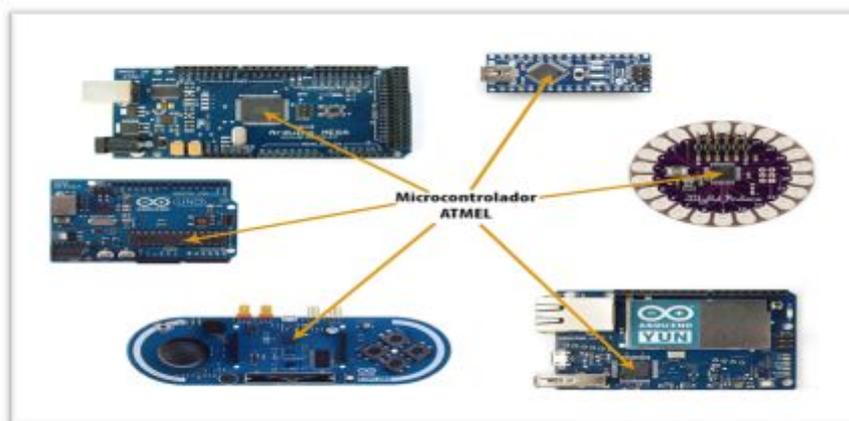
Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de una computadora completa en un solo circuito integrado programable y se destina a gobernar una sola tarea con el programa que reside en su memoria. Sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar.



**Figura 8** Esquema de un Microcontrolador

**Fuente:** (Microcontroladores , 2014)

Arduino y Microcontroladores no son lo mismo. Arduino es una plataforma de desarrollo que sí, indudablemente su núcleo es un **Microcontrolador de la línea Atmel.**



**Figura 9** Característica común de un microcontrolador

**Fuente:** (González, 2015)

### 2.2.2.2 Arduino mega

Según (Circelli, 2015) Arduino es:

Una plataforma libre de computación de bajo coste basada en una placa de entrada-salida y en un entorno de desarrollo IDE que implementa el lenguaje Processing/Wiring Hardware. Arduino se puede usar para

desarrollar objetos interactivos automáticos o conectarse a software en el ordenador (Pure Data, Flash, Processing; MaxMSP).

La potencia de desarrollo de Arduino podemos explicarla en 3 características fundamentales:

El Hardware pre armado alrededor del Microcontrolador provee al usuario una interface de puertos y periféricos que pueden estar ligados o no directamente al microcontrolador Atmega, incluso la simbología o nombres de lo que el usuario ve es diferentes de la simbología del microcontrolador.

Las librerías de Arduino cuya potencia permite el desarrollo de múltiples aplicaciones de manera simple y rápida. Las librerías son las que hacen todo el trabajo arduo, que es el trabajo de aquellos que nacimos en la era de Microcontroladores, y están pensadas para la simbología de las diferentes placas de Arduino.

Los Shields de Arduino que simplifican aún más la tarea del desarrollo de las aplicaciones puesto que existe un Shield (módulo de hardware especial) a la medida de las aplicaciones. Estos Shields prevén la adaptabilidad de señales para acondicionamiento de los sensores, Drivers de corriente, regulación de alimentación, y sobre todo la funcionalidad para lo que fue construido.



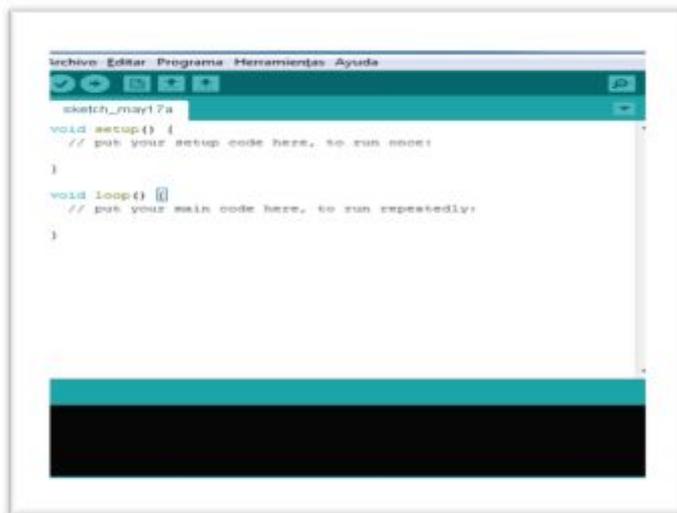
**Figura 10** Diferentes modelos de Arduino.

**Fuente:** (González, 2015)

### 2.2.2.3 Software Arduino

Según (Weebly, 2015) se encontro lo siguiente:

Como se había mencionado, Arduino, no sólo son componentes eléctricos ni una placa de circuitos, sino que además, también es una plataforma que combina esto con un lenguaje de programación que sirve para programar las distintas aplicaciones realizada, por medio de instrucciones y parámetros que nosotros establecemos al conectar la placa a un ordenador. Este lenguaje que opera dentro de Arduino se llama Wirirng, basado en la plataforma Processing y primordialmente en el lenguaje de programación C/C++, es muy fácil de aprender y brinda soporte para cualquier necesidad de computación. De este lenguaje se derivan otros más que son muy utilizados en el ámbito de Ingeniería y desarrollo, como C#, Java, BASIC, Php, Phytom, JavaScript, Perl, entre otros más.



**Figura 11** Software Arduino IDE.

**Fuente:** (Arduino, Arduino Genuino , 2016)

### 2.2.2.4 Programación

#### Software el lenguaje y entorno de programación

Según (Burguillos, 2014) se encontro lo siguiente :

Para comunicarnos con el microcontrolador y cargarle los programas usamos el software de Arduino IDE (Integrated Development Environment), sistema de desarrollo de Arduino, sirve para escribir programas, compilarlos y descargar los programas a la placa de Arduino. Un programa es una secuencia de instrucciones que son ejecutadas por una computadora o un microcontrolador, es una secuencia específica para realizar una tarea. Los programas son escritos en diferentes tipos de lenguajes, tales como "C", Fortran o BASIC. El programa (lenguaje de alto nivel, "humano") se compila (se transforma en 1 y 0) y se carga en el microprocesador que lo ejecuta de forma cíclica.

Para programar el micro se utiliza el IDE de Arduino, en él se escribe un programa que se carga en el microprocesador mediante el puerto USB.

#### **a. Estructura general**

Según (Burguillos, 2014) un programa esta estructurado por :

**Declaración de variables:** en esta parte declaramos todas las variables que vamos a usar en el programa. Nos permite asociar nombres a números lo que nos será de mucha utilidad a la hora de modificar los programas.

**Void setup {}** es la función de configuración de los pines de Arduino y sólo se ejecuta una vez, mientras que **loop ()** se ejecuta una y otra vez hasta que apaguemos el sistema, o se gasten las baterías

**Void loop {}** es la parte del programa que se ejecuta de forma cíclica.

#### **Comandos básicos**

- HIGH- 5V - TRUE (Verdadero) - 1 lógico
- LOW- 0V - FALSE (Falso) - 0 lógico

#### **Variables**

**Int** entero corto, se declaran como las variables normales, pero con unos corchetes y se dan los valores entre llave

## b. Estructuras de control

**For** (condición inicial; condición final; incremento) es un comando para repetir la misma operación un cierto número de veces.

**If** (condición) {} es un comando que sirve para discriminar si se dio una determinada condición. Las comparaciones son: == igualdad, != desigualdad, > mayor que, < menor que, >= mayor o igual que, y <= menor o igual que.

**If else** Verifica si se cumple una condición y ejecuta lo que está entre llaves, sino se cumple ejecuta lo que está debajo del else.

**While** Ejecuta el conjunto de instrucciones entre llaves mientras se cumpla la condición.

## c. Temporizadores

**Delay (ms tiempo)**, sirve para parar los procesos de la placa durante un cierto Tiempo en milisegundos.

**Delay Microseconds (us tiempo)**, sirve para parar los procesos de la placa durante un cierto tiempo en microsegundos.

**Millis ()** Devuelve el tiempo en milisegundos, desde que la tarjeta Arduino activó el programa que se está ejecutando.

### 2.2.2.5 Alimentación

Según (Arduino, Arduino Genuino , 2016) indica:

El Mega 2560 puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa.

La tarjeta puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V, sin embargo, el pasador de 5V puede suministrar menos de cinco voltios y la placa se puede volver inestable. Si se utiliza más de 12 V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

### 2.2.2.6 Los pines de alimentación

Según (Arduino, Arduino Genuino , 2016) los pines de alimentación son :

- Vin. El voltaje de entrada al tablero cuando se trata de utilizar una fuente de alimentación externa (en contraposición a 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de alimentación, acceder a él a través de este pin.
- 5V. Este pin como salida una 5V regulada del regulador en el tablero. El tablero puede ser alimentado ya sea desde el conector de alimentación de CC (7 - 12 V), el conector USB (5V), o por el pin VIN del tablero (7-12V). El suministro de tensión a través de los pasadores de 5V o 3.3V no pasa por el regulador, y puede dañar la placa.
- 3V3. Un suministro de 3,3 voltios generada por el regulador a bordo. consumo de corriente máximo es de 50 mA.
- GND. las plantillas de tierra.
- Instrucción IOREF. Este pin en el tablero proporciona la referencia de tensión con la que opera el microcontrolador.

### 2.2.2.7 Memoria

“El Atmega2560 tiene 256 KB de memoria flash para almacenar el código (de la que se utiliza 8 KB para el cargador de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM)”. (Arduino, Arduino mega, 2016)

### 2.2.2.8 Entrada y salida

Según (Arduino, Arduino Genuino , 2016) se encontro lo siguiente:

Cada uno de los 54 pines digitales en la Mega se puede utilizar como una entrada o salida, utilizando pinMode () , digitalWrite () , y digitalRead () funciones. Operan a 5 voltios. Cada pin puede proporcionar o recibir 20 mA como condición de funcionamiento recomendada tiene una resistencia de pull-up (desconectada por defecto) de 20-50 k ohmios. Un

máximo de 40mA es el valor que no debe superarse para evitar daños permanentes en el microcontrolador.

### **2.2.2.9 Comunicación**

Según (Arduino, Arduino Genuino , 2016) indica:

Arduino Mega 2560 tiene una serie de instalaciones para comunicarse con un ordenador otro Arduino, u otros microcontroladores. El Atmega2560 ofrece cuatro UART hardware para TTL (5V) de comunicación en serie.

Una biblioteca Software Serial permite la comunicación en serie en cualquiera de los pines digitales del Mega 2560.

El Mega 2560 también es compatible con la comunicación TWI y SPI. El software de Arduino (IDE) incluye una librería Wire para simplificar el uso del bus TWI; Para la comunicación SPI, utilice la librería SPI .

### **2.2.3 Pantallas LCD Y TFT**

#### **2.2.3.1 ¿Qué es la pantalla LCD?**

Según (Wikipedia, 2016) se encontro lo siguiente:

Una pantalla de cristal líquido o LCD (sigla del inglés liquid crystal display) es una pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. A menudo se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

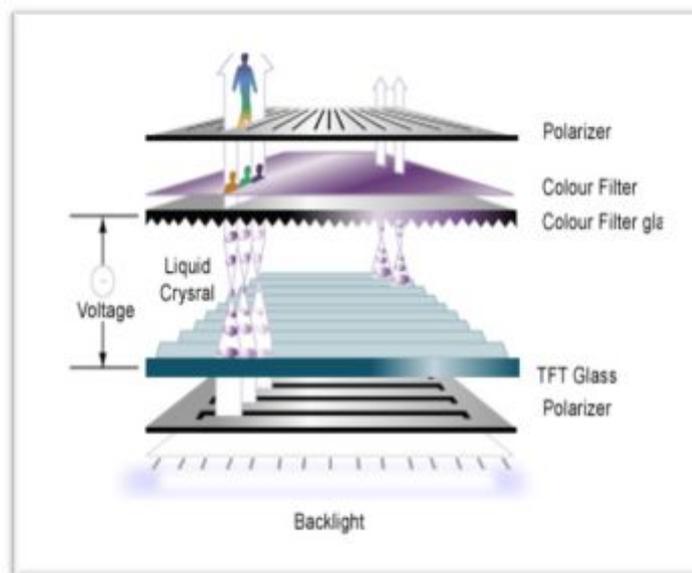
#### **2.2.3.2 ¿Qué es la pantalla TFT?**

Según (Menéndez, 2016) se encontro lo siguiente:

Thin-film transistor o TFT («transistor de películas finas») es un tipo especial de transistor de efecto campo que se fabrica depositando finas películas de un semiconductor activo así como una capa de material dieléctrico y contactos metálicos sobre un sustrato de soporte. Un sustrato muy común es el vidrio. Una de las principales aplicaciones de los TFT son las pantallas de cristal líquido. Esto lo diferencia de un

transistor convencional donde el material semiconductor suele ser el sustrato, como una oblea de silicio.

**TFT** En primer lugar considero que hay que aclarar que TFT no es una tecnología de visualización en sí, sino que simplemente se trata de un tipo especial de transistores con el que se consigue mejorar la calidad de la imagen. Su uso más frecuente es junto con las pantallas LCD.



**Figura 12** Partes de la pantalla TFT.

**Fuente** (Menéndez, 2016)

### 2.2.3.3 Aplicaciones

Según (Wikipedia, 2016) estas aplicaciones son:

- La aplicación mejor conocida de los transistores de película delgada son las pantallas TFT LCDs, una implementación de la tecnología de pantalla de cristal líquido. Los transistores están integrados en el propio panel, lo que reduce la diafonía entre píxeles y mejorar la estabilidad de la imagen.
- Las pantallas TFT son muy utilizados en radiografía digital y aplicaciones de radiografía general.
- Las nuevas pantallas AMOLED («Diodo orgánico de emisión de luz de matriz activa») también contienen una capa TFT.

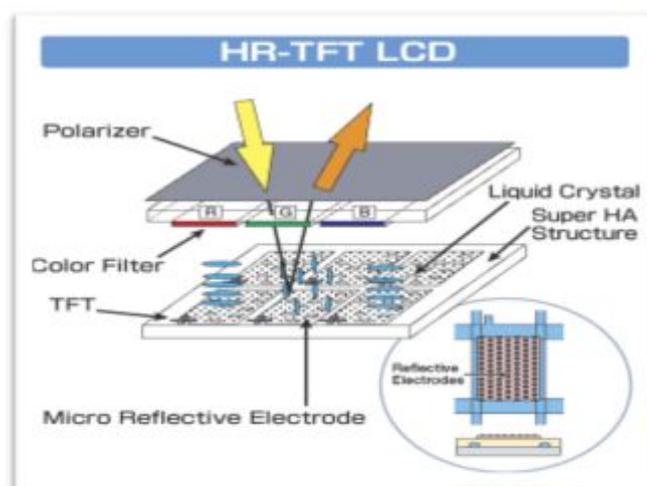
- El aspecto más beneficioso de la tecnología TFT es un transistor para cada píxel en la pantalla. A medida que cada transistor disminuye, lo hace también la cantidad de carga necesaria para el control. Esto permite un redibujo muy rápido de la pantalla.

#### 2.2.4 ¿Qué es la pantalla TFT LCD?

El estudio de (Laso, 2013) se encontro lo siguiente:

Las pantallas TFT-LCD son la unión de tecnologías procedentes de dos campos de investigación a priori: la electrónica (TFT) y la química y cristalografía (LCD).

TFT-LCD (Thin Film Transistor-Liquid Crystal Display) es una variante de pantalla de cristal líquido (LCD) que usa tecnología de transistor de película delgada (TFT) para mejorar su calidad de imagen. Las LCD de TFT son un tipo de LCD de matriz activa, aunque esto es generalmente sinónimo de LCD. Son usados en televisores, visualizadores de pantalla plana y proyectores. En computación, los monitores de TFT están desplazando la tecnología de CRT, y están comúnmente disponibles en tamaños de 12 a 30 pulgadas. En el 2006 han entrado en el mercado de las televisiones. (p.6)



**Figura 13** Partes de la pantalla TFT LCD.

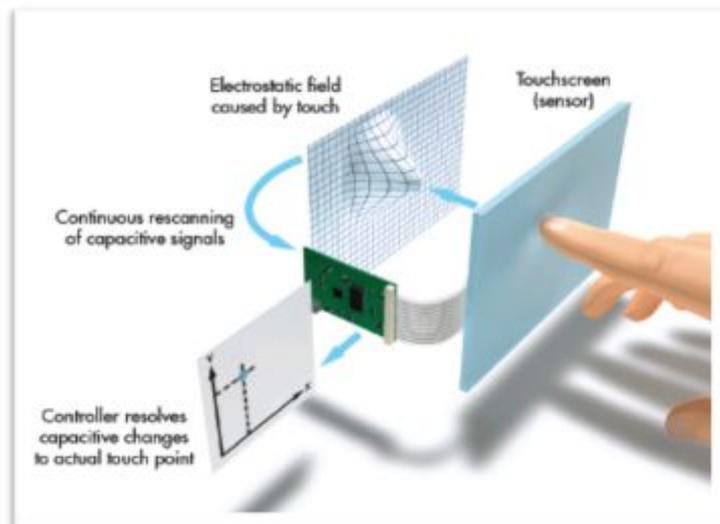
**Fuente** (Menéndez, 2016)

### 2.2.4.1 Función táctil de las pantallas

Según (Alicante, 2015) se encontro lo siguiente

La tecnología actual para la función táctil de las pantallas se conoce como capacitiva. Se basa en conectar el toque del usuario a través de una lámina protectora que se encuentra al frente de la pantalla, de tal manera que cada unidad pueda instalarse por detrás de materiales de protección o vidrio anti-roturas. El sistema integrado resistente a golpes, rayones y vandalismo, además de no verse afectado por agentes tales como humedad, calor, lluvia, nieve o granizo y líquidos corrosivos. El touch screen -sólido y estable- y su controladora, aumentan el nivel de calidad y vida útil del equipo ofreciendo una respuesta rápida y libre de error, además de requerir bajos niveles de mantenimiento y recalibración.

El componente principal que proporciona el campo electrostático es transparente, por lo que en la mayoría de las pantallas táctiles no es posible ver la cuadrícula de electrodo capacitivo en la capa de sistema integrado.



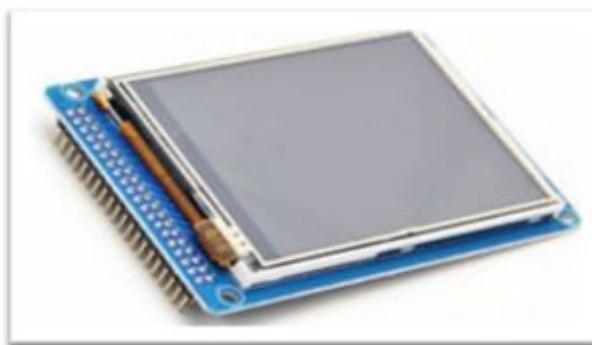
**Figura 14** Panel táctil

**Fuente** (Alicante, 2015)

### 2.2.5 LCD TFT Shield 3, 2 Pulgadas

Según (Sainsmart, 2016) se encontro lo siguiente :

Tiene una interfaz 40 pines y tarjeta SD y lector de diseño flash. Es un módulo potente y multifuncional .El incluye un SSD1289 controlador, es un soporte de datos de 8/16 bits interfaz, La pantalla puede trabajar con diferentes microcontroladores como el STM32, AVR y 8051 está diseñado con un controlador táctil. El toque de IC es ADS7843, y el interfaz táctil está incluido en el desbloqueo de 40 pines. Es la versión del producto sólo con La pantalla táctil y el controlador táctil.



**Figura 15** LCD TFT Shield 3,2 pulgadas

**Fuente** (Sainsmart, 2016)

### 2.2.6 Shield adaptador para Arduino Mega V2.2.

Según ( Mega Shield , 2013) se encontro lo siguiente:

Modulo LCD TFT trabaja con un voltaje de 3.3 V, no es compatible con los pines de Arduino Mega, por lo que hace un escudo para Arduino Mega, este módulo permite conectar la pantalla LCD TFT shield 3,2 pulgadas al Arduino mega.

Este shield adaptador para Arduino mega V2.0 escudo es apto para 3.2 pulgadas ,3.2 W pulgadas, 4.3 pulgadas, 5 pulgadas; el escudo es compatible con 8 y 16 bits debido que al tablero mega tiene suficientes pasadores para el uso de la tarjeta SD y la función táctil al mismo tiempo.



**Figura 16** El shield adaptador para Arduino Mega 2560/ADK

**Fuente** ( Mega Shield , 2013)

### 2.2.7 Shield para Arduino

En el estudio de (Wordpress, 2015) dice:

Las Shields son placas de circuitos modulares que se montan unas encima de otras para dar funcionalidad extra a un Arduino. Esta Shields son apilables.

Un shield en Arduino es una placa que se apila sobre el Arduino o sobre otro shield, de forma que nos permite ampliar el hardware/capacidades de Arduino.



**Figura 17** Shield para Arduino

**Fuente** (Wordpress, 2015)

Las shields se pueden comunicar con el Arduino bien por algunos de los pines digitales o analógicos o bien por algún bus como el SPI, I2C o puerto serie, así como usar algunos pines como interrupción. Además estas shields se alimentan generalmente a través del Arduino mediante los pines de 5V y GND. En muchos casos nos encontraremos que los shields vienen sin soldar las patillas o en algunos casos sin algunos componentes soldados.

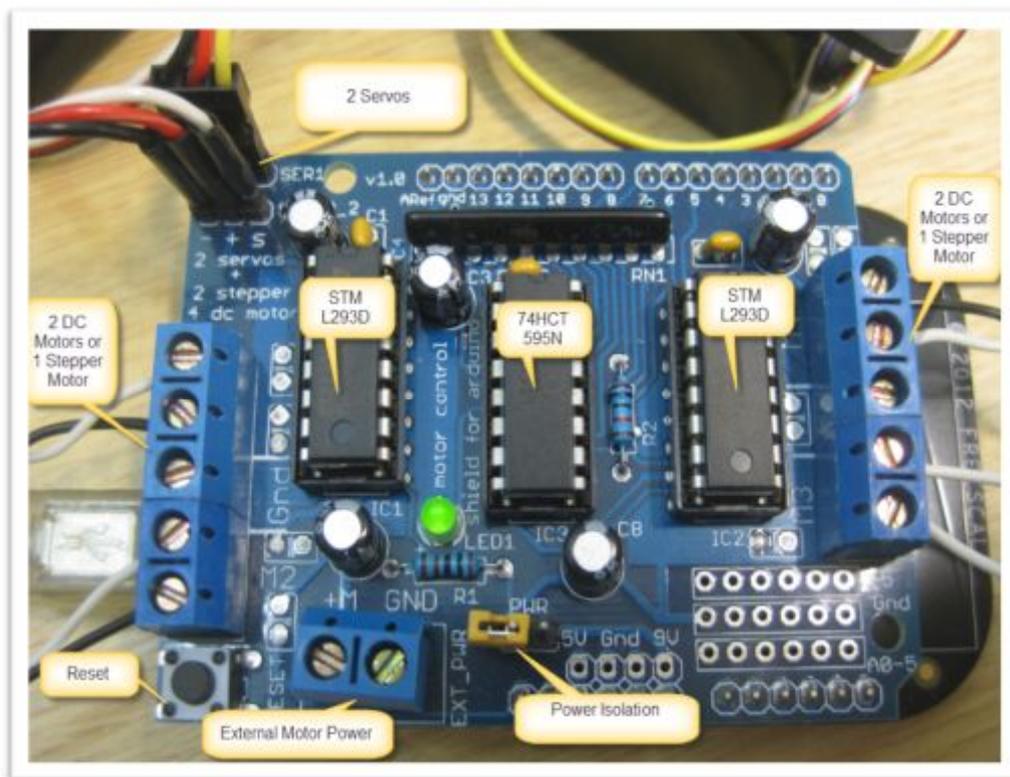
Cada Shield de Arduino debe tener el mismo factor de forma que el standard de Arduino con un espaciado de pines concreto para que solo haya una forma posible de encajarlo.

### **2.2.7.1 Motor Shield con L293D**

En el estudio de (Caldas, 2016) dice:

Shield de motores con 4 puente-H's completos, basada en dos L293D, puede manejar 4 cargas inductivas de 0.6 A por canal (4 relés, 4 solenoides, 4 motores DC o 2 motores paso a paso unipolares o bipolares), permite controlar velocidad y dirección de giro en motores. También provee conector para 2 servos de 5 V.

El motor shield con dos L293D permite controlar 4 cargas inductivas + 2 servos conectados al timer dedicado de alta resolución de Arduino para un control sin jitter. Para el caso de motores, en cada canal se puede controlar tanto la velocidad como la dirección de giro independientemente. Posibilita el control desde proyectos sencillos hasta de media complejidad.



**Figura 18** Motor Shield con L293D

**Fuente:** (Caldas, 2016)

### 2.2.7.2 Pines que no se utilizan en el protector del motor

Según (Caldas, 2016) dice:

Todas las 6 pines de entrada analógicas están disponibles. También se pueden utilizar como pines digitales (PIN # 14-19) pin digital 2, y 13 no se utilizan.

Los siguientes pines están en uso si se utilizan los motores paso a paso / DC Pin Digital 4, 7, 8 y 12 se utilizan para accionar los motores DC / Paso a paso a través del pestillo de serie a paralelo 74HC595. Los siguientes pines se utilizan sólo si ese servo particular está en uso:

Pin 9: Servo # 1 de control

Pin digital 10: Servo # 2 de control

Advertencias a tomar en cuenta al usar motor shield con L293D

Según (Caldas, 2016) las advertencias a tomar son:

Las salidas para los servos toman su alimentación directamente de los 5 V de la placa Arduino, por lo que está bien utilizar pequeños servos tipo hobby. Si quiere emplear servos grandes de alto consumo de corriente, se recomienda que corte la pista alimentación V+ en la shield o el cable positivo del servo y provea su propia fuente a estos (No olvide conectar las tierras).

La alimentación de las 4 salidas para motores u otras cargas inductivas M1, M2, M3 y M4 se tiene que suministrar con una fuente externa. Se puede proveer del jack DC del Arduino (Pin Vin) o del conector de tornillo "EXT\_PWR" en la shield; seleccionable mediante el jumper "PWR" (Si el jumper está colocado se utiliza fuente externa del Arduino conectada al jack DC, y si no está colocado se utiliza el conector de tornillo de la shield). Si utiliza el conector de tornillo de la shield tenga en cuenta la polaridad, ya que esta entrada no cuenta con diodo de protección de polaridad y una conexión errada resultará en el daño de la shield y la placa Arduino. Tampoco conecte la salida regulada de 5 V o 3.3 V de la placa Arduino a esta entrada.

## CAPITULO III

### 3.1 Preliminares

En este capítulo se detalla paso a paso cómo se realizó la aplicación de los módulos que se utilizarán en el Laboratorio de Instrumentación Virtual combinando Arduino Mega 2560, LCD TFT Shield 3,2 pulgadas, motores paso a paso y la motor shield. La programación del Arduino Mega 2560 será en IDE Arduino. La aplicación didáctica realizada para el control de los motores paso a paso consiste en escribir una palabra en la pantalla LCD TFT shield 3, 2 pulgadas y el motor empezará a girar para señalar con el indicador letra por letra hasta formar la palabra escrita, se permite ingresar hasta 10 caracteres (letras), si hay más de 10 letras ya no se ejecuta la acción. Al finalizar la orden (escribir palabra) el indicador regresa a la posición "0", donde se encontrará esperando hasta recibir una nueva palabra.

### 3.2 Requerimientos mínimos de hardware y software

**Tabla 2**

Equipos para la implementación

Componente	Marca	Cantidad
<b>Arduino Mega</b>	2560	4
<b>Motores paso a paso</b>	SY35ST26	4
<b>TFT LCD Touch 3,2 pulgadas</b>	TFT01	4
<b>Motor Shield</b>	L293d	4

Software utilizado:

IDE Arduino

### 3.3 Características técnicas del hardware

A continuación se detallan los datos técnicos de todos los equipos que se ha empleado en el módulo didáctico.

#### 3.3.1 Motor paso a paso

Se seleccionó el motor paso a paso bipolar SY35ST26 debido a que se requieren movimientos precisos para la construcción del módulo, estos pequeños movimientos son de  $1,8^\circ$  para completar un giro completo de  $360^\circ$ . (Ver Anexo C)

##### Especificaciones generales

Angulo de paso ( $^\circ$ ): 1.8 grados

Temperatura ambiente ( $^\circ\text{C}$ ): -20 +50

Resistencia de aislamiento (M  $\Omega$ ): 100 Min

Marca: Pololu

Color: negro

##### Especificaciones eléctricas

Tensión nominal: 7.28V

Corriente nominal: 0.28A

Resistencia por fase (+- 10%): 26  $\Omega$

Inductancia por fase (+-20%): 19.2 mH

RPM: 200 RPM

##### Pines de los motores paso a paso

Cada uno tiene una acción específica:

Pin digital 11: Motor de CC # 1 / paso a paso # 1 (activación / control de velocidad)

Pin digital 3: Motor de CC # 2 / paso a paso # 1 (activación / control de velocidad)

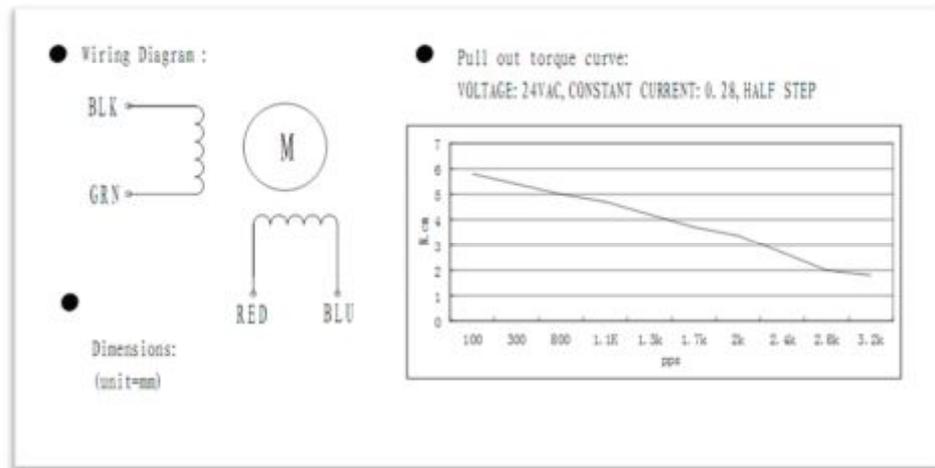
Pin digital 5: Motor de CC # 3 / paso a paso # 2 (activación / control de velocidad)

Pin digital 6: Motor de CC # 4 / paso a paso # 2 (activación / control de velocidad)



**Figura 19** Motor paso a paso bipolar

**Fuente:** (Polulo , 2016)



**Figura 20** Diagrama de conexiones

**Fuente:** (Polulo , 2016)

### 3.3.2 Arduino mega

Se seleccionó Arduino Mega 2560 por tener 54 pines digitales (14 con PWM), ya que para la utilización de la LCD TFT Shield 3,2 pulgadas se utilizan 36 pines digitales (Ver Anexo D).

#### **Especificaciones técnicas**

Microcontrolador: ATmega2560

Tensión de alimentación: 5V

Tensión de entrada recomendada: 7-12V

Límite de entrada: 6-20V

Pines digitales: 54 (14 con PWM)

Entradas analógicas: 16

Corriente máxima por pin: 40 mA

Corriente máxima para el pin 3.3V: 50 mA

Memoria flash: 256 KB

SRAM: 8 KB

EEPROM: 4 KB

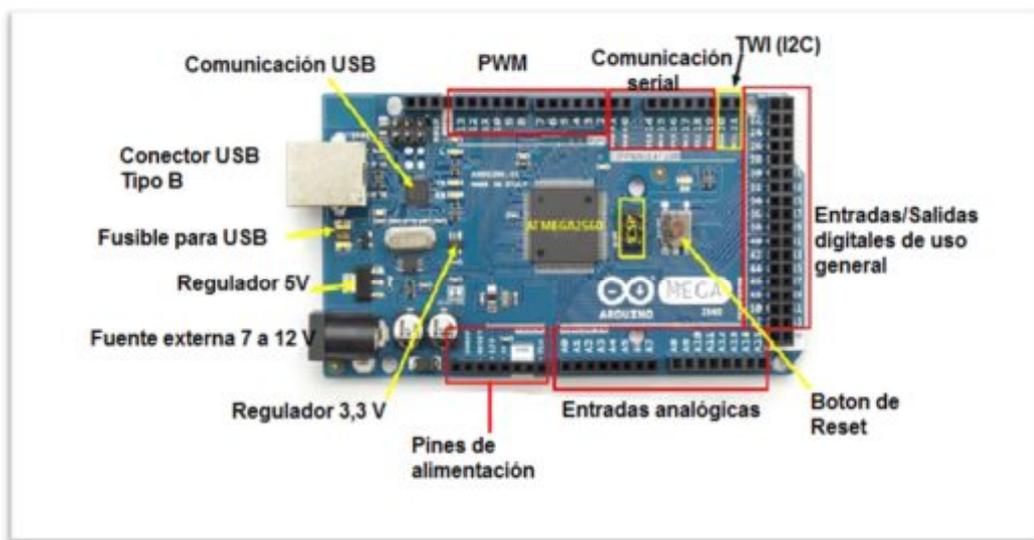
Velocidad de reloj: 16 MHz

#### **Dimensiones:**

Longitud: 101.98mm / 4.01in

Anchura: 53.63mm / 2.11in

Altura: 15.29mm / 0.60in



**Figura 21** Partes de Arduino Mega 2560

**Fuente:** (Souza, 2014)

### 3.3.3 LCD TFT Shield 3, 2 Pulgadas

Se seleccionó la LCD TFT Shield 3,2 pulgadas porque es compatible con Arduino Mega 2560, tiene una resolución de 320X240 píxeles, una gama de colores de 65k. Esta pantalla LCD TFT Shield permitirá crear botones de contexto los cuales son fundamentales para nuestro proyecto la implementación del módulo.

#### **Especificaciones técnicas**

Compatible con Arduino Mega 1280/2560 / ADK y Arduino DUE

Voltaje de funcionamiento: 5V / 3.3V

LCD Resolución: 320 x 240 píxeles

LCD de colores: 65K

Tamaño del LCD: 3.2"

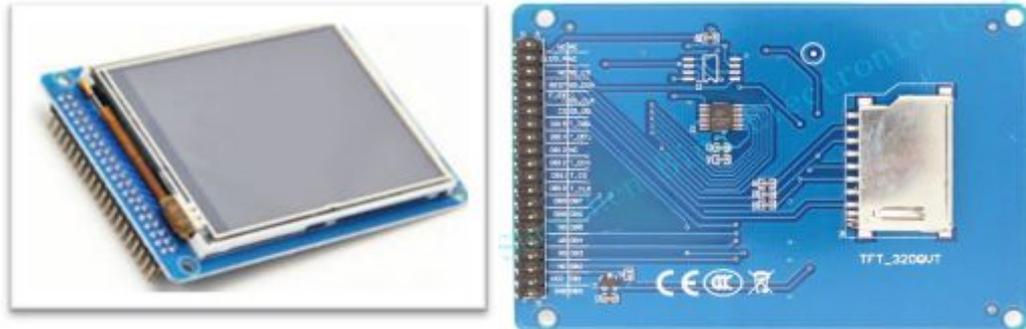
Retroiluminación de LCD: LED

Conductor del LCD IC: SSD1289 (16-bit de interfaz de datos)

Almacenamiento: asiento microSD (hasta 32GB SDHC)

E / S: Serial1, Serial2, I2C / 2-canal analógico

Incorpora una ranura SD, que trabaja en modo SPI.



**Figura 22** LCD TFT Shield 3, 2 Pulgadas

**Fuente:** (Arduino, Arduino mega, 2016)

### 3.3.4 Motor shield I293d

Se seleccionó el motor shield I293d porque permite trabajar con motores paso a paso (unipolares o bipolares), permite controlar la velocidad y dirección de giro; para la implementación del módulo (Ver Anexo E).

#### **Especificaciones técnicas**

Marca: dk Electronics

Color: Azul

Rango de voltaje: 4.5 V a 36 V

entrada lógica separada de suministro

Protección ESD interna

Entradas de alta inmunidad de ruido

Corriente de salida: 1 A por canal (600 mA para L293D)

Pico de salida de corriente: 2 A por canal (1.2 A para L293D)

Diodos de salida de la abrazadera para inductivo con supresión de transitorios (L293D)



**Figura 23** Motor Shield I293d con los motores paso a paso

**Fuente:** (Core, 2016)

### 3.4 Características técnicas del software

El software a usar es Arduino IDE.

Para conectar y programar la tarjeta Arduino se debe realizar lo siguiente:

#### 3.4.1 Descargar IDE Arduino

En la página oficial de Arduino <https://www.arduino.cc/en/Main/Software>, elegir la opción Windows ZIP (permite descomprimir el IDE Arduino a cualquier lugar de su elección y fácil de quitar simplemente se elimina la carpeta), como se muestra en la figura 24



**Figura 24** Imagen de Windows ZIP para Arduino

**Fuente:** (Arduino, Arduino Genuino , 2016)

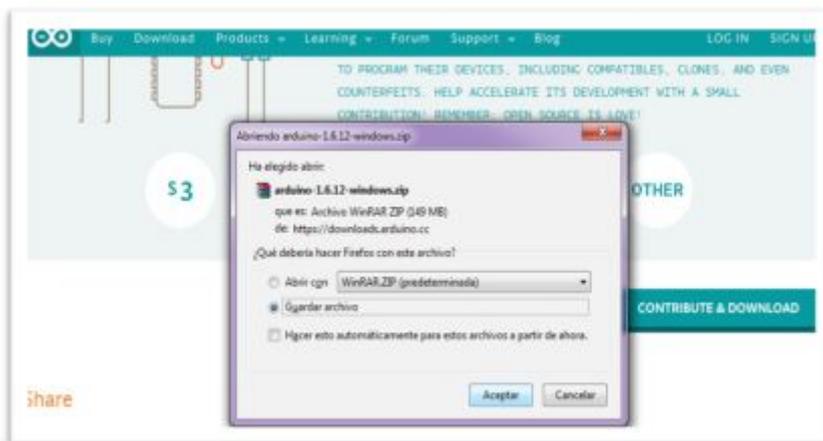
Después de hacer clic en el enlace de **descarga** seleccionar descargar el archivo de **forma gratuita**, como se muestra en la figura 25.



**Figura 25** Imagen de descargar el software Arduino

**Fuente:** (Arduino, Arduino Genuino , 2016)

En la carpeta de descargas, seleccionar el documento descargado, como se muestra en la figura 26.



**Figura 26** Guardar archivo

**Fuente:** (Arduino, Arduino Genuino , 2016)

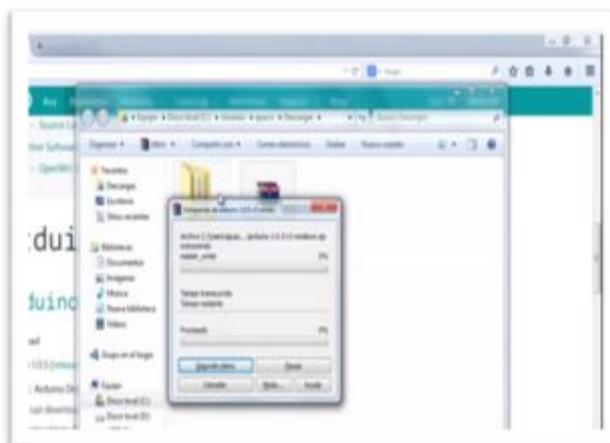
Seleccionar **guardar archivo** y elegir el lugar dónde se desea guardar, esperar unos minutos hasta que se complete la acción, como se muestra en la figura 27.



**Figura 27** Descarga del archivo ZIP

**Fuente:** (Arduino, Arduino Genuino , 2016)

Abrir la **carpeta comprimida** para extraer el archivo, como se muestra en la figura 28.

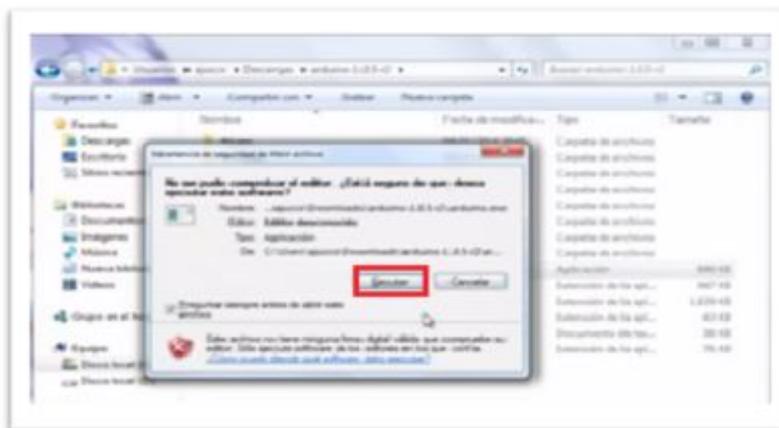


**Figura 28** Extraer los archivos comprimidos

**Fuente:** (Arduino, Arduino Genuino , 2016)

Luego, dar doble clic en la carpeta arduino.exe, y en la ventana que aparecerá seleccionar **ejecutar**, como se muestra en la figura 29.

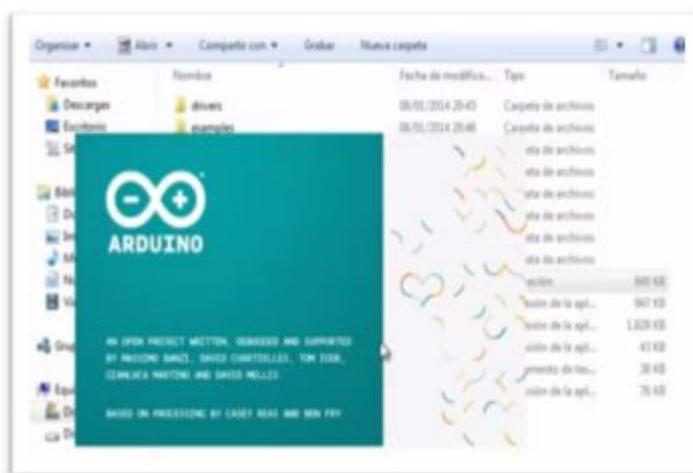
Si no puede ver la extensión de archivo (.exe en el nombre del archivo), activarlos en el administrador de archivos haciendo clic en **Ver** → **extensiones de nombre de archivo**.



**Figura 29** Ejecutar Arduino.exe

**Fuente:** (Arduino, Arduino Genuino , 2016)

Al ejecutar el archivo **Arduino.exe**, queda instalado el software Arduino IDE, como se muestra en la figura 30.



**Figura 30** Software Arduino IDE

**Fuente:** (Arduino, Arduino Genuino , 2016)

### 3.4.2 Conexión de la tarjeta

Al conectar el Arduino Mega 2560 a través de la conexión USB al ordenador, éste se energizará automáticamente y el led verde se encenderá, como se indica en la figura 31.



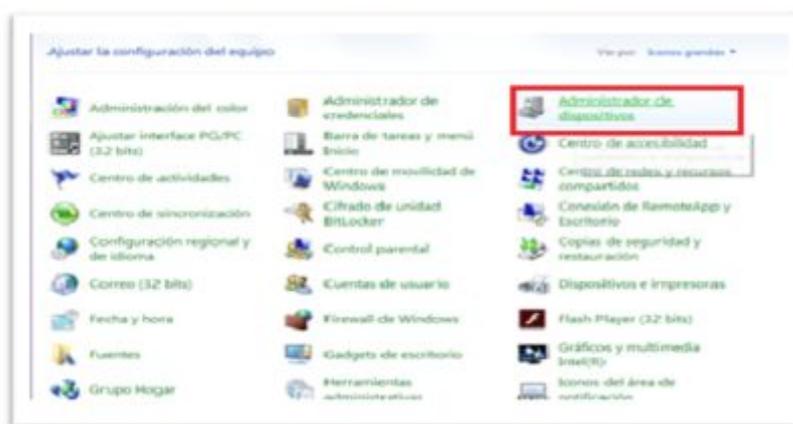
**Figura 31** Conexión USB entre la tarjeta Arduino Mega y la PC.

**Fuente:** (Alvarez, 2014)

### 3.4.3 Instalación de los controladores en la computadora

Al enchufar una placa Arduino en un equipo con Windows, es necesario instalar el controlador del puerto **Arduino Mega COM**. El controlador configura el Arduino como un puerto COM virtual siguiendo los pasos que se detallan a continuación:

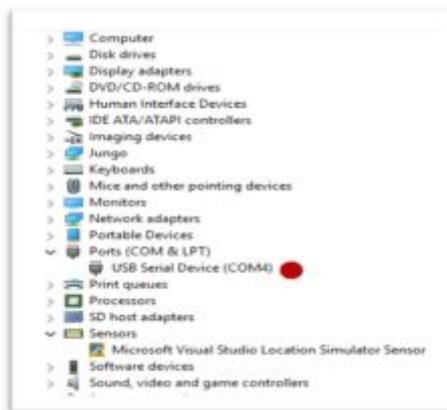
En la barra de inicio y abrir la ventana panel de control, dar doble clic en administrador de dispositivos, como se indica en la figura 32.



**Figura 32** Administración de dispositivos.

**Fuente:** (Arduino, Arduino Genuino , 2016)

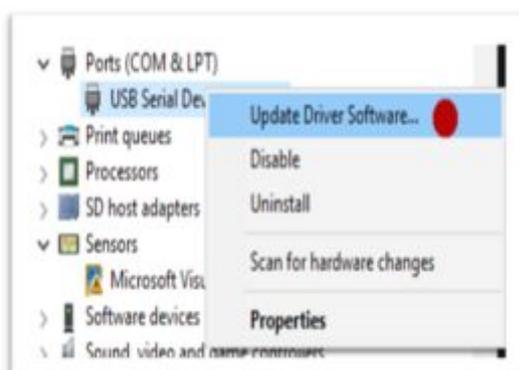
En la ventana que se abrirá, dar clic en **Puertos (COM y LPT)**, y seleccionar la opción **USB Serial Device (COM)**, como se indica en la figura 33.



**Figura 33** USB serial Device (COM)

**Fuente:** (Arduino, Arduino Genuino , 2016)

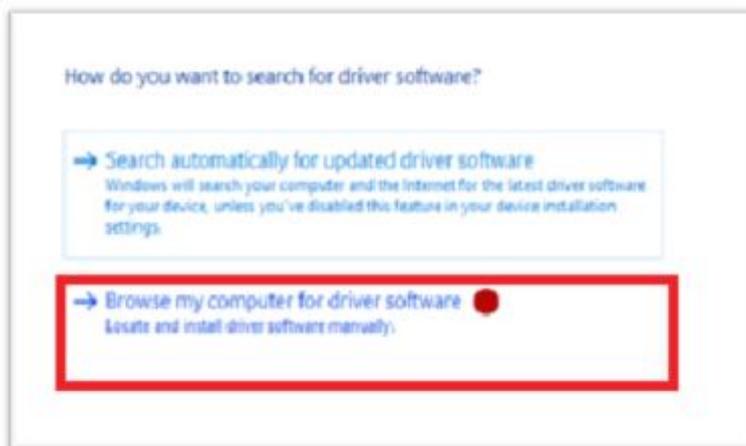
En el Administrador de dispositivos, haga clic en el **puerto COM 12** que se identificó como el Arduino (Arduino debe estar conectado al equipo) y seleccione **Actualizar software de controlador** en el menú que aparece, como se muestra en la figura 34.



**Figura 34** Actualizar software de controlador

**Fuente:** (Arduino, Arduino Genuino , 2016)

Se abrirá una nueva ventana en la cual se debe seleccionar **Buscar software de controlador** como se muestra en la figura número 35.



**Figura 35** Software de controlador

**Fuente:** (Arduino, Arduino Genuino , 2016)

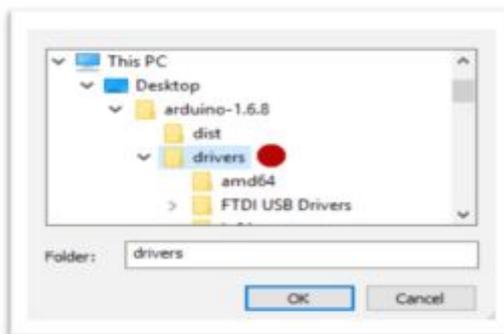
Se abre una nueva ventana de dialogo, en la cual se marca **incluir subcarpetas**, y luego se da clic en **examinar**, como se muestra en la figura 36.



**Figura 36** Examinar la carpeta

**Fuente:** (Arduino, Arduino Genuino , 2016)

Luego abrir la carpeta descomprimida anteriormente para la instalación de software Arduino IDE, en ésta se encontrará los drivers, hacer clic en la **Carpeta drivers** y a continuación dar clip en el botón **Aceptar** como se muestra en la figura 37.



**Figura 37** Instalación de los drivers

**Fuente:** (Arduino, Arduino Genuino , 2016)

Se mostrará un cuadro de diálogo **Actualizar software del controlador**, donde se dará clic en el botón **Siguiente** para iniciar la instalación del controlador, como se muestra en la figura 38.



**Figura 38** Actualizar software del controlador.

**Fuente:** (Arduino, Arduino Genuino , 2016)

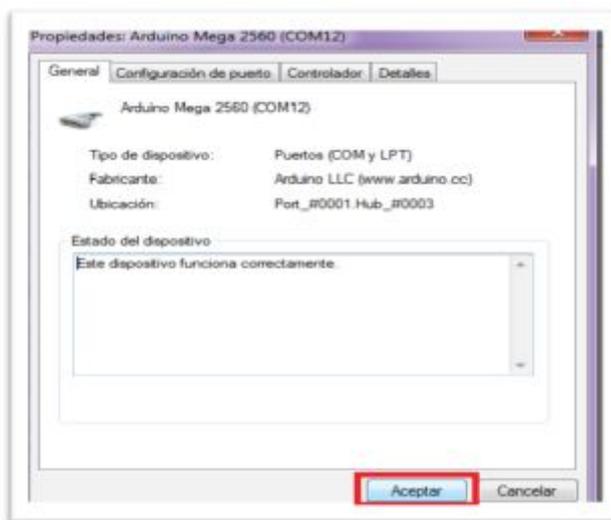
En la siguiente ventana, seleccionar **instalar**, como muestra la figura 39.



**Figura 39** Seguridad de Windows

**Fuente:** (Arduino, Arduino Genuino , 2016)

Después de que el controlador ha sido instalado, aparecerá un cuadro de diálogo que indica que el controlador ha sido actualizado con éxito y se mostrará el número de puerto **COM Arduino**, como se muestra en la figura 40.



**Figura 40** Puerto COM12 Arduino Mega

**Fuente:** (Arduino, Arduino Genuino , 2016)

Para verificar que el puerto COM esté instalado, ir al **Administrador de dispositivos**, el puerto COM se identifica ahora como un **Arduino**, como se muestra en la figura 41.

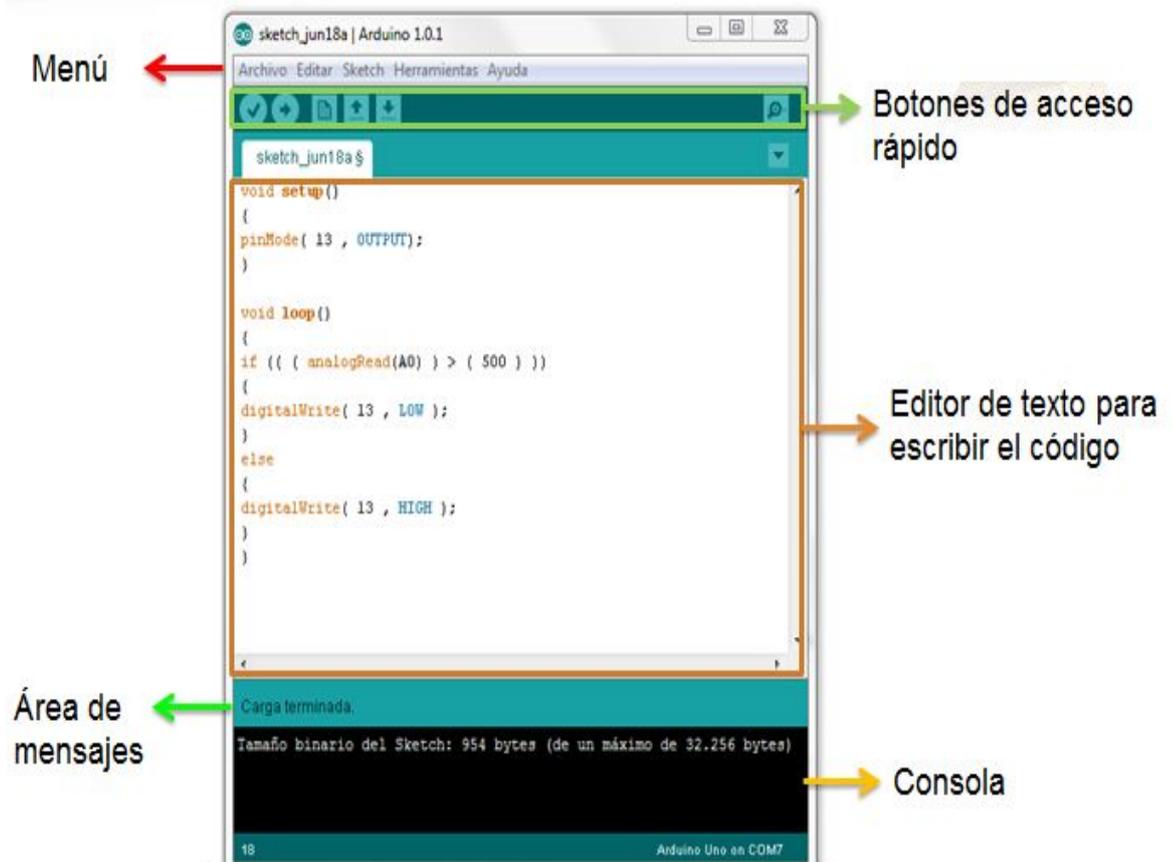


**Figura 41** Arduino Mega (COM)

**Fuente:** (Arduino, Arduino Genuino , 2016)

### 3.4.4 Identificación de los iconos en el software IDE Arduino

Para una forma fácil de trabajo se identificara los iconos básicos del entorno del software Arduino IDE, como se observa en la figura 42.



**Figura 42** Partes del software Arduino IDE.

**Fuente:** (Arduino, 2016)

### 3.4.5 Selección de la tarjeta Arduino mega que se usará

En la pestaña Herramientas de la barra de tareas, seleccionar Placa: "Arduino/Genuino Mega or Mega 2560" y después **Arduino Mega 2560**, como se indica en la figura 43.

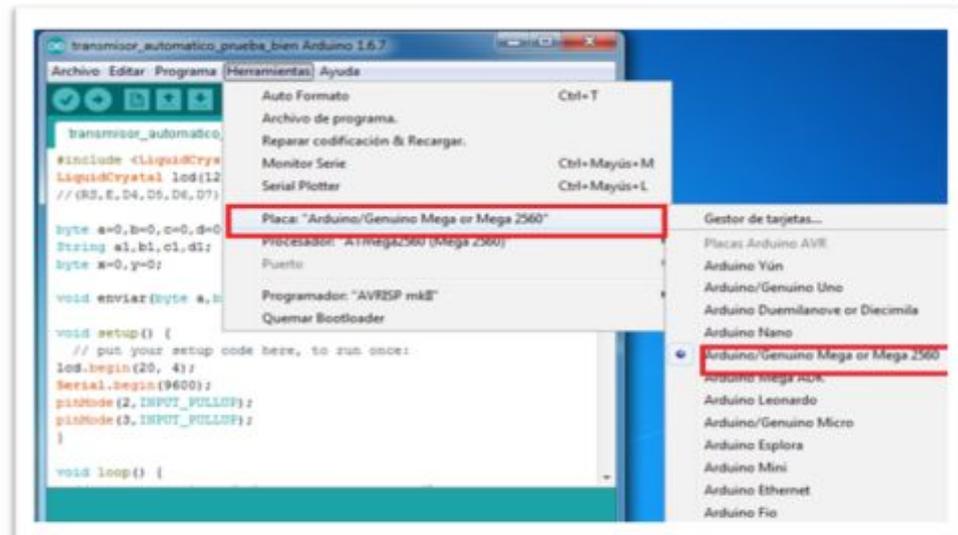


Figura 43 Seleccionar placa arduino mega.

Fuente: (Arduino, 2016)

### 3.4.6 Selección del puerto serial que usará Arduino

En la pestaña Herramientas de la barra de tareas, seleccionar el puerto serial con el que se trabajará, en este caso es el puerto **"COM 6(Arduino/Genuino Mega or Mega 2560)"**, como se muestra en la figura 44.

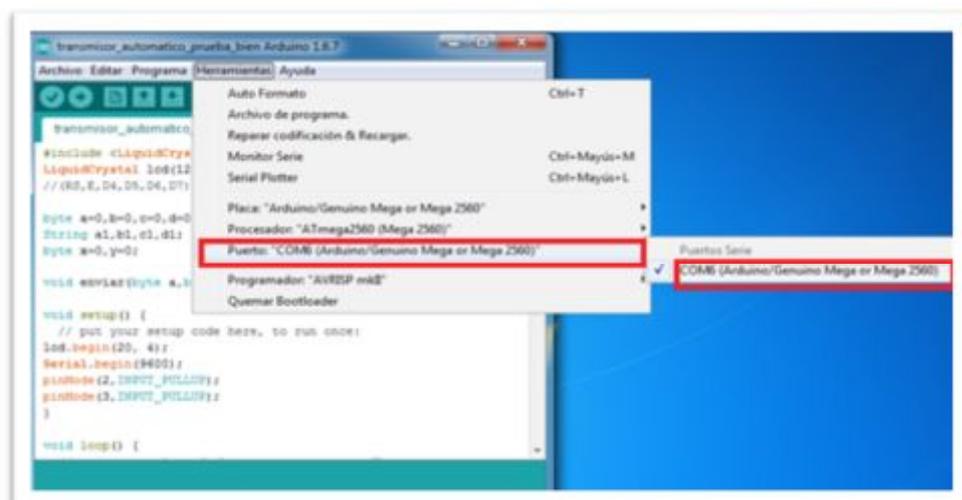


Figura 44 Puerto serial arduino mega.

Fuente: (Arduino, 2016)

### 3.4.7 Configuración del Motor Shield I293d

Para usar el motor shield se debe descargar las librerías necesarias como se indica a continuación:

#### 3.4.7.1 Descargar la librería para la motor shield L293d

La página <http://playground.arduino.cc/Main/LibraryList> descargar esta librería `#include <AFMotor.h>` para utilizar motor shield L293d y los motores paso a paso, como se muestra en la figura 45.

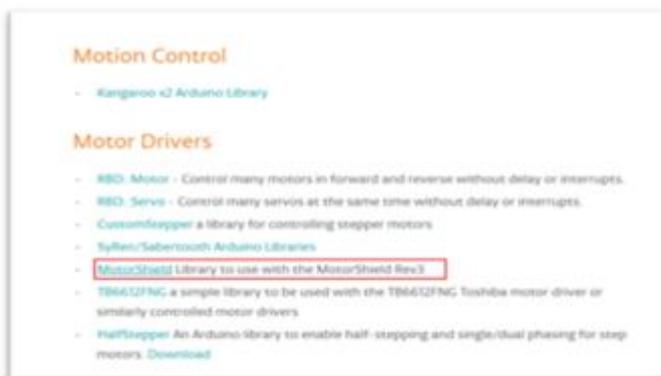


Figura 45 Librería para la motor shield I293d

Fuente: (Arduino, 2016)

#### 3.4.7.2 Cargar la librería para la Motor Shield I293d

Para cargar la librería, abrir el software Arduino IDE seleccionar en la barra de herramientas en la pestaña programa **importar librería**, se abre una ventana de dialogo donde se debe seleccionar **Añadir librería**, como se muestra en la figura 46.

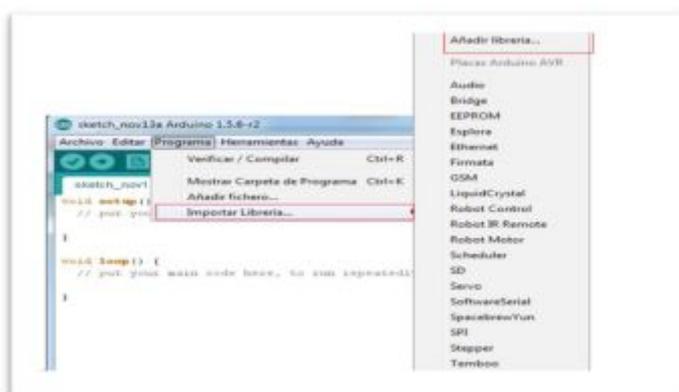
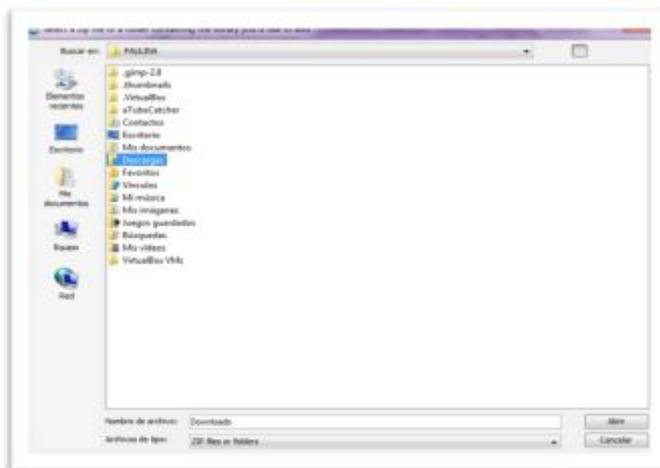


Figura 46 Seleccionar Importar librería

Fuente: (Arduino, 2016)

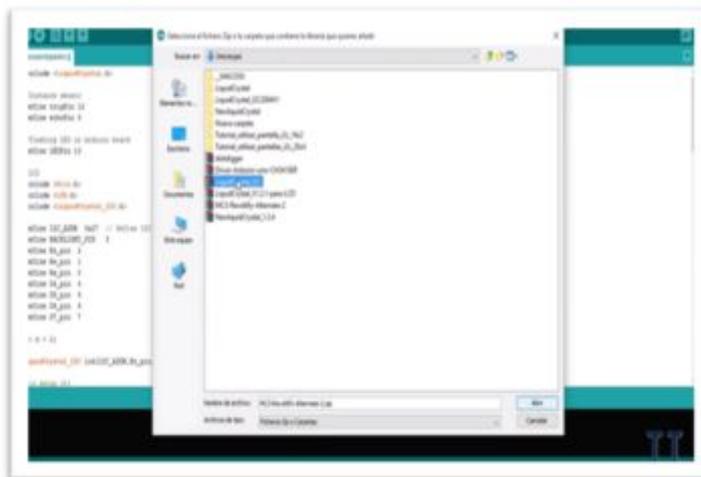
Dar clic en **Añadir librería**, se mostrará varias carpetas donde se debe seleccionar la carpeta **descargas**, como se muestra en la figura 47.



**Figura 47** Carpeta Descargas

**Fuente:** (Arduino, 2016)

Descomprimir el archivo con el nombre **motor shield I293d**, luego dar clic en la opción **abrir** con eso la librería ya está instalada como se observa en la figura 48.

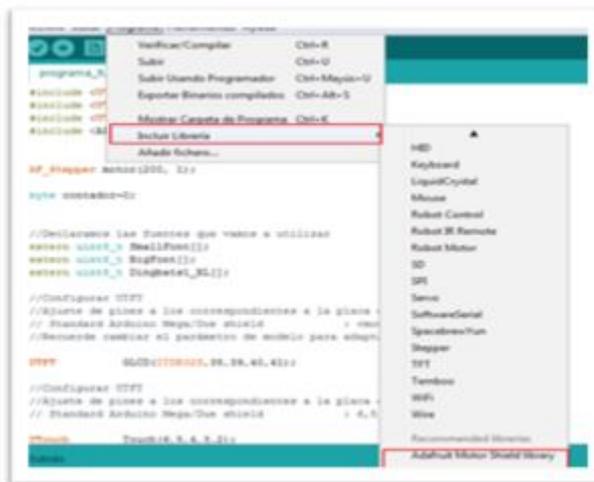


**Figura 48** Instalación de la librería motor shield I293d

**Fuente:** (Arduino, 2016)

Para verificar si la librería está instalada seleccionar **Programa** de la barra de herramientas del software Arduino IDE, se abre una nueva ventana de

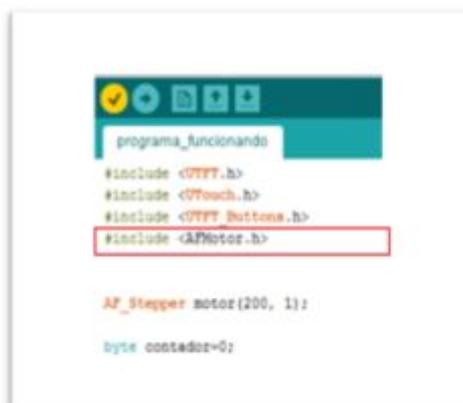
dialogo que abrirá seleccionar **incluir librería**, donde se mostrará la librería con el nombre **“Adafruit Motor Shield Library”** lo que confirma la figura 49.



**Figura 49** Verificación de la librería motor shield I293d

**Fuente:** (Arduino, 2016)

Para incluir la librería en el programa, se debe ir a la pestaña programa de la barra de herramientas en incluir librería y seleccionar **“Adafruit Motor Shield Library”** como se observa en la figura 50.



**Figura 50** Imagen del `#include <AFMotor.h>`.

**Fuente:** (Arduino, 2016)

### 3.4.8 Configuración de la TFT LCD Touch 3,2 pulgadas

Para conectar y programar la TFT LCD Touch 3,2 pulgadas se siguen los siguientes pasos:

### 3.4.8.1 Descargar las librerías

Descargar las librerías de la página

<http://playground.arduino.cc/Main/LibraryList>. Para utilizar la Touch, la pantalla LCD y los botones las librerías que deben cargarse son:

```
#include <UTFT.h>           // librería para la pantalla
#include <UTouch.h>         // librería para la touch
#include <UTFT_Buttons.h>   // librería para los botones
```

### 3.4.8.2 Cargar las librerías

Para cargar las librerías en Arduino abrir el software Arduino IDE realizar los pasos indicados para la librería de la motor shield I293d pero en este caso se cargara 3 librerías que son para la Touch, la LCD y los botones como se muestra en la figura 51.



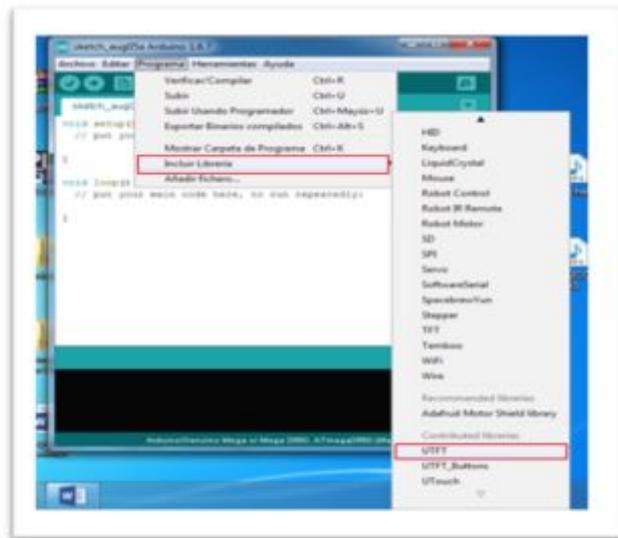
Figura 51 Abrir el software arduino IDE

Fuente: (Arduino, 2016)

### 3.4.8.3 Verificar las librerías si están cargadas

#### Librería para la pantalla

Seleccionar en la barra de herramientas la opción **programa** del software Arduino IDE se abre una ventana de dialogo indicar la opción **incluir librería**, en la cual se debe escoger la librería **UTFT** esta librería es para todo lo relacionado con la pantalla LCD, como lo confirma la figura 52.

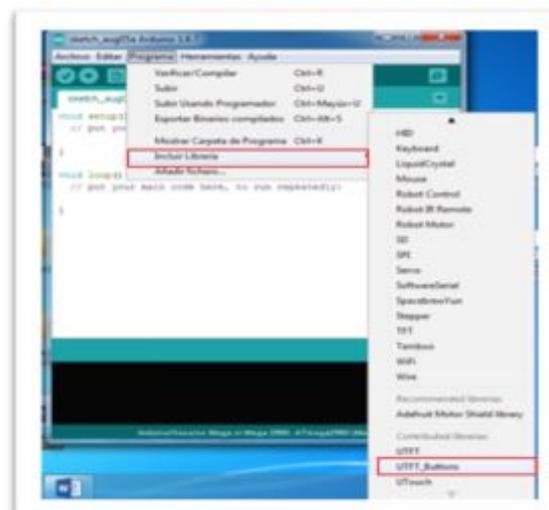


**Figura 52** Librería UTFT para la pantalla

**Fuente:** (Arduino, 2016)

### Librería para los botones

Seleccionar en la barra de herramientas la opción **programa** del software Arduino IDE se abre una ventana de dialogo indicar la opción **incluir librería**, en la cual se debe escoger la librería **UTFT\_Butons** esta librería permite crear botones, como son botones de contexto; se muestra en la figura 53.



**Figura 53** Librería UTFT\_Butons.h para los botones

**Fuente:** (Arduino, 2016)

## Librería para la touch

Seleccionar en la barra de herramientas la opción **programa** del software Arduino IDE, se abre una ventana de dialogo escoger la opción **incluir librería**, la cual se debe escoger la librería **UTouch** esta librería activa la pantalla táctil, como se observa en la figura 54.



**Figura 54** Libreria UTouch

**Fuente:** (Arduino, 2016)

Para finalizar seleccionar las 3 librerías con las cuales se trabajará en la programación, como lo confirma la figura 55.

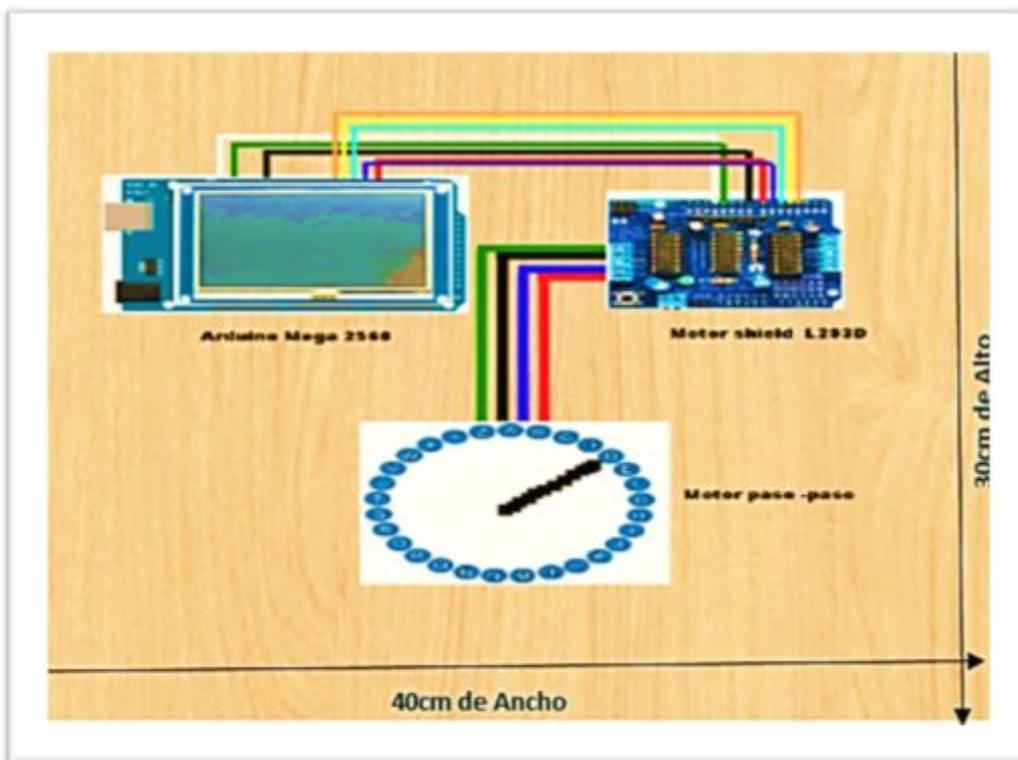


**Figura 55** Librerias para trabajar en arduino

**Fuente:** (Arduino, 2016)

### 3.5 Implementación del módulo

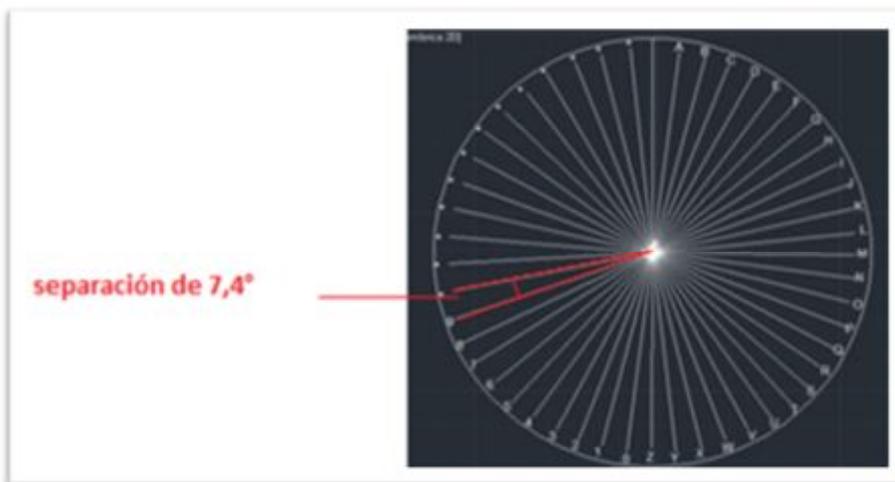
Para un mejor desempeño de los estudiantes al momento de realizar sus prácticas, los módulos tendrán conexiones externas mismas que permitan observar lo que los estudiantes están realizando. En la Figura 56 se muestra el diseño del módulo a realizar en el cual se colocaran los diferentes dispositivos a utilizar.



**Figura 56** Diseño del módulo

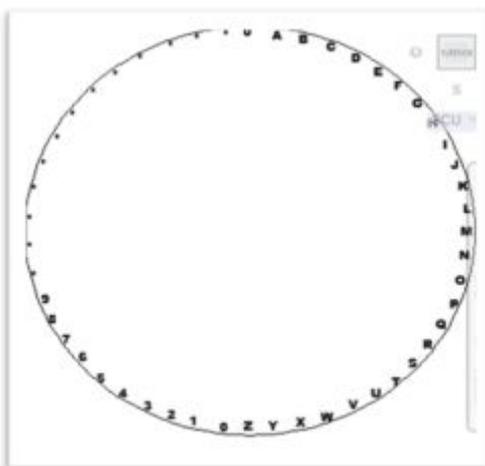
#### 3.5.1 Construcción del módulo

Para la construcción del módulo se realizaron las siguientes actividades:  
Se diseñó en AutoCAD la cubierta para los motores paso a paso, el cual consiste en imprimir cada letra del alfabeto en forma de una ruleta de la A a la Z y los números del 0 al 9, Con una separación de 7.4 grados como se muestra en la figura 57.



**Figura 57** Diseño en AutoCAD para la cubierta

Luego se imprimirá en un CD la cubierta realizada en AutoCAD para los motores paso a paso como se observa en la siguiente figura 58.



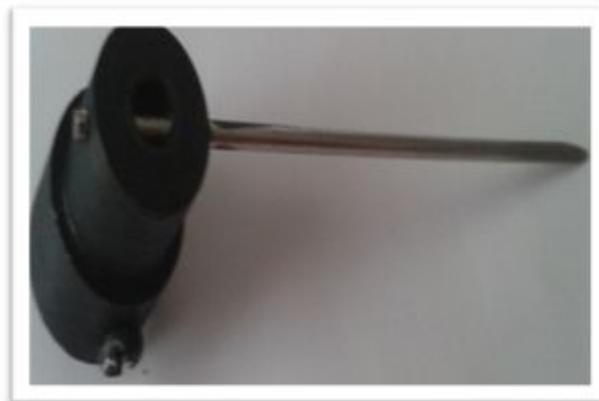
**Figura 58** Cubierta en AutoCAD para imprimir

Se realizó la perforación en los CD's para sujetar los motores paso a paso los cuales servirán como base, como se muestra en la figura 59.



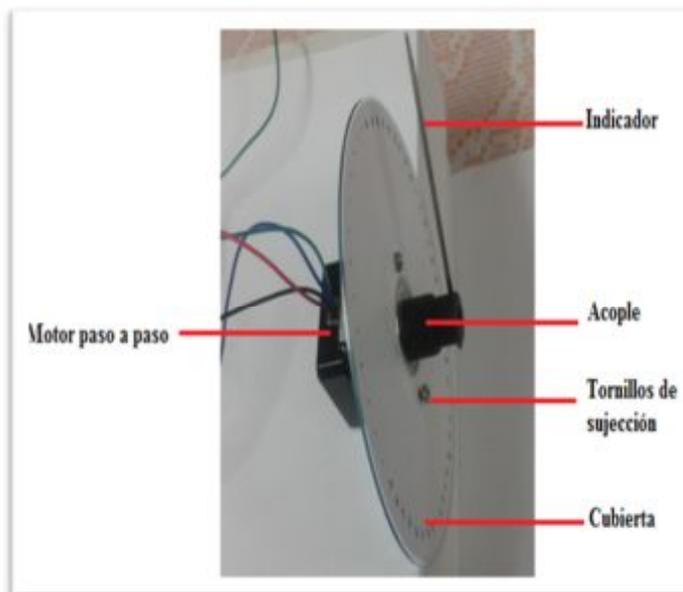
**Figura 59** Perforación de los CD's

Se realizó un acople para cada uno de los motores paso a paso, el cual servirá como indicador de cada letra al momento que el motor de un paso (a manera de una ruleta) como se muestra en la figura 60.



**Figura 60** Acople para los motores paso a paso

Finalmente, la base de los motores paso a paso queda, como se indica en la figura 61.



**Figura 61** Base para los motores paso a paso

Con los tableros ya recortados para los módulos, se procedió a perforar cada tablero para colocar los dispositivos que conforman cada módulo.

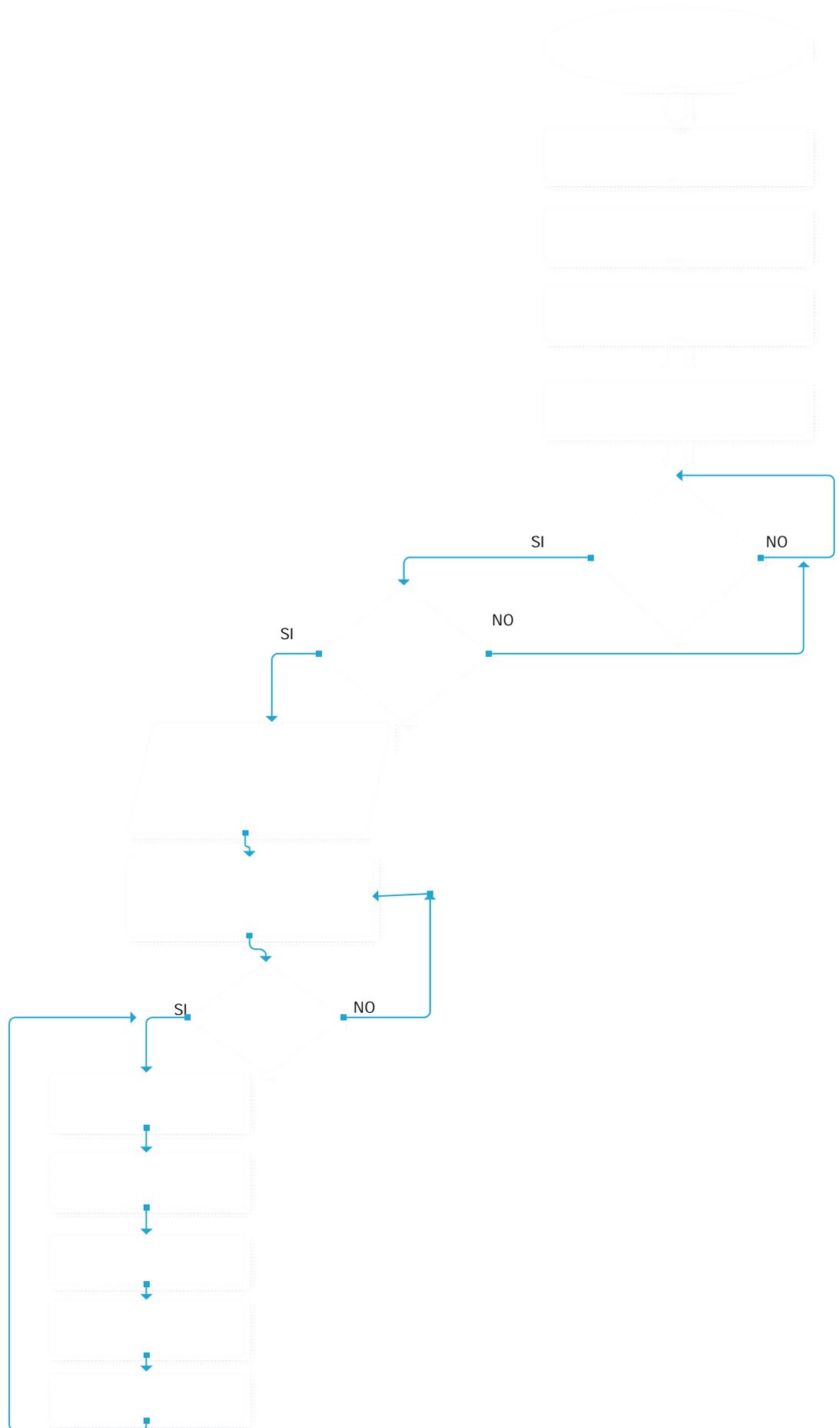
Ya elaboradas las perforaciones de cada tablero, se realizó la distribución de los dispositivos (Arduino Mega, Motor shield, motores paso a paso) formando así el respectivo módulo, como indica la figura 62.



**Figura 62** Módulo Armado

### 3.5.2 Diagrama de flujo de la programación

Representa los pasos ordenados a seguir en la programación.



### 3.5.3 Programación

Se establece la programación que es de importancia para el correcto desempeño de las acciones a programar en el ingreso de hasta 10 letras. Programación completa ver en (ANEXO A). La programación para ingresar 4 palabras ver en el (ANEXO B).

#### ***Para la programación en la TFT LCD Touch 3 ,2 pulgadas***

```
#include <URTouch.h>           // librería para la touch
#include <UTFT.h>              // librería para la pantalla
#include <UTFT_Buttons.h>      // librería para los botones
#include <AFMotor.h>           // librería para los motores paso a paso

#define RESTART asm("jmp 0x0000") //Salte a la localidad CERO de la memoria de
programa

#define ancho    31           //Alto de las teclas
#define alto     30           //Ancho de las teclas
#define hspace   1           //Espacio horizontal entre teclas
#define vspace   2           //Espacio vertical entre teclas
#define voffset  113         //Posición vertical del teclado
#define fila0y0  (alto+vspace)*0 + voffset //((30+2)*0 + 113 = 113  Coordenada Y superior
de la fila 0

extern uint8_t BigFont[];      //Incluyo los tipos de letra
extern uint8_t SmallFont[];    //Incluyo los tipos de letra
extern uint8_t GroteskBold24x48[]; //Incluyo los tipos de letra

int tx, ty;                    //Sirve para obtener la coordenada de donde se presionó la
pantalla
char letra;                    //La letra que se presiono
String texto = "";             //El texto que se va formando
int fc[3];                     //Color del relleno
int lc[3];                     //Color de la línea
int bc[3];                     //Color de fondo atrás de la letra

UTFT    myGLCD(ITDB32S,38,39,40,41); //Configuración de la UTFT modelo
URTouch myTouch(6,5,4,3,2);         //Configuración de la URTouch modelo
AF_Stepper motor(200, 1);           //Revoluciones del motor

void setup() {
```

```

myGLCD.InitLCD();           // Iniciar pantalla
myGLCD.clrScr();           // Limpiar pantalla
myGLCD.setFont(BigFont);   // Tamaño y tipo de letra
myTouch.InitTouch();       // Iniciar Touch
myTouch.setPrecision(PREC_MEDIUM); // Precisión de la touch

```

### ***Pantalla de presentación***

```

void caratula() {
boolean continuar = false;
int xBtn = 119;           //Oposición en X del botón continuar
int yBtn = 179;           //Oposición en Y del botón continuar
int wBtn = 200;           //Ancho del botón continuar
int hBtn = 60;           //Alto del botón continuar
myGLCD.setFont(BigFont); //Cambio el tipo de letra
drawBanner( 0, 10, 319, 10, "UNIDAD DE GESTION DE" );
drawBanner( 0, 25, 319, 10, " TECNOLOGIAS" );
drawBanner( 0, 40, 319, 10, "LATAACUNGA" );
int largo = texto.length(); //Cuantas letras tiene el texto a ser mostrado en la tecla
wfont = myGLCD.getFontXsize(); //Obtengo el ancho de la letra

```

### ***Dibujo el teclado en la pantalla***

```

void drawKeyboard() {           //Dibuja el teclado en pantalla
myGLCD.setFont(BigFont);       // Tipo de letra
drawButton( colu0x0, fila0y0, ancho, alto, "Q" ); //Dibujo el botón "Q" con el ancho , alto de
                                                la fila y al columna
drawButton( colu1x0, fila0y0, ancho, alto, "W" ); //Dibujo el botón "W" con el ancho , alto de
                                                la fila y al columna
myGLCD.setFont(SmallFont);
drawButton( colu8x0, fila2y0, (ancho + hspace)*2 - hspace, alto, "Enter" ); //Dibujo el botón
                                                "Enter" con el ancho , alto de la fila y al
                                                columna
int largo = texto.length();     //Cuantas letras tiene el texto a ser mostrado en la tecla
wfont = myGLCD.getFontXsize(); //Obtengo el ancho de la letra

```

### ***MANEJO DEL MOTOR***

```

void string2index(String cadena, int LENGTH) { //Recibe el texto y calcula sus índices
Serial.print("LENGTH: ");                   // Imprime el texto

```

```

for (int i = 0; i < LENGTH; i++){ //En este bucle separo el texto letra por letra y calculo el
                                índice de cada letra

//int letra = cadena[i];          // restando 64, está basado
Serial.println(cadena[i]);        // en el código ASCII
//Serial.print(" = ");           // donde la letra "A" mayúscula
//Serial.println(letra - 64);     // se corresponde al numero
//Serial.println();               // 65
indice[i] = cadena[i] - 64;
}

void indices (int indice[], int LENGTH) { //Sirve para calcular cuantos pasos hay que
                                          moverse de acuerdo a cada secuencia de letras
index[0] = indice[0]; //Obtengo la primera
index[0] *= 4; //multiplico el índice de la letra por 4,
               para que se corresponda con el paso del motor
for (int i = 1; i < LENGTH; i++){ //En este bucle se calculan los pasos en función de la
                                   letra anterior

index[i] = indice[i] - indice[i-1]; // se van almacenando en la variable
index[i] *= 4;                      // index
Serial.print ("Puntero: ");
Serial.println(index[i]);
}
puntero(index, LENGTH);
}

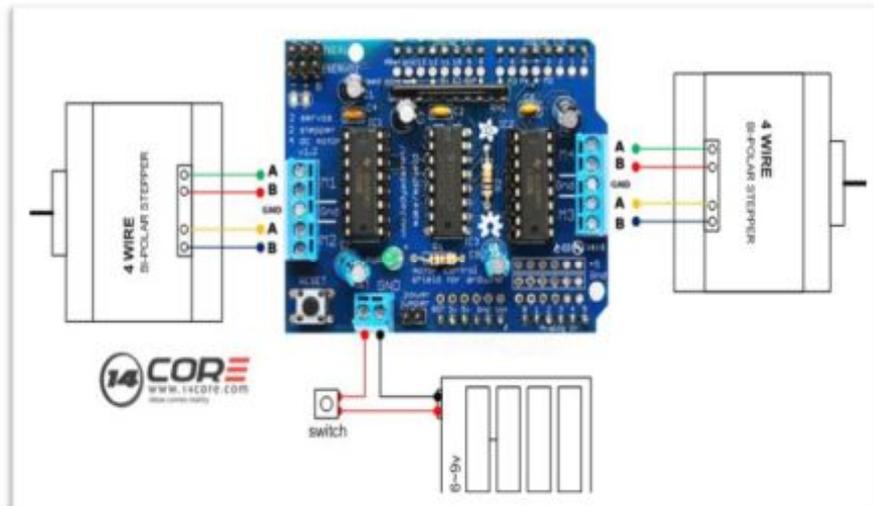
void puntero (int pasos[], int LENGTH) { //Le indico cuantos pasos moverse, y cuantas
                                          letras son

//pasos puede ser positivo o negativo. Si es positivo es antihorario, si es negativo es horario
for (int i = 0; i < LENGTH; i++){ //Aplico esto para cada letra
motor.release();                  //Libero el motor
if (pasos[i] > 0)                  //Verifico si es horario o antihorario el movimiento
motor.step(pasos[i],BACKWARD, SINGLE);
else
motor.step(abs(pasos[i]),FORWARD, SINGLE);
delay(3000);                       //Apunto a la letra durante 3 segundos
}
}

```

### 3.6 Conexiones

#### Motor shield I293d con la fuente de alimentación



**Figura 63** Motor shield con fuente

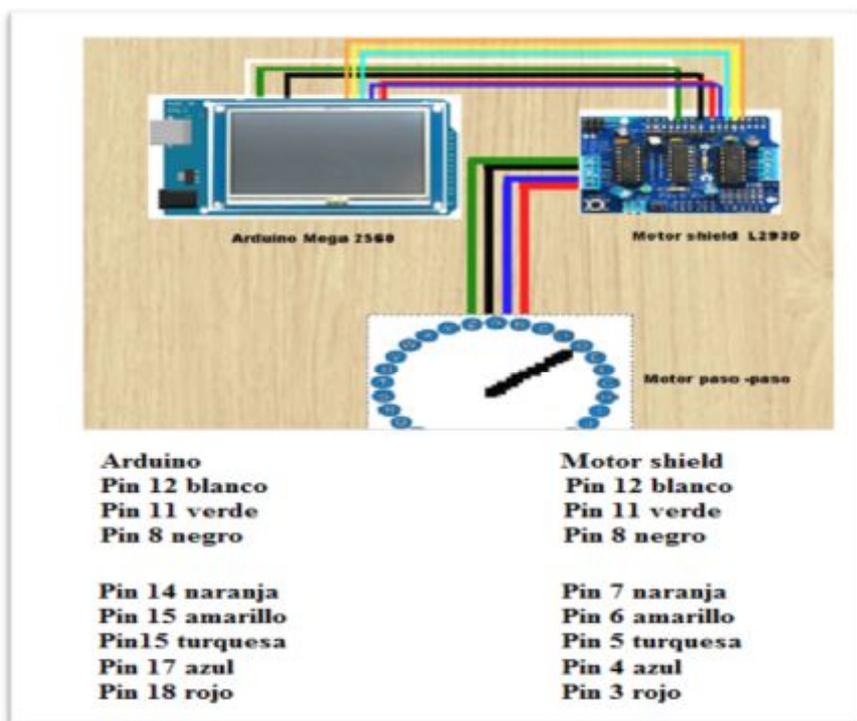
**Autor:** (Core, 2016)



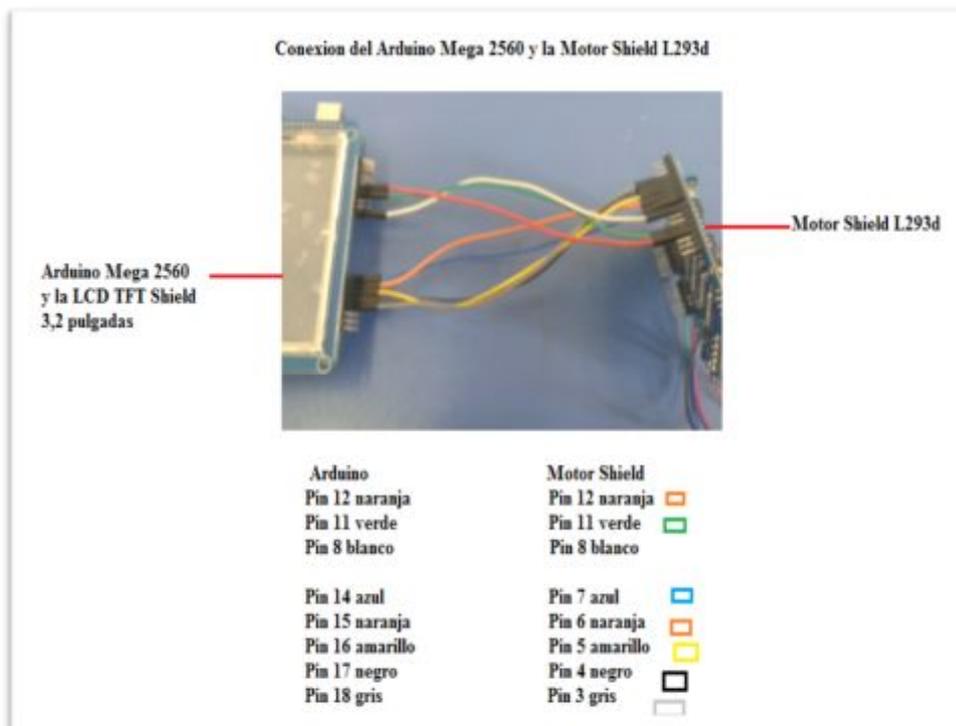
**Figura 64** Arduino Mega, Shield y LCD TFT Touch

**Autor:** (Arduino, 2016)

## Conexión de Arduino mega y la shield L293d



**Figura 65** Pines de conexión



**Figura 66** Pines de alimentación

### 3.7 Operación de la pantalla

Presentación de la pantalla, consta de un título en la parte superior y en la parte inferior derecha el botón continuar, como se muestra en la figura 67.



**Figura 67** Pantalla de inicio

Para observar su funcionamiento se debe presionar el botón CONTINUAR >>, como indica la figura 68.



**Figura 68** Teclado

Se abre una nueva interfaz con la aplicación realizada un teclado, la cual consiste en escribir una palabra (HOLA) misma que será deletreada en el módulo, como se muestra en la figura 69.



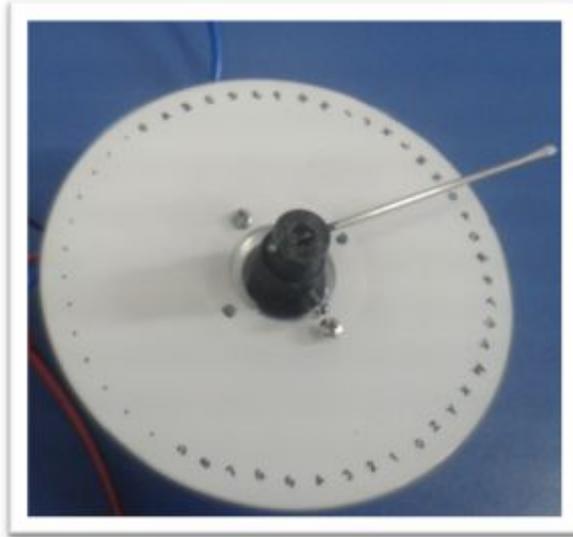
**Figura 69** Palabra HOLA

Dar **Enter** en el teclado esperar unos segundos el teclado se pondrá en Rojo, como se muestra en la figura 70.



**Figura 70** Clic en ENTER

El teclado en color rojo indica que se está ejecutando la acción (indicador girando de acuerdo a las letras que contiene la palabra presionada (**HOLA**), como se muestra en la figura 71.



**Figura 71** Deletreo de la palabra HOLA

Siempre se debe encerrar el indicador al momento de empezar el funcionamiento de la pantalla.

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

- Se indagó las características y el funcionamiento de la LCD TFT Shield 3,2 pulgadas, permitiendo comprender de mejor manera todos los parámetros necesarios para un correcto funcionamiento de la misma y así poder implementar el modulo didáctico.
- Se implementó un módulo que permite el control de dispositivos electrónicos en el cual se realizó las conexiones, alimentación de acuerdo a los datos establecidos en las hojas técnicas, este módulo ayudara a mejorar el proceso de enseñanza – aprendizaje ya que el módulo quedará en el laboratorio de Instrumentación Virtual para que los alumnos realicen prácticas .
- La aplicación que se realizó para el control de los motores paso a paso consiste en escribir una palabra en la pantalla LCD TFT shield 3.2 pulgadas; y el motor empezará a girar y señalar con el indicador letra por letra hasta formar la palabra escrita, se permite ingresar hasta 10 caracteres (letras), si hay más de 10 letras ya no se ejecuta la acción. Al finalizar la orden (escribir palabra) el indicador regresa a la posición “0”, donde se encontrará esperando hasta recibir una nueva palabra.

## 4.2 Recomendaciones.

- Revisar las especificaciones y características técnicas de los dispositivos a utilizar, para realizar una correcta conexión y evitar que se quemen o dañen por malas conexiones.
- Tener mucho cuidado con los errores de sintaxi en la programación ya que, son los más comunes.
- Para trabajar con motores P-P en especial los unipolares o bipolares es necesario poder identificar cuáles son las bobinas A, B, C, y D, para esto simplemente medimos las resistencias de cada una de las bobinas.
- Para el uso de la motor shield se necesita de una fuente externa de 7,4 V
- Verificar que el software Arduino IDE contenga estas 4 librerías de suma importancia para su respectiva programación, mismas que permiten controlar la pantalla táctil, los botones, motor shield L293d y los motores paso a paso estas librerías son:

```
#include <URTouch.h>           // librería para la touch
#include <UTFT.h>               // librería para la pantalla
#include <UTFT_Buttons.h>      // librería para los botones
#include <AFMotor.h>           //librería para los motores paso a
paso
```

- Verificar que todos los terminales de alimentación del módulo, estén funcionando correctamente para evitar cortocircuitos.
- Siempre se debe encerrar el indicador al momento de empezar el funcionamiento de la pantalla.

## GLOSARIO DE TÉRMINOS

### A

**AMOLED**      Diodo orgánico de emisión de luz de matriz activa

### C

**CA**              Corriente alterna

**CC**              Corriente continúa

### E

**EEPROM**      Electrically Erasable Programmable Read-Only Memory

### F

**FFD**            Dispositivo de funcionalidad completa

### G

**GND**            Son los pines a tierra de la placa, el negativo

**GTS**            Ranura de Tiempo Garantizado

### I

**IOREF**        Sirve para que la placa reconozca el tipo de alimentación

**ID**              PAN ID

**IDE**            Entorno de desarrollo integrado

**IEEE**         Instituto de Ingenieros Eléctricos, Electrónicos

### K

**KB**             Kilobytes

**Kbps**         Kilobites por segundos

### L

<b>LCD</b>	Liquid crystal display
<b>L293D</b>	Driver de 4 canales una corriente de 600mA
<b>P</b>	
<b>P-P</b>	Motor paso a paso
<b>R</b>	
<b>RESET</b>	Por este pin se puede reiniciar la placa
<b>S</b>	
<b>SRAM</b>	Memoria estática de acceso aleatorio (o RAM estática)
<b>T</b>	
<b>TTL (5V)</b>	Lógica transistor a transistor trabaja con 5V
<b>TWI</b>	Training Within Industry
<b>TFT</b>	Transistor de películas finas
<b>TFT-LCD</b>	Thin Film Transistor-Liquid Crystal Display
<b>U</b>	
<b>ULN2803</b>	Existen 8 transistores NPN Darlington
<b>USB</b>	Bus Universal en Serie
<b>V</b>	
<b>Vin</b>	Voltaje de entrada

## REFERENCIAS BIBLIOGRAFICAS

- Alicante, U. d. (2015). *GitBook*. Obtenido de GitBook: [https://mastermoviles.gitbooks.io/tecnologias2/content/caracteristicas\\_y\\_hardware\\_de\\_los\\_dispositivos\\_moviles.html](https://mastermoviles.gitbooks.io/tecnologias2/content/caracteristicas_y_hardware_de_los_dispositivos_moviles.html)
- Alvarez, I. (Febrero de 2014). *Batraco*. Obtenido de <http://batrako.blogspot.com/2013/02/posibilidades-del-arduino-en-simracing.html>
- Amazon. (2016). *Ingeniería y Diseño Electrónico*. Obtenido de Ingeniería y Diseño Electrónico: [https://www.amazon.es/SainSmart-Pantalla-Arduino-adaptado-tarjeta/dp/B008HWL8QS/ref=sr\\_1\\_6?ie=UTF8&qid=1481432516&sr=8-6&keywords=Arduino+Tft+Lcd](https://www.amazon.es/SainSmart-Pantalla-Arduino-adaptado-tarjeta/dp/B008HWL8QS/ref=sr_1_6?ie=UTF8&qid=1481432516&sr=8-6&keywords=Arduino+Tft+Lcd)
- Android*. (6 de mayo de 2012). Obtenido de La web del Android: <http://www.todorobot.com.ar/tutorial-sobre-motores-paso-a-paso-stepper-motors/>
- Arduino. (Marzo de 2016). Obtenido de Arduino cc
- Arduino. (Marzo de 2016). *Arduino Genuino* . Obtenido de Arduino Genuino : <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- Arduino. (2016). *Arduino mega*. Obtenido de arduino.cc: <https://www.arduino.cc/en/Main/arduinoBoardMega>
- Arduino. (2016). *Arduino Motor Shield* . Obtenido de Arduino Motor Shield : <http://playground.arduino.cc/Main/AdafruitMotorShield>
- Burguillos. (2014). *Teoría de arduino* .
- Caldas. (2016). *Caldas Electronicos*. Obtenido de Caldas Electronicos: <http://www.electronicoscaldas.com/shields-escudos-arduino/511-shield-de-motores-para-arduino-shd-mstepper.html>
- Circelli, G. (14 de Febrero de 2015). *Panama Hitek* . Obtenido de Panama Hitek : <http://panamahitek.com/microcontroladores-o-arduino-no-es-lo-mismo/>

- Core. (2016). *Arduino Motor Shield*. Obtenido de Arduino Motor Shield:  
<http://www.14core.com/controlling-the-l293d-unomega-shield-with-stepper-motor/>
- Diy Makers. (28 de 12 de 2013). *Diy Makers*. Obtenido de Diy Makers:  
<http://diymakers.es/mover-motores-paso-paso-con-arduino/>
- Diy Makers. (2013). *Diy Makers Motores paso a paso* . Obtenido de Diy Makers Motores paso a paso : <http://diymakers.es/mover-motores-paso-paso-con-arduino/>
- Fernando, L. M. (2014). Definición y Funcionamiento de Microprocesadores y Microcontroladores. En L. M. Fernando, *Sistemas Digitales 3* (pág. p.2).
- González, A. G. (20 de Mayo de 2015). *Panama hitek* . Obtenido de Panama hitek : <http://panamahitek.com/que-es-arduino-y-para-que-se-utiliza/>
- Laso, D. W. (2013). *Asignatura de Perifericos* . E3086221.
- Menéndez, R. (8 de julio de 2016). *Monitores TFT*. Obtenido de Monitores TFT:  
<http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Monitores-TFT.html>
- Microcontroladores . (Diciembre de 2014). Obtenido de <http://losmicrocontroladores.blogspot.com/>
- Polulo . (2016). *Polulo* .
- Sherlin, H. (2014). Obtenido de <http://sherlin.xbot.es/microcontroladores/introduccion-a-los-microcontroladores/que-es-un-microcontrolador>
- Shield Adaptador V2.2. (10 de Abril de 2013). *Shield Adaptador V2.2*. Obtenido de Shield Adaptador V2.2: [http://www.dx.com/es/p/lcd-tft01-mega-shield-v2-0-adapter-module-white-200503#.V\\_WzYMM-OVO](http://www.dx.com/es/p/lcd-tft01-mega-shield-v2-0-adapter-module-white-200503#.V_WzYMM-OVO)
- Souza. (2014). *embarcados.com*. Obtenido de <http://www.embarcados.com.br/arduino-mega-2560/stepper/stepping-motors>
- stepper / stepping motors. (2012). *stepper / stepping motors*. Obtenido de [http://www.teleco.com.br/es/tutoriais/es\\_tutorialzigbee/pagina\\_2.asp](http://www.teleco.com.br/es/tutoriais/es_tutorialzigbee/pagina_2.asp)
- Victor, G. F. (21 de Abril de 2013). *Fundamentos de Robotica*. Obtenido de Fundamentos de Robotica:

[http://platea.pntic.mec.es/vgonzale/cyr\\_0204/cyr\\_01/robotica/sistema/motores\\_p-p.htm#funcionamiento](http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/motores_p-p.htm#funcionamiento)

Weebly. (2015). *rduinodhtics*. Obtenido de rduinodhtics:  
<http://arduinohtics.weebly.com/>

Wikipedia. (09 de 12 de 2016). *Wikipedia* . Obtenido de Wikipedia :  
[https://es.wikipedia.org/wiki/Pantalla\\_de\\_cristal\\_L%C3%ADquido](https://es.wikipedia.org/wiki/Pantalla_de_cristal_L%C3%ADquido)

Wordpress. (23 de 03 de 2015). *wordpress*. Obtenido de shiel arduino:  
<https://aprendiendoarduino.wordpress.com/2015/03/23/shields-para-arduino/>

# ANEXOS



