

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:  
INGENIEROS ELECTRÓNICOS**

**TEMA:  
AUTOMATIZACIÓN DE UNA MÁQUINA MANUAL FABRICADORA DE  
BOBINAS DE CABLE DE COBRE MEDIANTE CONTROL BASADO EN  
VISIÓN ARTIFICIAL.**

**AUTORES:  
ACHIG QUISHPE JUAN CARLOS  
PAZMIÑO HIDALGO ANGEL FERNANDO**

**TUTOR:  
JAVIER ARTURO ROJAS URBANO**

**Quito, agosto 2018**

## DECLARACIÓN DE DERECHOS DE AUTOR

Nosotros Juan Carlos Achig Quishpe y Angel Fernando Pazmiño Hidalgo, con documento de identificación N° 172066696-3 y N° 172493428-4 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “AUTOMATIZACIÓN DE UNA MÁQUINA MANUAL FABRICADORA DE BOBINAS DE CABLE DE COBRE MEDIANTE CONTROL BASADO EN VISIÓN ARTIFICIAL”, mismo que ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

 Juan Carlos Achig Quishpe Cédula: 172066696-3	 Angel Fernando Pazmiño Hidalgo Cédula: 172493428-4
---	---

Quito, agosto 2018

## **DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR**

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico, “AUTOMATIZACIÓN DE UNA MÁQUINA MANUAL FABRICADORA DE BOBINAS DE CABLE DE COBRE MEDIANTE CONTROL BASADO EN VISIÓN ARTIFICIAL” realizado por Juan Carlos Achig Quishpe y Angel Fernando Pazmiño Hidalgo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, agosto 2018



Javier Arturo Rojas Urbano

C.I. 020189802-0

## **DEDICATORIA**

Dedico este proyecto a mi madre Maria Hermelinda Quishpe Llumiquinga en el cielo quien gracias a sus bendiciones durante todo el tiempo de preparación ha sido una inspiración. A mi padre Rafael Achig quien ha sido un aporte fundamental incondicional quien con su paciencia y consejo estuvo ahí para brindarme la fuerza necesario y seguir adelante, A mis hermanos, Carmita, Néstor, Edgar Achig, cada uno de ellos brindándome su ayuda, su consejo y sobre todo el valor de responsabilidad para terminar con esta carrera y sueño tan importante en mi vida.

Juan Carlos Achig Quishpe

Dedico este proyecto a mis padres por todo el apoyo que me han brindado a lo largo de toda mi vida, por sembrar en mí la responsabilidad para el cumplimiento de esta meta, ellos son mi guía, mi motor, mi fortaleza, mi ejemplo de perseverancia y de vida. A mis hermanos Daya y Pato con los que puedo contar siempre, que han aportado con su carisma y apoyo para la culminación del proyecto. A mi familia que por su unidad me han enseñado a trabajar en grupo y a salir adelante. A mis mejores amigos Fernando y Mauricio por su consejo y apoyo a lo largo de nuestra amistad. Y a mi novia Katherine por ser parte fundamental de mi vida y estar en los momentos que más la he necesitado.

Angel Fernando Pazmiño Hidalgo

## **AGRADECIMIENTO**

Agradecemos a la Universidad Politécnica Salesiana que junto con sus grandes docentes ha permitido formarnos como profesionales adquiriendo los conocimientos necesarios para poder desenvolvemos en el ámbito laboral con bases bien forjadas, así también formarnos como personas teniendo en cuenta siempre los valores éticos y profesionales para ser antes de buenos profesionales, excelentes personas.

Agradecemos al Ing. Fernando Naranjo quien colaboró con nosotros durante el proceso de realización del presente proyecto trabajando junto con nosotros brindando consejo, así como técnicas que permitan tener un proyecto de calidad.

También agradecemos a nuestro tutor de proyecto de titulación, la Ing. Javier Arturo Rojas Burbano por su asesoría durante la realización del proyecto técnico; también su guía durante la corrección de detalles, aclaramiento de dudas y sugerencias brindadas para el apropiado desarrollo del proyecto de titulación.

¡Gracias a ustedes!

## ÍNDICE DE CONTENIDO

DECLARACIÓN DE DERECHOS DE AUTOR	i
DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR	ii
DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE DE CONTENIDO	v
ÍNDICE DE FIGURAS	viii
ÍNDICE DE TABLAS	x
RESUMEN	xi
ABSTRACT	xii
INTRODUCCIÓN	xiii
CAPÍTULO 1	1
ANTECEDENTES	1
1.1. Problema de estudio	1
1.2. Justificación	1
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
1.4. Metodología	2
1.5. Beneficiarios	3
CAPÍTULO 2	4
FUNDAMENTACIÓN TEÓRICA	4
2.1. Bobina o Inductor	4
2.1.1. Características de la bobina.	4
2.1.1.1. Inductancia.	4
2.1.1.2. Permeabilidad Magnética (M)	4
2.1.1.3. Factor de Calidad (Q)	5

2.1.2.	Dimensionamiento	5
2.1.3.	Construcción de una bobina	5
2.1.4.	Tipos de bobinas de cobre	5
2.1.4.1.	Núcleo de aire.	5
2.1.4.2.	Núcleo sólido	6
2.2.	Proceso para la construcción de bobinas.	6
2.2.1.	Proceso Manual	7
2.2.2.	Proceso Semiautomático	8
2.2.3.	Proceso Automático	9
2.3.	Sistemas de control de calidad	10
2.3.1.	Control de resistencia óhmica	11
2.3.2.	Control de resistencia de aislamiento.	11
2.3.3.	Visión artificial.	11
2.3.3.1.	Definición.	12
2.3.3.2.	Aplicaciones	12
2.3.3.3.	Control de calidad con visión artificial.	12
2.3.3.4.	Filtros aplicados para visión artificial.	14
CAPÍTULO 3		15
DESARROLLO DEL PROYECTO		15
3.1.	Desarrollo del hardware.	15
3.1.1.	Sistema mecánico.	15
3.1.1.1.	Motor NEMA 23.	16
3.1.1.2.	Motor NEMA 17	17
3.1.2.	Sistema de adquisición de datos.	18
3.1.2.1.	Driver A4988	18
3.1.2.2.	Sensor final de carrera.	20
3.1.3.	Sistema de control de calidad	21
3.1.3.1.	Sony Camara eye PS3.	21
3.1.4.	Sistema de control y visualización.	22
3.1.4.1.	Controlador Arduino	23
3.1.4.2.	Raspberry PI Model B	23
3.1.4.3.	LCD 20 X 4	24

3.1.4.4. Pantalla de control Rasberry.	24
3.1.4.5. Teclado matricial 4x4	25
3.1.5. Fuente de alimentación.	25
3.1.5.1. Fuente de voltaje.	26
3.2. Desarrollo del software.	27
3.2.1. Algoritmo desarrollado en Arduino	28
3.2.2. Algoritmo desarrollado en Raspberry PI	30
CAPÍTULO 4.	33
ANÁLISIS DE PRUEBAS Y RESULTADOS	33
4.1. Prueba de Panel de control e ingreso de la información.	33
4.2. Pruebas de bobinado	34
4.3. Prueba de control de calidad	35
4.4. Prueba de aislamiento.	37
CONCLUSIONES	40
RECOMENDACIONES	42
REFERENCIAS	44

## ÍNDICE DE FIGURAS

Figura 2.1 Bobina núcleo de aire de diferente calibre. ....	6
Figura 2.2 Máquina bobinadora manual para fabricación de bobinas .....	7
Figura 2.3 Máquina eléctrica para fabricación de bobinas automáticas. ....	8
Figura 2.4 Máquina bobinadora automática.....	9
Figura 2.5 Proceso de control de calidad basado por visión artificial .....	13
Figura 3.1 Diagrama de bloques para funcionamiento de la máquina bobinadora automática .....	15
Figura 3.2 Motor NEMA 23 ORIENTAL MOTOR .....	16
Figura 3.3 Motor a pasos NEMA 17 .....	17
Figura 3.4 Estructura física Driver A4988. ....	19
Figura 3.5 Diagrama de conexión del driver A4988 a un motor a pasos.....	20
Figura 3.6 Diseño sensor interruptor final de carrera. ....	20
Figura 3.7 Cámara PS3 para visión artificial .....	21
Figura 3.8 Estructura física de tarjeta Arduino Mega. ....	23
Figura 3.9 Raspberry PI 3 model B.....	24
Figura 3.10 LCD 20 x 4 .....	24
Figura 3.11 PiTFT Raspberry 3.2'' .....	25
Figura 3.12 Teclado matricial 4 x4. ....	25
Figura 3.13 Fuente de Voltaje.....	26
Figura 3.14 Diagrama de conexiones de hardware de la máquina.....	27
Figura 3.15. Principio de funcionamiento de máquina bobinadora automática.....	27

Figura 3.16 Flujograma del algoritmo realizado en el Arduino.....	29
Figura 3.17 Flujograma del algoritmo realizado en la Raspberry PI.....	31
Figura 4.1 Panel frontal de la maquina bobinadora automática.....	33
Figura 4.2 Fabricación de bobina en la maquina automatizada.....	34
Figura 4.3 Bobinas construidas con diferentes calibres de cable de cobre AWG.....	35
Figura 4.4 Visión artificial en proceso de bobinado.....	36
Figura 4.5 Análisis estadístico de pruebas de visión artificial.....	37
Figura 4.6 Medición de resistencia óhmica en bobinas construidas.....	38

## ÍNDICE DE TABLAS

Tabla 3.1 Características Motor NEMA 23. ....	17
Tabla 3.2 Características técnicas motor a pasos NEMA 17 .....	18
Tabla 3.3 Tabla de características del final de carrera .....	21
Tabla 3.4 Características técnicas cámara PS3 .....	22
Tabla 4.1 Tabla de cálculos para resistencia de bobinas fabricadas .....	38

## **RESUMEN**

El presente proyecto técnico se desarrollará con la finalidad de automatizar una máquina manual fabricante de bobinas de cobre, en base a un control de visión artificial. Para la automatización de esta máquina manual primero se realizó una caracterización para conocer cada una de las partes necesarias para la automatización como motores, sensores, controladores, etc. Los controladores serán una tarjeta Arduino Mega y una Raspberry PI para la parte del control con visión artificial, el sistema controla básicamente tres motores a pasos, en el cual el primero se encarga de rotar el eje de la máquina y el segundo se encargará de guiar el cable en el carrete y el tercero en la corrección del cable en caso de existir un error en el proceso.

El ingreso de datos al sistema y la visualización se la realizará a partir de un teclado matricial 4x4 y una LCD 4X20 respectivamente, se ingresa el número de vueltas y calibre de cable. El control de testeo se lo realiza en base a una prueba de resistencia óhmica y se comprueba que no exista pérdida de aislamiento del cable.

Para el control en el proceso mismo de la elaboración de las bobinas, se utilizará un módulo Raspberry PI 3 junto con una cámara, que permite observar el proceso de construcción de la bobina y cuando exista algún desperfecto en la misma, detendrá el proceso y realizará las correcciones en el proceso de bobinado.

## **ABSTRACT**

The present technical project will be developed with the purpose of automating a manual copper coil manufacturing machine, based on an artificial vision control. For the automation of this manual machine, a characterization was first performed to know each of the parts necessary for automation such as motors, sensors, controllers, etc. The controllers will be an Arduino Mega card and a Raspberry PI for the part of the control with artificial vision, the system basically controls three stepper motors, in which the first one is in charge of rotating the axis of the machine and the second one will be in charge of guiding the cable in the reel and the third in the correction of the cable in case of an error in the process.

The input of data to the system and the visualization will be made from a 4x4 matrix keyboard and a 4X20 LCD respectively, the number of turns and cable gauge is entered. The test control is performed based on an ohmic resistance test and it is verified that there is no loss of cable insulation.

For the control in the process of the coils elaboration, a Raspberry PI 3 module will be used together with a camera, which allows to observe the process of construction of the coil and when there is a fault in it, it will stop the process and perform the corrections in the winding process.

## INTRODUCCIÓN

En el proceso de fabricación de bobinas automáticamente permite obtener un producto de mayor calidad, así como también con un funcionamiento acorde a las características de las bobinas estándar, además de reducir el tiempo en comparación a la fabricación de bobinas con máquinas manuales debido a que estas no cuentan con la precisión necesaria lo cual hace que el proceso se tenga que corregir varias veces

En el primer capítulo se detalla el problema de estudio, su justificación, objetivos y metodología los mismos que son fundamentales para el desarrollo del proyecto.

En el segundo capítulo se desarrolla la fundamentación teórica de este proyecto, en el cual se definirán conceptos relacionados tanto con el sistema de control, así como los dispositivos que intervienen en todo el proceso.

El tercer capítulo se desarrolla el nuevo sistema para la fabricación de bobinas de forma automática, aquí se detallará el sistema de control, sistemas de testeo, que será utilizado, también se mostrara simulaciones y código fuente en el cual se muestra la lógica que tiene el proceso de fabricación.

En el cuarto capítulo se mostrará las pruebas de funcionamiento y el análisis de resultados de la máquina de bobinado automático, el cual sustentan la validación del proyecto

técnico

# CAPÍTULO 1

## ANTECEDENTES

En este primer capítulo se describe y desarrolla el problema de estudio, justificación, objetivos a cumplir del proyecto, la metodología del proyecto, así como también los beneficiarios en el proyecto.

### 1.1. Problema de estudio

La fabricación de bobinas de diferentes dimensiones es una actividad utilizada cuando se requiere construir o reemplazar un bobinado en dispositivos como transformadores, electroválvulas, motores, relés contactores, generadores, etc. En la actualidad se observa que el sector industrial del país que utiliza los dispositivos antes mencionados no cuenta con máquinas bobinadoras, al no disponer del equipo para el mantenimiento optan por reemplazar dichos dispositivos o la contratación de servicios de bobinado, provocando altos costos económicos debido al tiempo que tardan en dar la solución al mismo. Para realizar el arrollamiento del nuevo cable esmaltado se utilizan en la mayoría de los casos máquinas de banco manuales las cuales necesitan del operador para contabilizar el número de vueltas y realizar el arrollamiento sin cruces, lo cual hace que el tiempo de este sea muy largo para realizar el trabajo además de que se pueden presentar problemas como que el cable se rompa. Estas máquinas manuales a su vez son imprecisas porque el arrollamiento que realizan posee sobre monturas del cable o cruces, la rigidez del arrollamiento no es la requerida y esto causa problemas en el funcionamiento de la bobina. Por consiguiente, se plantea la pregunta ¿Cómo se podría optimizar el proceso de enrollamiento de cable para fabricar bobinas sin cruces o monturas, de tal manera que no exista la rotura del alambre ni un número incorrecto de vueltas, en un tiempo más corto al utilizado en las máquinas bobinadoras manuales?

### 1.2. Justificación

Al automatizar una máquina manual que fabrica bobinas se mejora los parámetros que intervienen en la construcción tales como el calibre, número de vueltas, etc., esta cuenta con un tensor de cable y se puede tener un control del alambre esmaltado que asegurará el correcto arrollamiento con capas perfectas, además con la

implementación de la visión artificial se detecta los cruces del alambre y permite corregirlos teniendo así un bobinado rápido y de alta calidad, optimizando el proceso de bobinado con respecto a la fabricación de bobinas de forma manual.

### **1.3. Objetivos**

#### **1.3.1. Objetivo General**

Desarrollar la automatización de una máquina manual fabricadora de bobinas de cable de cobre mediante control basado en visión artificial.

#### **1.3.2. Objetivos Específicos**

- Realizar la caracterización de la máquina bobinadora manual para determinar las variables de control y establecer los sensores y actuadores que se pueden utilizar en el desarrollo de la automatización
- Desarrollar un algoritmo de programación que permita el control de los sensores y actuadores involucrados en la automatización de la máquina
- Desarrollar un algoritmo de visión artificial que permita efectuar el control del arrollamiento de las bobinas fabricadas
- Implementar los sensores, circuitos de potencia y actuadores en la máquina bobinadora manual para automatizarla.
- Implementar un sistema de testeo en base al método de prueba de resistencia óhmica de los devanados para verificar que no existan fallos en las bobinas.
- Realizar pruebas con la máquina puesta en marcha para llevar a cabo calibraciones utilizando el testeo mediante resistencia óhmica.

### **1.4. Metodología**

Para el diseño adecuado del proyecto se ha tomado en consideración algunos puntos importantes que se detallan a continuación:

- **De acuerdo con la profundidad del estudio.** Se utilizará una investigación descriptiva que va a permitir realizar cambios de acuerdo como se avance con la documentación según el proceso de investigación.

- **De acuerdo con la intervención del investigador en la investigación.** La observación será parte importante ya que se establecerá un circuito que demostrará el funcionamiento del sistema de control de la máquina.
- **Inductivo – Deductivo.** La investigación se la realizará tanto en documentos, libros, así como también en artículos que sirvan de ayuda para la adquisición de información y aporten con la realización del proyecto.
- **Experimental.** La comprobación se la realizará a partir de prácticas para los elementos que intervengan en el diseño y construcción del sistema de control de la máquina automatizada.

### **1.5. Beneficiarios**

Los beneficiarios del proyecto son principalmente industrias que utilizan en su mayoría dispositivos electromagnéticos, donde al sufrir una avería de sus bobinados, se puede realizar un nuevo bobinado y reemplazarlo. Además, el proyecto también beneficia a microempresas que se dedican exclusivamente a la fabricación de bobinas.

## CAPÍTULO 2

### FUNDAMENTACIÓN TEÓRICA

Este capítulo describe la fundamentación teórica en que se basa el proyecto, donde se tiene en cuenta las principales generalidades y características de los controladores, actuadores, sensores.

#### **2.1. Bobina o Inductor**

Las bobinas es un circuito eléctrico, de una o más vueltas, normalmente más o menos circulares o cilíndricas, de un cable portador de corriente diseñado para producir un campo magnético o para proporcionar resistencia eléctrica o inductancia; en el último caso, una bobina también se llama bobina de choque. Un núcleo de hierro suave colocado dentro de una bobina produce un electroimán. Una bobina cilíndrica que mueve un émbolo dentro de ella por variaciones en la corriente a través de la bobina se conoce como solenoide. (Diaz Morales, 2015)

##### **2.1.1.Características de la bobina.**

Entre las características importantes de una bobina se presentan las siguientes:

- Inductancia
- Permeabilidad
- Factor de calidad

##### **2.1.1.1.Inductancia.**

Perez Sandoval (2015) manifiesta que: “Es la propiedad de un conductor eléctrico por el cual un cambio en la corriente a través de él induce una fuerza electromotriz tanto en el propio conductor” (p.1).

##### **2.1.1.2.Permeabilidad Magnética (M)**

Vargas (2014) manifiesta: “Es una característica que tiene gran influencia sobre el núcleo de las bobinas respecto del valor de la inductancia de estas. Los materiales ferromagnéticos son muy sensibles a los campos magnéticos y producen unos valores altos de inductancia.” (p.1).

### **2.1.1.3.Factor de Calidad (Q)**

Esta característica en las bobinas de cobre permite caracterizar los componentes no ideales de la misma. Así para una bobina real se tiene en cuenta la resistencia del cable, por ende, al tener un valor alto en Q significa tener un valor pequeño de resistencia lo cual hace que el comportamiento sea lo más parecido a una bobina real.

### **2.1.2.Dimensionamiento**

A la hora de fabricar una bobina se debe tomar en cuenta la potencia con la que esta va a funcionar, para lo cual es necesario elegir un diámetro en el cable que no limite la corriente circundante en ella.

Así también para el cálculo de la L (inductancia) se considera una relación directa con la permeabilidad (capacidad de los materiales para absorber fuerzas magnéticas), el número de vueltas que conforman la bobina y el diámetro del núcleo donde se arrolla el cable y es inversamente proporcional a la longitud o espacio entre espiras.

### **2.1.3.Construcción de una bobina**

Toda bobina está formada por un cable de cobre esmaltado de un calibre AWG determinado arrollado sobre un carrete, el cual será el núcleo de la bobina construida, así también el número de vueltas que compone la bobina es una característica principal para su elaboración, la cual permitirá crear un mayor o menor campo magnético

### **2.1.4.Tipos de bobinas de cobre**

Dentro de las bobinas que se pueden construir tenemos las bobinas fijas y variables.

Entre las bobinas fijas se tiene:

- Con núcleo de aire.
- Núcleo sólido

#### **2.1.4.1.Núcleo de aire.**

En este tipo de bobinas se arrolla el conductor sobre un soporte hueco el cual posteriormente se retira y da al núcleo un aspecto similar a un muelle. Se la conoce

también como solenoide, su longitud aumenta de acuerdo con el número de vueltas que estas tengan. Se utilizan en casos en los cuales la corriente y la potencia son muy elevadas. (Diaz Morales, 2015)

En la Figura 2.1 se muestra una bobina con núcleo de aire.

Figura 2.1 Bobina núcleo de aire de diferente calibre.



Bobinas núcleo de aire. Fuente: (ELECTRONICA FACIL, 2014)

#### **2.1.4.2. Núcleo sólido**

Este tipo de bobinas poseen un valor alto de inductancia en comparación al núcleo de aire, debido a que tienen un nivel muy alto de permeabilidad magnética. Estas son muy utilizadas para eliminar frecuencias bajas, especialmente en circuitos sintonizadores de radio. (Diaz Morales, 2015)

A continuación, se detalla los tipos de bobinas:

- Bobina de ferrita
- Bobina de ferrita de núcleo de abeja
- Bobinas de ferrita para SMD
- Bobina de núcleo toroidal.

#### **2.2. Proceso para la construcción de bobinas.**

Para la construcción de bobinas se toma en cuenta el proceso en el cual se va a arrollar un cable de cobre esmaltado de un determinado calibre AWG sobre un núcleo, la cual tendrá un determinado número de espiras según las especificaciones que requiera, estas bobinas no deben tener sobre monturas de sus espiras ya que esto

causara una pérdida de flujo magnético al momento que circule corriente sobre ella. (Albuja, 2012)

De manera general para el proceso de construcción de bobinas se tomará en consideración las diferentes características constructivas, así como también se usará una máquina bobinadora entre las cuales se presenta:

- Manual
- Semiautomática
- Automática.

### **2.2.1. Proceso Manual**

En este proceso se cuenta con la intervención directa del operario, quien guía el alambre en el molde y debe dar también movimiento rotacional a la bobinadora de banco manual, el operario lleva la cuenta del número de espiras de cada bobina y su respectivo paso. En la Figura 2.2 se muestra una máquina bobinadora manual con la cual se ejecuta este proceso de bobinado.

Figura 2.2 Máquina bobinadora manual para fabricación de bobinas



Máquina para fabricar bobinas manuales. Fuente: (SERVOR, 2014)

Esta máquina posee las siguientes características:

- Aluminio.
- Dimensiones: 500x190x310 mm
- Peso: 5kg.

- Accesorios de serie: Soporte rectangular de moldes 285mm Tirantes porta moldes Juego de moldes lineales tipo 40 Ø. Juego de moldes lineales tipo 60 Ø. Cuentavueltas mecánico de cinco cifras. Palanca de bloqueo del eje. ∪
- Accesorios opcionales: Serie completa de moldes lineales tipo 40 Ø - 150 Ø. Serie completa de moldes concéntricos tipo T10 - T40.

### 2.2.2. Proceso Semiautomático

Para este proceso el aporte del operario para la construcción de la bobina es mínima, debido a que cuenta con una máquina eléctrica dispuesta con un control semiautomático lo cual facilita al momento de maniobrar con los arrollamientos de las bobinas, en este proceso se incorporan parámetros a la máquina como número de espiras, la velocidad a la cual se realiza el proceso, con lo cual se obtiene un mejor bobinado en comparación al proceso manual. En la Figura 2.3 se muestra la estructura física de una máquina eléctrica para el proceso de bobinado semiautomático.

Figura 2.3 Máquina eléctrica para fabricación de bobinas semiautomática.



Máquina eléctrica fabricante de bobinas magnéticas. Fuente: (SERVOR, 2014)

Dentro de las características constructivas y de funcionamiento de esta máquina están las siguientes:

- Aluminio.
- Dimensiones: 730x810x1490 mm.

- Peso: 125kg.
- Velocidad: 100 - 195 - 200 - 390 - 450 - 900 r.p.m.
- Para bobinar motores de 50 hp máximo.
- Accesorios de serie: 2 Soportes circulares: 300 Ø - 500 Ø Tirantes porta moldes Juego de moldes lineales tipo 40 Ø. Juego de moldes lineales tipo 80 Ø. Cuentavueltas digital con preselección. Motor con freno. Mesa de acero con cajón.
- Accesorios opcionales: Serie completa de moldes lineales tipo 40 Ø - 150 Ø. Serie completa de moldes concéntricos tipo T10 - T40.

### 2.2.3. Proceso Automático

En este caso se cuenta con una máquina bobinadora completamente automática, la cual se encarga de todo el proceso de fabricación de las bobinas, utilizando diferentes dispositivos de control que reduce de manera significativa la intervención de un operario. Las especificaciones de fabricación de la bobina están dadas mediante un software o mediante un teclado externo, el cual permite parametrizar las diferentes características para el diseño de la bobina. En la Figura 2.4 se muestra la estructura física de una máquina bobinadora automática.

Figura 2.4 Máquina bobinadora automática



Máquina bobinadora automática marca SMT. Fuente: (SERVOR, 2014)

Dentro de las características técnicas y físicas se tiene las siguientes:

- Velocidad de bobinado: 3200 r/min

- Distancia de transferencia: Max 170mm
- Presión de aire: Min 0.4 MPa
- Fuente de alimentación: 220 V/50/60Hz 0.75Kw

### 2.3. Sistemas de control de calidad

Todos los dispositivos electrónicos necesitan ser verificados en las diferentes etapas de su fabricación, y para esto es necesario utilizar un sistema de control de calidad y dar un mayor valor a los productos fabricados.

Antes de detallar los diferentes sistemas de control de calidad que se tienen presentes para la fabricación de bobinas, se va a dar una breve definición de lo que es control de calidad.

- **Control de calidad.** Gracia Padilla (2018), se manifiesta que “Es el conjunto de los mecanismos, acciones y herramientas realizadas para detectar la presencia de errores” (p. 1).

Las fabricaciones de bobinas necesitan siempre de un sistema de pruebas avanzado y fiable con el cual se mantenga un control continuo al momento de fabricación de bobinas de cobre esmaltado, debido a que de existir un cruce o un espaciado muy grande entre espiras causará un daño en su funcionamiento, es decir afecta directamente en la producción de flujo magnético haciendo que este se esparza.

Una vez fabricadas las bobinas deben ser sometidas a un conjunto de pruebas las cuales reafirmaran la calidad de estas, todas las pruebas que se realizan forman parte del control de calidad.

Dentro de los tipos de controles de calidad que se pueden implementar para la construcción de bobinas de cobre se tiene los siguientes:

- Control de resistencia óhmica
- Control de resistencia de aislamiento
- Visión artificial

### **2.3.1. Control de resistencia óhmica**

Permite comprobar si el arrollamiento se encuentra firme al carrete, así también podemos conocer si hay pérdida de cobre o identificar contactos falsos, espiras que se encuentran en corto circuito y que existan espiras abiertas en algún punto de la bobina que pueda afectar con el funcionamiento de la misma, se realizan una serie de medidas en base a un circuito previamente armado como es el puente de Kelvin el cual arroja valores de resistencia óhmica para que sean comparados con valores estándares y así se conoce si la bobina se encuentra en condiciones óptimas para su operación. (FLUKE, 2015)

### **2.3.2. Control de resistencia de aislamiento.**

Este tipo de control permite verificar que el aislamiento del cable que forma la bobina cumple con la resistencia mínima soportable bajo alguna operación a la cual pueda ser sometida la bobina, esta resistencia mínima esta parametrizada según el estándar de IEEE 43-2000 (R2006), así también se comprueba que no exista una inadecuada conexión entre espiras con esto se avala el buen diseño y no existan defectos en las mismas. Este control consiste en aplicar voltaje directo en los extremos de la bobina, con ayuda del Mega óhmetro se mide voltaje y corriente que circula por el dispositivo, aplicando una ley de ohm ( $R=V/I$ ) se conoce el valor de resistencia al existir fugas de corrientes en valores de mili o microamperios, al comparar este valor de resistencia con el estándar, se puede dar como correcta o no la bobina fabricada. (FLUKE, 2015)

### **2.3.3. Visión artificial.**

La visión artificial permite realizar una inspección continua con ayuda de una cámara de alta definición en procesos en los cuales se pueda presentar fluctuaciones, al capturar estas fluctuaciones con la cámara e ingresa a un proceso de codificación de imágenes donde se compara con un patrón lo cual permite corregir y continuar con el proceso en un tiempo muy corto y dando al final un producto con características de funcionamiento correctas. (Universidad politecnica de Madrid, 2017)

### **2.3.3.1. Definición.**

Tornero (2015) se manifiesta que: “La visión por computador persigue la correcta adquisición e interpretación de la información visual adquirida por las cámaras, en áreas tales como: máquinas-herramienta, medicina, análisis de documentos, navegación autónoma, tele operación de robots, etc.” (p.1)

### **2.3.3.2. Aplicaciones**

Dentro de las aplicaciones por visión artificial se tiene:

- Inspección para embalaje con visión artificial
- Control de calidad y optimización en la industria farmacéutica
- Defectos de pintura
- Controles de tiempo con visión artificial
- Robótica guiada por visión artificial.

### **2.3.3.3. Control de calidad con visión artificial.**

Los sistemas de control de calidad basados en visión artificial permiten capturar imágenes en tiempo real para posteriormente procesar analizar e interpretar y dar una retroalimentación en tiempo. Esto permite deshacer o rechazar los productos que no cumplen con los estándares de producción es decir se encuentran dañados. (Gracia Padilla, 2018)

La iluminación dentro de los sistemas de control de calidad con visión artificial juega un papel muy importante para la captación de imágenes.

El sistema de visión artificial sigue el siguiente patrón el cual se muestra en la Figura 2.5

Figura 2.5 Proceso de control de calidad basado por visión artificial

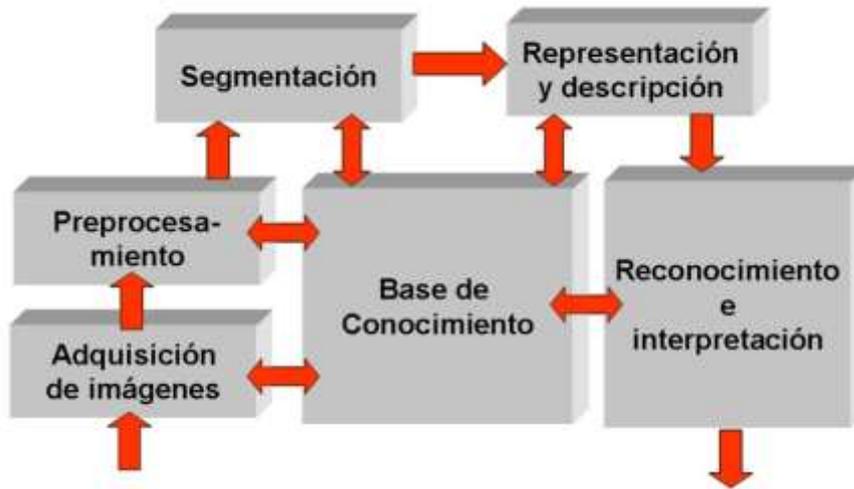


Diagrama de etapas de la visión artificial. Fuente: (Vives, 2014)

- **Adquisición de imágenes.**

Se trata de la primera etapa del proceso el cual es obtener la imagen visual a ser analizada.

- **Preprocesamiento:**

Permite la eliminación de señales externas tales como ruido. (Cavarrubias, 2017).

Para esto se presentan diferentes métodos para el procesamiento de imágenes:

- **Obtención de características del objeto por color.** - sacar datos relevantes en formato RGB de objetos, para crear características que puedan ser utilizadas por otros procesos para llegar a un resultado. (Cavarrubias, 2017)
- **Detección de bordes.** - La característica principal de usar el contorno en este proceso permite presentar la transición de claro u oscuro o viceversa. (Cavarrubias, 2017)
- **Detección de bordes en color.** – Este proceso trata de una imagen monocromática, y se hace una combinación de todas ellas, considerándose el máximo o el promedio además este método se plantea para detectar bordes de piezas no convexas y convexas. (Cavarrubias, 2017)
- **Segmentación:** Divide a una imagen en objetos los cuales son de utilidad para nosotros. (Cavarrubias, 2017)

- **Representación y descripción.** En esta etapa permite obtener características esenciales y de particular interés para diferenciar un objeto de otro. (Cavarrubias, 2017)
- **Reconocimiento e interpretación:** Permite asociar un significado a un conjunto de objetos ya reconocidos. (Cavarrubias, 2017)

#### **2.3.3.4. Filtros aplicados para visión artificial.**

La aplicación de filtros en la visión artificial es específicamente para suavizar la imagen es decir reducir la variación de intensidad entre píxeles, eliminar ruido y realzar los bordes. (GTOCOMA, 2014)

Entre los tipos de filtros usados se presenta los siguientes:

- **Filtros Gausseanos.** - Modela el ruido producido por los circuitos electrónicos o ruido de los sensores por falta de iluminación y/o altas temperaturas. (GTOCOMA, 2014)
- **Filtros realzantes.** - Estos filtros permiten destacar los detalles finos de la imagen entre sus aplicaciones más utilizadas esta la detección de bordes. (GTOCOMA, 2014)
- **Filtros pasa bajo.** - La aplicación del filtro paso bajo supone la cancelación parcial del ruido, pero también la pérdida de la propia información de la imagen. (GTOCOMA, 2014)

## CAPÍTULO 3

### DESARROLLO DEL PROYECTO

Para este capítulo se precisa y describe la ubicación, así como también la función de cada dispositivo involucrado en la máquina bobinadora automática, condiciones en las cuales las bobinas deben ser diseñadas, los algoritmos utilizados para la automatización de la máquina. Se divide en dos partes, las mismas que son:

- Desarrollo del hardware
- Desarrollo del software.

#### 3.1. Desarrollo del hardware.

En la Figura 3.1 se muestra un diagrama de bloques de los sistemas que forman parte de la máquina bobinadora

Figura 3.1 Diagrama de bloques para funcionamiento de la máquina bobinadora automática

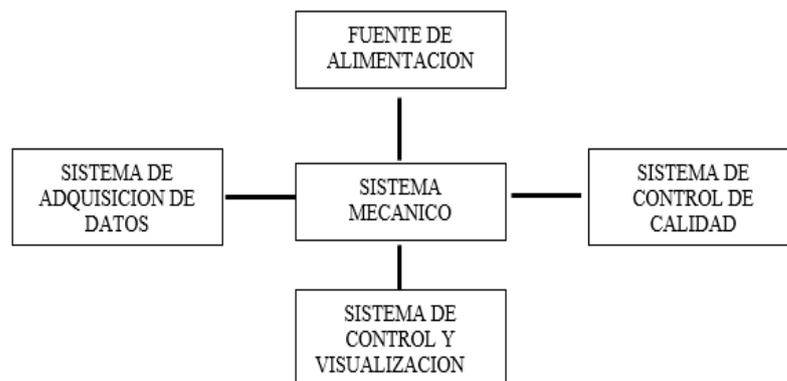


Diagrama de bloques de partes que forman parte de la máquina bobinadora. Elaborado por: Achig Juan, Pazmiño Fernando.

##### 3.1.1. Sistema mecánico.

Para la máquina bobinadora se consideran al menos 2 motores, que se encargan de la rotación y traslación, es decir la utilización de un eje X, y está involucrado un tornillo sin fin y es un carril en el cual se genera el movimiento de traslación y el segundo eje será al que se sujeta la pieza o carrete donde se va a bobinar, estos dos motores deben tener un giro coordinado para el proceso de bobinado.

Los motores ideales para realizar este trabajo en la máquina bobinadora son los tipos pasos que permiten ser controlados por impulsos eléctricos logrando desplazamientos angulares discretos, es decir avanza una serie de grados dependiendo del sistema de control. Este tipo de máquinas requieren de motores potentes que necesitan de una corriente igual o inferior a la que se encuentre marcada en su chapa. La demanda de corriente de dichos motores delimitará la elección de la etapa de potencia para su control, así como de la fuente de alimentación necesaria. De acuerdo con estos parámetros se considera los motores NEMA 23 y NEMA 17 para la construcción de los ejes de la máquina.

EL motor NEMA 23 al tener una mayor potencia formara parte del sistema de rotación y el motor NEMA 17 se encarga de la traslación. A continuación, se describe las características técnicas de los motores.

#### **3.1.1.1. Motor NEMA 23.**

Este motor posee una fuerza y precisión que permite que la máquina acepte diferentes calibres de cables. En la Figura 3.2 se muestra el motor NEMA 23.

Figura 3.2 Motor NEMA 23 ORIENTAL MOTOR



Motor a pasos NEMA 23 Oriental Motor modelo PH266-01GK-C55. Fuente: (ORIENTAL MOTOR, 2017)

En la Tabla 3.1 se muestran las características del motor NEMA 23.

Tabla 3.1 Características Motor NEMA 23.

<b>Voltaje DC</b>	6 V
<b>Corriente</b>	1.2 A
<b>Velocidad</b>	1.8°/STEP
<b>Fases</b>	2
<b>Pasos</b>	200 por vuelta
<b>Torque</b>	14Kg/cm

Características técnicas de motor NEMA 23. Modelo PH266-01GK Fuente (ORIENTAL MOTOR, 2017)

### 3.1.1.2.Motor NEMA 17

Se utiliza 2 motores del modelo PX243G01-01 el cual permite la rotación del tornillo para la traslación del usillo y permita un bobinado continuo y preciso dependiendo del tipo de cable. En la Figura 3.3 se muestra el motor a pasos NEMA 17

Figura 3.3 Motor a pasos NEMA 17



Motor a pasos NEMA 17 modelo PX234G01-01A. Fuente: (ORIENTAL MOTOR, 2017)

En la Tabla 3.2 se muestra las características de este motor, las cuales fueron consideradas para el diseño de la máquina fabricadora de bobinas.

Tabla 3.2 Características técnicas motor a pasos NEMA 17

<b>Voltaje DC</b>	4V
<b>Corriente</b>	0.6 A
<b>Velocidad</b>	0.1°/STEP
<b>Fases</b>	2
<b>Torque</b>	3.2 Kg/cm
<b>Pasos</b>	200 por vuelta

Características técnicas de motor NEMA 17. Modelo PX234G01-01A Fuente (ORIENTAL MOTOR, 2017)

### **3.1.2.Sistema de adquisición de datos.**

Una vez conocido los motores que se van a utilizar para la automatización de la parte mecánica, para el arranque de estos se necesita que sus bobinados sean excitadas de una forma determinada para que giren, este trabajo se lo realiza con la ayuda de un driver, este debe estar formado por un circuito lógico el cual genera la secuencia y una etapa de potencia en un mismo chip, además debe tener un control de corriente ajustable para establecer la salida de corriente máxima, para con esto lograr que los motores tenga una mayor velocidad, al estar formado por un puente H permita que en el mismo driver se genere una inversión de giro instantánea.

Bajo estos parámetros se utilizará el driver A4988 que cumple con estos requerimientos.

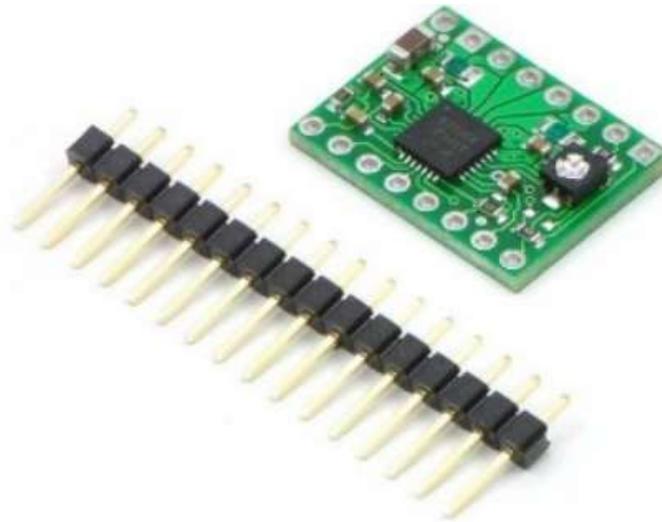
Así también es necesario de sensores los cuales emitirán una señal hasta el controlador, el mismo que permita delimitar el movimiento de traslación del carro bobinador y este no se descarrile, para esto se considera unos sensores finales de carrera que envían una señal lógica que cumple con este propósito.

#### **3.1.2.1.Driver A4988**

Esta placa es el controlador de motor paso a paso bipolar A4988 cuenta con protección de limitación de corriente ajustable, sobre corriente y sobre temperatura, y cinco diferentes resoluciones de microstep (hasta 1/16 de paso). Funciona de 8 V a 35V y puede entregar hasta aproximadamente 1 A por fase sin un disipador de calor o flujo de aire forzado (tiene una capacidad de 2 A por bobina con suficiente

refrigeración adicional). En la Figura 3.4 se muestra la estructura y composición del driver A4988.

Figura 3.4 Estructura física Driver A4988.



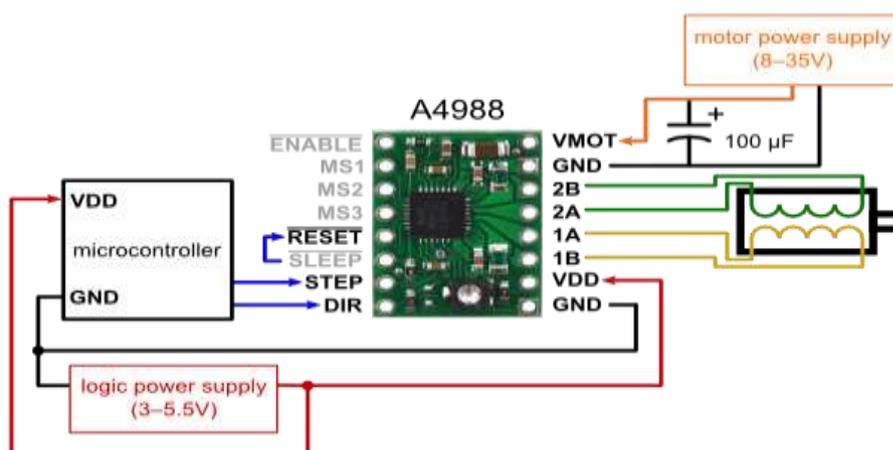
Driver Pololu A4988, Fuente: (ELECTRONICLAB, 2015)

Dentro de las especificaciones de este driver se encuentran las siguientes:

- Interfaz de control de paso y dirección simple
- Cinco resoluciones de paso diferentes: paso completo, medio paso, cuarto de paso, octavo paso y decimosexto paso
- El control de corriente ajustable le permite establecer la salida de corriente máxima con un potenciómetro, que le permite usar voltajes por encima del voltaje nominal de su motor paso a paso para lograr tasas de pasos más altas
- Control inteligente de corte que selecciona automáticamente el modo correcto de caída de corriente (decaimiento rápido o decaimiento lento)
- Apagado térmico por sobre temperatura, bloqueo de bajo voltaje y protección de corriente cruzada Protección contra cortocircuitos y carga en cortocircuito.

Este driver puede conectarse a los motores de cuatro, seis, y ocho pasos, este driver soporta limitaciones activas de corriente, el cual mediante un potenciómetro permite ajustar el límite de corriente, en la Figura 3.5 se muestra la conexión del driver a un motor a pasos.

Figura 3.5 Diagrama de conexión del driver A4988 a un motor a pasos.



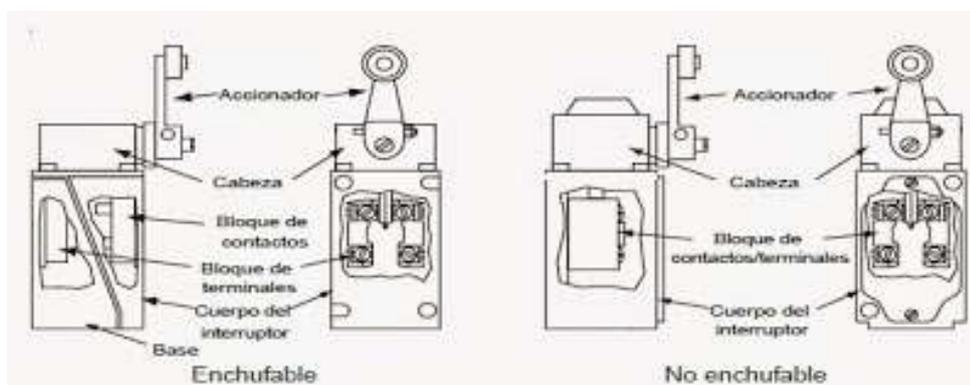
Funcionamiento y conexión de driver Pololu A4988 a motor a pasos. Fuente: (ELECTRONICLAB, 2015)

### 3.1.2.2. Sensor final de carrera.

Los sensores finales de carrera en cada extremo serán una referencia para determinar el límite de bobinado, para esto cuando el carro de la traslación que lleva el cable ha llegado a su extremo activa el sensor, con lo cual se envía una señal al controlador para que este proceda frenarlo o caso contrario se emita al motor el cambio de giro para que se regrese bobinando las vueltas que sean necesarias de acuerdo a los parámetros dados.

En la Figura 3.6 se muestra la estructura de un sensor final de carrera tipo rodillo.

Figura 3.6 Diseño sensor interruptor final de carrera.



Sensor final de carrera capas enchufable y no enchufable, Fuente: (Caniparoli, 2012)

En Tabla 3.3 se detalla las características del sensor final de carrera.

Tabla 3.3 Tabla de características del final de carrera

<b>Voltaje máximo</b>	250 VAC
<b>Corriente máxima</b>	3 A
<b>Temperatura</b>	-20 + 80 grados C
<b>Contactos</b>	1 NC y 1 NO
<b>Contacto móvil</b>	Palanca con roldana

Características del sensor final de carrera. Fuente: (Electronica Embajadores, 2017)

### 3.1.3.Sistema de control de calidad

El sistema de control de calidad está basado en visión artificial donde se involucra una cámara de alta definición con lente regulable que tenga un buen funcionamiento en caso de existir poca luz para la captura de imágenes, los niveles de zoom sean lo suficientemente claros. Es necesario que la captura de imágenes sea lo más clara posible debido al procesamiento que posteriormente se lo realizará en un controlador, esta cámara debe tener un bajo consumo de energía debido a que estará siempre encendida al tratarse de un proceso de control de calidad continuo.

De acuerdo con estas especificaciones, compatibilidad y costos se utilizará la cámara PS3, que cumple con los requerimientos para el proceso de visión artificial.

#### 3.1.3.1.Sony Camara eye PS3.

La cámara PS3 posee una resolución muy alta ideal para la visión artificial así mismo trabaja de manera eficiente en condiciones de baja luz y no refleja pérdida de bordes de los objetos capturados, en la Figura 3.7 se muestra la estructura física de la cámara.

Figura 3.7 Cámara PS3 para visión artificial



Playstation Eye SCEH-7010571. Fuente: (PlayStation, 2014)

En la Tabla 3.4 se muestra las características de la cámara por las cuales fue elegida para el proceso de visión.

Tabla 3.4 Características técnicas cámara PS3

<b>Conexión</b>	USB
<b>Funcionamiento</b>	Plug and Play
<b>Resolución</b>	8 megapíxeles
<b>Consumo de energía</b>	5V, máximo 500mA

Características cámara PlayStation PS3. Fuente: (PlayStation, 2014)

#### **3.1.4.Sistema de control y visualización.**

Para el proceso de control y visualización se requiere de dispositivos controladores amigables y que permitan diseñar los algoritmos de manera fácil, rápida y comprensible para el correcto funcionamiento de la máquina, para esto tienen 2 controladores, en donde en uno se llevará a cabo el sistema automatizado de la parte mecánica (motores), la parte de adquisición de datos, y la visualización, mientras que en el segundo controlador se realizará plenamente el sistema de control de calidad.

Para el primer controlador se elige utilizar una tarjeta Arduino, este simplifica el proceso de utilizar varios microcontroladores para el sistema, lo que permite reducir costos, al ser multiplataforma facilita la creación de algoritmos, ya que el software Arduino puede ser utilizado en diferentes sistemas operativos, específicamente Windows y se trata de software libre, las salidas y entradas digitales que posee hace que la generación de algoritmos sea basada en 0 y 1 sin complicar la programación, además Arduino cuenta con librerías precargadas que permiten omitir ciertos códigos en este caso específicamente para la visualización.

En el segundo controlador se realizará el proceso de control de calidad, específicamente se desarrollará visión artificial, por lo tanto, se necesita de un procesador más potente y rápido que el que se tiene en Arduino, debido a que se realizará codificación de imagen continua este controlador debe tener un bajo consumo de energía y de bajo costo. para el desarrollo del algoritmo de visión artificial se usará el controlador Raspberry PI que cuenta con un procesador lo suficientemente potente para la captura de imágenes de manera continua, para la

ejecución de los comandos se realiza en base a lenguajes ya conocidos y prácticos, específicamente PHYTON, lo que facilita la secuencia y comprensión del algoritmo. Además, permite una comunicación directa con Arduino.

Para la visualización se requiere de una pantalla LCD donde se mostrará los datos ingresados por medio de un teclado matricial para el proceso de bobinado, así también se requiere de una pantalla Touch, y permita ejecutar el algoritmo del sistema de visión artificial.

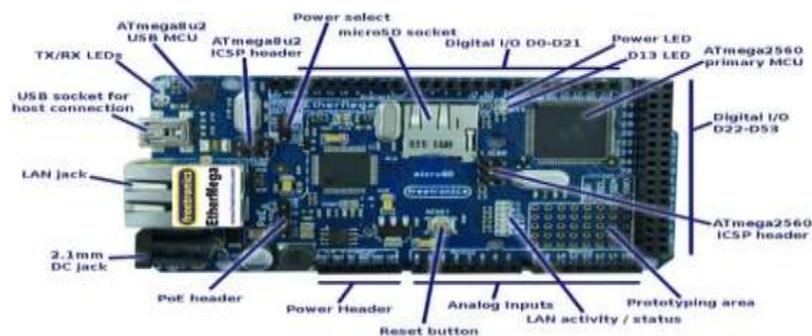
A continuación, se detallan las características principales de los 2 controladores y de los dispositivos de visualización

#### 3.1.4.1. Controlador Arduino

El Arduino es el encargado del control de los actuadores, sensores y visualización, es quien recibe los parámetros de construcción para la bobina, estos parámetros son receptados por el Arduino mediante dispositivos de control externos que posteriormente hacen que se emita las acciones para los actuadores. (ARDUINO, 2016)

En la Figura 3.8 se muestra la estructura física (hardware) así como también los elementos y partes que la conforman.

Figura 3.8 Estructura física de tarjeta Arduino Mega.



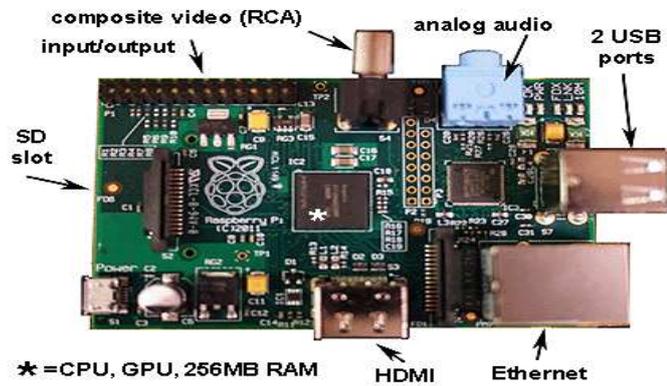
Estructura tarjeta Arduino, Fuente: (Mega ARDUINO 2016)

#### 3.1.4.2. Raspberry PI Model B

Se trata de una placa computadora (SBC) de bajo costo basado en ARM con 4 núcleos administrable por software libre, permite la conexión con el Arduino Mega 2560 mediante puerto serial para transmisión y recepción de datos.

En la Figura 3.9 se muestra el hardware y las partes que lo componen. (Universidad politecnica de Madrid, 2017)

Figura 3.9 Raspberry PI 3 model B

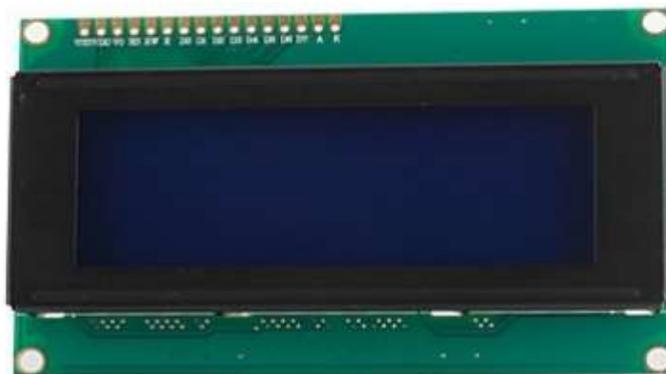


Arquitectura Raspberry PI 3 Model B, Fuente: (Universidad Politecnica de Valencia 2013).

#### 3.1.4.3.LCD 20 X 4

Para visualizar los datos se utiliza una LCD 20 x 4 que se encarga de registrar los datos de numero de vueltas, así como también datos de corriente voltaje resistencia, calibre de cable AWG. En la Figura 3.10 se muestra el LCD 20 x 4. (I+D, 2014)

Figura 3.10 LCD 20 x 4

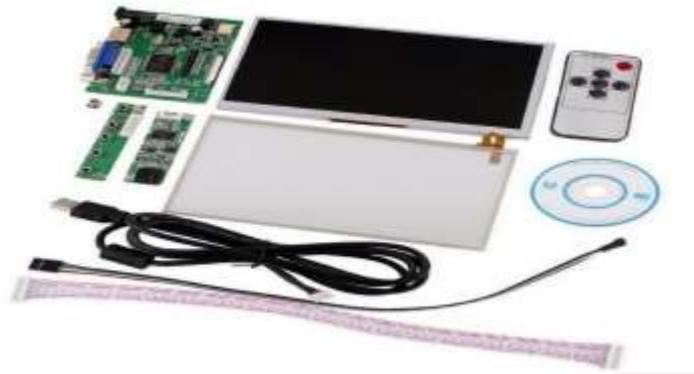


Módulo Display LCD 4x 20. Fuente: (I+D, 2014)

#### 3.1.4.4.Pantalla de control Raspberry.

La pantalla es una touchscreen 3.5'' que permite dar un arranque del proceso la visión artificial. En la Figura 3.11 se muestra el modelo PiTFT.

Figura 3.11 PiTFT Raspberry 3.2''



Pantalla Touch HMI del Equipo, Fuente: (ELECTRONICLAB, 2015)

Este modelo permite una conexión directa con la Raspberry PI y mediante las aplicaciones de la touch poner en funcionamiento el control del proceso de visión artificial. (ELECTRONICLAB, 2015)

#### 3.1.4.5. Teclado matricial 4x4

Este dispositivo permite ingresar los datos necesarios para el proceso de bobinado, así como también se puede ingresar al sistema. En la Figura 3.12 se muestra el modelo de teclado matricial. (Circuitos Electronicos, 2016)

Figura 3.12 Teclado matricial 4 x4.



Teclado matricial para ingreso de datos. Fuente: (Circuitos Electronicos, 2016)

#### 3.1.5. Fuente de alimentación.

Para este sistema se considera una fuente de alimentación que brinde diversos voltajes específicamente 5, 9, 12 y 24 V, así también se considera corrientes de

salidas de hasta 15 A, lo cual será suficiente para poner en movimiento los motores, la fuente debe tener un sistema de ventilación interno para evitar sobrecalentamientos en la placa que conforman la fuente. Bajo estas características se considera el uso de una fuente de alimentación de computador, que brinde estos voltajes y diferentes rangos de corriente con lo que se puede trabajar para la alimentación del sistema completo.

### **3.1.5.1.Fuente de voltaje.**

Se considera una fuente de voltaje variable de la cual se obtiene 5, 12, 24 V con corrientes de 1,5 hasta 2 A, los cuales permiten alimentar a todos los componentes de la maquina bobinadora involucrados es decir sensores, actuadores, controladores. En la Figura 3.13 se muestra la fuente utilizada.

Figura 3.13 Fuente de Voltaje



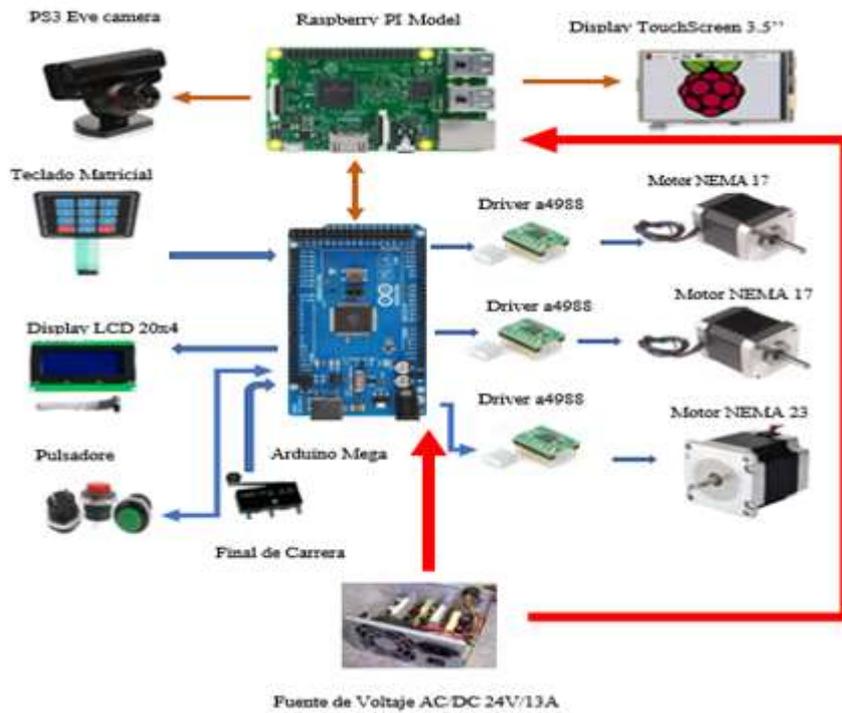
Fuente de voltaje DC AC, Elaborado por: Juan Achig, Fernando Pazmiño.

Los actuadores consumen un máximo de 2 A para lo cual la fuente de alimentación nos brinda una corriente de hasta 15 A.

Una vez determinado cada uno de los dispositivos que forman parte de la máquina bobinadora se desarrolla un diagrama de conexión en base a su funcionamiento. En la Figura 3.14 se muestra el diagrama.

Cada uno de los controladores, sensores, actuadores que son parte del funcionamiento, así como la conexión de estos según las flechas.

Figura 3.14 Diagrama de conexiones de hardware de la máquina.

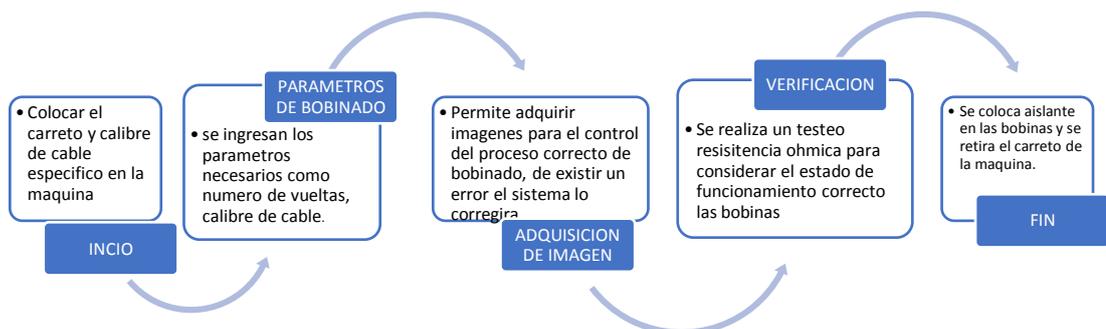


Descripción y conexión de los dispositivos involucrados en la máquina, Elaborado por: Juan Achig, Ángel Pazmiño.

### 3.2. Desarrollo del software.

El desarrollo del software consta de las siguientes etapas mostradas en la Figura 3.15, para el proceso de bobinado automático

Figura 3.15. Principio de funcionamiento de máquina bobinadora automática



Etapas de funcionamiento para fabricación automática de bobinas. Elaborado por: Juan Achig,  
Fernando Pazmiño.

- **Inicio.** – En esta primera etapa se coloca en la máquina en el eje de rotación el carrete en el cual se arrollará la bobina ajustando el cable de cobre también al eje de traslación donde se tendrá la tensión necesaria del cable para llevarlo de izquierda a derecha según el número de vueltas dado.
- **Parámetros de bobinado.** - En esta etapa se ingresa mediante un teclado los datos necesarios que se va a construir la bobina, es decir, número de vueltas, calibre del cable con el que se va a fabricar la bobina estos se podrán visualizar en el display que se encuentra integrado a la máquina.
- **Adquisición de imágenes.** – En esta etapa la cámara va capturando imágenes durante el proceso de arrollamiento del cable, se realiza un escaneo continuo en caso de existir una montura o una alteración en la fabricación de la bobina que pueda causar el mal funcionamiento de esta, lo que hace es parar el proceso corregir el error y continuar hasta finalizar la fabricación de la bobina.
- **Verificación.** - Una vez terminado el proceso de elaboración de la bobina se le realiza una prueba de resistencia óhmica el cual de acuerdo con un parámetro de estandarización de bobinas permite conocer las características de resistencia son los correctos para el funcionamiento.
- **Fin.** -Finalmente se coloca una cinta protectora en el bobinado y se retira la misma de la máquina.

### 3.2.1.Algoritmo desarrollado en Arduino

Una vez conocido el procedimiento del sistema de automatización del bobinado, el algoritmo desarrollado en Arduino permite controlar los motores que conforman la máquina, en este caso tenemos 3, el programa principalmente está desarrollado en



AL iniciar el programa implementado en el Arduino se inicializa el teclado matricial y la pantalla LCD, con lo cual se muestra un mensaje en la pantalla con la opción de realizar una nueva bobina, una segunda opción puede ser ingresada, pero esta se hace solo desde la Raspberry PI mediante comunicación serial con el Arduino para recibir el valor del calibre del cable y el número de vueltas de la bobina.

Con los valores de calibre de cable y el número de vueltas de la bobina válidos, el programa entra en la fase de bobinado, haciendo que los motores giren en sentido horario, los mismo que se han sincronizado para que al realizar 3600 pasos giren una vuelta cada uno, dentro de los pasos de los motores se realiza el monitoreo de las señales de los sensores, al activarse la señal del sensor izquierdo el motor 2 (motor guía) cambia su sentido de giro a sentido anti horario y al activarse la señal del sensor derecho el motor 2 cambia su sentido de giro a horario; al detectar la señal de reversa enviada desde la Raspberry PI por error en el bobinado se invierte el sentido de giro de los tres motores hasta llegar al error del bobinado, lo corrige y continúa normalmente con el proceso; La señal de control se activa cada diez vueltas y es enviada a la Raspberry PI para que realice el control del proceso por visión artificial; El pulsador de Pausa ayuda a detener el proceso de bobinado. Una vez que se ha bobinado el número de espiras que se digitó en la interfaz se envía una señal de culminación de bobinado desde el Arduino a la Raspberry PI para que detenga el programa de visión artificial.

### **3.2.2. Algoritmo desarrollado en Raspberry PI**

El algoritmo desarrollado en el dispositivo Raspberry PI es el de visión artificial, el cual mediante la cámara se realiza una captura de imágenes continua, estas imágenes son procesadas mediante el método de borde antes estudiado y son filtradas con un filtro gausseano desarrollado en la programación, al momento que la cámara registra un borde sobrepuesto hace que el sistema se detenga, poniendo en funcionamiento el tercer motor y se genera un movimiento contrario al del bobinado, corrigiendo la sobreposición del cable y continuar con el proceso de bobinado, la Raspberry también se encuentra en comunicación serial con el Arduino, es decir se puede ingresar los datos de construcción de una bobina mediante la pantalla touch FTTP e iniciando el proceso de bobinado.

En la Figura 3.17 se presenta el flujograma del algoritmo diseñado en la Raspberry PI.

Figura 3.17 Flujograma del algoritmo realizado en la Raspberry PI

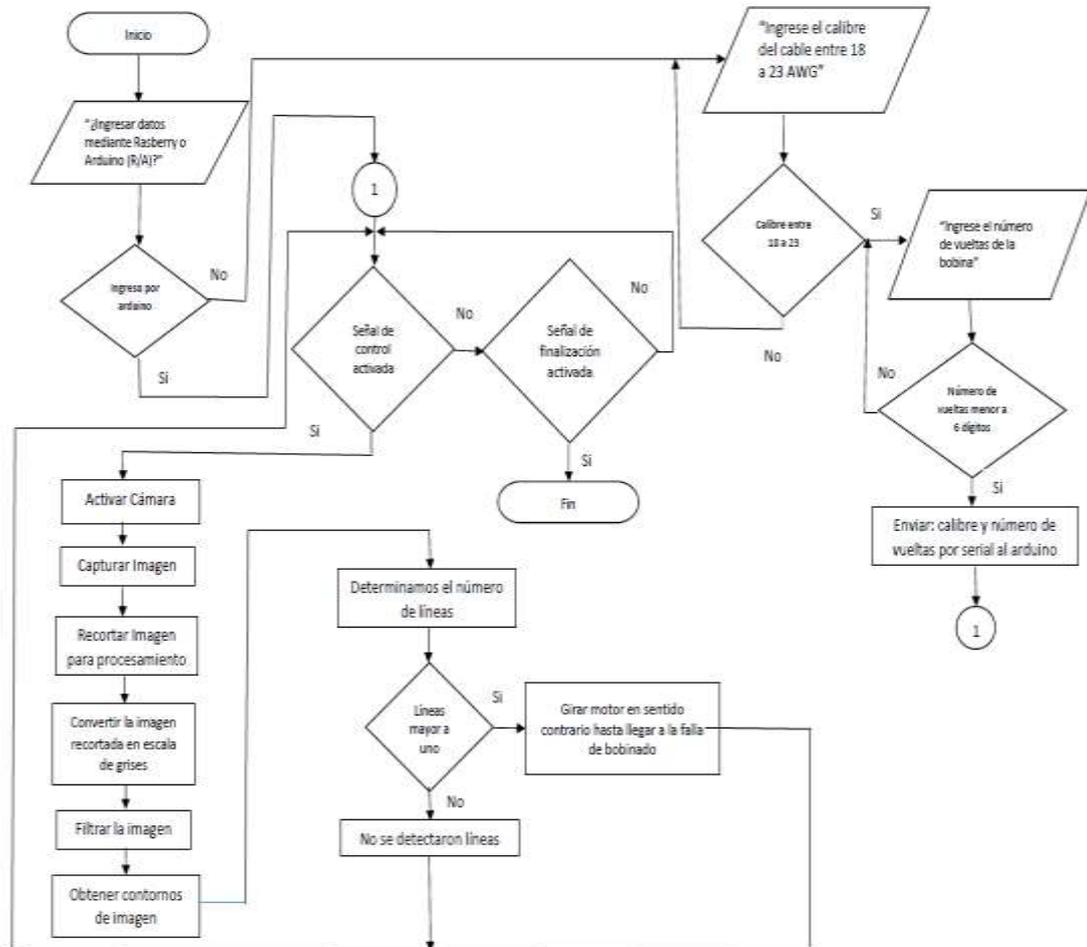


Diagrama de flujo de la Raspberry PI. Elaborado por: Juan Achig, Fernando Pazmiño.

Al iniciar el programa en la Raspberry PI se muestra un mensaje para seleccionar si se desea ingresar los datos de calibre de cable y el número de vueltas de la bobina por el teclado matricial conectado al Arduino o para ingresar estos datos mediante el teclado de la Raspberry PI y posteriormente enviarlos mediante comunicación serial al Arduino. Una vez que el bobinado esté en proceso y llegue la señal de control desde el Arduino a la Raspberry PI empieza el control por visión artificial, la cámara conectada a la Raspberry PI se activa y toma una captura de la imagen. La imagen capturada se recorta de tal manera que se pueda observar un cierto número de espiras de la bobina dependiendo el calibre del cable, a esta imagen recortada se la convierte

a escala de grises debido a que proporciona facilidad para detectar características. A esta imagen se la filtra para realizar un suavizado.

Una vez realizado el procesamiento de la imagen se obtiene los bordes de la imagen filtrada. A partir de este proceso se determinan las líneas existentes en la imagen. La finalidad de este proceso es determinar si la bobina en su parte superior forma una línea recta, caso contrario si hay un cruce de cable o una separación muy grande entre espiras se formará más de una línea lo que hace que los motores giren en sentido contrario hasta llegar al fallo de la bobina, lo corrige y continúa con el bobinado.

## CAPÍTULO 4.

### ANÁLISIS DE PRUEBAS Y RESULTADOS

En este capítulo se detalla todas las pruebas del proceso de bobinado automático y los resultados obtenidos para validar su funcionamiento.

Se realiza las siguientes pruebas:

- Prueba de panel de control e ingreso de la información
- Pruebas de bobinado
- Pruebas de control de calidad.
- Pruebas de aislamiento.

#### 4.1. Prueba de Panel de control e ingreso de la información.

Esta prueba se la ejecuta para verificar el funcionamiento de la interfaz, la cual permite el ingreso de datos por parte del usuario, se realizó a todos los menús que están involucrados en la interfaz, probando varias veces el teclado matricial, la visualización en la LCD, así como la adquisición de datos en el sistema. También se corrige los datos ingresados y se inició y paró el proceso en varias ocasiones bobinando y sin bobinar.

En la Figura 4.1 se muestra el panel de control (LCD, TECLADO, BOTON DE INICIO Y PARO).

Figura 4.1 Panel frontal de la máquina bobinadora automática



Panel frontal para ingreso de datos. Elaborado por: Juan Achig, Fernando Pazmiño.

Después de evaluar todos los procedimientos, el sistema permitió el ingreso de la información sin errores, al igual que la visualización siempre fue clara y las órdenes y valores comprendidos por completo por lo que se puede afirmar que el panel de control funciona de manera correcta.

#### **4.2. Pruebas de bobinado**

Se realiza para evaluar el funcionamiento de la máquina al realizar el bobinado con los parámetros ingresados a través del panel de control, para esto se considera una muestra de 10 bobinas, construidas con los mismos parámetros y calibre de cable, para poder establecer comparaciones, la prueba se repite con las mismas condiciones que van desde el calibre 23 AWG al 18 AWG, considerados los de mayor frecuencia a trabajar por la máquina. Todas las bobinas son de 100 vueltas y se monitorea durante todo el proceso para identificar posibles diferencias en el bobinado.

En la Figura 4.2 se muestra la máquina bobinadora puesta en marcha de acuerdo con los parámetros previamente ingresados.

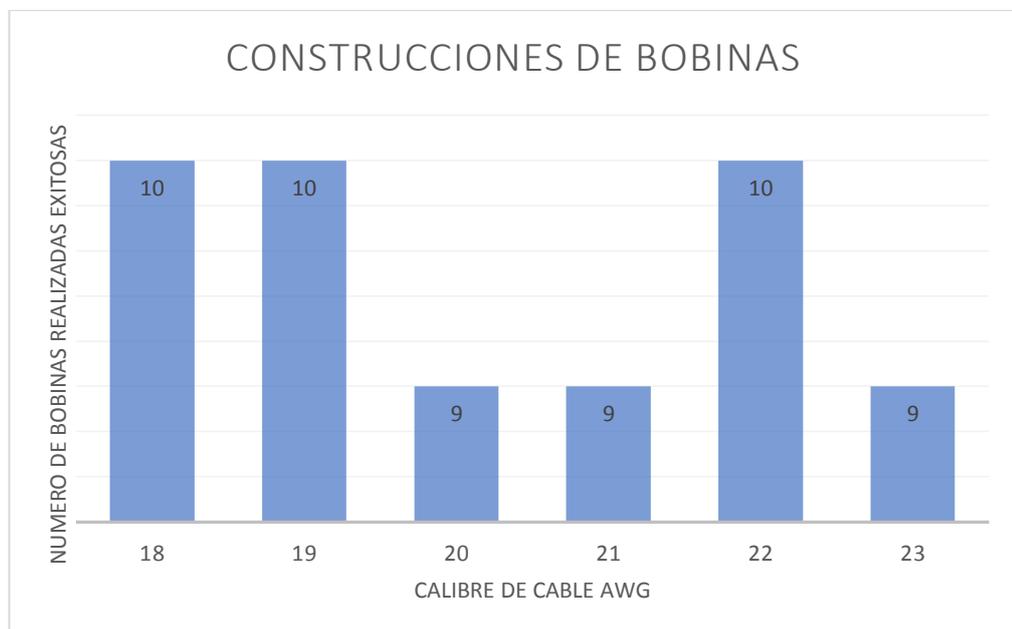
Figura 4.2 Fabricación de bobina en la máquina automatizada



Fabricación de bobina con parámetros establecidos. Elaborado por: Juan Achig, Fernando Pazmiño.

Una vez desarrolladas las bobinas y de acuerdo con la muestra establecida, se desarrolla un estudio estadístico para determinar la eficiencia de bobinado de la máquina considerando que si al bobinar se obtuvo las 100 vueltas completas y sin espacios entre vueltas. Para el calibre 23 AWG se obtuvo una eficiencia del 90% ya que solo una de las bobinas salió mal, debido a que se observó un espacio entre vueltas al iniciar el bobinado, este problema fue recurrente para cada calibre de cable, encontrando como solución la realización de un ajuste previo de los ejes de traslación cada vez que se cambie el calibre del cable a utilizar. Sin embargo, al utilizar calibres de cable mayores a 18 AWG, el problema ya no se repite, sin requerir de un ajuste previo, debido a que su sección es mayor. De esta manera para continuar con la prueba, con los demás calibres de cable, se realizó el ajuste y se obtuvo una eficiencia del 100% en la fabricación del bobinado.

Figura 4.3 Bobinas construidas con diferentes calibres de cable de cobre AWG



Análisis estadístico de construcción de bobinas. Elaborado por: Juan Achig, Fernando Pazmiño.

### 4.3. Prueba de control de calidad

Se realiza para evaluar el funcionamiento del sistema de visión artificial. Para esto se considera que el enfoque y la definición en la iluminación sea la adecuada, para que la captura de imágenes sea clara, con esto se puede realizar una corrección en el bobinado en caso de existir un cruce entre arrollamientos, o espacios entre bobinado, para esto se considera una muestra de 10 bobinas, construidas con los mismos

parámetros y calibre de cable de las pruebas de bobinado anteriores. En este caso se forzará n-veces el error del bobinado colocando un obstáculo entre el cable y el eje rotacional y así se genere la sobre montura de las espiras y el sistema entre en acción y realice el respectivo control.

En la Figura 4.4 se muestra una imagen tomada directamente por la cámara PS3 EYE involucrada en el sistema de visión artificial al momento del bobinado mostrando una línea color verde que es la referencia para el funcionamiento del control de calidad.

Figura 4.4 Visión artificial en proceso de bobinado

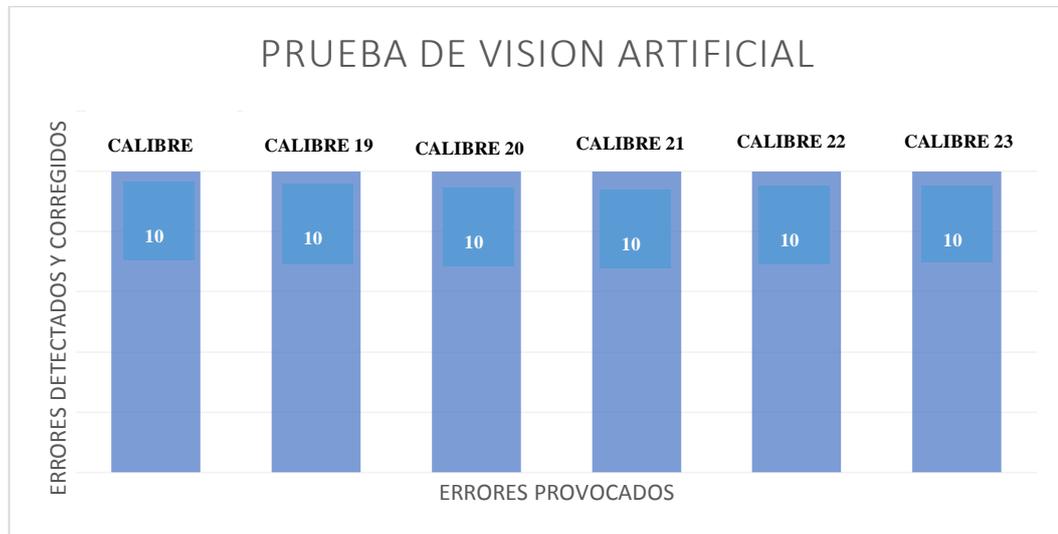


Supervisión de bobinado con visión artificial. Elaborado por: Juan Achig, Angel Pazmiño

Una vez desarrolladas las bobinas y puesto a prueba el funcionamiento del sistema de visión artificial, donde se provocó una sobre montura cada 5 vueltas en cada bobina construida, para esto se desarrolla un análisis estadístico para determinar la eficiencia de la visión artificial. Para el calibre 23 AWG hasta el calibre 21 AWG se obtuvo una eficiencia del 100% del sistema de visión artificial, al provocar el error se detectó de manera inmediata y se produjo su posterior corrección en el bobinado, para los calibres 20 AWG hasta el 18 AWG se provocó una cantidad mayor de errores en el bobinado de igual manera cada 5 vueltas, con lo cual el sistema tuvo una eficiencia del 100% para calibres con secciones más gruesas. En la Figura 4.5 se

muestra el estudio estadístico la detección de errores encontrados por parte de la visión artificial.

Figura 4.5 Análisis estadístico de pruebas de visión artificial



Análisis estadístico de errores detectados y corregidos durante el proceso de bobinado. Elaborado por: Juan Achig, Fernando Pazmiño.

#### 4.4. Prueba de aislamiento.

Esta prueba se realiza para comprobar el estado físico de las bobinas específicamente el aislamiento del cable, esta prueba se la realizar mediante la ayuda de un óhmetro para medir la resistencia de la bobina, se tomó una muestra de 10 medidas por bobina que se realizara en cada calibre de cable.

En la Figura 4.6 se muestra la medición de la resistencia de la bobina al terminar el proceso de bobinado

Figura 4.6 Medición de resistencia óhmica en bobinas construidas



Prueba de resistencia óhmica en bobinas fabricadas. Elaborado por: Juan Achig, Fernando Pazmiño.

Una vez desarrolladas las pruebas. Se realiza un análisis estadístico en el cual en la Tabla 4.1 se muestra los resultados obtenidos al compararlos con la resistencia teórica que debería tener la bobina de acuerdo con el Anexo 1. En el calibre 18 AWG se muestra un error muy bajo, muestra que para los posteriores calibres el porcentaje de error va aumentando. De acuerdo con el análisis de los resultados permite determinar que el error es debido al rose que existe al momento de arrollar el cable tanto en el eje de traslación como con el eje de rotación o su vez al activar el control de calidad por visión artificial, lo cual hace que se genere un retroceso en el cable, reduciendo esmaltado del cable y variando la resistencia, sin embargo, el error no supera el 5% siendo aceptable el rendimiento de la máquina.

Tabla 4.1 Tabla de cálculos para resistencia de bobinas fabricadas

MEDICIONES DE RESISTENCIAS DE BOBINAS DE CABLES						
CALIBRE (AWG)	LONGITUD (m)	RESISTENCIA ( $\Omega$ )	SECCION	RESISTENCIA CALCULADA	RESISTENCIA TEORICA	ERROR %
18	5,14	0,1	0,7854	0,11256429	0,1116	0,8640667
19	4,83	0,1	0,6362	0,13058157	0,1325	1,4478655
20	4,29	0,2	0,5027	0,14678337	0,1421	3,2958267
21	5,3	0,2	0,3959	0,23026016	0,2342	1,6822516
22	4,58	0,3	0,3117	0,25273018	0,25638	1,4235941
23	4,24	0,3	0,258	0,28266666	0,27153	4,1014498

Comparación y cálculo de error por resistencia del cable para bobinas fabricadas.

Elaborado por: Juan Achig, Angel Pazmiño



## CONCLUSIONES

- Se implementó el prototipo de máquina bobinadora automática utilizando un sistema de supervisión a través de visión artificial, permitiendo tener bobinas de diferentes calibres de cable y de diferente número de vueltas en un tiempo más corto en comparación con una máquina manual de bobinado y eliminando también el número de bobinas en mal estado que se pueden generar al bobinar con esta máquina manual.
- La caracterización de la máquina manual de bobinas para el rediseño de una máquina automática permite conocer las necesidades del sistema y definir las variables de control calibre de cable, número de vueltas para la fabricación de la bobina de tal manera que el sistema tenga un funcionamiento eficiente.
- El desarrollo del algoritmo permite que la máquina realice las funciones específicas de manera automática y se utilizó una plataforma robusta basada en software libre, como es Arduino 1.6.5 y procesamiento de imágenes al utilizar el software de Python 2.7 la cual genera una programación amigable y eficiente con la cual se cubren los requerimientos de la máquina bobinadora automática y hace que los sensores y actuadores se pongan en acción.
- La visión artificial que se utiliza en la máquina fabricadora de bobina automática permite la verificación de la calidad del bobinado, logrando así no tener cruces o sobre monturas en las espiras, disminuyendo el tiempo de fabricación de bobinas y teniendo una mayor cantidad de bobinas en buen estado.
- Al realizar el control mediante visión artificial se aplicó el método de bordes de imagen obteniendo así los límites del objeto requerido, se utiliza un sistema de iluminación de luz directa para que en cualquier punto de la bobina se pueda reconocer los bordes superiores, detectando si la bobina tiene monturas y corrigiéndolas al activar los actuadores que permiten realizar el desbobinado.
- Al realizar el control de resistencia tanto puesta en marcha como al finalizar la fabricación de las bobinas se genera una cierta pérdida en la resistencia del cable lo cual no genera un mal funcionamiento en las mismas ya que al obtener un error porcentual se maneja un 5% máximo, y las bobinas fabricadas en los diferentes calibres no sobrepasa este valor.



## RECOMENDACIONES

- Para mejorar el procesamiento de imágenes en visión artificial se puede utilizar una cámara Ultra HD la cual hace que se tenga un mayor número de píxeles y con esto podría realizar un mejor procesamiento de las imágenes ganando más detalles que pueden ser útiles para el proceso de automatización.
- Para la fabricación de bobinas de cable de cobre esmaltado es necesario tomar en cuenta además del calibre del cable, el espacio que ocupa el aislante, para que el carrete esté dimensionado correctamente para el número de vueltas deseadas y no haya pérdidas de espiras.
- Se recomienda utilizar la adquisición de imágenes para reemplazar a los sensores finales de carrera que al activarse envían una señal para que se realice el cambio de giro del motor que hace el movimiento del eje de traslación, de este modo se tendría una mayor utilidad de la cámara ya que es un sensor que simula el ojo humano, es decir la imagen podría indicar si el cable está llegando a los extremos y si es el caso se debería cambiar el sentido de giro del motor.
- Se recomienda crear un mini cuarto oscuro totalmente sellado con iluminación eficientemente distribuida que permita que la captura de imagen para el proceso de visión artificial sea más nítida y que no interfieran factores como variación de la luz del medio ambiente al momento de capturar la imagen.
- Para obtener mayor velocidad en el bobinado se recomienda reemplazar los motores a pasos por motores DC con encoder, debido que al tener motores a pasos, estos se deben sincronizar y se lo hace con respecto al motor de menor velocidad y al hacer que los motores a pasos tengan más velocidad, es decir, aumentar la frecuencia de excitación a la bobina para generar un paso puede suceder que el tiempo de excitación a la bobina no sea el suficiente y por ende no se realice el paso deseado.
- Para evitar problemas con la fuente de alimentación para los controladores, circuitos y actuadores se recomienda diseñar una fuente de tal manera que esté en rangos tanto de voltaje como de corriente que consume el quipo y con esto los actuadores, sensores y controladores funcionen eficientemente, ya

que especialmente los motores debido a la carga que tienen pueden consumir mayor corriente.

- Al utilizar un controlador como la Raspberry PI, se podría realizar todo el sistema de control y así evitar un panel frontal de ingreso de datos, ya que al tener una pantalla touch se puede hacer el ingreso desde este dispositivo.

## REFERENCIAS

- Albuja, E. (2012). *Construcción de una rebobinadora semiautomática para la confección de bobinas en los motores eléctricos trifásicos de dos y cuatro polos*. Quito.
- Aranda, D. (2014). *Electronica: Conceptos básicos y diseño de circuitos*. Buenos Aires.
- ARDUINO. (23 de septiembre de 2016). *ARDUINO.CL*. Recuperado el 19 de diciembre de 2017, de <http://arduino.cl/arduino-mega-2560/>
- BricoGeek. (23 de septiembre de 2017). *BricoGeek*. Obtenido de <http://tienda.bricogeek.com/motores-paso-a-paso/639-motor-paso-a-paso-14-kg-cm.html>
- Caceres, J. (23 de julio de 2013). *Aprendiendo arduino*. Recuperado el 2017 de 12 de 27, de *Aprendiendo arduino*: <https://aprendiendoarduino.wordpress.com/2015/03/23/microcontroladores/>
- Caniparoli, J. (2012). *Sensor de final de carrera*. Mexico.
- Cavarrubias, M. (2017). *Procesamiento de imágenes*. Mexico: UDLAP.
- Circuitos Electronicos. (15 de Julio de 2016). *Foros de electronica Basica*. Obtenido de <http://www.circuitoselectronicos.org/2011/03/teclado-matricial-4x4.html>
- Diaz Morales, A. (2015). *Inductores*. Bogota.
- Electronica Embajadores. (14 de ABRIL de 2017). *FINAL DE CARRERA MEDIANO CON PALANCA ROLDAN*. Obtenido de Características Final de Carrera IT41R13: <https://www.electronicaembajadores.com/es/Productos/Detalle/IT41R13/interruptores-conmutadores-pulsadores/finales-de-carrera/final-de-carrera-mediano-con-palanca-con-roldana>
- ELECTRONICLAB. (12 de octubre de 2015). *Ingeniería y diseño Electronico*. Obtenido de <https://electronilab.co/tienda/driver-motor-paso-a-paso-1a-a4988-pololu/>
- Fink, D. (1981). *Manual práctico de electricidad ingenieros*. Barcelona: REVERTE.
- I+D. (23 de febrero de 2014). *Electronica*. Obtenido de <http://www.didacticaselectronicas.com/index.php/optoelectronica/displays-lcd/display-lcd-caracteres-4x20-detail>
- IES Bellavista. (2015). *Introducción a los sistemas de control programado*. Mexico.

- LOGICELECTRONIC. (2016). *Inspeccion automatizada mediante sistemas de vision*. Lima.
- NATIONAL INSTRUMENTS. (12 de marzo de 2016). *National Instruments*.  
Obtenido de <http://sine.ni.com/nips/cds/view/p/lang/es/nid/203937>
- Nuñez, P. (2012). *Controlador Logico Programable*. Riobamba.
- ORIENTAL MOTOR. (12 de agosto de 2017). *Oriental Motor*. Obtenido de  
<https://catalog.orientalmotor.com/category/brushless-dc-motor-speed-control-systems>
- Perez Sandoval, J. A. (2015). *Inductancia*. Mexico.
- PlayStation. (2014). *Uso de la camara Playstation Eye*. Madrid.
- Tornero, J. (2015). *Control de Calidad mediante vision artificial*. Valencia.
- Universidad politecnica de Madrid. (2017). *Vision Artificial. Guia de aprendizaje*, 139.
- Universidad Politecnica de Valencia. (18 de diciembre de 2013). *museo informatica*.  
Obtenido de museo informatica:  
<http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>
- Vargas, K. (2014). *BOBINAS Y CONDENSADORES*. Lima.
- Vives, L. (2014). *Vision Artificial: Aplicacion de filtros y segmentacion en imagenes de hoja de cafe*. Mexico: Ciencia y tecnologica e Innovacion.

## **ANEXOS**

**Anexo 1.** Tabla de características físicas y eléctricas de cable de cobre AWG

Equiv. Aprox. A.W.G.	DIMENSIONES CONDUCTOR (d)				COBRE			
	Estandar (ESSEX)	Ø [mm]	Tolerancia [± mm]	Sección [mm <sup>2</sup> ]	Masa [g/m]	Resistencia		
						Minima [Ω/m]	Nominal [Ω/m]	Maxima [Ω/m]
33	*	0,180	0,003	0,02545	0,2262	0,6444	0,6718	0,7007
32	*	0,200	0,003	0,03142	0,2793	0,5237	0,5441	0,5657
31	*	0,224	0,003	0,03941	0,3503	0,4188	0,4338	0,4495
30	*	0,250	0,004	0,04909	0,4364	0,4345	0,4482	0,4628
29	*	0,280	0,004	0,06158	0,5474	0,3676	0,3776	0,3882
28	*	0,315	0,004	0,07793	0,6928	0,3121	0,3193	0,327
27	*	0,355	0,004	0,09898	0,8799	0,3674	0,3727	0,3782
26	*	0,400	0,005	0,12566	1,1172	0,3316	0,3136	0,3407
25	*	0,450	0,005	0,15904	1,4139	0,3042	0,3075	0,3109
24	*	0,500	0,005	0,1963	1,746	0,28462	0,28706	0,28959
23	*	0,560	0,006	0,2463	2,19	0,26736	0,2694	0,27153
22	*	0,630	0,006	0,3117	2,771	0,25335	0,25484	0,25638
21	*	0,710	0,007	0,3959	3,52	0,2198	0,2318	0,2342
20	*	0,800	0,008	0,5027	4,469	0,1305	0,1401	0,1421
19	*	0,900	0,009	0,6362	5,656	0,1317	0,1320	0,1325
18	*	1,000	0,010	0,7854	6,982	0,1116	0,1176	0,1116

## Anexo 2. Descripción y detalle de la distribución de pines en Arduino.

PIN	DESCRIPCIÓN	ACCIÓN
2	Inicia el bobinado después de ingresar los datos	Botón Inicio
3	Recibe la señal del final de carrera de la izquierda	Sensor izquierdo
4	Recibe la señal del final de carrera de la derecha	Sensor derecho
5	Detiene el bobinado hasta que vuelva ser pulsado	Botón pausa/continuar
8	Proporciona la señal al driver que controla los pasos del motor 2	Pasos del motor 2 (Guía)
9	Envía la señal al driver que controla el sentido de giro del motor 2	Dirección de giro del motor 2 (Guía)
10	Envía la señal al driver que controla los pasos del motor 1	Pasos del motor 1 (Bobinador)
11	Envía la señal al driver que controla los pasos del motor 3	Pasos del motor 1 (Recogedor)
12	Envía la señal al driver que controla el sentido de giro del motor 1	Dirección de giro del motor 1 (Bobinador)
13	Envía la señal al driver que controla el sentido de giro del motor 3	Dirección de giro del motor 3 (Recogedor)
20	Envía una señal a la Raspberry Pi para que realice el proceso de visión artificial	Control para la realizar visión artificial
21	Recibe una señal de la Raspberry Pi y hace que cambie el sentido de giro de los motores 1 y 3	Retroceso del cable
22	Pin 1 de fila del teclado matricial	Filas del teclado matricial
23	Pin 2 de fila del teclado matricial	
24	Pin 3 de fila del teclado matricial	
25	Pin 4 de fila del teclado matricial	
26	Pin 1 de columna del teclado matricial	Columnas del teclado matricial
27	Pin 2 de columna del teclado matricial	
28	Pin 3 de columna del teclado matricial	
29	Pin 4 de columna del teclado matricial	
34	Pin RS de la pantalla LCD 20 x 4, se selecciona el dispositivo para su uso	RS de la pantalla LCD
35	Pin Enable de la pantalla LCD 20 x 4, habilita o deshabilita el LCD	Enable de la pantalla LCD
36	Pin 1 de datos que se envía del Arduino a la LCD	Datos por donde se envía la información del Arduino a la LCD
37	Pin 2 de datos que se envía del Arduino a la LCD	
38	Pin 3 de datos que se envía del Arduino a la LCD	
39	Pin 4 de datos que se envía del Arduino a la LCD	
44	Envía una señal a la Raspberry Pi cada vez que se realiza una vuelta de bobinado	Contador de vuelta positivo
45	Envía una señal a la Raspberry Pi cada vez que se realiza una vuelta de retroceso	Contador de vuelta negativo
46	Envía una señal a la Raspberry Pi indicando que el bobinado ha finalizado	Fin

PIN	DESCRIPCIÓN	ACCIÓN
5	Envía una señal desde la Raspberry Pi hacia el Arduino para que cambie el sentido de giro de los motores 1 y 3	Retroceso del cable
12	Recibe una señal desde el Arduino para que realice el proceso de visión artificial	Control por visión artificial
16	Recibe una señal desde el Arduino cada vez que se realiza una vuelta de bobinado	Contador de vuelta positivo
20	Recibe una señal desde el Arduino cada vez que se realiza una vuelta de retroceso	Contador de vuelta negativo
21	Recibe una señal desde el Arduino indicando que el bobinado ha finalizado	Fin

**Anexo 3.** Descripción y distribución para pines de Raspberry PI Model B

**Anexo 4.** Código de programación en Arduino que permite el control de motores y visualización juntamente con entrada de parámetros.

Código del arduino:

```
#include <LiquidCrystal.h>
#include <Keypad.h>
```

```
const int steppin1 = 10; // declaramos el pin 10 del arduino como step (motor grande)
const int dirpin1 = 12; // declaramos el pin 12 del arduino con dir (motor grande)
const int steppin2 = 8; // declaramos el pin 8 del arduino como step (motor pequeño)
const int dirpin2 = 9; // declaramos el pin 9 del arduino con dir (motor pequeño)
const int steppin3 = 11; // declaramos el pin 52 del arduino como step (motor de ayuda y retro)
const int dirpin3 = 13; // declaramos el pin 53 del arduino con dir (motor de ayuda y retro)
const int tiempo = 450; // declaramos la variable tiempo como (2 milisegundos)
const int tiempo1 = 6000; // declaramos la variable tiempo como (2 milisegundos)
const int encender = 2; // declaramos el pin 2 del arduino como encender
const int senizq = 3; // declaramos el pin 3 para el sensor límite de la izquierda
const int sender = 4; // declaramos el pin 4 para el sensor límite de la derecha
const int pausa = 5; // declaramos el pin 5 para el botón de pausa
const int retro = 48; // declaramos el pin 21 como entrada de la señal de la raspberry pi
const int control=47; // declaramos el pin 20 como salida de señal para que la raspberry haga el control de calidad
const int masvuelta=44; // declaramos el pin 44 como contador de vueltas positivas de la raspberry en modo serial
const int menosvuelta=45; // declaramos el pin 45 como contador de vueltas negativas de la raspberry en modo serial
const int fin=46; // declaramos el pin 46 como fin de bobinado, el mismo que indica a la raspberry que se termino el bobinado
```

```
LiquidCrystal lcd(34, 35, 39, 38, 37, 36); //Inicializamos la librería con los pines del arduino a utilizar (RS, ENABLE, D4-D7)
```

```
const byte FILA = 4; //Número de filas
```

```

const byte COL = 4; //Número de columnas
char keys[FILA][COL]={
  {'1','2','3','A'}, // configuración del teclado 4x4
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte filaPins[FILA]={22, 23, 24, 25}; // Pines de las filas en el arduino mega (teclado)
byte colPins[COL]={26, 27, 28, 29}; // Pines de las columnas en el arduino mega (teclado)
Keypad keypad=Keypad(makeKeymap(keys), filaPins, colPins, FILA, COL); // Inicializamos el keypad

char key=' '; //Inicialización de variable key para almacenar dato del keypad
int numvueltas=0; // valor real del numero de vueltas
int cont=0; // cuenta el numero de digitos del numero
char entrada[9]; //vector de 9 digitos para el valor de entrada de vueltas
char numchar[9]; //vector donde se almacena el valor de entrada del numero de vueltas
char numvser[6]; //Vector de 6 digitos para el valor de entrada de vueltas por serial
char numvserchar; //Variable donde se almacena el valor del numero de vueltas ingresado por
serial
int contnum=0; //Variable donde se guarda el numero de digitos que tiene el num. de vueltas serial

char entradacal[9]; //Vector de 2 digitos para el valor de entrada del calibre
char calchar[9]; //Vector donde se almacena el valor de la entrada del calibre
char calibreser[9]; //Vector donde se guardará el valor de la entrada calibre por serial
char calserchar; //Variable donde se almacena el valor del calibre ingresado por serial
int calser=0; //Valor real del calibre ingresado opor serial
int calibre=0; //valor real del calibre AWG
int contcal=0; //Valor del contaor del vector entrada calibre
int contpausa=0;
int paso123=0;

bool entrarbucle0=true; //valor para entrar al bucle de selección de bobinado o puebas
bool entrarbucle1=true; //valor para entrar al bucle de calibre
bool entrarbucle1ser=true; //Valor para entrar al bucle serial de calibre
bool entrarbucle2=true; //valor para entrar al bucle de numero de vueltas
bool entrarbucle2ser=true; //valor para entrar al bucle serial de numero de vueltas
bool entrarbucle3ser=true; //Valor para entrar al bucle que obtiene el numero de digitos del num.
de vueltas serial
bool entrarpausa=false; //valor para entrar al bucle pausa
bool entrarretro=false; //valor para entrar al condicional retro
bool entrarencender=true; //Valor para entrar al bucle que contiene el condicional de start
bool bucleconti=true; //Valor para entrar al bucle que espera a que la raspberry envíe un valor
para hacer una nueva bobina

char conti='n'; //Variable donde se guarda el valor serial enviado por la raspberry para hacer
una nueva bobina

int vueltasretro=0; //numero de vueltas que gira el motor principal en reversa.
int numvueltasactual=0; // variable donde se almacena el valor del numero de vueltas cuando el pin
retro se active.
int vueltas=0; //Numero de vueltas a visualizarse en la lcd.
int contretro=0; //contador del numero de vueltas de retro.

int pulsos; //Numero de pulsos del motor 2 (motor guía)

```

```

int opcion; //Número para escoger la opcion de bobinado o pruebas

void setup() {
  Serial.begin(9600);
  //Serial.print('o');
  pinMode(steppin1, OUTPUT); // declaramos como salidas
  pinMode(dirpin1, OUTPUT); // declaramos como salida
  pinMode(steppin2, OUTPUT); // declaramos como salidas
  pinMode(dirpin2, OUTPUT); // declaramos como salida
  pinMode(steppin3, OUTPUT); // declaramos como salidas
  pinMode(dirpin3, OUTPUT); // declaramos como salida
  pinMode(encender, INPUT); // declaramos como entrada
  pinMode(senizq, INPUT); // declaramos como entrada
  pinMode(sender, INPUT); // declaramos como entrada
  pinMode(pausa, INPUT); // declaramos como entrada
  pinMode(control, OUTPUT); // declaramos como salida
  pinMode(masvuelta, OUTPUT); // declaramos como salida
  pinMode(menosvuelta, OUTPUT); // declaramos como salida
  pinMode(retro, INPUT); // declaramos como entrada
  pinMode(fin, OUTPUT); //declaramos como salida

  lcd.begin(20, 4); //Colocamos en numero de filas y columnas de la LCD
  lcd.setCursor(0, 0); //Ubicación del curso
  lcd.print(" U.P.S. "); //Texto que se mostrará en la pantalla
  lcd.setCursor(0, 1); //Ubicación del curso
  lcd.print(" BOBINADORA "); //Texto que se mostrará en la pantalla
  lcd.setCursor(0, 2); //Ubicación del curso
  lcd.print(" AUTOMATICA "); //Texto que se mostrará en la pantalla
  delay(1000);
  lcd.clear();
}

void loop() {

  while(entrarbucle0==true){
    lcd.setCursor(0, 0); //Ubicación del cursor
    lcd.print("1.- FABRICAR BOBINA"); //Texto que se mostrará en la pantalla
    lcd.setCursor(0, 2); //Ubicación del cursor
    lcd.print("2.- REALIZAR PRUEBAS"); //Texto que se mostrará en la pantalla
    while(entrarbucle0==true){
      char key= keypad.getKey(); //Si la tecla es presionada esta es almacenada en la variable key
      if (Serial.available() > 0){
        char dato=Serial.read();
        if(dato == '3'){
          opcion=3;
          entrarbucle0=false;
        }
      }
    }
    if(key != NO_KEY && key=='1'){
      opcion=1;
      entrarbucle0=false;
    }
    if(key != NO_KEY && key=='2'){
      opcion=2;
      entrarbucle0=false;
    }
  }
}

```

```

}
}

switch(opcion){
case 1:
  lcd.clear();
  while(entrarbucl1==true){
  lcd.setCursor(0, 0); //Ubicación del cursor
  lcd.print("INGRESE EL CALIBRE"); //Texto que se mostrará en la pantalla
  lcd.setCursor(0, 1); //Ubicación del cursor
  lcd.print("AWG ENTRE 23 A 18:"); //Texto que se mostrará en la pantalla
  //////////////////////////////////// TOMA DE DATOS POR TECLADO DEL CALIBRE DEL CABLE
  ////////////////////////////////////
  calibre=Ingresarnum18a23();
  pulsos=ContadorPulsos(calibre);
  vueltasretro=ContRetro(calibre);
  }

  ////////////////////////////////////
  ////////////////////////////////////
  while(entrarbucl2==true){
  lcd.setCursor(0, 0); //Ubicación del cursor
  lcd.print("INGRESE EL NUMERO DE"); //Texto que se mostrará en la pantalla
  lcd.setCursor(0, 1); //Ubicación del cursor
  lcd.print("VUELTAS DE LA BOBINA"); //Texto que se mostrará en la pantalla
  //////////////////////////////////// TOMA DE DATOS POR TECLADO DEL NUMERO DE VUELTAS
  ////////////////////////////////////
  numvueltas=Ingresarnum();
  }
  //////////////////////////////////// MOTORES
  ////////////////////////////////////
  ///
  digitalWrite(dirpin1, HIGH); // pone en bajo , o en 0 logico para que gire en sentido
antihorario (MOTOR 1)
  digitalWrite(dirpin2, LOW); // pone en alto, o en 1 logico para qe gire de tal manera que avanza
a la derecha. (MOTOR 2)
  digitalWrite(dirpin3, LOW); // pone en bajo , o en 0 logico para que gire en sentido antihorario
(MOTOR DE AYUDA Y RETRO)
  while(entrarencender==true){
  if (digitalRead(encender)==HIGH){ // ingresa al condicional si el botor de star se presiona
  lcd.clear();
  lcd.setCursor(0, 0); //Ubicación del cursor
  lcd.print("CALIBRE AWG: "); //Texto que se mostrará en la pantalla
  lcd.setCursor(13, 0); //Ubicación del cursor
  lcd.print(calibre); //Texto que se mostrará en la pantalla
  lcd.setCursor(0, 1); //Ubicación del cursor
  lcd.print("NUM. VUELTAS: "); //Texto que se mostrará en la pantalla
  lcd.setCursor(15, 1); //Ubicación del cursor
  lcd.print(numvueltas); //Texto que se mostrará en la pantalla
  for(int numvm1=0; numvm1 < numvueltas; numvm1++){ // el bucle se ejecuta hasta cumplir
el numero de vueltas numreal
  vueltas=numvm1;
  lcd.setCursor(0, 2); //Ubicación del cursor
  lcd.print(vueltas); //Texto que se mostrará en la pantalla
  for (int pasom1 = 0; pasom1 < 3600; pasom1++){ // entramos a un nuevo lazo pero ahora la
unica diferencia es que

```

```
// que dara 3600 pasos o 1 vueltas de 360 grados para el MOTOR
```

2.

```
    if(digitalRead(pausa)==HIGH){          // si pulsa el boton pausa, la variable entrarpause
cambia a verdadero y con eso entra al bucle
        entrarpause=true;
        delay(500);
    }
    while(entrarpause==true){              //se ejecuta el bucle cuando entrarpause=true, con
esto se evita que los motores giren, es decir,
        lcd.setCursor(7, 2);              // el proceso se detiene y se muestra en la lcd el mensaje
de "PAUSA"
        lcd.print("PAUSA");
        if(digitalRead(pausa)==HIGH){
            entrarpause=false;
            lcd.setCursor(5, 2);
            lcd.print("      "); delay(500);}
    }

    if(digitalRead(retro)==HIGH){          // si se activa el pin retro, entra al condicional para
girar el motor en sentido contrario.
        digitalWrite(dirpin1, LOW);
        digitalWrite(dirpin3, HIGH);      //el motor de ayuda y retro gira en sentido contrario
        digitalWrite(dirpin2,!digitalRead(dirpin2));
        numvueltasactual=numvm1;
        pasom1=0;
        entrarretro = true;
    }
    digitalWrite(control,LOW);
    if (digitalRead(senizq)==HIGH){        // el condicional entra solo cuando el sensor izq. sea
pulsado y el motor 2
        digitalWrite(dirpin2, LOW);      // gira de tal modo de avanzar hacia la derecha.
        contpausa=1;
    }
    if (digitalRead(sender)==HIGH){        // el condicional entra solo cuando el sensor der. sea
pulsado y el motor 2
        digitalWrite(dirpin2, HIGH);     // gira de tal modo de avanzar hacia la izquierda.
        contpausa=1;
    }
    digitalWrite(stepin1, HIGH);          //Se enciende el pin de pasos del los tres motores, en el
motor 2(Guía) solo realizar un cierto numero de pulsos.
    if(pasom1%9==0){
        digitalWrite(stepin3, HIGH);}
    if(pulsos>pasom1){
        digitalWrite(stepin2, HIGH);
    }
    delayMicroseconds(tiempo);            // Se espera un tiempo para que reconozca la siguiente
accion
    digitalWrite(stepin1, LOW);           //Se apaga el pin de pasos del los tres motores, en el
motor 2(Guía) solo realizar un cierto numero de pulsos.
    if(pasom1%9==0){
        digitalWrite(stepin3, LOW);}
    if(pulsos>pasom1){
        digitalWrite(stepin2, LOW);
        if(contpausa==1){
            delayMicroseconds(tiempo);
            contpausa=0;
        }
    }
}
```

```

    }
    delayMicroseconds(tiempo);
}

if(numvm1>5 && numvm1%vueltasretro==0){
    digitalWrite(control,HIGH);
}
//else{digitalWrite(control,LOW);}
if(entrarretro==true){
    if(contretro<vueltasretro){
        numvm1=numvm1-2;
        contretro++;
    }
    if(contretro==vueltasretro){
        numvm1=numvueltasactual-vueltasretro;
        contretro=0;
        digitalWrite(dirpin1, HIGH);
        digitalWrite(dirpin3, LOW);
        digitalWrite(dirpin2,!digitalRead(dirpin2));
        entrarretro=false;
    }
}
}
}
entrarbucle0=true;
entrarbucle1=true;
entrarbucle2=true;
entrarencender=false;
lcd.clear();
lcd.setCursor(0, 1); //Ubicación del cursor
lcd.print("FIN DEL BOBINADO"); //Texto que se mostrará en la pantalla
digitalWrite(fin,HIGH);
//lcd.setCursor(0, 2); //Ubicación del cursor
//lcd.print(entradacal); //Texto que se mostrará en la pantalla
//lcd.setCursor(0, 3); //Ubicación del cursor
//lcd.print(entrada); //Texto que se mostrará en la pantalla
delay(5000);
digitalWrite(fin,LOW);
lcd.clear();
//memset(entradacal,0,sizeof(entradacal));
//memset(entrada,0,sizeof(entrada));
}
}
break;

case 2:
    lcd.clear();
    lcd.setCursor(0, 1); //Ubicación del cursor
    lcd.print("PRUEBAS!!!!!!"); //Texto que se mostrará en la pantalla
    delay(2000);
    lcd.clear();
    entrarbucle0=true;
break;

case 3:

    lcd.clear();

```

```

    lcd.setCursor(0, 0); //Ubicación del cursor
    lcd.print("CONEXION SERIAL"); //Texto que se mostrará en la pantalla
    delay(500);
    while(entrarbucle1ser==true){ //Bucle para tomar los datos del calibre de forma serial desde
la raspberry
    if (Serial.available() > 0){ //Si el puerto serial esta disponible entra al condicional
    calserchar=Serial.read(); //Se recibe bite a bite el valor del calibre
    calibreser[0]=calserchar;
    delay(500);
    calserchar=Serial.read();
    calibreser[1]=calserchar;
    calibre=atoi(calibreser); //Se convierte el valor char del calibre a entero
    }

    if(calibre != 0){ //Si el valor del calibre es diferente de cero, es decir se a recibido el
valor serial sale del bucle del calibre.
    entrarbucle1ser=false;
    }

    pulsos=ContadorPulsos(calibre); //En la variable pulsos se gurada el valor de numero de
pulsos por vuelta que debe dar el motor guía(motor 2)
    vueltasretro=ContRetro(calibre); //En esta variable guardamos el numero de vueltas que
volverá la maquina cada vez que haya un fallo, esta depende del calibre
    }

    while(entrarbucle2ser==true){
    while(entrarbucle3ser==true){
    if(Serial.available(>0){
    char contnumchar[]={'0'};
    contnumchar[0]=Serial.read();
    contnum=atoi(contnumchar);
    if(contnum != 0){
    entrarbucle3ser=false;
    }
    }
    }
    }

    if (Serial.available() > 0){
    for(int i=0; i<contnum; i++){
    numvserchar=Serial.read();
    numvser[i]=numvserchar;
    //lcd.setCursor(i, 1); //Ubicación del cursor
    //lcd.print(numvser[i]); //Texto que se mostrará en la pantalla
    }

    numvueltas=atoi(numvser);
    }
    //lcd.setCursor(0, 3); //Ubicación del cursor
    //lcd.print(numvser); //Texto que se mostrará en la pantalla
    //lcd.setCursor(10, 3); //Ubicación del cursor
    // lcd.print(numvueltas); //Texto que se mostrará en la pantalla
    delay(3000);
    if(numvueltas != 0){
    entrarbucle2ser=false;
    }
    }
}

```

```

digitalWrite(dirpin1, HIGH); // pone en bajo , o en 0 logico para que gire en sentido
antihorario (MOTOR 1)
digitalWrite(dirpin2, LOW); // pone en alto, o en 1 logico para qe gire de tal manera que avanza
a la derecha. (MOTOR 2)
digitalWrite(dirpin3, LOW); // pone en bajo , o en 0 logico para que gire en sentido antihorario
(MOTOR DE AYUDA Y RETRO)
//while(entrarencender==true){
//if (digitalRead(encender)==HIGH){ // ingresa al condicional si el botor de start se presiona
lcd.clear();
lcd.setCursor(0, 0); //Ubicación del cursor
lcd.print("CALIBRE AWG: "); //Texto que se mostrará en la pantalla
lcd.setCursor(13, 0); //Ubicación del cursor
lcd.print(calibre); //Texto que se mostrará en la pantalla
lcd.setCursor(0, 1); //Ubicación del cursor
lcd.print("NUM. VUELTAS: "); //Texto que se mostrará en la pantalla
lcd.setCursor(15, 1); //Ubicación del cursor
lcd.print(numvueltas); //Texto que se mostrará en la pantalla
for(int numvm1=0; numvm1 < numvueltas; numvm1++){ // el bucle se ejecuta hasta cumplir
el numero de vueltas numreal
vueltas=numvm1;
lcd.setCursor(0, 2); //Ubicación del cursor
lcd.print(vueltas); //Texto que se mostrará en la pantalla
for (int pasom1 = 0; pasom1 < 3600; pasom1++){ // entramos a un nuevo lazo pero ahora la
unica diferencia es que
// que dara 3600 pasos o 1 vueltas de 360 grados para el MOTOR

```

2.

```

if(digitalRead(pausa)==HIGH){ // si pulsa el boton pausa, la variable entrarpause
cambia a verdadero y con eso entra al bucle
entrarpause=true;
delay(500);
}
while(entrarpause==true){ //se ejecuta el bucle cuando entrarpause=true, con
esto se evita que los motores giren, es decir,
lcd.setCursor(7, 2); // el proceso se detiene y se muestra en la lcd el mensaje
de "PAUSA"
lcd.print("PAUSA");
if(digitalRead(pausa)==HIGH){
entrarpause=false;
lcd.setCursor(5, 2);
lcd.print(" "); delay(500);}
}
digitalWrite(control,LOW);
if(digitalRead(retro)==HIGH){ // si se activa el pin retro, entra al condicional para
girar el motor en sentido contrario.
digitalWrite(dirpin1, LOW); //se invierte el giro del motor 1
digitalWrite(dirpin3, HIGH); //Se invierte el giro del motor de ayuda y retro
digitalWrite(dirpin2,!digitalRead(dirpin2));
digitalWrite(menosvuelta, HIGH); // se enciende el pin para avisar que las vueltas
decrecen en modo serial.
numvueltasactual=numvm1; //Guardo en la variable en numero de vueltas actual.
pasom1=0;
entrarretro = true;
}
if (digitalRead(senizq)==HIGH){ // el condicional entra solo cuando el sensor izq. sea
pulsado y el motor 2
digitalWrite(dirpin2, LOW); // gira de tal modo de avanzar hacia la derecha.
contpausa=1;

```

```

    }
    if (digitalRead(sender)==HIGH){           // el condicional entra solo cuando el sensor der. sea
pulsado y el motor 2
        digitalWrite(dirpin2, HIGH);        // gira de tal modo de avanzar hacia la izquierda.
        contpausa=1;
    }

    digitalWrite(masvuelta,LOW);           //el pin que cuenta las vueltas positivas en modo serial
se apaga

    digitalWrite(stepin1, HIGH);           //Se enciende el pin de pasos del los tres motores, en el
motor 2(Guía) solo realizar un cierto numero de pulsos.
    if(pasom1%8==0){
        digitalWrite(stepin3, HIGH);}
    if(pulsos>pasom1){
        digitalWrite(stepin2, HIGH);
    }
    delayMicroseconds(tiempo);           // Se espera un tiempo para que reconozca la siguiente
accion
    digitalWrite(stepin1, LOW);           //Se enciende el pin de pasos del los tres motores, en el
motor 2(Guía) solo realizar un cierto numero de pulsos.
    if(pasom1%8==0){
        digitalWrite(stepin3, LOW);}
    if(pulsos>pasom1){
        digitalWrite(stepin2, LOW);
        if(contpausa==1){
            delayMicroseconds(tiempo);
            contpausa=0;
        }
    }
    delayMicroseconds(tiempo);
}
digitalWrite(masvuelta,HIGH);           //se enciende el pin de vueltas positivas en modo serial
if(numvm1>5 && numvm1%vueltasretro==0){
    digitalWrite(control,HIGH);
}
if(entrarretro==true){
    if(contretro<vueltasretro){
        numvm1=numvm1-2;
        contretro++;
    }
}
if(contretro==vueltasretro){
    numvm1=numvueltasactual-vueltasretro;
    contretro=0;
    digitalWrite(dirpin1, HIGH);
    digitalWrite(dirpin3, LOW);
    digitalWrite(dirpin2,!digitalRead(dirpin2));
    digitalWrite(menosvuelta,LOW);
    entrarretro=false;
}
}
}
}
entrarbucle1ser=true;
entrarbucle2ser=true;
entrarbucle3ser=true;
//entrarencender=false;
lcd.clear();

```

```

    lcd.setCursor(0, 1); //Ubicación del cursor
    lcd.print("FIN DEL BOBINADO"); //Texto que se mostrará en la pantalla
    digitalWrite(fin,HIGH);
    //lcd.setCursor(0, 2); //Ubicación del cursor
    //lcd.print(entradacal); //Texto que se mostrará en la pantalla
    //lcd.setCursor(0, 3); //Ubicación del cursor
    //lcd.print(entrada); //Texto que se mostrará en la pantalla
    while(bucleconti==true){
    if (Serial.available() > 0){
        conti=Serial.read();
        if(conti=='s'){
            bucleconti=false;
        }
    }
    }
    //delay(5000);
    digitalWrite(fin,LOW);
    lcd.clear();
    //memset(calibreser,0,sizeof(calibreser));
    //memset(numvser,0,sizeof(numvser));
    //}
    //}

    entrarbucle0=true;
    opcion=0;
    break;

    // default:
    //lcd.clear();
    // lcd.setCursor(0, 1); //Ubicación del cursor
    // lcd.print("INGRESE EL VAL. 1 o 2"); //Texto que se mostrará en la pantalla
    // delay(2000);
    // lcd.clear();
    // entrarbucle0=true;
    // break;
}
}
}

```

```

//////////////////////////////////// FUNCION INGRESAR NUMERO ENTERO ENTRE 18-
23////////////////////////////////////

```

```

int Ingresarnum18a23(){
    bool entrar = true;
    int numreal;
    char espacio = ' ';
    int num;

    while (entrar == true){
        char key= keypad.getKey(); //Si la tecla es presionada esta es almacenada en la variable key
        if (key != NO_KEY && (key=='1' || key=='2' || key=='3' || key=='4' || key=='5' || key=='6' || key=='7' ||
        key=='8' || key=='9' || key=='0')){ // entra al condicional si se presiona digitos del 0 al 9
            entradacal[contcal]=key; // guarda el valor de la tecla presionada en el vector entrada
            lcd.setCursor(contcal, 2); //Ubicación del cursor
            lcd.print(key); //El valor de la variable es impreso en el lcd
            contcal++; // se incrementa el contador en 1, el contador indica la posición del vector.
        }
    }
}

```

```

if(contcal > 2){
    memset(entradacal,0,sizeof(entradacal)); //Borra el contenido del vector
    Error();
    contcal=0;
}
}
if (key != NO_KEY && key=='A'){ // entra al condicional si se presiona el digito 'A'
    contcal--; // se decrementa el contador en 1, el contador indica la posición del vector.
    if(contcal < 0)
        contcal=0;
    entradacal[contcal]=' '; // guarda el valor de la tecla presionada en el vector entrada
    lcd.setCursor(contcal, 2); //Ubicación del cursor
    lcd.print(espacio); //El valor de la variable es impreso en el lcd
}
if(key != NO_KEY && key=='D'){ // entra al condicional si se presiona: D
    if(entradacal[1]=='D'){
        memset(entradacal,0,sizeof(entradacal)); //Borra el contenido del vector
        Error();
        contcal=0;
    }
    num = atoi(entradacal);
    contcal=0;
    if(num==18 || num==19 || num==20 || num==21 || num==22 || num==23){
        contcal=0; // el contador vuelve a cero
        for(int i = 0; i < 2; i++) { calchar[i] = entradacal[i]; } //Se guarda en el vector numchar los datos
        del vector entrada
        memset(entradacal,0,sizeof(entradacal)); //Borra el contenido del vector
        entrar=false;
        entrarbucle1=false;
        lcd.clear();
    }
    else{
        Error();
        contcal=0;
    }
}
}
numreal = atoi(calchar); // convierte a entero
memset(calchar,0,sizeof(calchar)); //Borra el contenido del vector
return numreal;
}
}

```

```

////////////////////////////////////////// FUNCION INGRESAR NUMERO
ENTERO//////////////////////////////////////////

```

```

int Ingresarnum(){
    bool entrar = true;
    int numreal;
    char espacio = ' ';
    while (entrar == true){
        char key= keypad.getKey(); //Si la tecla es presionada esta es almacenada en la variable key
        if (key != NO_KEY && (key=='1' || key=='2' || key=='3' || key=='4' || key=='5' || key=='6' || key=='7' ||
        key=='8' || key=='9' || key=='0')){ // entra al condicional si se presiona digitos del 0 al 9
            entrada[cont]=key; // guarda el valor de la tecla presionada en el vector entrada
            lcd.setCursor(cont, 2); //Ubicación del cursor
            lcd.print(key); //El valor de la variable es impreso en el lcd
            cont++; // se incrementa el contador en 1, el contador indica la posición del vector.
            if(cont >6){

```

```

    memset(entrada,0,cont); //Borra el contenido del vector
    Error();
    cont=0;
}
}
if (key != NO_KEY && key=='A'){ // entra al condicional si se presiona el digito 'A'
    cont--; // se decrementa el contador en 1, el contador indica la posición del vector.
    if(cont < 0)
        cont=0;
    entrada[cont]=' '; // guarda el valor de la tecla presionada en el vector entrada
    lcd.setCursor(cont, 2); //Ubicación del cursor
    lcd.print(espacio); //El valor de la variable es impreso en el lcd
}
if(key != NO_KEY && key=='D'){ // entra al condicional si se presiona: A,B,C,D,*,#
    if(entrada[1]=='D'){
        memset(entrada,0,sizeof(entrada)); //Borra el contenido del vector
        Error();
        cont=0;
    }
    cont=0; // el contador vuelve a cero
    for(int i = 0; i < 8; i++) { numchar[i] = entrada[i]; } //Se guarda en el vector numchar los datos del
vector entrada
    memset(entrada,0,sizeof(entrada)); //Borra el contenido del vector
    entrar=false;
    entrarbucle2=false;
    entrarencender=true;
    lcd.clear();
    lcd.setCursor(0, 3); //Ubicación del cursor
    lcd.print(" PRESIONE START "); //Texto que se mostrará en la pantalla
}
    numreal = atoi(numchar); // convierte a entero
    return numreal;
}
}
}

```

```

////////////////////////////////////// CONTADOR DE PULSOS DEL SEGUNDO MOTOR (MOTOR
GUÍA) ////////////////////////////////////////

```

```

int ContadorPulsos(int cal){
    int numreal; //Valor que retornará la función
    float constante; //Valor que contiene la constante de del tornillo sin fin
    float numpulsos; //Valor de pulsos con punto flotante
    float valormm; //Valor en mm del calibre AWG ingresado

    if(cal==18){ //Si el calibre es 18 el valor en mm es 1.024
        valormm=1.077;}

    if(cal==19){ //Si el calibre es 19 el valor en mm es 0.912
        valormm=0.963;}

    if(cal==20){ //Si el calibre es 20 el valor en mm es 0.812
        valormm=0.884;}

    if(cal==21){ //Si el calibre es 21 el valor en mm es 0.723
        valormm=0.770;}

    if(cal==22){ //Si el calibre es 22 el valor en mm es 0.664
        valormm=0.686;}
}

```

```

if(calibre==23){          //Si el calibre es 23 el valor en mm es 0.573
  valormm=0.617;}

  constante=(valormm*22)/(25.4); //Valor de multiplicar en valor en mm del cable por la constante
del tornillo de 22pulgadas
  numpulsos=3600*constante; //la constante por el numero de pulsos para hacer girar una vuelta a
los motores
  numreal=round(numpulsos); //Redondea el valor del numero de pulsos
  return numreal;
}

////////////////////////////////////// CONTADOR DE VUELTAS CUANDO HAYA UN ERROR EN EL
BOBINADO ////////////////////////////////////////
int ContRetro(int cal){
int vuelt;
  if(cal==18){vuelt=5;}
  if(cal==19){vuelt=6;}
  if(cal==20){vuelt=7;}
  if(cal==21){vuelt=8;}
  if(cal==22){vuelt=9;}
  if(cal==23){vuelt=10;}
  return vuelt;
}

////////////////////////////////////// MENSAJE DE ERROR
//////////////////////////////////////
void Error(){
  for(int i=0 ; i<3 ; i++){
    lcd.setCursor(0, 2); //Ubicación del cursor
    lcd.print("  ERROR  "); //Se muestra el texto ERROR en la pantalla
    delay(500);
    lcd.setCursor(0, 2); //Ubicación del cursor
    lcd.print("      "); // Se muestra un espacio en blanco para simular un parpadeo
    delay(500);
  }
}

```

## Anexo 6. Características Arduino Mega

<b>Microcontrolador</b>	ATmega 2560
<b>Voltaje de entrada</b>	5- 12V
<b>Entradas análogas</b>	16
<b>Velocidad de reloj</b>	16 MHz
<b>Entradas/Salidas digitales</b>	54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM)

### **Anexo 7. Características técnicas controlador Raspberry PI**

<b>Procesador</b>	Chipset Broadcom BCM2835, ARM1176JZF-S a 700 MHz
<b>Procesador Grafico</b>	VideoCore IV
<b>Memoria RAM</b>	512 MB
<b>Conexión</b>	USB 2.0
<b>Salida Analógica</b>	RCA
<b>Salida LAN</b>	RJ45

### **Anexo 8. Características de LCD**

Alimentación	5V
Formato de presentación	20 caracteres x 4 líneas.
Interfaz	Analógica
Tamaño	9,8 cm x 6 cm x 1,2 cm

### **Anexo 9. Características técnicas de la pantalla Touch**

<b>Formato</b>	1024 x 600
<b>Tipo</b>	TFT
<b>Tipo de pantalla</b>	Resistiva
<b>Área Activa</b>	154 x 86 mm
<b>Luminosidad</b>	250 nits

### **Anexo 10. Características de la fuente de voltaje**

<b>Voltaje entrado</b>	110 / 220 V
<b>Voltaje de salida</b>	5-24 V
<b>Corriente</b>	15 A

### **Anexo 11. Manual de usuario**

La máquina bobinado automático está conectada a una fuente de alimentación de 120 VAC, esta fuente tiene un sistema de conversión AC/DC donde se alimentará también los controladores y actuadores a 5 y 12v respectivamente.

El panel de control de la maquina representado en la figura está compuesto de:

1. Pulsador de emergencia
2. Pulsador de marcha

3. Teclado matricial
4. LCD 20X4

Panel de control



Panel de control maquina bobinadora automática. Elaborado por: Juan Achig, Angel Pazmiño

### **Pulsador de emergencia**

Este elemento de control sirve para interrumpir el proceso de manera inmediata cortando la alimentación hacia los actuadores, este pulsador se queda enclavado durante el tiempo que se requiera para corregir el inconveniente en la máquina, una vez corregido se puede volver a pulsar y continuar con el proceso.

### **Pulsador de marcha**

Este pulsador permite al igual que con el teclado iniciar el proceso de bobinado una vez determinados los parámetros de construcción de la bobina.

### **Teclado matricial**

Este es un dispositivo electrónico que permite ingresar tanto datos numéricos como ordenes de acción de programación.

Menú de arranque para ingreso de datos



Menú principal de maquina bobinadora automática. Elaborado por: Juan Achig, Angel Pazmiño

### **Tecla D**

Esta tecla permite dar enter para ingresar los datos o pasar al siguiente menú

### **Tecla A**

Esta tecla borra los datos ingresados en el caso de que sean erróneos y poder ingresar los correctos.

Con el menú de la figura el cual se muestra una vez que la maquina ha sido encendida se tiene las opciones

1. FABRICAR BOBINA
2. REALIZAR PRUEBAS

Cuando se toma la opción 1 fabricar bobina nos pide que ingresemos el calibre del cable con el cual vamos a trabajar para la fabricación de la bobina.



Ingreso de calibre de cable desde el 18 AWG hasta el 23AWG. Elaborado por: Juan Achig, Angel Pazmiño

Se ingresa el calibre del cable mediante el teclado matricial y una vez que se ingresa se da enter es decir la letra D, lo siguiente es ingresar el número de vueltas con el cual queremos que cuente la bobina es decir el número de paso que va a dar el motor.

Ingreso de numero de vueltas para la fabricación de bobina



Ingreso de numero de vueltas. Elaborado por: Juan Achig, Angel Pazmiño

Al ingresar el número de vueltas y dar al botón enter nos indica que le demos al botón start lo cual lo podemos hacer mediante el teclado matricial con la letra D o con el botón verde del panel principal.

Una vez que el sistema comienza con el conteo de vueltas en el LCD se mostrara un mensaje de fin de proceso

Para la inicialización del software de la Raspberry vamos a tener la siguiente interfaz

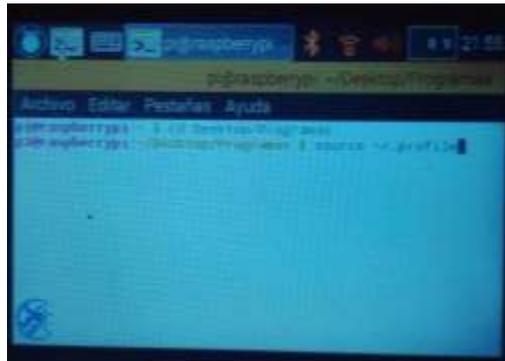
Interfaz gráfica FTP



Interfaz gráfica para funcionamiento de visión artificial mostrado en FTP. Elaborado por: Juan Achig, Angel Pazmiño

En esta pantalla principal representada en la FTP conectada a la raspberry PI ingresamos la función LXTERMINAL el cual nos permite seleccionar el archivo el cual contiene la programación del logaritmo de visión artificial.

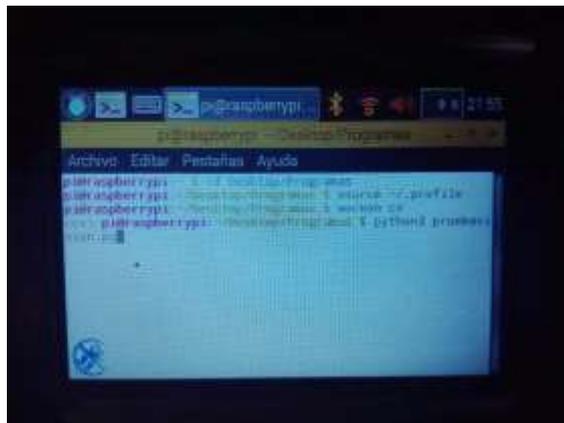
### Ingreso de comando para llamada de archivo de visión artificial



Llamada del archivo para el arranque de visión artificial. Elaborado por: Juan Achig, Angel Pazmiño

Colocamos el comando `cd` seguida de la dirección donde se encuentra el archivo. Una vez detectado el archivo aparece el entorno de visión artificial. Verificamos que estamos en el entorno y damos el comando `work CV`, posterior a esto digitamos `python 3` seguido del nombre del programa o algoritmo.

### Entorno de visión artificial para comunicación serial



Entorno para comunicación serial y visión artificial. Elaborado por: Juan Achig, Angel Pazmiño

Una vez culminado este proceso se tiene 2 opciones para el funcionamiento de la maquina donde se puede ingresar los parámetros de la bobina (número de vueltas, calibre de cable) por medio de la raspberry o también mediante el teclado matricial.

