

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA

CARRERA DE INGENIERÍA ELECTRÓNICA

*Trabajo de titulación previo
a la obtención del título de
Ingeniero Electrónico*

PROYECTO TÉCNICO CON ENFOQUE INVESTIGATIVO:

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO
PARA OPTIMIZAR LA FASE DE IMPREGNACIÓN DE TINTA EN EL
PROCESO DE SERIGRAFÍA APLICADO AL ESTAMPADO DE
TAFILETES PARA SOMBREROS DE PAJA TOQUILLA EN LA
MICROEMPRESA JO & MI CONFECCIONES”**

AUTORES:

ANGEL FABIÁN ORTIZ GONZÁLEZ

ROMMEL XAVIER VINTIMILLA AVILA

TUTOR:

ING. JORGE GIOVANNI SAGBAY SACAQUIRIN, MsC.

CUENCA - ECUADOR

2021

CESIÓN DE DERECHOS DE AUTOR

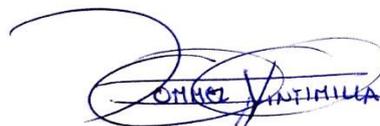
Nosotros, Angel Fabián Ortiz González con documento de identificación N° 0103945655 y Rommel Xavier Vintimilla Avila con documento de identificación N° 0104870506, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA OPTIMIZAR LA FASE DE IMPREGNACIÓN DE TINTA EN EL PROCESO DE SERIGRAFÍA APLICADO AL ESTAMPADO DE TAFILETES PARA SOMBREROS DE PAJA TOQUILLA EN LA MICROEMPRESA JO & MI CONFECCIONES”**, mismo que ha sido desarrollado para optar por el título de *Ingeniero Electrónico*, 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 la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, marzo del 2021



Angel Fabián Ortiz González
C.I. 0103945655



Rommel Xavier Vintimilla Avila
C.I. 0104870506

CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA OPTIMIZAR LA FASE DE IMPREGNACIÓN DE TINTA EN EL PROCESO DE SERIGRAFÍA APLICADO AL ESTAMPADO DE TAFILETES PARA SOMBREROS DE PAJA TOQUILLA EN LA MICROEMPRESA JO & MI CONFECCIONES”**, realizado por Angel Fabián Ortiz González y Rommel Xavier Vintimilla Avila, obteniendo el *Proyecto Técnico con enfoque investigativo*, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, marzo del 2021



Ing. Giovanni Sagbay, MsC.
C.I. 0702639063

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Angel Fabián Ortiz González con documento de identificación N° 0103945655 y Rommel Xavier Vintimilla Avila con documento de identificación N° 0104870506, autores del trabajo de titulación: **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMATIZADO PARA OPTIMIZAR LA FASE DE IMPREGNACIÓN DE TINTA EN EL PROCESO DE SERIGRAFÍA APLICADO AL ESTAMPADO DE TAFILETES PARA SOMBREROS DE PAJA TOQUILLA EN LA MICROEMPRESA JO & MI CONFECCIONES”**, certificamos que el total contenido del *Proyecto Técnico con enfoque investigativo*, es de nuestra exclusiva responsabilidad y autoría.

Cuenca, marzo del 2021



Angel Fabián Ortiz González
C.I. 0103945655



Rommel Xavier Vintimilla Ávila
C.I. 0104870506

AGRADECIMIENTOS

Quiero agradecer a mis padres Angel Ortiz y Marina González quienes siempre me mostraron su apoyo incondicional, por su esfuerzo, amor, consejos que me ayudaron a lo largo de este tiempo para que pueda cumplir con esta meta.

Al Ing. Giovanni Sagbay quien nos brindó su apoyo y nos supo guiar de la mejor manera en este proyecto.

Angel Fabián Ortiz González

A Dios en primer lugar, por la vida y porque de ÉL procede toda sabiduría y entendimiento. A mis padres Romel Vintimilla y Lucia Avila por su amor, trabajo y sacrificio, por sus consejos y su apoyo en todo momento al igual que mis hermanos Byron y Paul. A mis abuelitos Luis Vintimilla y Eloisa Benavides por su amor, por su apoyo incondicional, por sus consejos, por confiar y creer en mí.

Rommel Xavier Vintimilla Avila

DEDICATORIAS

El presente proyecto de titulación se lo quiero dedicar a mis padres Angel Ortiz y Marina González, quienes siempre demostraron su apoyo y confianza en todos los aspectos para terminar con mi carrera universitaria.

A mis hermanos Jonathan Ortiz, Oscar Ortiz, Alejandro Rodas y a mi tía Pilar González quienes siempre me mostraron su apoyo incondicional.

Angel Fabián Ortiz González

A toda mi familia, que con sus oraciones y palabras de aliento me acompañaron en todo este tiempo. A mi amiga de muchos años Adriana Zapata, por creer siempre en mí, por extenderme su mano en momentos difíciles, por el cariño brindado durante todo este tiempo, por sus consejos y palabras de aliento. De igual manera a mis padres espirituales Fermin y Yamile Orellana, por sus oraciones y su guía espiritual para conocer el amor de Cristo que es la base en la que se sustenta todo en la vida.

Finalmente, a todos mis docentes, futuros colegas y amigos, dentro y fuera de la institución.

Rommel Xavier Vintimilla Avila

ÍNDICE GENERAL

AGRADECIMIENTOS.....	I
DEDICATORIAS.....	II
ÍNDICE GENERAL.....	III
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS.....	IX
RESUMEN.....	X
INTRODUCCIÓN.....	XI
ANTECEDENTES DEL PROBLEMA DE ESTUDIO.....	XII
JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES).....	XIV
OBJETIVOS.....	XV
OBJETIVO GENERAL.....	XV
OBJETIVOS ESPECÍFICOS.....	XV
CAPÍTULO 1: DESCRIPCIÓN DEL MÉTODO DE SERIGRAFÍA.....	1
1.1 ESTAMPADO.....	1
1.2 HISTORIA DE LA SERIGRAFÍA.....	1
1.3 SERIGRAFÍA.....	1
1.4 APLICACIONES DE LA SERIGRAFÍA.....	1
1.5 PROCEDIMIENTO DE LA SERIGRAFÍA MANUAL.....	2
1.5.1 TÉCNICA USADA PARA EL ESTAMPADO MANUAL.....	2
1.5.2 PRINCIPALES ELEMENTOS QUE INTERVIENEN EN EL PROCESO DE SERIGRAFÍA MANUAL.....	2
CAPÍTULO 2: ANÁLISIS DE LAS TECNOLOGÍAS Y MÉTODOS PARA LA AUTOMATIZACIÓN.....	6
2.1 TECNOLOGÍAS PARA LA AUTOMATIZACIÓN.....	6
2.1.1 TECNOLOGÍA CABLEADA.....	6
2.1.2 TECNOLOGÍA PROGRAMABLE.....	6
2.2 MÉTODOS PARA LA AUTOMATIZACIÓN.....	7
2.2.1 AUTOMATIZACIÓN FIJA.....	7
2.2.2 AUTOMATIZACIÓN PROGRAMABLE.....	7
2.2.3 AUTOMATIZACIÓN FLEXIBLE.....	8
2.3 FASES PARA AUTOMATIZAR UN PROCESO.....	9

2.3.1	ESPECIFICACIÓN	9
2.3.2	DISEÑO	10
CAPÍTULO 3: COMPARACIÓN TÉCNICO Y ECONÓMICO DE DISPOSITIVOS		11
3.1 DESCRIPCIÓN DE LOS DISPOSITIVOS QUE SE PODRÍAN UTILIZAR PARA LA AUTOMATIZACIÓN DE LA FASE DE IMPREGNACIÓN DE TINTA		11
3.1.1	MICROCONTROLADOR	11
3.1.2	CONTROLADOR LÓGICO PROGRAMABLE (PLC)	12
3.1.3	INTERFAZ HUMANO-MÁQUINA (HMI)	14
3.1.4	DRIVERS PARA MOTOR A PASOS	16
3.2 DESCRIPCIÓN DE ELEMENTOS PARA EL PROCESO DE AUTOMATIZACIÓN		18
3.2.1	MOTOR ELÉCTRICO DC.....	18
3.2.2	RELÉS O CONTACTORES.....	18
3.2.3	CILINDRO DE DOBLE EFECTO	19
3.2.4	ELECTROVÁLVULA	19
3.3 COMPARACIÓN DE CARACTERÍSTICAS ENTRE LOS DISPOSITIVOS PARA LA FASE DE IMPREGNACIÓN DE TINTA.....		20
3.3.1	COMPARACIÓN ENTRE MICROCONTROLADOR Y PLC.....	20
3.3.2	COMPARACIÓN ENTRE PANTALLAS HMI	21
3.3.3	COMPARACIÓN ENTRE DRIVERS	21
CAPÍTULO 4: DISEÑO DE HARDWARE Y SOFTWARE		23
4.1 DISEÑO DE LA ESTRUCTURA MECÁNICA		24
4.1.1	TENSADO DE CINTA	24
4.1.2	BRAZO PARA ESTAMPAR	25
4.1.3	TRASLADO DE CINTA	25
4.1.4	ESTRUCTURA MECÁNICA COMPLETA.....	26
4.2 DISEÑO DEL SISTEMA ELECTRÓNICO.....		27
4.2.1	SISTEMA DE CONTROL Y VENTILACIÓN	28
4.2.2	ALIMENTACIÓN	31
4.2.3	SENSADO Y ALARMA	31
4.2.4	SISTEMA ELECTRÓNICO COMPLETO	33
4.3 DISEÑO INTERFAZ GRÁFICA		36
4.3.1	INICIALIZACIÓN	36
4.3.2	MENÚ PRINCIPAL	37
4.3.3	AVISOS DE FALLOS	38
CAPÍTULO 5: PRUEBAS Y CORRECCIÓN DE ERRORES		41
5.1 CORRECCIÓN DE ERRORES.....		41
5.1.1	ERROR DE ESTAMPADO.....	41
5.1.2	ERROR MOTOR TRASLADO DE CINTA	42
5.1.3	ERROR EN LOS CILINDROS	43

5.2	PRUEBAS DE ESTAMPADO DE CADA TALLA.....	43
5.2.1	PRUEBAS DE LA TALLA 54.....	43
5.2.2	PRUEBAS DE LA TALLA 58.....	44
5.2.3	PRUEBAS DE LA TALLA 62.....	45
5.2.4	PRUEBAS DE LA TALLA 66.....	46
CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES		48
6.1	CONCLUSIONES.....	48
6.2	RECOMENDACIONES	50
ANEXOS		53
ANEXO A: DIMENSIONES DE LA ESTRUCTURA MECÁNICA		53
ANEXO B: ARMADO DE LA TARJETA ELECTRÓNICA		56
ANEXO C: PRUEBAS DE LA ESTRUCTURA MECÁNICA		58
ANEXO D: TABLA CÓDIGO DE ERRORES		62
ANEXO E: PROGRAMACIÓN DE LOS MICROCONTROLADORES.....		63

ÍNDICE DE FIGURAS

Figura 1. 1 Marco de madera en el cual se sujeta la malla que contiene el diseño que se desea transferir a la superficie	3
Figura 1. 2 Racle	4
Figura 3. 1 Arduino Mega 2560	12
Figura 3. 2 Pantalla Nextion NX4827T043-011	14
Figura 3. 3 Pantalla HMI Kinco 4'3.	15
Figura 3. 4 Driver HY-FIV28-5A	16
Figura 3. 5 Driver TB6560.....	17
Figura 3. 6 Relé 5VDC	18
Figura 3. 7 a) Cilindro de doble efecto con el vástago contraído. b) Cilindro doble efecto con el vástago completamente fuera	19
Figura 3. 8 Electroválvula 3 posiciones 2 salidas	19
Figura 4. 1 Diagrama de Bloques del funcionamiento del sistema.....	23
Figura 4. 2 Dispositivo para tensado de cinta	24
Figura 4. 3 Brazo para estampar	25
Figura 4. 4 Estructura para el traslado de cinta	26
Figura 4. 5 Estructura completa	27
Figura 4. 6 Arduino Mega de Altium Designer	28
Figura 4. 7 Sistema de control para las electroválvulas	29
Figura 4. 8 Conectores para conexión de pantalla y drivers	29
Figura 4. 9 Sistema de ventilación	30
Figura 4. 10 Salidas auxiliares y reset.....	30
Figura 4. 11 Entrada de las fuentes de 12-24VDC.....	31
Figura 4. 12 Conectores para los sensores y buzzer para las alarmas.....	32
Figura 4. 13 Leds indicadores	32
Figura 4. 14 PIC16F84A	33
Figura 4. 15 Diseño completo de la tarjeta electrónica en Altium Designer	33
Figura 4. 16 Primera capa de la PCB con su plano de tierra.....	34
Figura 4. 17 Segunda capa de la PCB y su plano de tierra	35

Figura 4. 18 PCB en 3 D de la tarjeta electrónica.....	35
Figura 4. 19 Pantalla de inicialización	36
Figura 4. 20 Segunda pantalla.....	37
Figura 4. 21 Menú principal.....	38
Figura 4. 22 Aviso de cinta agotada.....	38
Figura 4. 23 Aviso de revisar brazo	39
Figura 4. 24 Aviso falla motor traslado de cinta.....	39
Figura 4. 25 Aviso revisar cilindros.....	40
Figura 5. 1 Comparación de tiempo entre estampados manual y automatizado	44
Figura 5. 2 Estampado de la Talla 54.....	44
Figura 5. 3 Comparación de tiempo entre estampados manual y automatizado	45
Figura 5. 4 Estampado de la Talla 58.....	45
Figura 5. 5 Comparación de tiempo entre estampados manual y automatizado	46
Figura 5. 6 Estampado de la Talla 62.....	46
Figura 5. 7 Comparación de tiempo entre estampados manual y automatizado	47
Figura 5. 8 Estampado de la Talla 66.....	47
Figura A. 1 Medidas de la estructura de tensado de cinta.....	53
Figura A. 2 Medidas del brazo robótico.....	54
Figura A. 3 Medidas sección traslado de cinta	54
Figura A. 4 Medidas sección traslado de cinta	55
Figura B. 1 Soldado de los componentes en la tarjeta electrónica.....	56
Figura B. 2 Armado de los componentes en la tarjeta electrónica.....	57
Figura B. 3 Montaje de la tarjeta electrónica y sus dispositivos.....	57
Figura C. 1 Estructura mecánica	58
Figura C. 2 Manejo de la estructura mecánica.....	59
Figura C. 3 Manejo de la estructura mecánica.....	59
Figura C. 4 Marco y diseño para el estampado	60
Figura C. 5 Estampado en tafilete para 54 cm	60

Figura C. 6 Estampado en tafilete para 58 cm	61
Figura C. 7 Estampado en tafilete para 62 cm	61
Figura C. 8 Estampado en tafilete para 66 cm	61

ÍNDICE DE TABLAS

Tabla 1. 1 Dimensiones de marcos comunes	3
Tabla 3. 1 Características del Arduino Mega 2560	12
Tabla 3. 2 Características del PLC Logo V8.....	13
Tabla 3. 3 Características Pantalla Nextion modelo	15
Tabla 3. 4 Características de pantalla Kinco MT4230T	16
Tabla 3. 5 Características driver HY-DIV268N-5A	17
Tabla 3. 6 Características driver TB6560	17
Tabla 3. 7 Comparación entre Microcontrolador y PLC.....	20
Tabla 3. 8 Comparación entre pantallas HMI	21
Tabla 3. 9 Comparación entre drivers para motores nema.....	22
Tabla D. 1 Código de Errores	62

RESUMEN

El método de serigrafía es un proceso de impresión, nos permite en realizar la impregnación de un boceto previamente diseñado sobre una superficie, aplicando presión mediante una racle a la malla del marco la cual estará recubierta de tinta y sobre esta se encontrará el diseño para realizar el estampado.

La automatización es una tecnología que se utiliza para mejorar u optimizar un proceso, a través del desarrollo de un sistema que permita controlar y monitorear diferentes variables que intervienen en el proceso, garantizando la calidad del producto y aumentando la producción, reduciendo tiempos y costos, puede ser aplicado tanto en macroempresas como en microempresas.

Este proyecto se enfocó en realizar de manera automatizada el proceso de impregnación de tinta en tafiletes utilizando el método de serigrafía, en nuestro medio el uso de tecnología en procesos de producción tradicionales no es común y sobre todo poco asequible económicamente para microempresas, es por eso que surgió la iniciativa de aplicar tecnología de automatización.

Realizamos un estudio investigativo sobre los diferentes tipos de tecnologías y métodos para la automatización de un proceso, se tomó en cuenta las necesidades de la microempresa para elegir los procesos adecuados para la automatización de la fase de impregnación de tinta.

Para la implementación se optó por elementos de bajo costo en el mercado, los cuales están dentro de las capacidades económicas de una microempresa, ya que estas requieren calidad pero su producción es media en cuanto a la cantidad de estampados

Al realizar la implementación del sistema automatizado para la fase de impregnación de tinta se observaron resultados, tanto en la mejora del tiempo de producción en comparación con el estampado de la manera tradicional, además aumentó la producción general de la microempresa ya que el trabajo antes destinado a este proceso se lo enfoco a otras áreas de producción.

INTRODUCCIÓN

Empezaremos por investigar y conocer todos los aspectos referentes con el proceso de serigrafía que se utiliza en la empresa “Jo & Mi confección” para el estampado de tafiletes, estaremos al corriente de todas las fases que involucran el proceso, sin embargo, ya que el enfoque general del proyecto está orientado a la fase de impregnación de tinta pues será ahí en donde nos enfocaremos y buscaremos solucionar algunas de las limitaciones del proceso manual aplicando los conocimientos que poseemos para optimizar esta fase.

Luego de esto se procederá al análisis para determinar la mejor estrategia para llevar a cabo la automatización de la fase de impregnación de tinta lo cual dependerá de un análisis técnico de las variables que están involucradas en el proceso las principales son: tiempo, velocidad, fuerza y presión.

Posteriormente se realizará un estudio y comparación para determinar los elementos y dispositivos adecuados para la implementación del sistema automatizado. Dicho análisis se lo realizara bajo dos parámetros: las características de funcionamiento que deben ser adecuadas para el proceso y el costo. Ya que al ser destinado para una micro empresa este último parámetro debe ser lo más bajo posible sin que esto afecte la calidad del producto final.

Una vez identificados y adquiridos los elementos y dispositivos se procederá a desarrollar un firmware para el sistema de control, el cual será el encargado de gestionar los tiempos y las secuencias de accionamiento de cada elemento que integra el sistema. De igual manera dentro de esta fase se diseñará una interfaz gráfica de usuario que interactuará en conjunto con el sistema de control para facilitar el manejo y la lectura de las variables principales del sistema.

El siguiente paso será incorporar todos los elementos de desplazamiento, control y sensado en una estructura mecánica que nos permita realizar el proceso de impregnación de tinta, será en esta fase en la que se realice la identificación, ajuste o corrección de errores del funcionamiento en conjunto del sistema.

Finalmente se procederá a realizar un análisis y comparación de los tiempos finales del proceso antes y después de la implementación del sistema.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

La Real Academia de las Ciencias Físicas y Exactas define: “la automática como el conjunto de métodos y procedimientos para la substitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales” [1].

Citando de otra manera, la automatización utiliza diferentes tipos de tecnologías para realizar el control y monitoreo de diferentes procesos industriales, equipos, dispositivos o máquinas, las cuales cumplen funciones repetitivas con el objetivo de lograr que operen automáticamente y de esta manera reducir al máximo la intervención humana. Actualmente, la automatización se ha convertido en una herramienta indispensable para que las empresas puedan competir en el mercado, ya que las empresas quieren afianzarse y de esta manera mejorar sus ganancias [2].

En el país la automatización de procesos en el área de la industria textil ha tenido importantes adelantos debido a la importación de máquinas con tecnología destinada al aumento de la producción a través de la automatización [3], sin embargo ciertas áreas no cuentan con los recursos financieros para la adquisición de dicha tecnología, debido a esto surge la necesidad de muchas medianas y pequeñas empresas en adquirir sistemas de bajo costo en comparación con las grandes inversiones de otras empresas. Dicha necesidad se enfoca en cubrir la automatización de micro procesos y de esta forma aumentar la producción.

La serigrafía es un sistema de impresión que surge por el interés del hombre en marcar una diferencia y dejar un legado de sus diferentes actividades [4]. El origen de este método de impresión se remonta a la antigua China; sin embargo, esta apareció por primera vez en Europa y Estados Unidos a principios del siglo XXI. En la década de los 60 la serigrafía toma un gran impulso en los países de latino américa al ser un método versátil que permite imprimir en muchos materiales con lo cual llego a ser incluido en muchos sectores productivos [5].

Hoy en día dentro de la industria textil se continúa utilizando este método de impresión para el estampado en distintos materiales, sin embargo, en nuestro país todavía el uso de la automatización no ha despuntado en todos los sectores, especialmente en las PyMES, esto lo podemos decir debido a que el proceso de serigrafía todavía se lo está realizando de forma artesanal en muchos talleres y microempresas, lo que implica la pérdida de recursos (tiempo y mano de obra) los cuales se redujeron al desarrollar y fabricar un sistema automatizado, el cual nos permitió evidenciar mejoras en este proceso.

JUSTIFICACIÓN (IMPORTANCIA Y ALCANCES)

El Ecuador es un país con un sector productivo textil muy amplio lo cual implica que la serigrafía es una necesidad en la mayoría de empresas y PyMES que se desempeñan en dicho sector [6]. En los últimos años se ha podido notar un incremento en la importación de maquinaria que satisfaga los requerimientos de las empresas dedicadas al sector textil teniendo una incidencia directa sobre el precio del producto final ya que a nivel nacional hay pocas empresas que producen maquinaria con nueva tecnología destinada para este sector.

A pesar de los avances tecnológicos en la automatización de procesos, los PyMES y muchos talleres independientes a nivel nacional todavía utilizan serigrafía con métodos manuales generando un gran gasto en mano de obra y ocupando gran cantidad de tiempo; generalmente se necesita por lo menos de dos personas para poder cumplir con este trabajo de impresión textil.

Este panorama podría cambiar con el desarrollo de sistemas de automatización de bajo costo destinados a mejorar este y muchos otros procesos dentro de la industria textil, con lo cual se reduciría el tiempo y mano de obra apuntando a un más que probable aumento de producción lo cual significaría una posible disminución del costo final del producto debido a la inclusión de nuevas tecnologías.

Es por este motivo que la micro empresa “Jo & Mi CONFECCIONES” ubicada en la ciudad de Cuenca específicamente en el sector de la ciudadela El Rosal ha tenido el interés de adquirir un sistema que optimice su actual proceso y de esta manera lograr aumentar su productividad mediante la disminución de tiempo y mano de obra en el proceso. La principal actividad que desarrolla la micro empresa es el estampado de tafletes mediante el proceso de serigrafía. Por petición del propietario se nos dio a conocer de la necesidad de optimizar una etapa determinada del proceso de estampado mediante la inclusión de tecnología. Dicha etapa es denominada como la fase de impregnación de tinta, la misma que será detallada en la sección correspondiente al marco teórico referencial.

OBJETIVOS

OBJETIVO GENERAL

- Diseñar e implementar un sistema automatizado para optimizar la fase de impregnación de tinta en el proceso de serigrafía enfocado al estampado en tafiletes para sombreros de paja toquilla.

OBJETIVOS ESPECÍFICOS

- Estudiar y conocer el proceso de estampado con el método de serigrafía.
- Analizar la tecnología, los métodos y requerimientos necesarios para llevar a cabo la automatización del proceso.
- Realizar un estudio de comparación técnico y económico para determinar los elementos y dispositivos adecuados para el desarrollo del proyecto.
- Diseñar el hardware y software para la fase de impregnación de tinta.
- Realizar pruebas, afinamiento y corrección de errores del sistema automatizado.

CAPÍTULO 1: DESCRIPCIÓN DEL MÉTODO DE SERIGRAFÍA

1.1 ESTAMPADO

El estampado es un proceso de impresión de diseños la cual utiliza una plancha grabada y entintada sobre una hoja, tela, cuero, cerámica, etc. los estampados poseen diseños de cualquier tipo los cuales pueden ser coloridos o a blanco y negro dependiendo del diseño que se requiere en el material [7].

1.2 HISTORIA DE LA SERIGRAFÍA

El proceso de serigrafía es una técnica de impresión que fue utilizada por antiguas culturas como son los romanos, griegos, chinos, egipcios los cuales han dejado restos arqueológicos que muestran el uso de plantillas para realizar decoraciones de cerámicas, tejidos y paredes. Las plantillas utilizadas para estas decoraciones fueron implementadas en papeles, hojas de plantas e incluso las partes del cuerpo como las manos o los pies [7].

1.3 SERIGRAFÍA

Serigrafía proviene del latín sericum que significa seda y del griego graphe que significa escribir, en si el proceso de serigrafía es un método de impresión, el cual se utiliza para impregnar tinta de un boceto el cual se encuentra en una seda tensada colocada en un marco, esta impregnación de tinta se la puede realizar en cualquier tipo de material que se requiera [7].

1.4 APLICACIONES DE LA SERIGRAFÍA

Las aplicaciones de la serigrafía están relacionadas con el tipo de tinta que se requiere para la impresión y depende del material que se necesite estampar, a continuación se muestra algunas de sus aplicaciones:

- Reproducción de obras de arte, pinturas y diseños.
- Estampado en textiles como camisetas, bolsos y diferentes telas.
- Impresión en materiales como plásticos, maderas, aluminio, circuitos impresos.
- Decoración en cristales y cerámicas.
- Tableros de control, se utiliza para señalización en la industria. [8]

1.5 PROCEDIMIENTO DE LA SERIGRAFÍA MANUAL

El proceso de serigrafía manual está constituido por varios pasos secuenciales que interactuaran con los elementos que conforman la impresión textil junto con las habilidades del artesano.

Lo esencial en este proceso es tener definido el diseño del estampado, así como la pintura que se va a utilizar para la impresión, y en conjunto con la técnica se podrá obtener un estampado de buena calidad.

1.5.1 TÉCNICA USADA PARA EL ESTAMPADO MANUAL

La tinta tiene una cierta densidad por lo tanto a través de una presión manual sobre la racle se arrastra sobre la malla para que la pintura atraviese el diseño grabado y que de este modo quede impreso el diseño sobre el textil.

1.5.2 PRINCIPALES ELEMENTOS QUE INTERVIENEN EN EL PROCESO DE SERIGRAFÍA MANUAL

En esta sección se hablará sobre los principales elementos que se utiliza para el proceso de serigrafía manual.

1.5.2.1 MESA DE IMPRESIÓN

La mesa de impresión es la superficie en la cual se coloca el objeto que requiere ser estampado, las dimensiones tienen que ser de acuerdo con el marco que se va a utilizar para la impresión, el máximo tamaño no tiene que ser mayor a 30 cm por cada lado con referencia a los bordes del marco a utilizar [8].

1.5.2.2 EL MARCO

El objetivo de este elemento es fijar con firmeza la malla o seda, para dejar constituido el tamiz. Los marcos pueden ser fabricados tanto de madera o metal, pero los marcos de madera son de mayor preferencia en el sector artesanal, ya que estos marcos de madera se obtienen con facilidad y son sencillos para montar la seda [8].

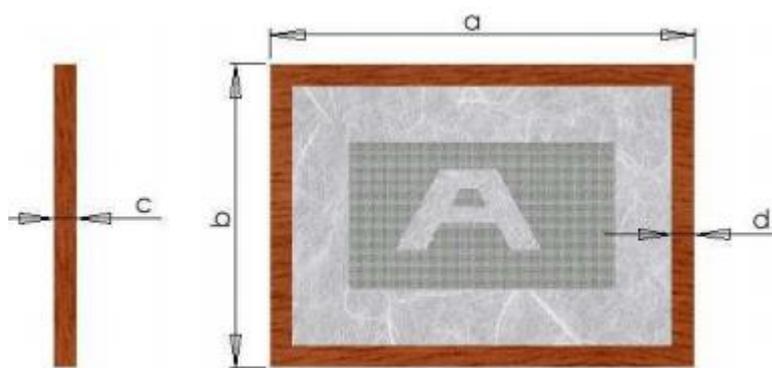


Figura 1. 1 Marco de madera en el cual se sujeta la malla que contiene el diseño que se desea transferir a la superficie

Fuente: [7]

Las dimensiones de marcos que se muestra en la siguiente tabla son las dimensiones más comunes o utilizadas en el sector artesanal.

Tabla 1. 1 Dimensiones de marcos comunes

Fuente: [8]

Dimensiones(cm)	A	B	C	D
Marco 1	58	50	4	4.5
Marco 2	58	50	4	4
Marco 3	50	40	4	4
Marco 4	40	25	3	2.5
Marco 5	30	25	2.5	1.5

1.5.2.3 MALLA O TAMIZ

El tamiz está formado por un tejido trenzado, el material utilizado para su fabricación es el nylon el cual se encuentra sujeto en un marco de madera para mantener la tensión [8].

La principal función de la malla es permitir el paso de la tinta por las zonas libres (área que contiene el diseño).

1.5.2.4 PINTURA PARA IMPRIMIR

La pintura que se utiliza para la impresión se le conoce con el nombre de plastisol la cual está formada por una mezcla de resina PVC el cual es un plastificante líquido, así como de otros componentes que dan color a la pintura [8].

1.5.2.5 RACLE

La racle se utiliza para desplazar la tinta que se encuentra en el marco a través de toda la superficie de la malla logrando de esta manera una mayor cobertura para el estampado.

La racle utiliza un caucho el cual puede ser de diferentes formas para realizar el estampado de cualquier diseño, las dimensiones del caucho van de acuerdo con las medidas del diseño que se desea imprimir.

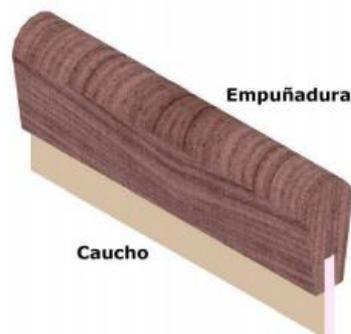


Figura 1. 2 Racle

Fuente: [8].

1.5.2.6 ÁREA DE IMPRESIÓN

En cuanto al área de impresión depende del diseño que se necesite imprimir pero se recomienda que deba estar comprendida entre 10cm a 15cm de largo y en el ancho debe ser de acuerdo a la cinta que se utilice.

CAPÍTULO 2: ANÁLISIS DE LAS TECNOLOGÍAS Y MÉTODOS PARA LA AUTOMATIZACIÓN

En este capítulo conoceremos sobre las tecnologías y métodos que se pueden utilizar para realizar la automatización de la fase de impregnación de tinta.

2.1 TECNOLOGÍAS PARA LA AUTOMATIZACIÓN

Las tecnologías utilizadas en la automatización se dividen en dos grupos: las cableadas y las programables [4].

2.1.1 TECNOLOGÍA CABLEADA

Los circuitos de esta tecnología están representados mediante esquemas para observar de manera gráfica el correcto funcionamiento del sistema, este tipo de tecnología se realizan de manera física, son económicas en montaje y mantenimiento [4].

Estas se dividen en:

- Eléctrica y electrónica
- Neumática
- Hidráulica
- Mecánica

2.1.2 TECNOLOGÍA PROGRAMABLE

La tecnología programable es un equipo electrónico programable está conformada por un conjunto de elementos que permite el control de un sistema [4]. Esta tecnología se desarrolló a partir de la aparición del microprocesador, y de los sistemas programables basados en este.

Esta tecnología se divide en:

- Ordenadores
- Microcontroladores
- PLC

2.2 MÉTODOS PARA LA AUTOMATIZACIÓN

Los métodos de la automatización industrial se dividen en tres clases:

- Automatización Fija
- Automatización Programable
- Automatización flexible.

2.2.1 AUTOMATIZACIÓN FIJA

La automatización fija es empleada en las industrias cuando se requiera un volumen de producción muy elevado por lo cual se justificaría la elevada inversión que se necesita para el diseño del sistema utilizado para la fabricación del producto [9].

La principal desventaja de la automatización fija es su vida útil ya que esta dependería de la cantidad de tiempo que el producto se encuentre en el en el mercado [9].

Sus principales características son:

- Se necesita una grande inversión inicial para el equipo de ingeniería.
- Altos índices de producción.
- No puede adaptarse a los cambios que se pueda generar en el producto [9].

2.2.2 AUTOMATIZACIÓN PROGRAMABLE

La automatización programable es utilizada en las empresas cuando la producción es baja y existe una gran diversidad de productos. En las industrias se realiza el diseño del sistema para la producción de acuerdo a las variaciones o configuraciones que pueda presentarse en el producto los cuales pueden surgir con el tiempo, por ejemplo requerimientos del cliente, por el mercado, entre otros. En esta automatización la configuración del producto se la puede realizar mediante softwares para PLC, microcontroladores, etc. [9]

Las características de la automatización programable son:

- Inversión para el equipo de ingeniería.

- Producción relativamente baja.
- Adaptabilidad del sistema para adaptarse a la configuración que pueda presentar el producto.
- Adecuado para producción en montones.

2.2.3 AUTOMATIZACIÓN FLEXIBLE

Este método se emplea en las industrias cuando se requiere de una producción media. Los sistemas flexibles se componen tanto de la automatización fija como de la programable [9].

Los sistemas flexibles en las industrias están formados por varias estaciones de trabajo interconectadas entre sí por sistemas de almacenamiento y manipulación de materiales, que son controlados por una computadora y un PLC [9].

Las características de la automatización flexible son:

- Se necesita una grande inversión para equipo de ingeniería.
- Producción continua de varios productos.
- Volumen de producción media.
- Adaptabilidad del sistema para adaptarse a la configuración que pueda presentar el producto.
- Capacidad para cambiar partes del programa sin perder tiempo de producción

Una vez analizado y conocido las tecnologías y los métodos de la automatización, se podría asegurar que las tecnologías que se utilizaran son la cableada y la programable ya que estos procesos se utilizan para controlar el sistema y son de bajo costo de montaje, en cuanto al método que se utilizaría sería la automatización programable debido a que la producción es baja y se podría realizar modificaciones en un futuro dependiendo de las necesidades que se presenten.

2.3 FASES PARA AUTOMATIZAR UN PROCESO

Se realizará una explicación de las fases que se necesita para automatizar un proceso en base al proyecto que se está realizando.

2.3.1 ESPECIFICACIÓN

2.3.1.1 CONOCIMIENTO DE LA PLANTA

En la empresa se pudo conocer que el proceso de estampado de tafiletes para sombreros de paja toquilla se la realiza de manera manual con un método tradicional, es por este motivo que surge la necesidad de automatizar ciertas áreas de este proceso en este caso la fase de impregnación de tinta será en donde se enfocará nuestro proyecto.

2.3.1.2 DEFINICIÓN DE LAS NECESIDADES

- Automatizar la fase de impregnación de tinta.
- Realizar automáticamente el proceso de arrastre y recolección del pigmento
- Desplazar el material en el que se va a estampar de manera automática
- Diseñar una secuencia que permita estampar en la cinta un diseño tomando en cuenta la técnica de serigrafía.

2.3.1.3 DETERMINACIÓN DE LAS VARIABLES A CONTROLAR

Las variables que intervendrán en el proyecto son velocidad y posición.

Velocidad: Esta variable es muy importante debido a que la calidad del estampado dependerá de la velocidad con la que la racle se desplaza sobre el diseño.

Posición: El estampado de tafiletes se lo realiza dejando un intervalo de separación, por lo que será necesario conocer la posición que deberá recorrer el material.

2.3.2 DISEÑO

2.3.2.1 ELECCIÓN DE LOS TIPOS DE SENSORES Y ACTUADORES

Los sensores y actuadores que se utilizaran son los siguientes:

Sensores:

- **Sensor de efecto hall:** se utiliza para detectar el movimiento del traslado de la cinta
- **Sensor de proximidad:** se eligió este tipo de sensor debido a que se necesita conocer cuando la cinta se hay terminada y así dar un aviso al trabajador.
- **Sensor de contacto:** este sensor se eligió debido a que se utilizara en los extremos del brazo robótico para así tener un inicio y un final para el área de estampado que se pueda utilizar.

Actuadores:

- **Electroválvula:** este actuador se eligió para controlar el accionar de los cilindros.
- **Cilindro doble efecto:** se utiliza para controlar la salida de la racle y el recogedor.
- **Motor a pasos:** se eligió los motores a paso ya que se necesitan precisión y fuerza para el estampado y el traslado de cinta

2.3.2.2 DEFINICIÓN DEL NÚMERO DE ENTRADAS Y SALIDAS

De acuerdo con los elementos que se van a utilizar para la automatización del proceso tenemos cinco entradas: para el sensor de proximidad, sensor reed, sensor de efecto hall y para los sensores de contacto; también tendremos 16 salidas las cuales estarán distribuidas para el control de las electroválvulas, los motores, ventilador, buzzer, leds indicadores, auxiliares para el sensor de efecto hall y para la pantalla HMI.

CAPÍTULO 3: COMPARACIÓN TÉCNICO Y ECONÓMICO DE DISPOSITIVOS

En este capítulo se describirá brevemente los dispositivos principales para la optimización de la fase de impregnación de tinta y así determinar mediante tablas comparativas en cuanto a aspectos económicos y técnicos cuál sería los elementos adecuados para el desarrollo del proyecto.

3.1 DESCRIPCIÓN DE LOS DISPOSITIVOS QUE SE PODRÍAN UTILIZAR PARA LA AUTOMATIZACIÓN DE LA FASE DE IMPREGNACIÓN DE TINTA

Se realizará una breve descripción de los dispositivos principales que se podrían utilizar para la aplicación de la fase de impregnación de tinta, para así obtener información de cada uno de los elementos y poder realizar una comparación entre cada uno de ellos.

3.1.1 MICROCONTROLADOR

Un microcontrolador es un circuito integrado que en su interior contiene o está conformado por las tres unidades básicas de un ordenador, los cuales son la memoria, el CPU o microprocesador y los periféricos de entrada y salida [10].

3.1.1.1 ARDUINO MEGA 2560

Es una tarjeta reprogramable la cual se encuentra conformada por un microcontrolador ATmega2560, una conexión USB, posee 16 entradas analógicas, 56 pines digitales que pueden funcionar como entrada/salida.

El lenguaje de programación que se utiliza para el Arduino es de alto nivel parecido al lenguaje de programación C++, es sencillo y fácil de aprender. Utiliza una plataforma de código abierto llamado Wiring, que a su vez se apoya en el lenguaje de programación Processing [11].

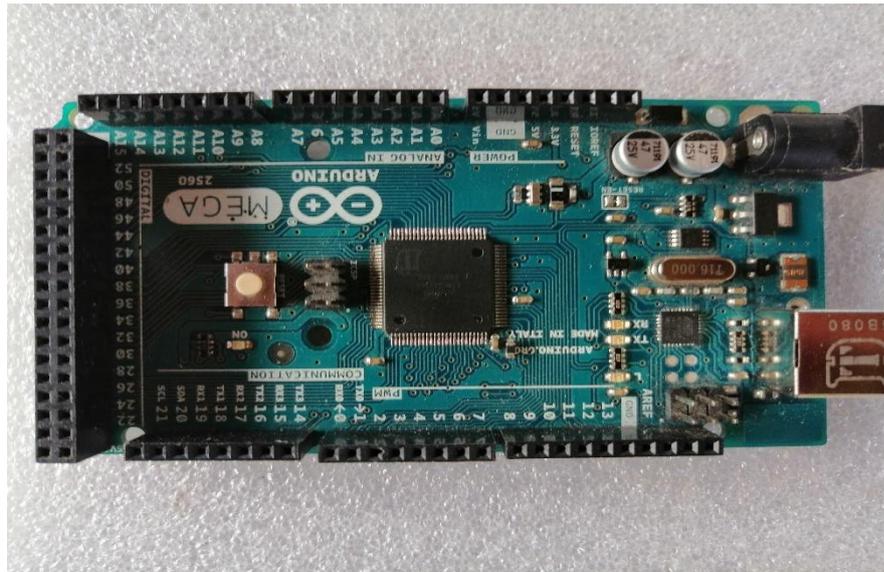


Figura 3. 1 Arduino Mega 2560

Fuente: Autor

Tabla 3. 1 Características del Arduino Mega 2560

Fuente: [11]

Características técnicas	
Voltaje de operación	5VDC
Tensión de entrada	7-12VDC
Pines de entrada-salida digital	54 pines (14 proveen salida PWM)
Entradas analógicas	16 pines
DC corriente pin I/O	40mA
DC corriente pin 3.3v	50mA
Memoria flash	256 KB
SRAM	8 KB
EEPROM	4 KB
Frecuencia de reloj	16MHz

3.1.2 CONTROLADOR LÓGICO PROGRAMABLE (PLC)

El PLC es un dispositivo que se diseñó para el control en tiempo real de procesos secuenciales que pueden ser ejecutados en un ambiente industrial su programación es de lenguaje no informático [12].

3.1.2.1 PLC SIEMENS LOGO V8

El PLC Logo V8 se desarrolló como una solución eficiente capaz de realizar tareas básicas de automatización. También posee una gran capacidad de integración en buses estándares industriales, por ejemplo con el estándar KNX el cual es un bus específico para la automatización de edificios [12].

Siemens LOGO está diseñado principalmente para la automatización a pequeña escala, por lo general Logo se utiliza para funciones básicas de automatización tales como en el hogar, además de esto permite controlar equipos de limpieza y tratamiento de superficies, realizar control de acceso, comandar máquinas dispensadoras, etc. [13]

Siemens LOGO cuenta con 8 entradas digitales, cuatro pueden ser utilizadas para uso analógico (0 V a 10 V) y cuatro como salidas digitales (relé) lo cual representa la relación entrada y salidas suficientes para un pequeño proyecto de automatización.

Tabla 3. 2 Características del PLC Logo V8

Características Técnicas del PLC	
Voltaje de operación	24 VDC
Tensión de entrada	5-8 VDC
Pines de entrada-salida digital	8/4
Entradas analógicas	Requiere módulo de expansión
DC corriente pin I/O	3- 10 A
DC corriente pin 3.3v	-
Memoria flash	-
SRAM	SD
EEPROM	2KB
Bloques	130
Memoria remanente	60
Frecuencia de reloj	43MHz
Precio	\$160

3.1.3 INTERFAZ HUMANO-MÁQUINA (HMI)

Para la fase de impregnación de tinta se ha decidido realizar una interfaz gráfica para una pantalla táctil la cual facilitara el manejo del sistema.

De acuerdo a los avances que se han presentado a lo largo de los años en los sistemas y herramientas que son utilizadas para el control y supervisión de varios procesos, es por esto que se realizó el diseño de una interfaz que sea de fácil uso para el operario [14].

En los sistemas de HMI se encuentran diferentes funciones como monitoreo, alarmas, supervisión, control, históricos [14].

3.1.3.1 PANTALLA NEXTION

La pantalla Nextion es una interfaz humano-máquina (HMI), esta combina un procesador con una pantalla táctil TFT, permite al operador un control y visualización de cualquier proceso, máquina, aplicación. Las aplicaciones principales de estas pantallas son en el campo de la electrónica de consumo y del Internet de las Cosas(IoT) [15].

La pantalla Nextion utiliza un solo puerto serie para comunicarse, además posee un software el cual es gratuito llamado Editor Nextion.

Nextion tiene 7 tipos de tamaños de pantallas (2.4", 2.8", 3.2", 3.5", 4.3", 5.0", 7.0"), es una solución HMI rentable ya que permite desarrollar una interfaz gráfica sencilla y eficiente [15].



Figura 3. 2 Pantalla Nextion NX4827T043-011

Fuente: Autor

Tabla 3. 3 Características Pantalla Nextion modelo

Fuente: [11]

Características técnicas pantalla Nextion	
Voltaje de operación	5VDC
Resolución	14, 6 pueden usarse como salidas PWM
Panel táctil	6 pines
Color	40mA
Destello	50mA
RAM	32 KB
MCU	2 KB
EEPROM	1 KB
Frecuencia de reloj	16MHz

3.1.3.2 PANEL HMI KINCO MT4230T



Figura 3. 3 Pantalla HMI Kinco 4'3.

Fuente: [16]

Kinco Automation es una empresa que se dedica a la producción y comercialización de productos de automatización industrial, garantizando que las soluciones sean completas y rentables, tiene una gran variedad de aplicaciones de procesamiento y de máquinas [16].

Kinco posee una línea completa de productos de automatización que incluyen interfaz hombre-máquina (HMI) las cuales se comercializan en el mercado local con un costo exequible para los desarrolladores de pequeños proyectos enfocados a la automatización industrial [16].

Tabla 3. 4 Características de pantalla Kinco MT4230T

Características técnicas pantalla Kinco MT4230T	
Voltaje de operación	24 VDC
Corriente de operación	-
Color	65536 colores
Resolución	480x272
Vida Útil	50000 horas
Peso	21000 g
Temperatura de trabajo	0~50°C
Temperatura de almacenamiento	-10~60°C
Humedad de trabajo	10 ~ 90% RH
Memoria RAM	64MB RAM
Memoria Flash	128MB FLASH
Tarjeta SD	Ninguna

3.1.4 DRIVERS PARA MOTOR A PASOS

Para la automatización del traslado de la cinta se utilizó un motor a pasos nema 23 y para el movimiento del brazo robótico se utilizó un motor a pasos nema 17 por lo cual se ve la necesidad de encontrar drivers que puedan ser utilizados para los dos motores.

3.1.4.1 DRIVER HY-DIV268N-5A

Este tipo de dispositivo es utilizado para controlar la velocidad y la dirección de motores nema 17 y 23. Es un controlador compatible con Arduino [17].



Figura 3. 4 Driver HY-FIV28-5A

Fuente: [17]

Tabla 3. 5 Características driver HY-DIV268N-5A

Características técnicas driver HY-DIV268N-5A	
Voltaje de operación	12-48 VDC
Corriente de salida	0.2-5A
Temperatura de operación	-10-45 °C
Control de micro pasos	1, 1/2, 1/4, 1/8, 1/16
Adecuado para motores	2-4 fases

3.1.4.2 DRIVER TB6560

Este controlador utiliza el chip TOSHIBA TB6560, el cual utiliza tecnología para controlar corriente sinusoidal pura, logrando que cualquier tipo de motor pueda trabajar con un menor ruido, evitando el calentamiento, el movimiento no es brusco y logra mejorar el rendimiento a gran velocidad en comparación con otros controladores en el mercado [18].

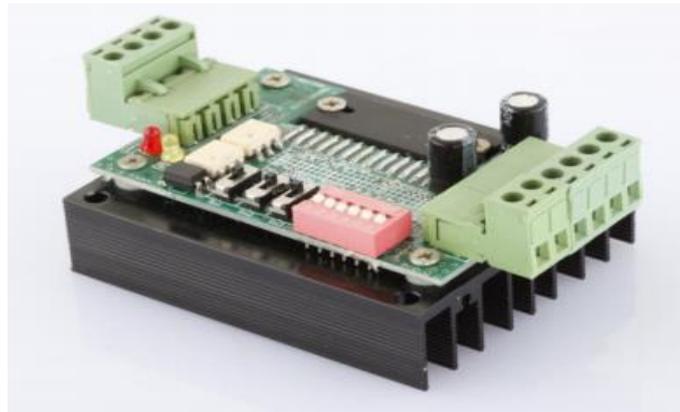


Figura 3. 5 Driver TB6560

Fuente: [18]

Tabla 3. 6 Características driver TB6560

Características técnicas driver TB6560	
Voltaje de operación	7-32 VDC
Corriente de salida	0.6-3A
Temperatura de operación	-10-45 °C
Control de micro pasos	1/2, 1/8 y 1/16
Adecuado para motores	2-4 fases

3.2 DESCRIPCIÓN DE ELEMENTOS PARA EL PROCESO DE AUTOMATIZACIÓN

En esta sección se mostrará una breve explicación de los elementos principales para la automatización de la fase de impregnación de tinta.

3.2.1 MOTOR ELÉCTRICO DC

Los motores eléctricos DC son dispositivos que generan energía mecánica a través de la energía eléctrica inducida, los motores pueden ser utilizados en aplicaciones industriales como: tornos, bombas de agua, bandas transportadoras, ventiladores, máquinas de papel, hornos, etc. [19].

El motor paso a paso es un dispositivo electromecánico el cual es capaz de transformar los pulsos eléctricos (señales digitales) en movimientos mecánicos. El eje de este tipo de motor realiza un desplazamiento angular por cada impulso eléctrico de entrada. Un motor paso a paso puede girar: de forma horaria como anti horaria, la cantidad de grados que se requiera [20].

3.2.2 RELÉS O CONTACTORES

El relé o contactor es un elemento electromagnético que tiene como función permitir o detener el paso de la corriente en cuanto se energice su bobina, puede controlar circuitos de salida de mayor potencia que la entrada. Los relés se los puede identificar en esquemas eléctricos con las letras KM [19].



Figura 3. 6 Relé 5VDC

Fuente: Autor

3.2.3 CILINDRO DE DOBLE EFECTO

Los cilindros de doble efecto poseen dos entradas para el aire comprimido los cuales permiten activar las cámaras que posee para realizar el avance y retroceso de su vástago o embolo, permitiendo realizar trabajos en ambos sentidos [21].

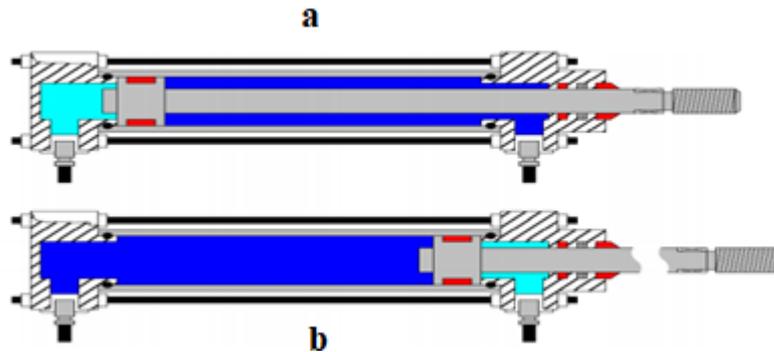


Figura 3. 7 a) Cilindro de doble efecto con el vástago contraído. b) Cilindro doble efecto con el vástago completamente fuera

Fuente: [21]

3.2.4 ELECTROVÁLVULA

La válvula electromecánica o electroválvula es diseñada para realizar el control del paso de cualquier fluido por una tubería. Este dispositivo posee dos tipos de posiciones abierto y cerrado. [22]. La electroválvula está conformado por dos partes: el solenoide y la válvula.



Figura 3. 8 Electroválvula 3 posiciones 2 salidas

Fuente: Autor

3.3 COMPARACIÓN DE CARACTERÍSTICAS ENTRE LOS DISPOSITIVOS PARA LA FASE DE IMREGNACIÓN DE TINTA

Se realizará una comparación entre algunos de los dispositivos propuestos en la sección 3.1 de acuerdo a su coste, características técnicas, características eléctricas y a partir de esta comparación se definirá los componentes adecuados a utilizar en el desarrollo del proyecto.

3.3.1 COMPARACIÓN ENTRE MICROCONTROLADOR Y PLC

Se podrá observar la comparación de las especificaciones, características eléctricas y precio entre el microcontrolador y el PLC, aunque los dos elementos tienen características similares se eligió el microcontrolador Arduino mega 2560 debido a que se ajusta a las necesidades para la realización del proyecto y también al menor costo que tiene el microcontrolador.

Tabla 3. 7 Comparación entre Microcontrolador y PLC

Comparación entre microcontrolador y PLC		
Características	Arduino Mega 2560	Siemens Logo v8
Voltaje de operación	5 VDC	24 VDC
Tensión de entrada	7-12 VDC	5-8 VDC
Pines de entrada-salida digital	54 pines (14 proveen salida PWM)	8/4
Entradas analógicas	16 pines	Requiere módulo de expansión
DC corriente pin I/O	40mA	3- 10 A
DC corriente pin 3.3v	50mA	-
Memoria flash	256 KB	-
SRAM	8 KB	SD
EEPROM	4 KB	2KB
Bloques	-	130
Memoria remanente	-	60
Frecuencia de reloj	16MHz	43MHz
Precio	\$25	\$160

3.3.2 COMPARACIÓN ENTRE PANTALLAS HMI

Se podrá observar en la siguiente tabla la comparación entre las especificaciones, características eléctricas y del precio entre la pantalla Nextion y la pantalla Kinco, como se puede apreciar en la tabla tanto la pantalla Nextion y Kinco tiene características parecidas por lo cual se seleccionó a la pantalla Nextion ya que se ajusta al microcontrolador Arduino mega 3560 anteriormente seleccionado.

Tabla 3. 8 Comparación entre pantallas HMI

Comparación entre las pantallas HMI		
Características	Nextion	Kinco MT4230T
Voltaje de operación	5 VDC	24 VDC
Corriente de operación	500mA	-
Color	65536 colores	65536 colores
Resolución	480 x 272	480x272
Vida Útil	>30000 horas	50000 horas
Peso	79.3g	21000 g
Temperatura de trabajo	-20 a 70°C	0~50°C
Temperatura de almacenamiento	-30 a 85°C	-10~60°C
Humedad de trabajo	10% a 90% RH	10 ~ 90% RH
Memoria RAM	3584 Byte	64MB RAM
Memoria Flash	16MB	128MB FLASH
Tarjeta SD	32GB	Ninguna
Precio	\$65	239

3.3.3 COMPARACIÓN ENTRE DRIVERS

En la siguiente tabla se podrá observar la comparación entre las especificaciones, características eléctricas y del precio entre los drivers para motores a pasos, como se puede apreciar en la tabla tanto el driver HY-DIV268N-5A y el driver TB6560 tienen características similares pero el costo difiere por lo cual se eligió el controlador con menor precio ya que satisface a las necesidades para los motores a pasos que se utilizara.

Tabla 3. 9 Comparación entre drivers para motores nema

Comparación entre los drivers		
Características	HY-DIV268N-5A	TB6560
Voltaje de operación	12-48 VDC	7-32 VDC
Corriente de salida	0.2-5A	0.6-3A
Temperatura de operación	-10-45 °C	-10-45 °C
Control de micro pasos	1, 1/2, 1/4, 1/8, 1/16	1/2, 1/8 y 1/16
Adecuado para motores	2-4 fases	2-4 fases
Precio	\$50	\$12

Una vez realizada la comparación entre los elementos principales se pudo conocer las características técnicas, así como su respectivo costo de cada uno de ellos, de acuerdo a las necesidades del proyecto y que a la vez va dirigido para una microempresa se optaron por los siguientes elementos para su implementación: Arduino Mega 2560, pantalla Nextion y driver TB6560.

CAPÍTULO 4: DISEÑO DE HARDWARE Y SOFTWARE

En el presente capítulo se realizará el diseño del hardware y software para la implementación del sistema automatizado para optimizar la fase de impregnación de tinta en el proceso de serigrafía, los diseños presentados en esta sección son: La estructura mecánica, el sistema electrónico y la interfaz gráfica.

Se puede apreciar en la figura 4.1 el diagrama de bloques del funcionamiento del sistema automatizado para la fase de impregnación de tinta para tafíletes.

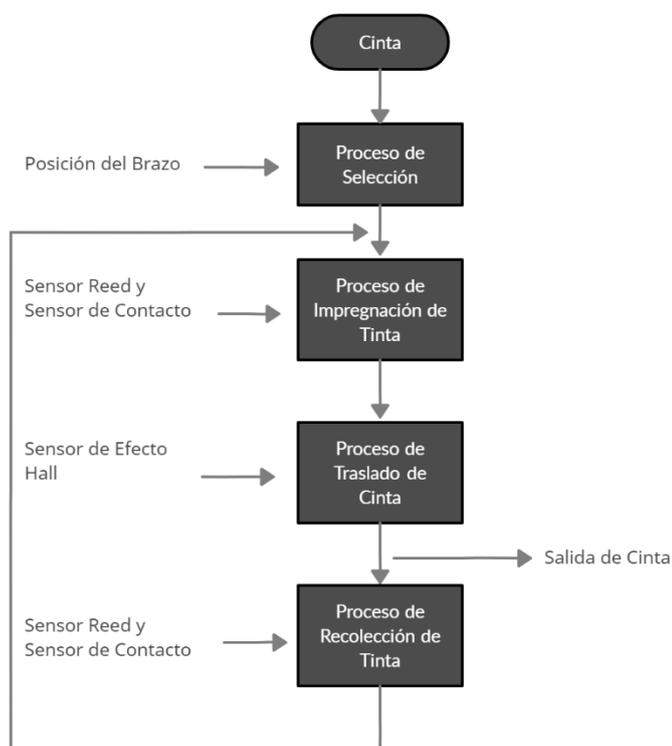


Figura 4. 1 Diagrama de Bloques del funcionamiento del sistema

4.1 DISEÑO DE LA ESTRUCTURA MECÁNICA

En esta sección se podrá observar el diseño en 3D realizado en Autodesk Inventor Profesional de la estructura mecánica, también se muestra en la sección de anexos los planos en 2D con sus respectivas medidas.

La estructura mecánica se divide en tres secciones:

- Tensado de cinta
- Brazo para estampar
- Traslado de cinta

4.1.1 TENSADO DE CINTA

En esta sección se podrá observar el elemento que se utiliza para realizar el tensado de la cinta, es la parte inicial de la estructura.

Este elemento cuenta con ejes móviles para desplazar y de esta manera obtener la tensión adecuada para la cinta, evitando inconvenientes al momento del traslado, también cuenta con guías elaboradas en grilon, los cuales serán utilizados para que la cinta pueda recorrer por lugar que se requiere.



Figura 4. 2 Dispositivo para tensado de cinta

Fuente: Autor

4.1.2 BRAZO PARA ESTAMPAR

A través de esta estructura se puede realizar el estampado y recolección de la tinta, debido a que el brazo está conformado por los cilindros que realizarán estas funciones, el desplazamiento del brazo será de izquierda a derecha, para realizar este movimiento se utiliza un motor a pasos nema 17.

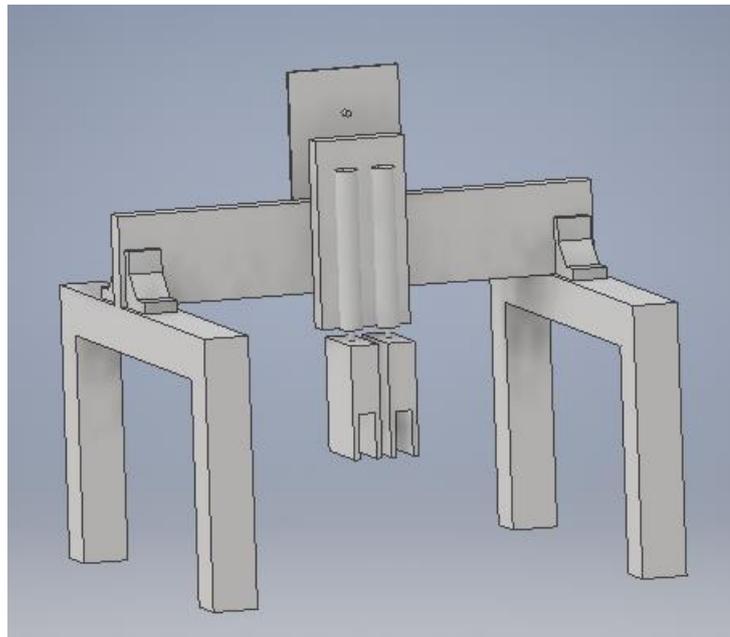


Figura 4. 3 Brazo para estampar

Fuente: Autor

4.1.3 TRASLADO DE CINTA

El rollo de cinta se encuentra ubicado al lado del elemento utilizado para el tensado la cual es la parte inicial de la estructura. El traslado de la cinta se la debe realizar desde la parte inicial hacia la parte final de la estructura mecánica en donde se encuentra dos rodillos, para realizar el traslado se coloca entre los rodillos a la cinta para que la presión entre ellos la pueda jalar sin ningún inconveniente por todas las áreas que conforman la estructura. Para el traslado de la cinta se utiliza un motor a pasos nema 23 colocado en el rodillo inferior y esta a su vez moverá al rodillo superior logran así el traslado de la cinta.

Se puede observar en la figura 4.4 tanto el área para el estampado como el área para el tensado de la cinta, también se puede apreciar unos ejes con sus respectivas guías de grilon para que la cinta pueda ser trasladado correctamente y así evitar que se

desvié, observamos la posición del marco en que se sujeta una malla o tamiz con el diseño a estampar.

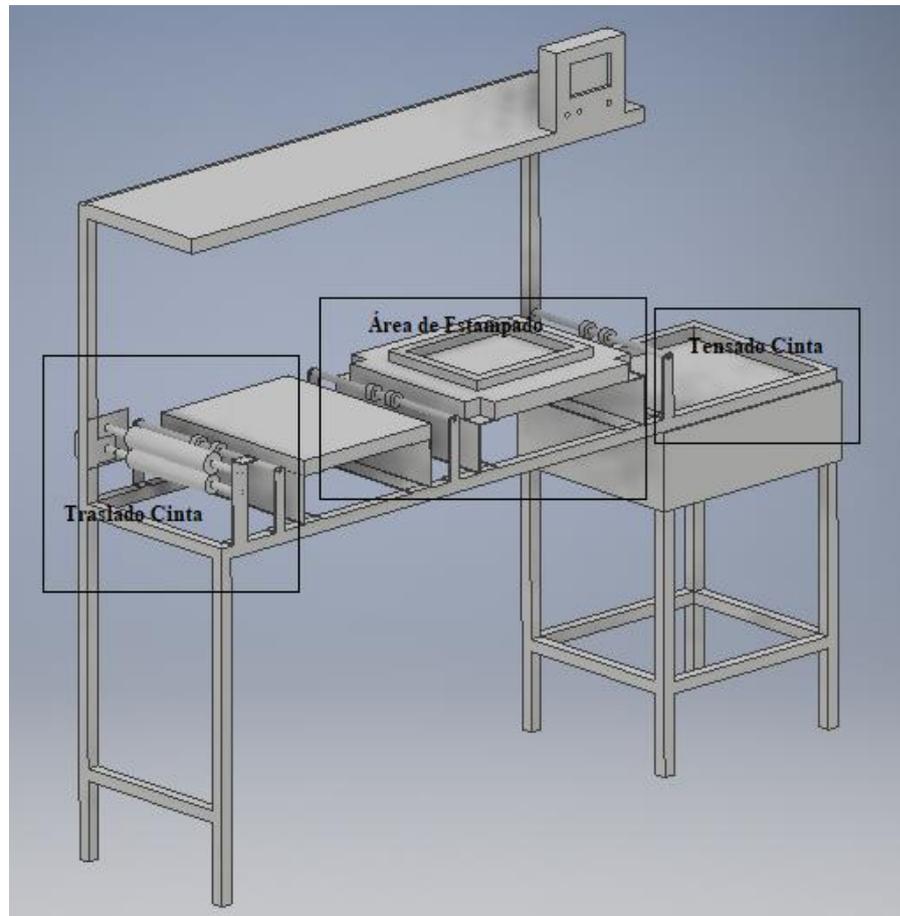


Figura 4. 4 Estructura para el traslado de cinta

Fuente: Autor

4.1.4 ESTRUCTURA MECÁNICA COMPLETA

Observaremos la estructura mecánica completa con todas las secciones anteriormente mencionadas.

La secuencia de toda la estructura es la siguiente: primero se coloca la cinta en los rodillos que se encuentran al final, en el área de estampado se encontrará el brazo el cual está conformado por dos cilindros que se accionaran mediante aire comprimido estos se utilizan para realizar el estampado y la recolección de la tinta, una vez accionado el cilindro de la racle el brazo recorrerá de izquierda a derecha, terminada esta secuencia se desactivara el cilindro y se realizará el traslado de la cinta, finalizado

el traslado se accionará el segundo cilindro para la recolección de la tinta y el brazo regresara a su posición inicial.

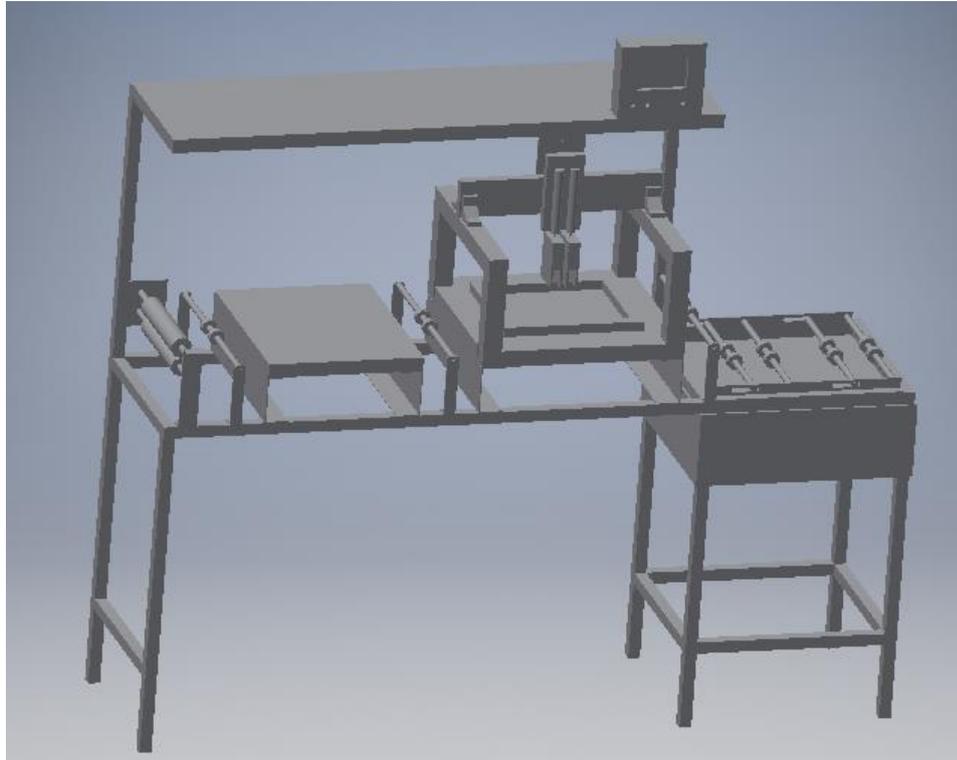


Figura 4. 5 Estructura completa

Fuente: Autor

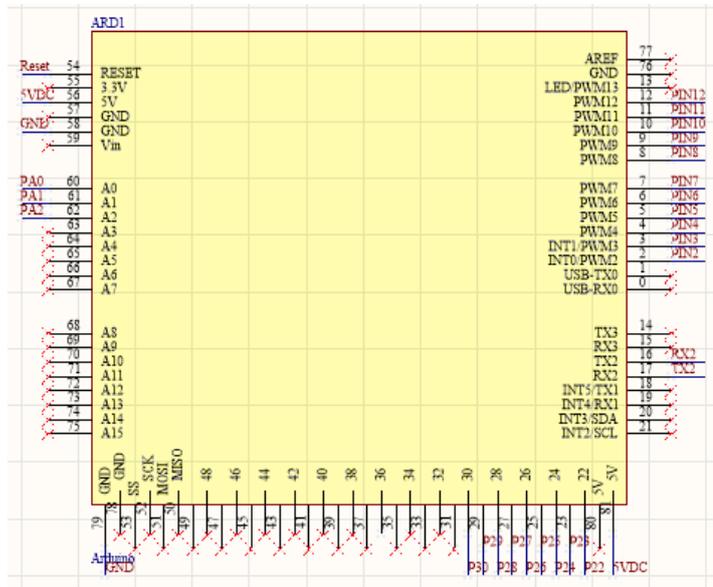
4.2 DISEÑO DEL SISTEMA ELECTRÓNICO

En esta sección encontramos el diseño del hardware para el sistema electrónico el cual está conformado por una tarjeta electrónica para el control del sistema, diseñado mediante el software Altium Designer, se puede apreciar tanto el esquemático como el PCB en 2D y 3D.

El sistema electrónico está dividido en tres secciones:

- Sistema de control
- Alimentación
- Sensado y alarma

En la figura 4.6 se puede observar el esquemático del microcontrolador Arduino mega 2560 en el programa Altium Designer con los respectivos pines que utilizaron.



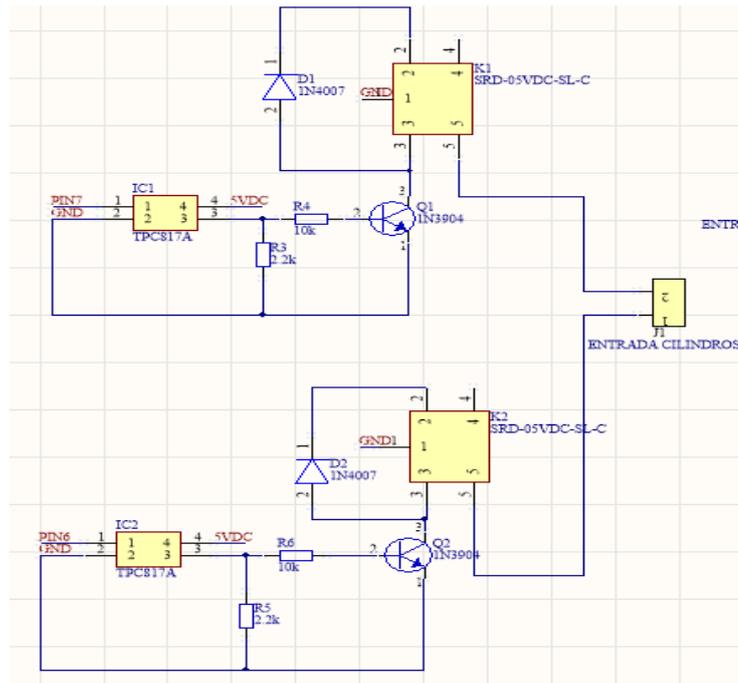


Figura 4. 7 Sistema de control para las electroválvulas

Fuente: Autor

En la figura 4.8 está constituido por tres conectores los cuales son utilizados como salidas para la conexión tanto para la pantalla Nextion como de los drivers tb6560 para los motores a paso nema 23 y 17.

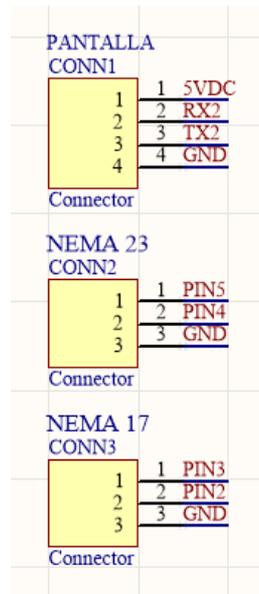


Figura 4. 8 Conectores para conexión de pantalla y drivers

Fuente: Autor

Se puede apreciar en la figura 4.9 el diseño realizado para la activación del ventilador el cual está conformado por relé, optoacoplador y transistor.

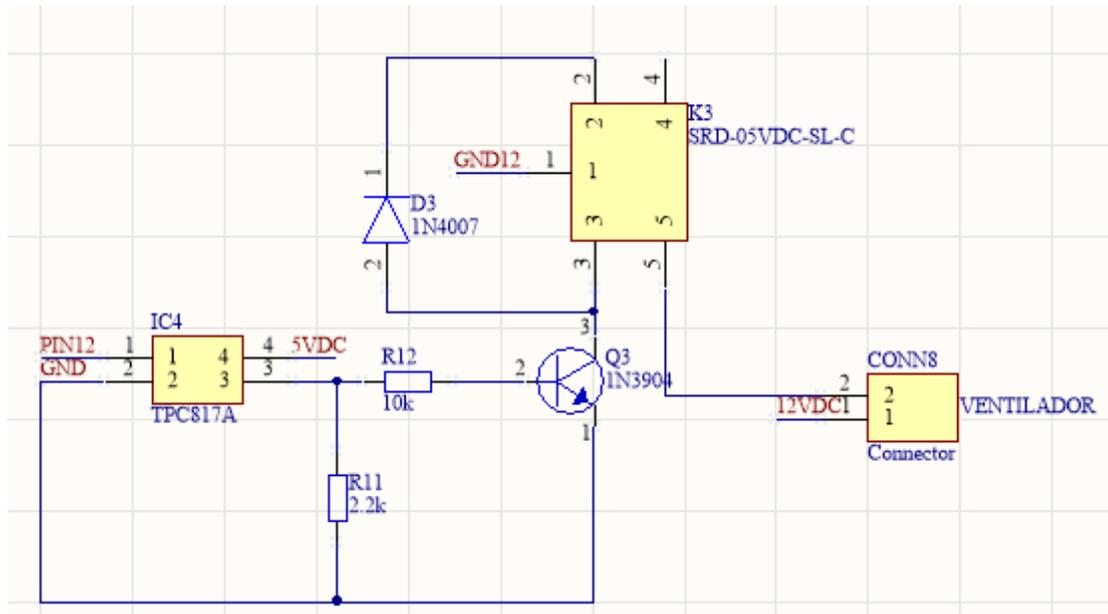


Figura 4. 9 Sistema de ventilación

Fuente: Autor

También encontraremos dos conectores adicionales, el primero posee salidas analógicas y digitales, estos se pueden utilizar como salidas auxiliares para futuras implementaciones, el segundo conector denotado con RESET se utiliza para reiniciar el microcontrolador.

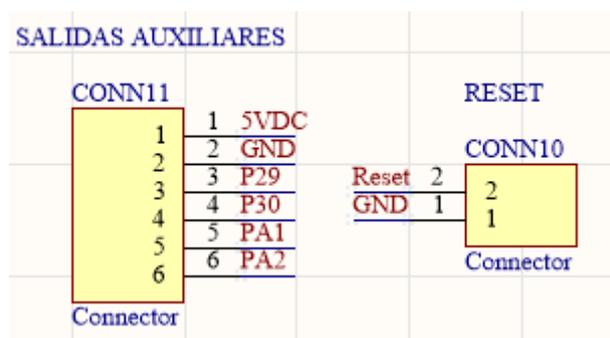


Figura 4. 10 Salidas auxiliares y reset

Fuente: Autor

4.2.2 ALIMENTACIÓN

Se puede observar en la figura 4.11 cinco conectores los de la izquierda se utilizan como entrada de la fuente de alimentación de 24VDC y cabe mencionar que el positivo está conectado a otro conector que se utiliza como salidas para las electroválvulas del sistema de control, los conectores de la derecha se utiliza como entrada de la fuente de alimentación de 12VDC y también se utilizan como salidas para alimentar a los drivers utilizados para el control de los motores a pasos.

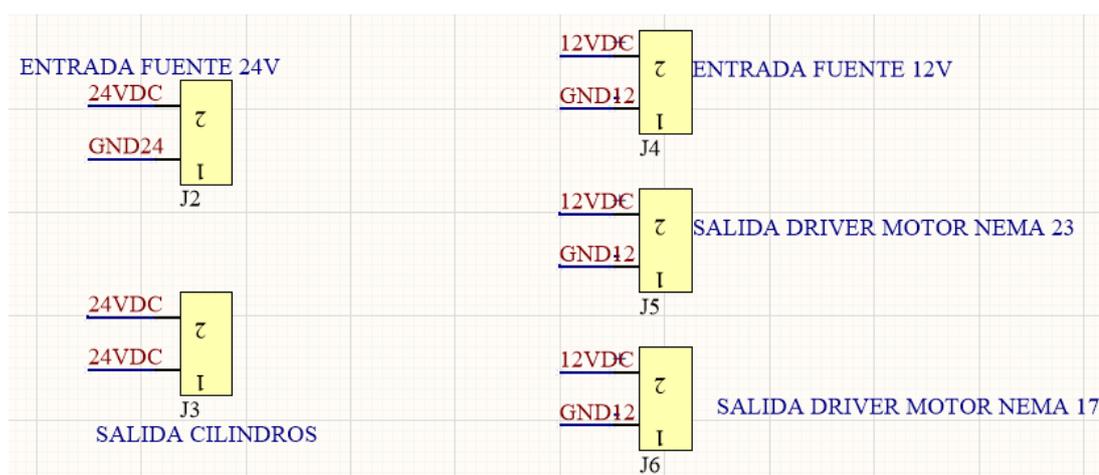


Figura 4. 11 Entrada de las fuentes de 12-24VDC

Fuente: Autor

4.2.3 SENSADO Y ALARMA

En la figura 4.12 observamos cinco conectores utilizados para los sensores, que nos permiten obtener información del comportamiento externo del sistema, entre ellos están un sensor de efecto hall, un señor reed, sensor sharp y un par de sensores de contacto. Adicional observamos la salida para un buzzer que se utiliza como alarma ante una posible falla en el sistema.

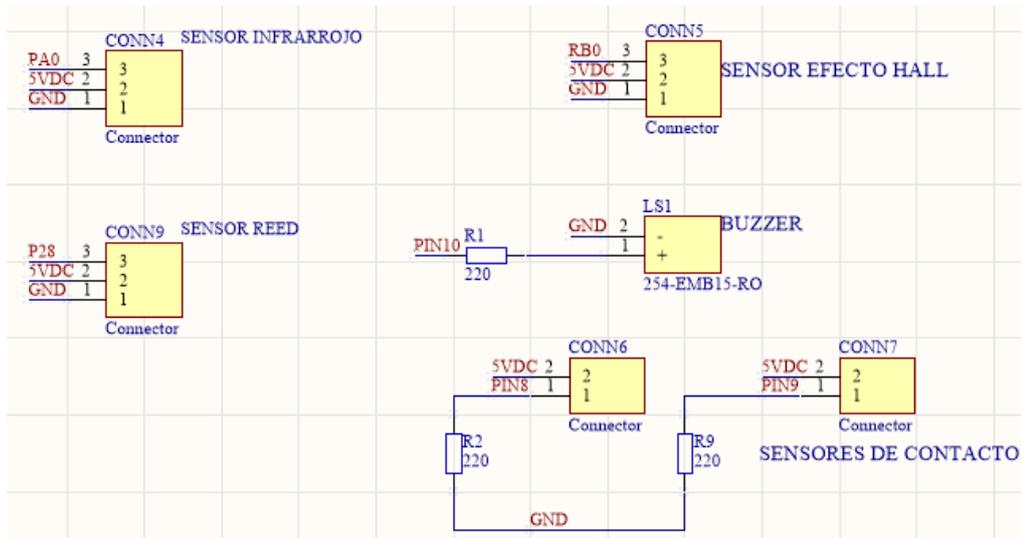


Figura 4. 12 Conectores para los sensores y buzzer para las alarmas

Fuente: Autor

También encontraremos dos leds que son utilizados como indicadores de correcto funcionamiento (led verde) o cuando se presente un error en el sistema (led rojo).

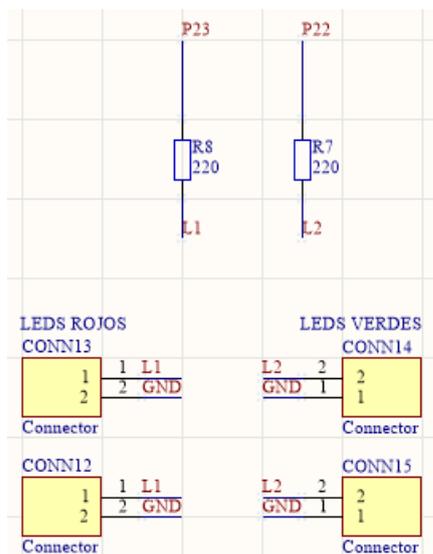


Figura 4. 13 Leds indicadores

Fuente: Autor

Para el control del sensor de efecto hall se utilizó un segundo microcontrolador el cual es el PIC16F84A y además se utilizó salidas del arduino que detectara el pic para así restringir cada dato que se obtenga del sensor en cada una de las tallas que se utilice para el estampado y así evitar inconvenientes en el funcionamiento.

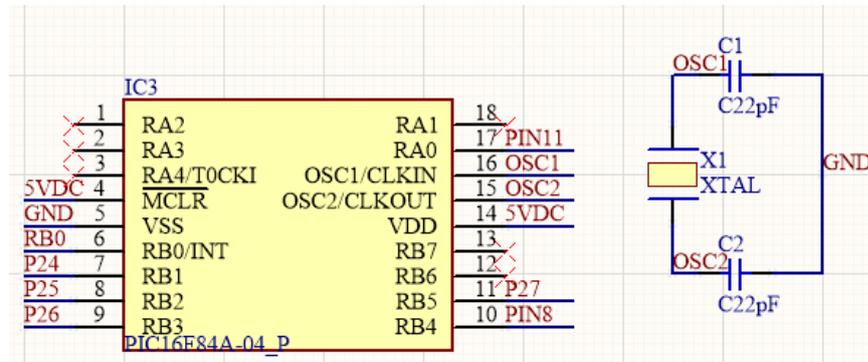


Figura 4. 14 PIC16F84A

Fuente: Autor

4.2.4 SISTEMA ELECTRÓNICO COMPLETO

En la siguiente imagen se puede observar el diseño completo del esquemático de las secciones anteriormente explicadas conectadas con el microcontrolador Arduino mega 2560

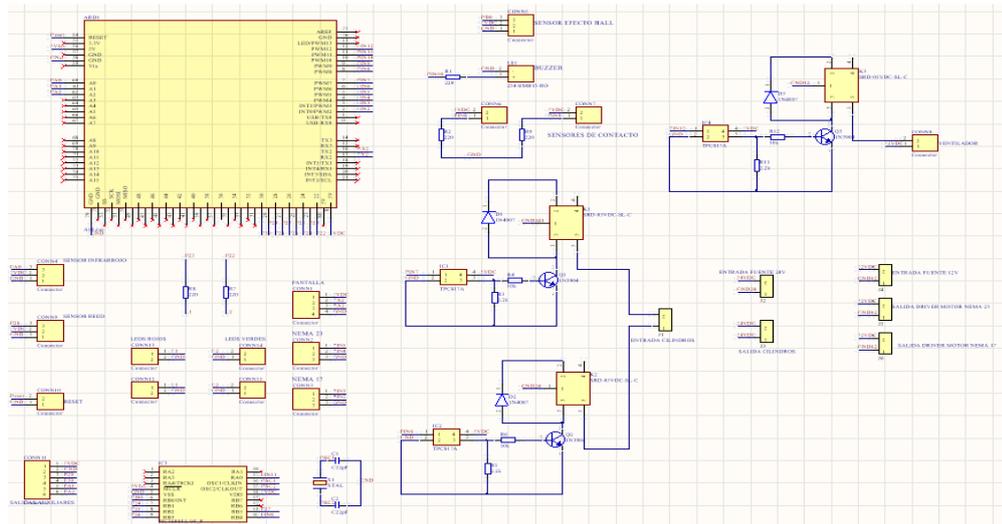


Figura 4. 15 Diseño completo de la tarjeta electrónica en Altium Designer

Fuente: Autor

En las siguientes figuras observaremos el diseño realizado de la PCB de dos capas tanto el 2D como el 3D.

En la figura 4.16 se observa la primera capa de la PCB la cual se la conoce como top layer, también se puede observar el plano de tierra realizado a esta capa.

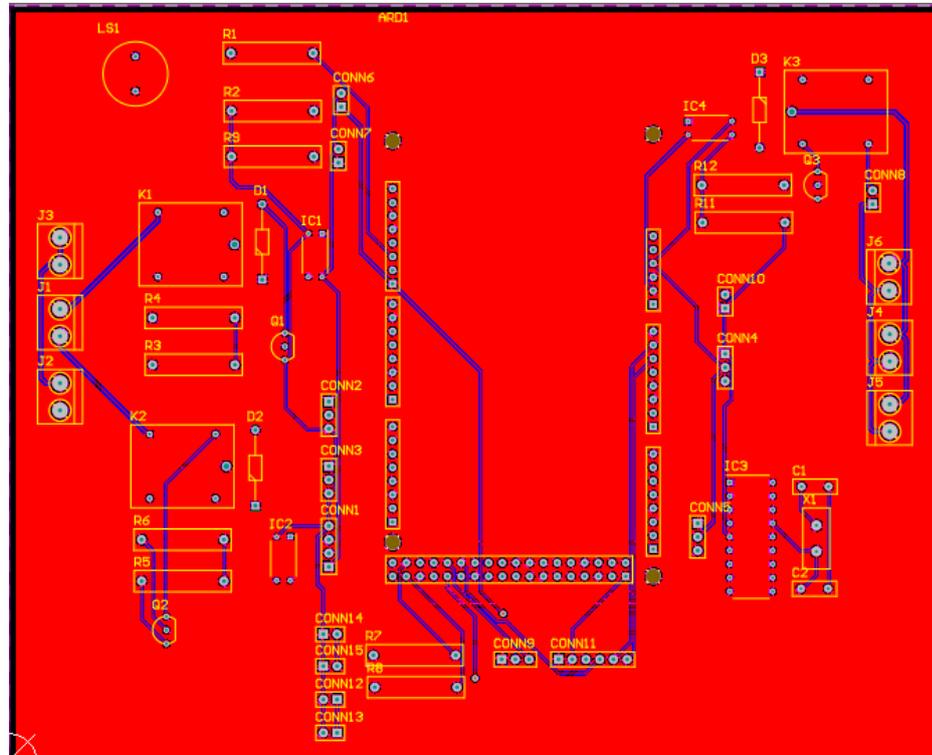


Figura 4. 16 Primera capa de la PCB con su plano de tierra

Fuente: Autor

En la figura 4.17 se observa la segunda capa de la PCB la cual se la conoce como bottom layer, también se puede observar el plano de tierra realizado a esta capa.

4.3 DISEÑO INTERFAZ GRÁFICA

En esta sección se realizara el diseño de la interfaz gráfica de la pantalla Nextion, esto se lo realizará mediante el software Nextion Editor.

La interfaz gráfica está dividida en tres secciones:

- Inicialización
- Menú principal
- Avisos de fallas

4.3.1 INICIALIZACIÓN

Esta sección está dividida en dos pantallas, en la primera se espera un tiempo de inicialización que una vez completada nos llevara a la página principal.



Figura 4. 19 Pantalla de inicialización

Fuente: Autor

La segunda pantalla consta de un botón (b0) el cual cumple con dos funciones la primera de mover el brazo a su posición inicial si este se encuentra en otra posición y la segunda de llevar a una tercera pantalla donde se encontrara el menú.



Figura 4. 20 Segunda pantalla

Fuente: Autor

4.3.2 MENÚ PRINCIPAL

En esta sección se encuentra el menú principal el cual está conformado por diez botones, cuatro textos y dos números.

El botón stop (b8) se utiliza para parar el proceso indiferentemente en que sección se encuentre del estampado y llevará al brazo a la posición inicial.

Los botones (b5) + y (b6) – se utilizan para elegir el número de estampados que se necesiten de cualquiera de las tallas, el cual podremos observar en el número (n1) arriba de los botones, cuando se llaga a la cantidad seleccionada el proceso de estampado se parara.

Los cuatro botones de talla (b1, b2, b3 y b4) se pueden utilizar ya sea en conjunto con los botones b5 y b6 o individualmente. Si se presiona únicamente el botón de una talla, inicia un proceso de estampado cíclico y no se detendrá hasta que se presione el botón de stop.

En el número (n2) se podrá apreciar el total de estampados que se han realizado, el botón de reset (b9) se utiliza para limpiar los números n1 y n2.

El botón de pause (b7) se puede utilizar en cualquier momento del estampado, al accionar este el sistema se detendrá en la sección en la que se encuentre y no realizara ninguna acción hasta que se pulse el mismo botón de talla que se utilizó al principio.

El botón de back (b10) se utiliza para regresar a la pantalla principal.



Figura 4. 21 Menú principal

Fuente: Autor

4.3.3 AVISOS DE FALLOS

Para la protección del proceso se ha implementado cuatro mensajes de error las cuales se visualizan como pantallas de advertencia, estos mensajes bloquean la continuidad del proceso hasta que sean corregidos.

El primer mensaje de error nos indica cuando la cinta se ha agotado, se lo realiza mediante un sensor infrarrojo.



Figura 4. 22 Aviso de cinta agotada

Fuente: Autor

El segundo mensaje de error nos alerta cuando el brazo no está funcionando correctamente, esto lo realizamos mediante dos sensores de contacto en los extremos del brazo los cuales detectan si el brazo llego a la posición final e inicial.



Figura 4. 23 Aviso de revisar brazo

Fuente: Autor

El tercer mensaje de error nos indica una falla de desplazamiento del motor a pasos encargado del traslado de la cinta, esto se puede suscitar por un daño en el motor o un bloqueo del mismo, la adquisición de esta señal se realizó mediante un sensor de efecto hall el cual detectará el campo magnético generado por un imán que se encuentra acoplado al eje del motor.



Figura 4. 24 Aviso falla motor traslado de cinta

Fuente: Autor

El último mensaje de error se visualiza cuando los cilindros para realizar el estampado o la recolección de tinta no descendieron para realizar su función, medida implementada a través de un sensor Reed que detectara si los cilindros bajaron o se accionaron.



Figura 4. 25 Aviso revisar cilindros

Fuente: Autor

CAPÍTULO 5: PRUEBAS Y CORRECCIÓN DE ERRORES

En este capítulo se indica el resultado de las pruebas realizadas al sistema, con esto se logró garantizar el correcto funcionamiento, además presentamos las respectivas correcciones de los errores suscitados.

5.1 CORRECIÓN DE ERRORES

Mientras se realizaba las pruebas se presentaron ciertos errores debido a diferentes causas, por lo cual en esta sección explicaremos los inconvenientes que se presentaron y como se corrigió cada uno de ellos.

Los errores encontrados fueron los siguientes:

5.1.1 ERROR DE ESTAMPADO

Para realizar el estampado se utiliza una racle que se encuentra ubicada en un cilindro, la racle está conformada por una goma, la cual presentó inconvenientes debido a la forma de la punta de la misma, es decir, en el área de contacto con el tamiz, por esa razón se realizó pruebas de estampado con tres diferentes diseños para que mantenga un contacto adecuado con la superficie que contiene el diseño.

Los diseños/formas utilizados en la goma fueron los siguientes:

a) GOMA RECTANGULAR

En la forma rectangular se presentaron algunos inconvenientes el primero fue que la racle se atrancaba a la mesa debido a su forma, impidiendo que el brazo realice su desplazamiento, el segundo error que se nos presento fue una falta de presión sobre la superficie de estampado.

b) GOMA EN PUNTA

Con los errores anteriormente mencionados se procedió a realizar a la goma de forma triangular o en punta, se realizó varias pruebas, sin embargo presunto un inconveniente el cual fue que la racle no realizaba la suficiente presión para el estampado por lo cual los diseños salían opacos o difuminados.

c) GOMA REDONDA

Por último, se realizó pruebas con la goma en forma redonda la cual fue la solución para los inconvenientes anteriormente mencionados ya que se pudo realizar estampados con la suficiente presión garantizando la calidad del estampado y no se obtuvo ningún problema en el desplazamiento del brazo, obteniendo un resultado satisfactorio.

5.1.2 ERROR MOTOR TRASLADO DE CINTA

En este error se presentaron dos problemas uno electrónico y otro mecánico por lo cual explicaremos los dos inconvenientes y su respectiva solución.

a) PROBLEMA MECÁNICO

El motor de traslado de cinta tiene colocado en su eje un rodillo y encima del eje se encuentra otro rodillo los cuales realizan la función de traslado de la cinta.

Debido a esto se presentó el inconveniente que el motor se trababa por el rodillo superior ya que los ejes se encontraban fijos y también porque se encontraban los rodillos demasiado juntos y se inmovilizaban entre sí.

Para solucionar este problema se incrementó el tamaño de los orificios en donde se encuentra los ejes para que se puedan mover con más libertad, también se colocó en los extremos del eje unos reguladores para incrementar o disminuir la distancia entre los rodillos.

b) PROBLEMA ELECTRÓNICO

En cuanto al problema electrónico empezó a mandar el aviso de revisar el motor de traslado de cinta debido que el sensor colocado para detectar esa falla no estaba mandando los datos en tiempo real al Arduino ya que el microcontrolador se permanecía en el bucle del motor por lo tanto omitía los cambios del sensor.

Para la solución a este problema se utilizó un segundo microcontrolador el cual es el pic16f84a para que detectara los datos del sensor mientras el Arduino este en el bucle del motor y así rectificar si hubo movimiento de los rodillos para el traslado de la cinta, y de esta manera lograr que mande el mensaje de error cuando se presente los inconvenientes correctos.

5.1.3 ERROR EN LOS CILINDROS

Este error se nos presentó debido a que los vástagos de los cilindros salían con demasiada fuerza y podrían dañar al marco y a la vez no se podía realizar un estampado de calidad.

Para la solución de esta falla se colocó estranguladores de aire para los cilindros para que los vástagos puedan bajar y subir con la velocidad adecuada.

5.2 PRUEBAS DE ESTAMPADO DE CADA TALLA

Una vez corregido todas las fallas que se presentaron se procedió a realizar las pruebas finales de nuestro proyecto, observaremos la calidad del estampado en cada una de las tallas.

Las pruebas realizadas fueron de cuatro tallas las cuales son las más comunes en estampados de tafilete sus medidas son 54, 58, 62 y 66 cm de largo, cabe recalcar que estas medidas son de la distancia que recorrerá la cinta mas no del diseño del estampado ya que este tiene una medida de 2.5 cm de ancho por 9.5 cm de largo.

5.2.1 PRUEBAS DE LA TALLA 54

Se realizó el estampado de un rollo de cinta de 88m y se obtuvo 162 estampados en un tiempo de 24 minutos con 18 segundos, también se lo realizo de la forma tradicional o manual y se obtuvo los mismos estampados, pero en un tiempo más largo el cual fue de 30 minutos con 40 segundos.

En la figura 5.1 se podrá observar la comparación con respecto a los tiempos entre los estampados tanto de la forma manual como de la automatizada.



Figura 5. 1 Comparación de tiempo entre estampados manual y automatizado

Fuente: Autor

En la figura 5.2 se puede observar la calidad del estampado en la talla 54



Figura 5. 2 Estampado de la Talla 54

Fuente: Autor

5.2.2 PRUEBAS DE LA TALLA 58

Se realizó el estampado de un rollo de cinta de 88m y se obtuvo 150 estampados en un tiempo de 22 minutos con 23 segundos, también se realizó de la forma tradicional o manual y se obtuvo los mismos estampados, pero en un tiempo más largo el cual fue de 28 minutos con 33 segundos.

En la figura 5.3 se podrá observar la comparación con respecto a los tiempos entre los estampados tanto de la forma manual como de la automatizada.



Figura 5. 3 Comparación de tiempo entre estampados manual y automatizado

Fuente: Autor

En la figura 5.4 se puede observar la calidad del estampado en la talla 58



Figura 5. 4 Estampado de la Talla 58

Fuente: Autor

5.2.3 PRUEBAS DE LA TALLA 62

Se realizó el estampado de un rollo de cinta de 88m y se obtuvo 141 estampados en un tiempo de 23 minutos con 28 segundos, también se lo realizó de la forma tradicional o manual y se obtuvo los mismos estampados, pero en un tiempo más largo el cual fue de 29 minutos con 16 segundos.

En la figura 5.5 se podrá observar la comparación con respecto a los tiempos entre los estampados tanto de la forma manual como de la automatizada.

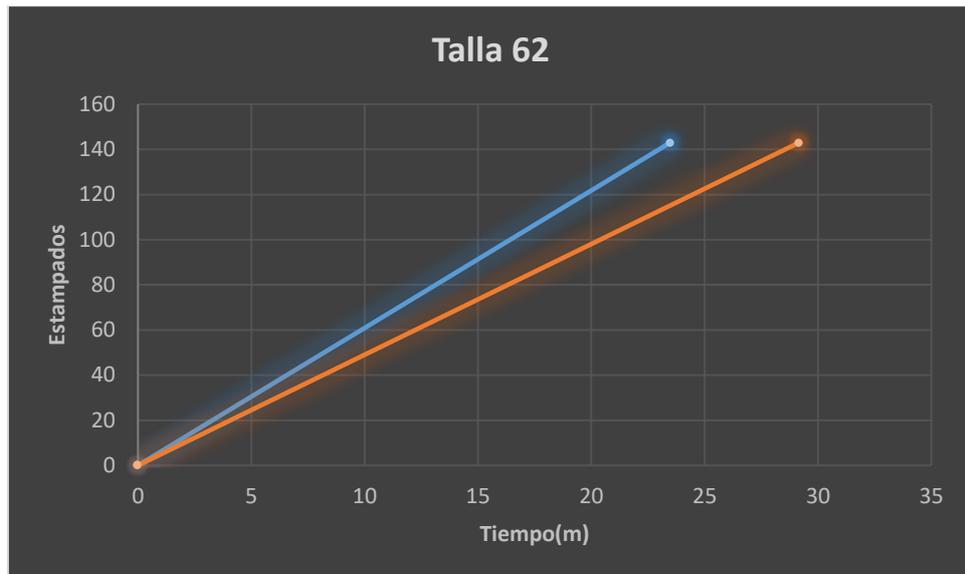


Figura 5. 5 Comparación de tiempo entre estampados manual y automatizado

Fuente: Autor

En la figura 5.6 se puede observar la calidad del estampado en la talla 62



Figura 5. 6 Estampado de la Talla 62

Fuente: Autor

5.2.4 PRUEBAS DE LA TALLA 66

Se realizó el estampado de un rollo de cinta de 88m y se obtuvo 133 estampados en un tiempo de 24 minutos con 23 segundos, también se lo realizó de la forma tradicional o manual y se obtuvo los mismos estampados, pero en un tiempo más largo el cual fue de 31 minutos con 25 segundos.

En la figura 5.7 se podrá observar la comparación con respecto a los tiempos entre los estampados tanto de la forma manual como de la automatizada.



Figura 5. 7 Comparación de tiempo entre estampados manual y automatizado

Fuente: Autor

En la figura 5.8 se puede observar la calidad del estampado en la talla 66



Figura 5. 8 Estampado de la Talla 66

Fuente: Autor

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

1. El proceso de estampado mediante el método de serigrafía es muy común actualmente y se utiliza para la impregnación de tinta en textiles, pudimos conocer que en las microempresas (Pymes) este proceso se lo realiza de manera artesanal, para este tipo de procedimiento se lo realiza utilizando una racle, tinta y un marco que estará conformada de una malla con el diseño que se requiere estampar en los textiles, las medidas del marco y de la malla dependerán del diseño que se utilice para él estampado.
2. Una vez que se estudió el caso de la microempresa la cual requería automatizar el proceso de impregnación de tinta en textiles (tafiletes) y analizado las tecnologías y métodos de la automatización, pudimos determinar la viabilidad de aplicar estos procesos a la construcción de nuestro proyecto, las tecnologías que se aplicaron fueron la cableada y programable, debido a que estos procesos nos permiten realizar el control del sistema mediante un microcontrolador y requieren de un bajo costo económico en materiales y equipos, así como del mantenimiento del sistema. El método que se utilizó fue la programable ya que nos permite realizar modificaciones futuras al sistema dependiendo de las necesidades que se puedan presentar.
3. Las pequeñas empresas ven limitada su producción ante los costos elevados de equipos tecnológicos en el mercado. Teniendo en cuenta este aspecto y la capacidad económica de las microempresas optamos por realizar una comparación de características técnicas y económicas entre equipos disponibles en el mercado, una vez realizado este análisis optamos por dispositivos de bajo costo sin que ellos represente una disminución de fiabilidad ni de calidad para la implementación de este proyecto ya que se obtuvo un resultado óptimo en el proceso de impregnación de tinta.

4. Al seleccionar los dispositivos y elementos que se utilizaran en el desarrollo del proyecto, se procedió a realizar el diseño del hardware y software para el control del sistema de impregnación de tinta. Se desarrolló una tarjeta electrónica, encargada de recibir información, procesar las señales y controlar los actuadores, todo esto a través de un microcontrolador Arduino Mega 2560, este utiliza un lenguaje de programación propio de alto nivel el cual es muy similar a C++, mediante la cual pudimos obtener el control sobre las variables del sistema a automatizar. Finalmente se implementó una interfaz gráfica de usuario la cual a través de botones nos permite controlar el proceso de estampado y obtener información relevante en tiempo real, además visualizar los errores que se pueden presentar en la máquina.

5. Las pruebas de afinamiento se realizaron después de colocar los componentes y dispositivos seleccionados en la tarjeta electrónica, después de haber realizado el cableado en la estructura y de realizar el montaje de la PCB con los drivers, al realizar las pruebas surgieron algunos inconvenientes que pudieron ser corregidos, el primer error que se presentó fue en el estampado debido a la forma de la goma en la racle, para mejorarlo se optó por dar una forma redonda a la goma, el segundo error ocurrió porque los cilindros para realizar el estampado y la recolección de tinta sus vástagos salían con demasiada fuerza, se procedió a colocar unos estranguladores para regular la fuerza de salida de los vástagos de los cilindros.

6. Los resultados obtenidos gracias a las pruebas fueron muy alentadores debido a que se pudo realizar los estampados con la misma calidad que de la forma tradicional, se pudo conseguir una disminución en el tiempo de trabajo, ya que este puede ser utilizado en realizar otras actividades de la microempresa y gracias a esto se pudo generar un aumento en la producción.

6.2 RECOMENDACIONES

1. Al llevar acabo el montaje de la tarjeta electrónica y los drivers utilizados en la estructura mecánica realizar una verificación de la alimentación tanto de la fuente de alimentación, el driver de tensión como del cargador utilizado en la placa para un correcto funcionamiento de los componentes y de esta manera evitar posibles daños que se puedan presentar por la mala alimentación.
2. En cuanto a la estructura del brazo mecánico, una mejora a tomar en cuenta es la posibilidad de implementar un mecanismo para regular la posición de los sensores de contacto y así lograr adaptabilidad para estampados de mayor longitud. También se podría desarrollar un mecanismo automatizado que nos permita colocar o distribuir la tinta plastisol sobre la malla mejorando así la producción de los estampados.
3. Para la fase de impregnación de tinta se puede colocar o implementar un mecanismo automatizado que nos permita realizar un pre secado de los estampados y de esta manera optimizar la calidad del producto final.
4. Se podría desarrollar un sistema de enrollado automático para los rollos de tafilete que se encuentren estampados, debido a que el operador este proceso lo realiza de forma manual, esta implementación implicaría ahorro de tiempo mejorando la producción.
5. En la interfaz gráfica implementada se podría mejorar desarrollando una nueva página dentro de la cual se podrá seleccionar nuevos botones con opciones de tallas no comunes, actualmente el sistema cuenta con opciones de selección de las tallas más solicitadas a la microempresa.
6. La automatización del sistema se puede mejorar implementando una batería externa de 9 VDC, esta sería de utilidad en cuanto se presente una falta de energía eléctrica en la red o ante una falta de alimentación a la tarjeta electrónica. Además utilizar un dispositivo de sensado para comprobar la alimentación de los drivers de los motores utilizados, evitando posibles conflictos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] P. Ponsa, “Diseño y automatización industrial,” Universidad Politécnica de Cataluña, Cataluña, 2010.
- [2] Aldakin, “Automatización Industrial y Robótica.” [Online]. Available: <http://www.aldakin.com/automatizacion-industrial-robotica-claves-exito/>. [Accessed: 04-Dec-2020].
- [3] J. Machado, ““Automatización de los procesos Productivos en la planta II División Partes y Piezas para la Empresa Indurama S.A.”” Universidad de Cuenca, Cuenca, 2016.
- [4] G. J. González, O. Asesorado, I. J. César, and C. Paiz, “Propuesta de diseño de un mecanismo automatizado de serigrafía para la empresa serigrafía dinámica,” Universidad de San Carlos de Guatemala, 2014.
- [5] J. Mor, “La serigrafía se adapta a los nuevos tiempos - Industria Gráfica,” 2013. [Online]. Available: <http://www.interempresas.net/Graficas/Articulos/115661-La-serigrafia-se-adapta-a-los-nuevos-tiempos.html>. [Accessed: 20-Jun-2019].
- [6] R. Córdova, “Diseño y construcción de un prototipo de una estación de máquina serigráfica textil automatizada para la empresa ortega de la barra,” ESPE, 2018.
- [7] A. Gallegos, “Estampado en textiles naturales en el ecuador,” Pontificia Universidad Católica del Ecuador, 2012.
- [8] A. E. Avila Illescas, “Diseño y construcción para la automatización de un sistema de serigrafía,” Universidad del Azuay, 2015.
- [9] D. Vera and J. William, “Sistema de automatización para la máquina tubetera de cartón,” Universidad Católica de Santiago de Guayaquil, 2015.
- [10] D. Ibrahim, “Chapter 5 - Simple PIC18 Projects BT - PIC Microcontroller Projects in C (Second Edition),” Newnes, 2014, pp. 167–171.
- [11] J. Mariela and T. Guallichico, “Diseño e implementación de un prototipo para un sistema electrónico temporizado en puertas y ventanas, empleando módulos Arduino, para el restaurante ‘El Pailón de Mushuñan’..,” Escuela Politécnica Nacional, 2017.
- [12] Siemens, “LOGO! - El Futuro de la Industria - Siemens.” [Online]. Available: https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/controladores_modulares/logo/pages/default.aspx. [Accessed: 08-Nov-2019].

- [13] E. Rousu, “¡Sistema de automatización de viviendas unifamiliares implementado con el logotipo de Siemens! 8,” Universidad de Ciencias Aplicadas de Jyväskylä, 2018.
- [14] Ing. Punzenberger_COPA-DATA, “Interfaz humano-máquina.” [Online]. Available: <https://www.copadata.com/es/productos/zenon-software-platform/visualizacion-control/que-significa-hmi-interfaz-humano-maquina-copa-data/>. [Accessed: 03-Dec-2020].
- [15] Nextion, “Nextion Manual,” 2019. [Online]. Available: <https://nextion.tech/>. [Accessed: 07-Nov-2019].
- [16] Kinco, “Kinco: proveedores líderes de soluciones de automatización de máquinas.” [Online]. Available: <https://en.kinco.cn/productById?kind=1&id=91>. [Accessed: 10-Nov-2019].
- [17] HY-DIV268N-5A, “HY-DIV268N-5A two phase hybrid stepper motor drive manual.” [Online]. Available: <https://datasheetspdf.com/datasheet/HY-DIV268N-5A.html>. [Accessed: 04-Dec-2020].
- [18] Alldatasheet, “Datasheet-TB6560 3Axis Stepper Motor Driver.” [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1180443/ETC/TB6560.html>. [Accessed: 03-Dec-2020].
- [19] C. Santiana, “Diseño e implementación de un módulo de entrenamiento con comunicación ethernet para el laboratorio de control y máquinas eléctricas AC,” Escuela Superior Politécnica de Chimborazo, 2012.
- [20] R. Vivanco, “Diseño e implementación de una máquina CNC laser para la impresión de plantillas de circuitos SMD,” Universidad Católica de Santiago de Guayaquil, Guayaquil, 2019.
- [21] M. Escalera and A. Rodríguez, “Actuadores Neumáticos,” 2015. [Online]. Available: <https://docplayer.es/331358-Actuadores-neumaticos.html>. [Accessed: 07-Nov-2019].
- [22] O. Chicaiza, “Construcción de un sistema de serigrafía para el estampado de sellos utilizando electroválvulas,” Instituto Tecnológico Superior Vida Nueva, Quito, 2019.

ANEXOS

ANEXO A: DIMENSIONES DE LA ESTRUCTURA MECÁNICA

En las siguientes figuras se podrá observar las medidas de cada una de las partes que conforman la estructura mecánica.

En la figura A.1 se puede observar las medidas de la estructura utilizada para realizar el tensado de la cinta.

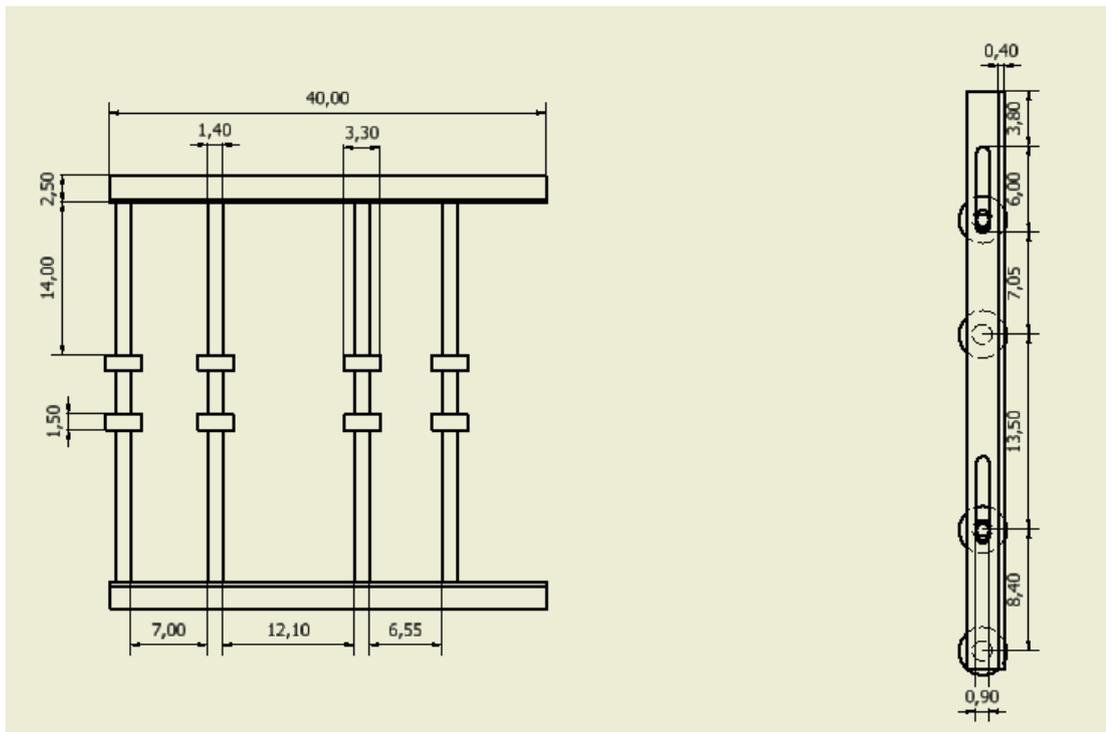


Figura A. 1 Medidas de la estructura de tensado de cinta

Fuente: Autor

En la figura A.2 se puede observar las medidas del brazo robótico utilizado para realizar el estampado, también se puede apreciar las medidas de los cilindros, la racle y el recogedor

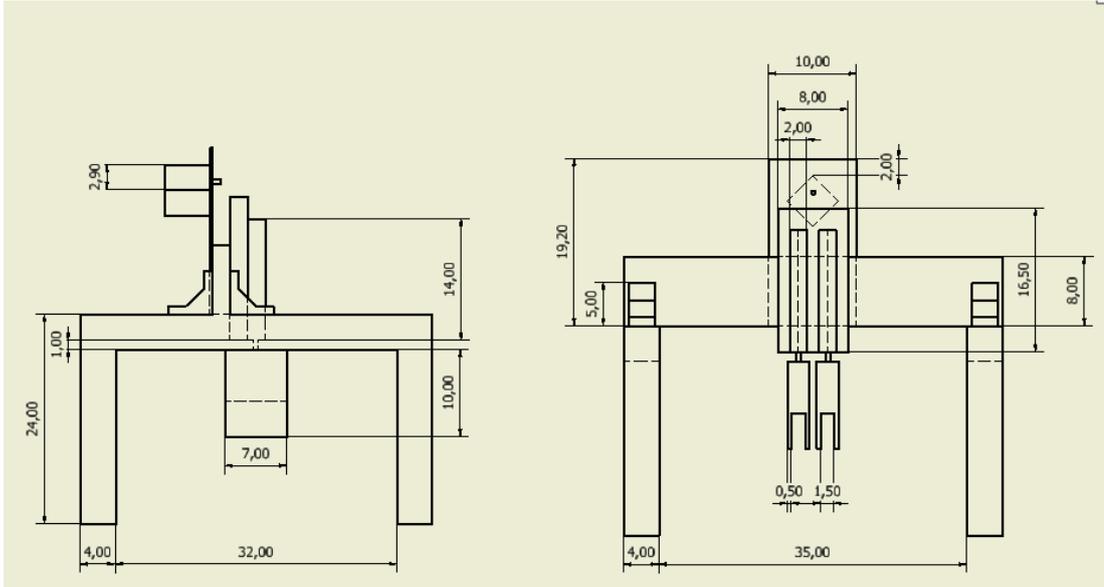


Figura A. 2 Medidas del brazo robótico

Fuente: Autor

En las figuras A.3 y A4 se puede observar las medidas de la sección de traslado de cinta en cual abarca el área para el estampado en donde va a estar el brazo robótico, también se aprecia las medidas de el alto de la estructura como el ancho y también de los ejes.

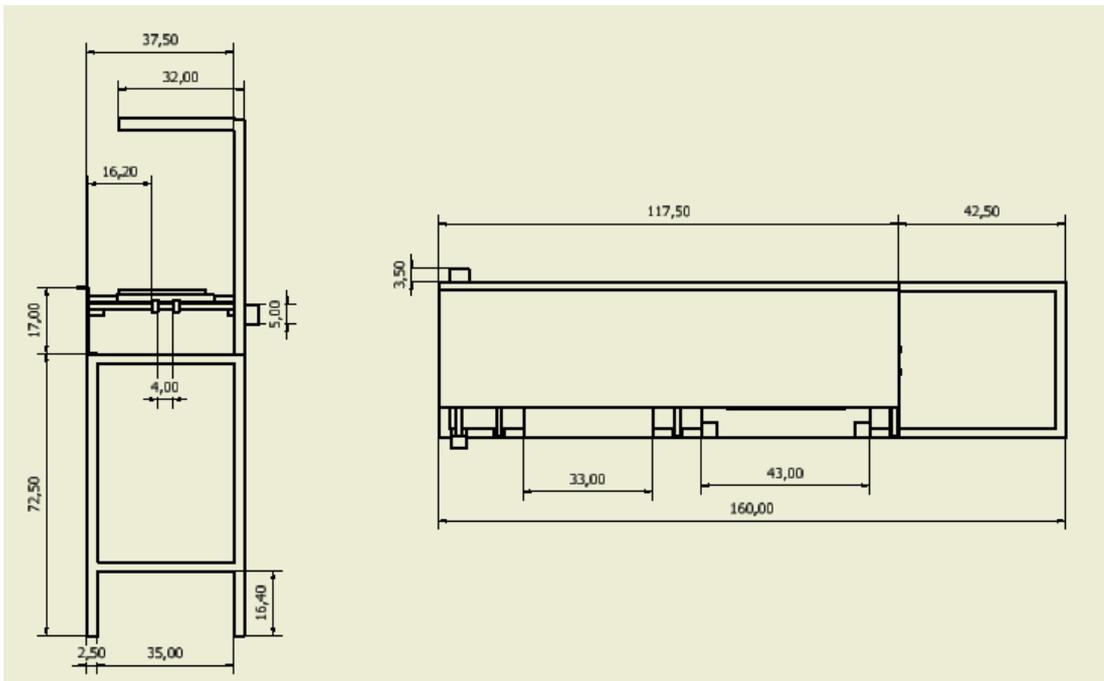


Figura A. 3 Medidas sección traslado de cinta

Fuente: Autor

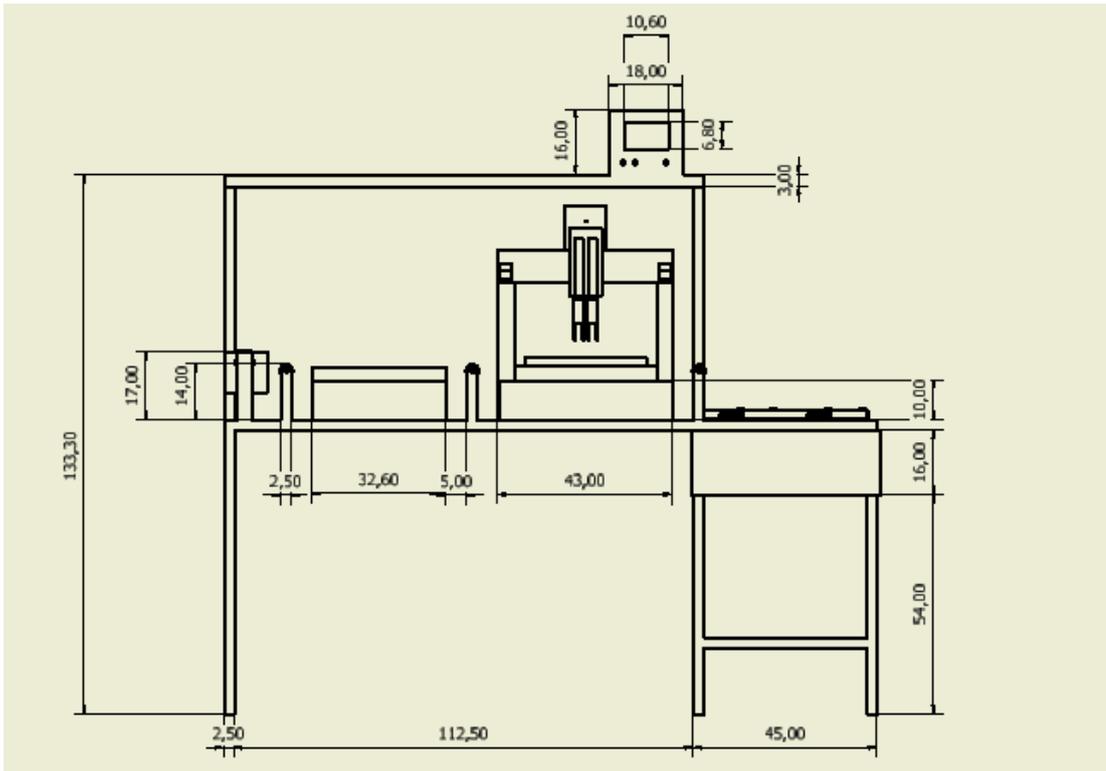


Figura A. 4 Medidas sección traslado de cinta

Fuente: Autor

ANEXO B: ARMADO DE LA TARJETA ELECTRÓNICA

En esta sección se podrá observar el armado y el soldado de cada uno de los componentes que se utilizaron para la tarjeta electrónica.

En la figura B.1 se observa el soldado de los componentes de la tarjeta electrónica.

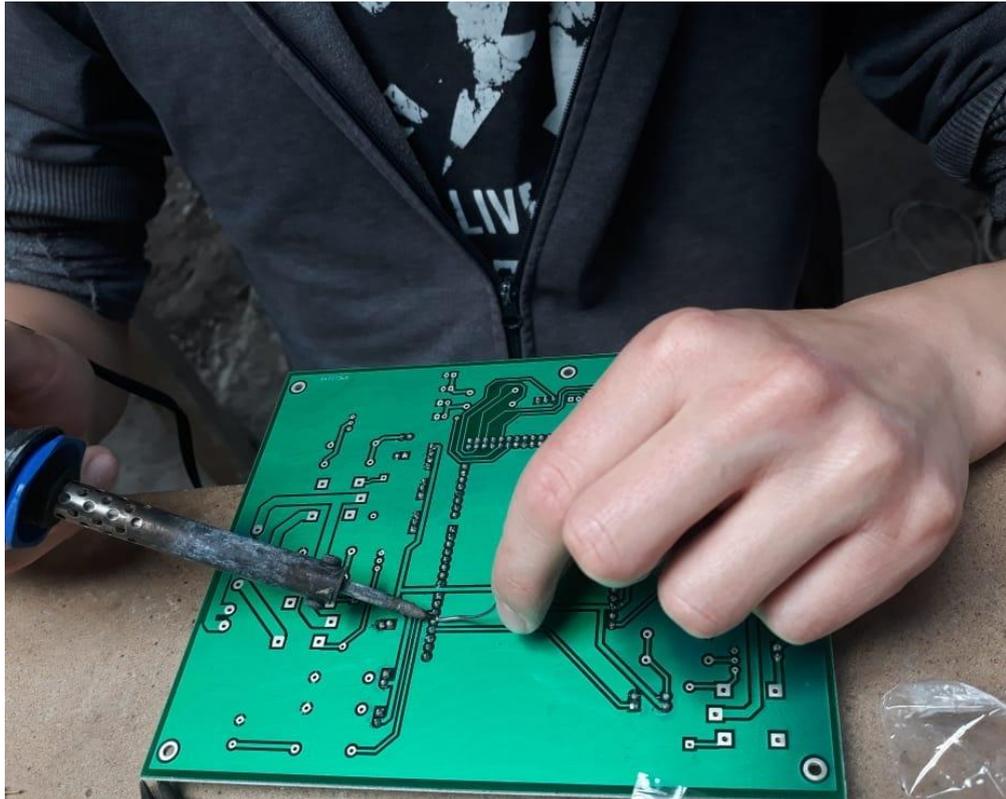


Figura B. 1 Soldado de los componentes en la tarjeta electrónica

Fuente: Autor

En la figura B.2 se puede observar los componentes que se soldaron en la tarjeta electrónica.

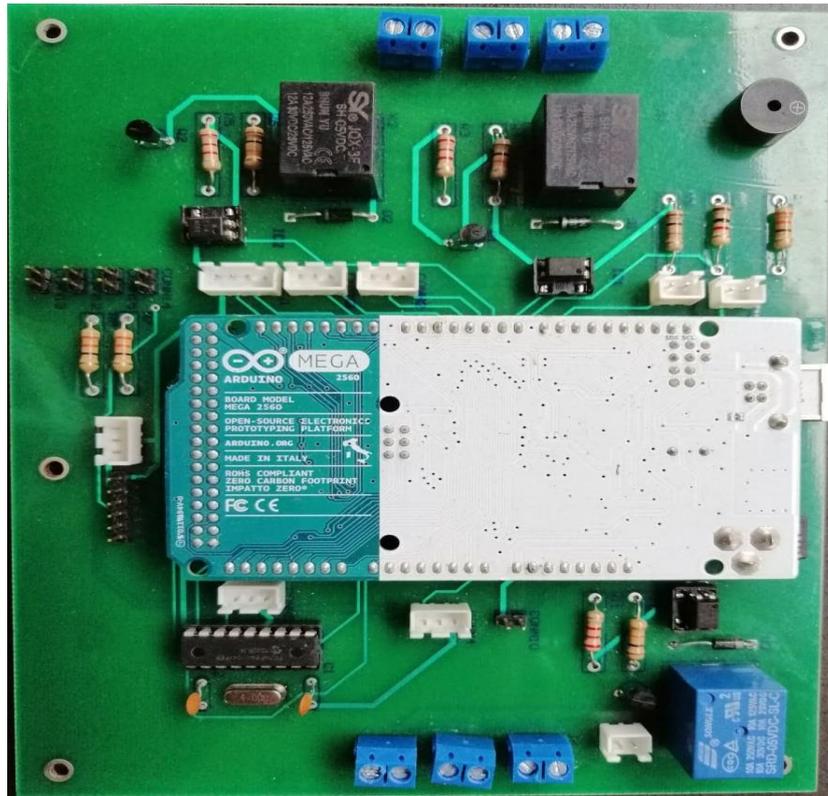


Figura B. 2 Armado de los componentes en la tarjeta electrónica

Fuente: Autor

En la figura B.3 se observa el montaje final de la tarjeta electrónica y de los dispositivos que conforman el sistema de control.

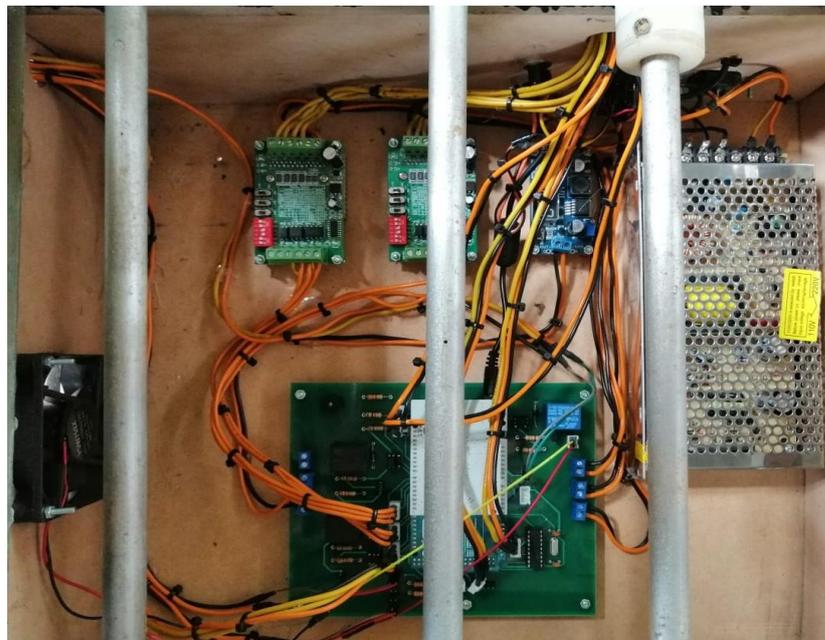


Figura B. 3 Montaje de la tarjeta electrónica y sus dispositivos.

Fuente: Autor

ANEXO C: PRUEBAS DE LA ESTRUCTURA MECÁNICA

En esta sección se mostrara la estructura mecánica finalizada con todos sus elementos, el diseño utilizado para el estampado, el manejo de la estructura mecánica y las diferentes medidas de los tafiletos.

Se puede observar en la figura C.1 la estructura mecánica completa lista para realizar pruebas.



Figura C. 1 Estructura mecánica

Fuente: Autor

Se puede observar en las figuras C.2 y C.3 el manejo de la estructura mecánica por parte de los estudiantes encargados en su desarrollo.

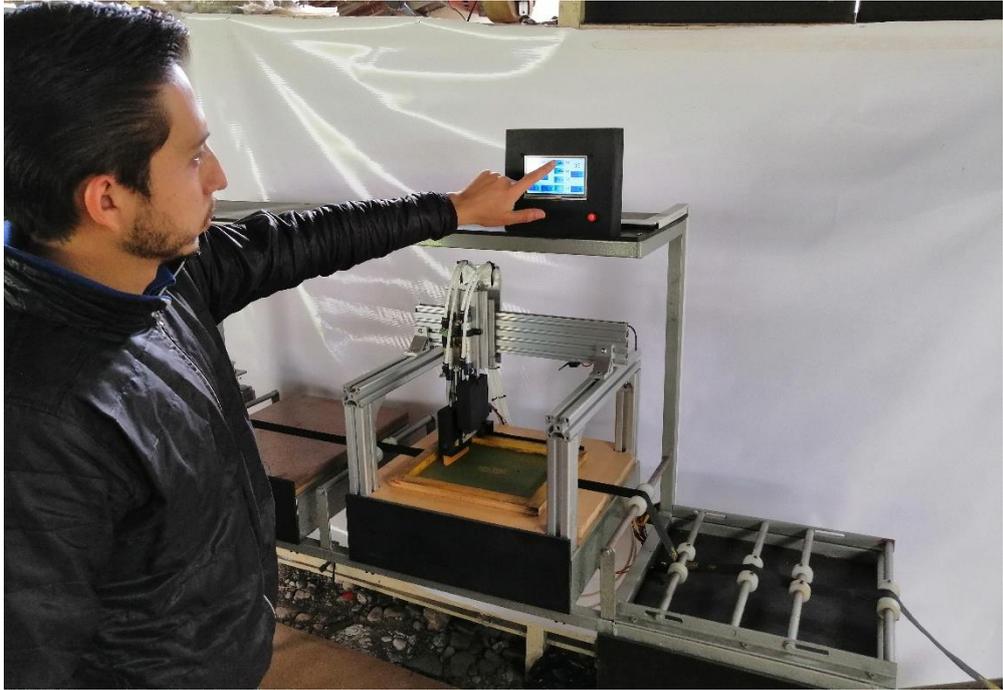


Figura C. 2 Manejo de la estructura mecánica

Fuente: Autor

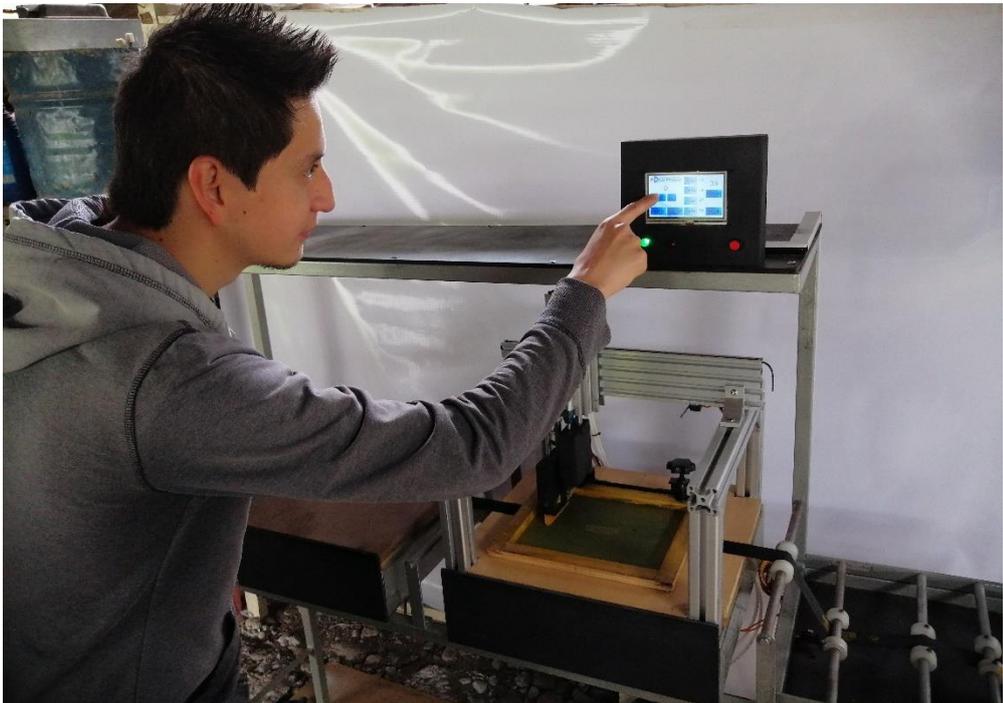


Figura C. 3 Manejo de la estructura mecánica

Fuente: Autor

La figura C.4 se observa tanto el marco que se utilizó como el diseño en la seda que se empleó para realizar los estampados en los tafletes.



Figura C. 4 Marco y diseño para el estampado

Fuente: Autor

Se puede observar en la figura C.5 el diseño estampado en el taflete así como también se observa la distancia de la cinta en este caso 54 cm.



Figura C. 5 Estampado en taflete para 54 cm

Se puede observar en la figura C.6 el diseño estampado en el taflete así como también se observa la distancia de la cinta en este caso 58 cm.



Figura C. 6 Estampado en taflete para 58 cm

Fuente: Autor

Se puede observar en la figura C.7 el diseño estampado en el taflete así como también se observa la distancia de la cinta en este caso 62 cm.



Figura C. 7 Estampado en taflete para 62 cm

Fuente: Autor

Se puede observar en la figura C.8 el diseño estampado en el taflete así como también se observa la distancia de la cinta en este caso 66 cm.



Figura C. 8 Estampado en taflete para 66 cm

Fuente: Autor

ANEXO D: TABLA CÓDIGO DE ERRORES

En esta sección se podrá observar una tabla con los códigos de los errores y su respectivo mensaje que se utilizaron para los avisos de fallas que se pueden visualizar en la pantalla.

Tabla D. 1 Código de Errores

Fuente: Autor

CÓDIGO DE ERRORES		
N°	Código	Mensaje
1	JOMICE001	Aviso cinta agotada
2	JOMICE002	Aviso Revisar Brazo para el Estampado
3	JOMICE003	Aviso Falla Motor de Traslado de Cinta
4	JOMICE004	Aviso Revisar Electroválvulas, Cilindros o Compresor

ANEXO E: PROGRAMACIÓN DE LOS MICROCONTROLADORES

En esta sección se podrá observar la programación realizada en el software Arduino IDE para el Arduino Mega 2560 y también la programación realizada en el software Mickro C pro for PIC para el pic16f84a.

PROHRAMACION PARA EL ARDUINO MEGA 2560

```
//////LOBRERIAS IMPLEMENTADAS////////////////////////////////
```

```
#include "TimerThree.h" //libreria timer
```

```
#include "Nextion.h" //libreria nextion para arduino
```

```
#include "NexButton.h" //libreria nextion para arduino
```

```
#include "EEPROM.h" //libreria memoria
```

```
//////DECLARACION DE LOS PINES////////////////////////////////
```

```
#define pos 2
```

```
#define vel 3
```

```
#define pos2 4
```

```
#define vel2 5
```

```
#define rele 6
```

```
#define rele1 7
```

```
#define finc 8
```

```
#define finc1 9
```

```
#define buzzer 10
```

```
#define hall 11
```

```
#define ventilador 12
```

```
#define led 22
```

```
#define led1 23
```

```
#define auxp 24
```

```
#define auxp1 25
```

```
#define auxp2 26
```

```
#define auxp3 27
```

```

#define reed 28

#define stepsPerRevolution 1600

////DECLARACION DE LAS VARIABLES A UTILIZAR////////

int aux=0, aux1=0, aux2=0, aux3=0, segundo=0, segundo1=0, segundo2=0,
segundo3=0; ///variables y auxiliares para los tiempos de cada talla

int auxbmt54=0, aux54=0, auxbmt58=0, aux58=0, auxbmt62=0, aux62=0,
auxbmt66=0, aux66=0; ///auxiliares para funcionamiento de cada talla

int auxec=0, auxec1=0, auxem=0, auxem1=0, auxemc=0, auxf=0, auxf1=0, auxfc1=0,
auxhall=0, auxhall1=0, auxhall2=0, auxeca=0, auxmb=0, auxbf=0, auxstop=0;
///auxiliares para mensajes de error

int auxesa=0,auxesa1=0,auxreed=0, auxhall3=0, auxreed1=0, segreed1=0, segreed=0;

int segf=0, auxem2=0, segm=0, segmfc1=0, seghall=0, seghall1=0, seghall2=0,
seghall3=0, segbf=0, segstop=0; ///auxiliares para los tiempos de mensaje de error

int number=0, val1=0; ///variables para cantidad de estampados

uint32_t valt54=0, valt58=0, valt62=0, valt66=0; ///variables para obtener datos de la
pantalla

byte val=0; ///variable para guardado de datos en memoria eeprom

////////DECLARACION DE LOS COMPONENTES DE LA PANTALLA//////////

////////DECLARACION DE NUMBERS//////////

NexNumber n1 = NexNumber(2, 9, "n1"); //contador de estampados

NexNumber n2 = NexNumber(2, 15, "n2"); //contador de estampados

////////DECLARACION DE VARIABLES PRA LOS MENSAJES DE
ERROR//////////

NexVariable va0 = NexVariable (1, 3, "va0"); //llamar página aviso conta agotada

NexVariable va1 = NexVariable (1, 4, "va1"); //llamar página aviso revisar brazo
estampado

NexVariable va2 = NexVariable (2, 18, "va2"); //llamar página aviso conta agotada

NexVariable va3 = NexVariable (2, 19, "va3"); //llamar página aviso revisar brazo
estampado

NexVariable va4 = NexVariable (2, 20, "va4"); //llamar página aviso falla motor
traslado cinta

NexVariable va5 = NexVariable (2, 21, "va5"); //llamar página aviso falla cilindros

////////DECLARACION DE LOS BOTONES//////////

```

```

NexButton b0 = NexButton(1, 1, "b0"); // botón menú
NexButton b1 = NexButton(2, 1, "b1"); // botón talla 54
NexButton b2 = NexButton(2, 2, "b2"); // botón talla 58
NexButton b3 = NexButton(2, 3, "b3"); // botón talla 62
NexButton b4 = NexButton(2, 4, "b4"); // botón talla 66
NexButton b5 = NexButton(2, 11, "b5"); // botón +
NexButton b6 = NexButton(2, 10, "b6"); // botón -
NexButton b7 = NexButton(2, 13, "b7"); // botón pause
NexButton b8 = NexButton(2, 14, "b8"); // botón stop
NexButton b9 = NexButton(2, 16, "b9"); // botón reset
NexButton b10 = NexButton(2, 12, "b10"); // botón back

```

```
char buffer[100] = {0};
```

```
NexTouch *nex_listen_list[] =
```

```

{
    &b0, &b1, &b2, &b3, &b4, &b5, &b6, &b7, &b8, &b9, &b10, ///declaración de los
    botones
    NULL
};

```

```
////////FUNCIONES DE CADA UNO DE LOS BOTONES//////////
```

```
void b0PopCallback(void *ptr)    /// botón menú
```

```

{
    val = EEPROM.read(0);
    n2.setValue(val);
    auxf1=1;
    val1=0;
    n1.setValue(val1);
    auxmb=1;

```

```

segm=0;
}
void b1PopCallback(void *ptr)    ///botón talla 54
{
aux=1;
aux1=0;
aux2=0;
aux3=0;
auxt54=1;
auxt58=0;
auxt62=0;
auxt66=0;
auxbmt58=0;
auxbmt62=0;
auxbmt66=0;
auxhall=0;
auxhall1=0;
auxhall2=0;
auxhall3=1;
auxreed=1;
segundo1=0;
segundo2=0;
segundo3=0;
auxf1=0;
segf=0;
auxbf=1;
auxstop=0;
digitalWrite(auxp, LOW);
digitalWrite(auxp1, LOW);

```

```
digitalWrite(auxp2, LOW);  
digitalWrite(auxp3,HIGH);  
}
```

```
void b2PopCallback(void *ptr)    /// botón talla 58  
{  
  aux=0;  
  aux1=1;  
  aux2=0;  
  aux3=0;  
  aux54=0;  
  aux58=1;  
  aux62=0;  
  aux66=0;  
  auxbmt54=0;  
  auxbmt62=0;  
  auxbmt66=0;  
  auxhall=1;  
  auxhall1=0;  
  auxhall2=0;  
  auxhall3=0;  
  auxreed=1;  
  segundo=0;  
  segundo2=0;  
  segundo3=0;  
  auxf1=0;  
  segf=0;  
  auxbf=1;
```

```
auxstop=0;
digitalWrite(auxp, HIGH);
digitalWrite(auxp1, LOW);
digitalWrite(auxp2, LOW);
digitalWrite(auxp3,LOW);
}
```

```
void b3PopCallback(void *ptr)  /// botón talla 62
{
aux=0;
aux1=0;
aux2=1;
aux3=0;
auxt54=0;
auxt58=0;
auxt62=1;
auxt66=0;
auxbmt54=0;
auxbmt58=0;
auxbmt66=0;
auxhall=0;
auxhall1=1;
auxhall2=0;
auxhall3=0;
auxreed=0;
auxreed1=1;
segundo=0;
segundo1=0;
```

```
segundo3=0;
auxf1=0;
segf=0;
auxbf=1;
auxstop=0;
digitalWrite(auxp, LOW);
digitalWrite(auxp1, HIGH);
digitalWrite(auxp2, LOW);
digitalWrite(auxp3,LOW);
}
```

```
void b4PopCallback(void *ptr)    /// botón talla 66
```

```
{
aux=0;
aux1=0;
aux2=0;
aux3=1;
auxt54=0;
auxt58=0;
auxt62=0;
auxt66=1;
auxbmt54=0;
auxbmt58=0;
auxbmt62=0;
auxhall=0;
auxhall1=0;
auxhall2=1;
auxhall3=0;
auxreed=0;
```

```

auxreed1=1;
segundo=0;
segundo1=0;
segundo2=0;
auxf1=0;
segf=0;
auxbf=1;
auxstop=0;
digitalWrite(auxp, LOW);
digitalWrite(auxp1, LOW);
digitalWrite(auxp2, HIGH);
digitalWrite(auxp3,LOW);
}

```

```

void b5PopCallback(void *ptr) { //botón +
    auxbmt54=1;
    auxbmt58=1;
    auxbmt62=1;
    auxbmt66=1;
    number += 1;
    n1.setValue(number);
}

```

```

void b6PopCallback(void *ptr){ //botón -
    auxbmt54=1;
    auxbmt58=1;
    auxbmt62=1;
    auxbmt66=1;
    number -= 1;
}

```

```

n1.setValue(number);
}

void b7PopCallback(void *ptr)    /// botón pause
{
aux=0;
aux1=0;
aux2=0;
aux3=0;
}

void b8PopCallback(void *ptr)    /// botón stop
{
aux=0;
aux1=0;
aux2=0;
aux3=0;
auxt54=0;
auxt58=0;
auxt62=0;
auxt66=0;
auxbmt54=0;
auxbmt58=0;
auxbmt62=0;
auxbmt66=0;

auxreed=0;
auxreed1=0;
segundo=0;
segundo1=0;

```

```

segundo2=0;
segundo3=0;
auxf1=1;
val1=0;
n1.setValue(val1);
auxbf=0;
auxstop=1;
digitalWrite(rele, LOW);
digitalWrite(rele1, LOW);
}
void b9PopCallback(void *ptr)    /// botón reset
{
aux=0;
aux1=0;
aux2=0;
aux3=0;
auxt54=0;
auxt58=0;
auxt62=0;
auxt66=0;
auxbmt54=0;
auxbmt58=0;
auxbmt62=0;
auxbmt66=0;
segundo=0;
segundo1=0;
segundo2=0;
segundo3=0;
auxf1=1;

```

```

val1=0;

n1.setValue(val1);

val=0;

n2.setValue(val);

digitalWrite(rele, LOW);

digitalWrite(rele1, LOW);

}

void b10PopCallback(void *ptr)    /// botón back

{

aux=0;

aux1=0;

aux2=0;

aux3=0;

auxt54=0;

auxt58=0;

auxt62=0;

auxt66=0;

auxbmt54=0;

auxbmt58=0;

auxbmt62=0;

auxbmt66=0;

segundo=0;

segundo1=0;

segundo2=0;

segundo3=0;

auxf1=1;

val1=0;

n1.setValue(val1);

auxbf=0;

```

```

digitalWrite(rele, LOW);
digitalWrite(rele1, LOW);
}
void setup() {

nexInit();//inicializacion de la pantalla
Serial.begin (115200);

/////LLAMADO DE CADA UNA DE LAS FUNCIONES DE LOS BOTONES//////////
b0.attachPop(b0PopCallback, &b0); //botón menú
b1.attachPop(b1PopCallback, &b1); //botón talla 54
b2.attachPop(b2PopCallback, &b2); //botón talla 58
b3.attachPop(b3PopCallback, &b3); //botón talla 62
b4.attachPop(b4PopCallback, &b4); //botón talla 66
b5.attachPop(b5PopCallback, &b5); //botón +
b6.attachPop(b6PopCallback, &b6); //botón -
b7.attachPop(b7PopCallback, &b7); //botón pause
b8.attachPop(b8PopCallback, &b8); //botón stop
b9.attachPop(b9PopCallback, &b9); //botón reset
b10.attachPop(b9PopCallback, &b10); //botón back

////////DECLARACIÓN DE PINES COMO ENTRADA O SALIDA//////////
pinMode(pos, OUTPUT); // Es el Giro, en Bajo “0” lógico Sentido Horario y “1”
Anti Horario.
pinMode(vel, OUTPUT); // Velocidad de Operación, Controlada por Pulsos.
pinMode(pos2, OUTPUT); // Es el Giro, en Bajo “0” lógico Sentido Horario y “1”
Anti Horario.
pinMode(vel2, OUTPUT); // Velocidad de Operación, Controlada por Pulsos.
pinMode(rele, OUTPUT); // salida para la activación de las electrovalvulas.
pinMode(rele1, OUTPUT); // salida para la activación de las electrovalvulas.
pinMode(finc, INPUT); // entrada para los sensores de contacto

```

```

pinMode(finc1, INPUT); //entrada para los sensores de contacto
pinMode(buzzer, OUTPUT); //salida para la alarma
pinMode(hall, INPUT); //entrada para el sensor de aviso de traslado de cinta
pinMode(ventilador, OUTPUT); //salida para la ventilación
pinMode(led, OUTPUT); //led indicador de encendido
pinMode(led1, OUTPUT); //led indicador para errores
pinMode(auxp, OUTPUT); //auxiliar para el pic
pinMode(auxp1, OUTPUT); //auxiliar para el pic
pinMode(auxp2, OUTPUT); //auxiliar para el pic
pinMode(auxp3, OUTPUT); //auxiliar para el pic
pinMode(reed, INPUT_PULLUP); //entrada para aviso de cilindros
/////INICIALIZACION DE LA LIBRERIA TIMMER3////////
Timer3.initialize(1000000);/////tiempo en segundos
Timer3.attachInterrupt(Timmer3);/////funcion timmer3
}
/////DECLARACIÓN DE LA FUNCIÓN PARA LOS TEMPORIZADORES////////
void Timmer3()
{
  if(aux==1){ //auxiliar tiempo talla 54
    segundo++;
  }
  if(aux1==1){ //auxiliar tiempo talla 58
    segundo1++;
  }

  if(aux2==1){ //auxiliar tiempo talla 62
    segundo2++;
  }
}

```

```

if(aux3==1){ //auxiliar tiempo talla 66
    segundo3++;
}
if(auxf==1){ //auxiliar tiempo brazo
    segf++;
}
if(auxem2==1){ //auxiliar error brazo
    segm++;
}
if(auxfc1==1){ //auxiliar error brazo
    segmfc1++;
}
if(auxhall==1){ //auxiliar error traslado de cinta
    seghall++;
}
if(auxhall1==1){ //auxiliar error traslado de cinta
    seghall1++;
}
if(auxhall2==1){ //auxiliar error traslado de cinta
    seghall2++;
}
if(auxhall3==1){ //auxiliar error traslado de cinta
    seghall3++;
}
if(auxreed==1){ //auxiliar error cilindros
    segreed++;
}
if(auxreed1==1){ //auxiliar error cilindros
    segreed1++;
}

```

```

}
if(auxbf==1){
    segbf++;
}
if(auxstop==1){
    segstop++;
}
}

void loop() {

digitalWrite(ventilador, HIGH);//encendido del ventilador
digitalWrite(led, HIGH); //encendido led correcto funcionamiento
digitalWrite(led1, LOW); //apagado led advertencia
nexLoop(nex_listen_list); //llamado de la lista de las variables de la pantalla

////////TALLA 54//////////
if(auxbmt54==1 && aux54==1){
    auxf=0;
    segf=0;

    n1.getValue(&valt54); //se obtiene el valor de la pantalla del número n1 y se guarda
    en valt54

    if(segundo==1){
        digitalWrite(rele, HIGH);
    }

    if(segundo==2){
        digitalWrite(pos, LOW); // Sentido Horario
        for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
            // las cuatro lineas dan como resultado un paso:

```

```

digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo==3){
    digitalWrite(rele, LOW);
}
if(segundo==4){
    digitalWrite(pos2, HIGH); // Sentido Antihorario
    for (int i = 0; i < 5.43 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
        digitalWrite(vel2, HIGH);
        delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
        digitalWrite(vel2, LOW);
        delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo==7){
    digitalWrite(rele1, HIGH);
}
if(segundo==8){
    digitalWrite(pos, HIGH); // Sentido Antihorario
    for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
        digitalWrite(vel, HIGH);
        delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
        digitalWrite(vel, LOW);

```

```

delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo==9){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo==9){
    val+=1;
    val1+=1;
    n2.setValue(val); //se envia el numero de estampados al numer 2 de la pantalla
    segundo=0;
}
EEPROM.update(0, val); // se guarda el numero de estampados en la EEPROM del
arduino
if(val1==valt54){
    val1=0;
    n1.setValue(val1);
    aux=0;
    segundo=0;
    auxbmt54=0;
    aux54=0;
    auxbf=0;
}
}
else if(auxbmt54==0 && aux54==1){
    auxf=0;
    segf=0;
    if(segundo==1){
        digitalWrite(rele, HIGH);
    }
}

```

```

    if(segundo==2){
digitalWrite(pos, LOW); // Sentido Horario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
    }
}
if(segundo==3){
digitalWrite(rele, LOW);
}
if(segundo==4){
digitalWrite(pos2, HIGH); // Sentido Antihorario
for (int i = 0; i < 5.43 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo==7){
digitalWrite(rele1, HIGH);
}
if(segundo==8){
digitalWrite(pos, HIGH); // Sentido Antihorario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

// las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo==9){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo==9){
    val+=1;
    n2.setValue(val);
    segundo=0;
}
EEPROM.update(0, val);
}
////////////////////////////////////
////////TALLA 58////////////////////////////////
if(auxbmt58==1 && auxt58==1){
    auxf=0;
    segf=0;
    n1.getValue(&valt58);
    if(segundo1==1){
        digitalWrite(rele, HIGH);
    }
    if(segundo1==2){
        digitalWrite(pos, LOW); // Sentido Horario
        for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo1==3){
    digitalWrite(rele, LOW);
}
if(segundo1==4){
    digitalWrite(pos2, HIGH); // Sentido Antihorario
    for (int i = 0; i < 5.83 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo1==7){
    digitalWrite(rele1, HIGH);
}
if(segundo1==8){
    digitalWrite(pos, HIGH); // Sentido Antihorario
    for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos

```

```

digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo1==9){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo1==9){
    val+=1;
    val1+=1;
    n2.setValue(val);
    segundo1=0;
}
EEPROM.update(0, val);
if(val1==valt58){
    val1=0;
    n1.setValue(val1);
    aux1=0;
    segundo1=0;
    auxbmt58=0;
    aux58=0;
    auxbf=0;
}
}else if(auxbmt58==0 && aux58==1){
    auxf=0;
    segf=0;
    if(segundo1==1){
        digitalWrite(rele, HIGH);
    }
}

```

```

    if(segundo1==2){
digitalWrite(pos, LOW); // Sentido Horario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
    }
}
if(segundo1==3){
digitalWrite(rele, LOW);
}
if(segundo1==4){
digitalWrite(pos2, HIGH); // Sentido Antihorario
for (int i = 0; i < 5.83 * stepsPerRevolution; i++) {
    // These four lines result in 1 step:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo1==7){
digitalWrite(rele1, HIGH);
}
if(segundo1==8){
digitalWrite(pos, HIGH); // Sentido Antihorario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

// las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo1==9){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo1==9){
    val+=1;
    n2.setValue(val);
    segundo1=0;
}
EEPROM.update(0, val);
}
////////////////////////////////////
////////TALLA 62////////////////////////////////
if(auxbmt62==1 && aux62==1){
    auxf=0;
    segf=0;
    n1.getValue(&valt62);
    if(segundo2==1){
        digitalWrite(rele, HIGH);
    }
    if(segundo2==2){
        digitalWrite(pos, LOW); // Sentido Horario
        for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo2==3){
    digitalWrite(rele, LOW);
}
if(segundo2==4){
    digitalWrite(pos2, HIGH); // Sentido Antihorario
    for (int i = 0; i < 6.23 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo2==8){
    digitalWrite(rele1, HIGH);
}
if(segundo2==9){
    digitalWrite(pos, HIGH); // Sentido Antihorario
    for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos

```

```

digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo2==10){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo2==10){
    val+=1;
    val1+=1;
    n2.setValue(val);
    segundo2=0;
}
EEPROM.update(0, val);
if(val1==valt62){
    val1=0;
    n1.setValue(val1);
    aux2=0;
    segundo2=0;
    auxbmt62=0;
    aux62=0;
    auxbf=0;
}
}else if(auxbmt62==0 && aux62==1){
    auxf=0;
    segf=0;
    if(segundo2==1){
        digitalWrite(rele, HIGH);
    }
}

```

```

    if(segundo2==2){
digitalWrite(pos, LOW); // Sentido Horario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
    }
}
if(segundo2==3){
digitalWrite(rele, LOW);
}
if(segundo2==4){
digitalWrite(pos2, HIGH); // Sentido Antihorario
for (int i = 0; i < 6.23 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo2==8){
digitalWrite(rele1, HIGH);
}
if(segundo2==9){
digitalWrite(pos, HIGH); // Sentido Antihorario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

// las cuatro lineas dan como resultado un paso::
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo2==10){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo2==10){
    val+=1;
    n2.setValue(val);
    segundo2=0;
}
EEPROM.update(0, val);
}
////////////////////////////////////
////////TALLA 66////////
if(auxbmt66==1 && aux66==1){
    auxf=0;
    segf=0;
    n1.getValue(&valt66);
    if(segundo3==1){
        digitalWrite(rele, HIGH);
    }
    if(segundo3==2){
        digitalWrite(pos, LOW); // Sentido Horario
        for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo3==3){
    digitalWrite(rele, LOW);
}
if(segundo3==4){
    digitalWrite(pos2, HIGH); // Sentido Antihorario
    for (int i = 0; i < 6.63 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo3==9){
    digitalWrite(rele1, HIGH);
}
if(segundo3==10){
    digitalWrite(pos, HIGH); // Sentido Antihorario
    for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos

```

```

digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo3==11){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo3==11){
    val+=1;
    val1+=1;
    n2.setValue(val);
    segundo3=0;
}
EEPROM.update(0, val);
if(val1==valt66){
    val1=0;
    n1.setValue(val1);
    aux3=0;
    segundo3=0;
    auxbmt66=0;
    aux66=0;
    auxbf=0;
}
}else if(auxbmt66==0 && aux66==1){
    auxf=0;
    segf=0;
    if(segundo3==1){
        digitalWrite(rele, HIGH);
    }
}

```

```

    if(segundo3==2){
digitalWrite(pos, LOW); // Sentido Horario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
    }
}
if(segundo3==3){
digitalWrite(rele, LOW);
}
if(segundo3==4){
digitalWrite(pos2, HIGH); // Sentido Antihorario
for (int i = 0; i < 6.63 * stepsPerRevolution; i++) {
    // las cuatro lineas dan como resultado un paso:
digitalWrite(vel2, HIGH);
delayMicroseconds(190); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel2, LOW);
delayMicroseconds(190); // delay de X milisegundos
    }
}
if(segundo3==9){
digitalWrite(rele1, HIGH);
}
if(segundo3==10){
digitalWrite(pos, HIGH); // Sentido Antihorario
for (int i = 0; i < 2.53 * stepsPerRevolution; i++) {

```

```

// las cuatro lineas dan como resultado un paso:
digitalWrite(vel, HIGH);
delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
digitalWrite(vel, LOW);
delayMicroseconds(150); // delay de X milisegundos
}
}
if(segundo3==11){
    digitalWrite(rele1, LOW);
}
if(digitalRead(rele1)== LOW && segundo3==11){
    val+=1;
    n2.setValue(val);
    segundo3=0;
}
EEPROM.update(0, val);
}
////////////////////////////////////
/////////BOTÓN STOP////////////////////////////////////
if((digitalRead(finc1)== HIGH) && (auxf1==1)){
    auxf1=1;
    if(segf==3){
        digitalWrite(pos, HIGH); // Sentido Horario
        for (int i = 0; i < 2.6 * stepsPerRevolution; i++) {
            // las cuatro lineas dan como resultado un paso:
            digitalWrite(vel, HIGH);
            delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
            digitalWrite(vel, LOW);
            delayMicroseconds(150); // delay de X milisegundos

```

```

    }
}
}
if(segstop>=2){
digitalWrite(auxp, LOW);
digitalWrite(auxp1, LOW);
digitalWrite(auxp2, LOW);
digitalWrite(auxp3, LOW);
auxhall=0;
auxhall1=0;
auxhall2=0;
auxhall3=0;
}

////////////////////////////////////
//////////////////ERROR TRASLADO CINTA//////////////////
if(((digitalRead(hall)== LOW) && seghall==6){
    auxemc=1;
    while(auxemc==1){
        if(auxemc==1){
            va4.setValue(auxemc); //llamar página error traslado de cinta
            digitalWrite(led, LOW); //apagar led de encendido
            digitalWrite(led1, HIGH); //encender led de advertencia
        }
        tone(buzzer, 246, 250);
        delay(1000);
        tone(buzzer, 520, 250);
        delay(2000);
    }
}
}

```

```

if((digitalRead(hall)== LOW) && seghall1==7){
    auxemc=1;
    while(auxemc==1){
        if(auxemc==1){
            va4.setValue(auxemc);//llamar página error traslado de cinta
            digitalWrite(led, LOW);
            digitalWrite(led1, HIGH);
        }
        tone(buzzer, 246, 250);
        delay(1000);
        tone(buzzer, 520, 250);
        delay(2000);
    }
}

if((digitalRead(hall)== LOW) && seghall2==8){
    auxemc=1;
    while(auxemc==1){
        if(auxemc==1){
            va4.setValue(auxemc); //llamar página error traslado de cinta
            digitalWrite(led, LOW);
            digitalWrite(led1, HIGH);
        }
        tone(buzzer, 246, 250);
        delay(1000);
        tone(buzzer, 520, 250);
        delay(2000);
    }
}

if((digitalRead(hall)== LOW) && seghall3==6){

```

```

    auxemc=1;
while(auxemc==1){
    if(auxemc==1){
va4.setValue(auxemc); //llamar página error traslado de cinta
digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
    }
    tone(buzzer, 246, 250);
    delay(1000);
    tone(buzzer, 520, 250);
    delay(2000);
    }
}

////////////////////////////////////
/////////ERROR MOTOR Y FIN CARRERAS DEL BRAZO////////////////////////////////////
if(digitalRead(finc)== LOW){
auxem2=1;
auxfc1=1;
if(digitalRead(finc1)== LOW && segm>=10){
auxem=1;
auxem1=1;
    while((auxem==1) || (auxem1==1)){
        if((auxem==1) || (auxem1==1)){
va1.setValue(auxem); //llamar página error brazo
va3.setValue(auxem1); //lamar página error brazo
digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
        }
        tone(buzzer, 246, 250);

```

```

delay(1000);
tone(buzzer, 520, 250);
delay(2000);
}
}

}
if(digitalRead(finc)== HIGH){
  segm=0;
  auxmb=0;
  segmfc1=0;
  seghall=0;
  seghall1=0;
  seghall2=0;
  seghall3=0;
  segreed=0;
  segreed1=0;
  segstop=0;
}
if(digitalRead(finc1)== HIGH){
  segm=0;
  segbf=0;
}
if(digitalRead(finc1)== HIGH && segmfc1>=11){
  auxem=1;
  auxem1=1;
  while((auxem==1) || (auxem1==1)){
    if((auxem==1) || (auxem1==1)){
      va1.setValue(auxem); //lamar página error brazo

```

```

va3.setValue(auxem1); //lamar página error brazo
digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
    }
tone(buzzer, 246, 250);
delay(1000);
tone(buzzer, 520, 250);
delay(2000);
    }
}
if(digitalRead(finc)== HIGH && segbf>=5){
auxem=1;
auxem1=1;
while((auxem==1) || (auxem1==1)){
    if((auxem==1) || (auxem1==1)){
va1.setValue(auxem); //lamar página error brazo
va3.setValue(auxem1); //lamar página error brazo
digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
        }
tone(buzzer, 246, 250);
delay(1000);
tone(buzzer, 520, 250);
delay(2000);
    }
}
Serial.println(auxbf);
Serial.println(segbf);

```

```

////////////////////////////////////
////BOTÓN MENÚ////////////////////////////////////
if(auxmb==1){
    digitalWrite(pos, HIGH); // Sentido Antihorario
    for (int i = 0; i < 2.6 * stepsPerRevolution; i++) {
        // las cuatro lineas dan como resultado un paso:
        digitalWrite(vel, HIGH);
        delayMicroseconds(150); // delay define la Velocidad o tiempo entre pasos
        digitalWrite(vel, LOW);
        delayMicroseconds(150); // delay de X milisegundos
    }
}

////////////////////////////////////
////////ERROR SIN CINTA////////////////////////////////////
int ADC_SHARP=ADC0_promedio(20);
if(ADC_SHARP<200)
{
    auxec=1;
    auxec1=1;
    while((auxec==1) || (auxec1==1)){
        if((auxec==1) || (auxec1==1)){
            va0.setValue(auxec); //llamar página error sin cinta
            va2.setValue(auxec1); //llamar página error sin cinta
            digitalWrite(led, LOW);
            digitalWrite(led1, HIGH);
        }
        tone(buzzer, 246, 250);
        delay(1000);
    }
}

```

```

tone(buzzer, 520, 250);
delay(2000);
}
}

////////////////////////////////////
//////////ERROR REVISAR CILINDROS////////////////////////////////////
if((digitalRead(reed)== HIGH)){
    segreed=0;
    segreed1=0;
}
if((digitalRead(reed)== LOW) && segreed==3){
    auxeca=1;
    while(auxeca==1){
        if(auxeca==1){
            va5.setValue(auxeca); //llamar página error cilindros
            digitalWrite(led, LOW);
            digitalWrite(led1, HIGH);
        }
        tone(buzzer, 246, 250);
        delay(1000);
        tone(buzzer, 520, 250);
        delay(2000);
    }
}
if((digitalRead(reed)== LOW) && segreed>=5){
    auxeca=1;
    while(auxeca==1){
        if(auxeca==1){

```

```

va5.setValue(auxeca); //llamar página error cilindros
digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
    }
tone(buzzer, 246, 250);
delay(1000);
tone(buzzer, 520, 250);
delay(2000);
    }
}

if((digitalRead(reed)== LOW) && segreed1==3){
    auxeca=1;
    while(auxeca==1){
        if(auxeca==1){
va5.setValue(auxeca); //llamar página error cilindros
digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
        }
        tone(buzzer, 246, 250);
        delay(1000);
        tone(buzzer, 520, 250);
        delay(2000);
        }
    }

if((digitalRead(reed)== LOW) && segreed1>=7){
    auxeca=1;
    while(auxeca==1){
        if(auxeca==1){
va5.setValue(auxeca); //llamar página error cilindros

```

```

digitalWrite(led, LOW);
digitalWrite(led1, HIGH);
    }
tone(buzzer, 246, 250);
delay(1000);
tone(buzzer, 520, 250);
delay(2000);
    }
}

}

/////////////////////////////////////////////////////////////////
//////////ERROR SIN CINTA//////////////////////////////////////
int ADC0_promedio(int n)
{
    long suma=0;
    for(int i=0;i<n;i++)
    {
        suma=suma+analogRead(A0);
    }
    return(suma/n);
}

/////////////////////////////////////////////////////////////////

```

PROGRAMACION PARA EL PIC16F84A

La programación realizada para el PIC se lo utilizo para detectar los cambios que presente el sensor hall colocado en el motor de traslado de cinta y lograr que el aviso de error siempre mande en las condiciones correctas.

```
////DECLARACIÓN DE VARIABLES////////////////////////////////////
```

```
int aux = 0;
```

```
int aux1 = 0;
```

```
int aux2 = 0;
```

```
int aux3 = 0;
```

```
void main(void)
```

```
{
```

```
    INTCON.GIE = 0;
```

```
    TRISA.RA0 = 0;
```

```
    TRISB.RB0 = 1;
```

```
    TRISB.RB1 = 1;
```

```
    TRISB.RB2 = 1;
```

```
    TRISB.RB3 = 1;
```

```
    TRISB.RB4 = 1;
```

```
    TRISB.RB5 = 1;
```

```
    PORTA = 0;
```

```
    PORTB = 0;
```

```
    // INICIO
```

```
    while(1)
```

```
    {
```

```
        // ERROR TALLA 54
```

```
        while (PORTB.F5 == 1)
```

```
        {
```

```
            if(PORTB.F4 == 1)
```

```

{
  aux3 = 0;
}
while( (PORTB.F0 == 0) && (aux3 == 0) )
{
  PORTA.F0 = 0;
}
if(PORTB.F0 == 1)
{
  aux3++;
}
if(aux3 >= 1)
{
  PORTA.F0 = 1;
}
}
////////////////////////////////////
// ERROR TALLA 58
while (PORTB.F1 == 1)
{
  if(PORTB.F4 == 1)
  {
    aux = 0;
  }
  while( (PORTB.F0 == 0) && (aux == 0) )
  {
    PORTA.F0 = 0;
  }
  if(PORTB.F0 == 1)

```

```

{
    aux++;
}
if(aux >= 1)
{
    PORTA.F0 = 1;
}
}

////////////////////////////////////

// ERROR TALLA 62
while (PORTB.F2 == 1)
{
    if(PORTB.F4 == 1)
    {
        aux1 = 0;
    }
    while( (PORTB.F0 == 0) && (aux1 == 0) )
    {
        PORTA.F0 = 0;
    }
    if(PORTB.F0 == 1)
    {
        aux1++;
    }
    if(aux1 >= 1)
    {
        PORTA.F0 = 1;
    }
}
}

```

```

////////////////////////////////////
// ERROR TALLA 66
while (PORTB.F3 == 1)
{
if(PORTB.F4 == 1)
{
aux2 = 0;
}
while( (PORTB.F0 == 0) && (aux2 == 0) )
{
PORTA.F0 = 0;
}
if(PORTB.F0 == 1)
{
aux2++;
}
if(aux2 >= 1)
{
PORTA.F0 = 1;
}
}
////////////////////////////////////
PORTA = 0;
PORTB = 0;
aux = 0;
aux1 = 0;
aux2 = 0;
aux3 = 0;
}

```

}