



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



**TECNOLÓGICO
NACIONAL DE MÉXICO**

Instituto Tecnológico de Pabellón de Arteaga
Departamento de Ciencias Económico Administrativas

REPORTE FINAL PARA ACREDITAR RESIDENCIA PROFESIONAL DE LA CARRERA DE INGENIERÍA MECATRÓNICA .

PRESENTA: ISIDRO SANTILLÁN ZAMARRIPA

CARRERA: INGENIERÍA MECATRÓNICA

MEDIDOR DE TENDIDOS

Maco-Mex Industrial S.A. de C.V.



ING. ALDO MACIEL GARCÍA ORTEGA

ING. FERNANDO GARCÍA VARGAS

Junio 2021

I. Agradecimientos

Le agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, por brindarme una vida llena de aprendizajes, experiencias y sobre todo felicidad.

Le agradezco a mi madre Ma. Guadalupe por haberme ayudado en todo momento, por los valores que me ha inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. También quiero agradecer a mis hermanos Diana y Antonio por ser parte importante de mi vida y por su ayuda incondicional.

Agradezco a la empresa Maco-Mex Industrial S.A. de C. V. por brindarme la oportunidad de desarrollar mis residencias profesionales, por darme la oportunidad de crecer profesionalmente y personalmente. Así mismo agradezco al ingeniero Aldo Maciel por creer en mí, por compartir conocimientos conmigo y alentarme más en mi aprendizaje.

Agradezco al Instituto Tecnológico de Pabellón de Arteaga, a todos mis profesores y compañeros, por haber compartido conmigo sus conocimientos y experiencias que me acompañaran toda la vida.

II. Resumen

En el presente documento se desarrolla el seguimiento de elaboración de un controlador, este dispositivo tiene como propósito dar solución a un problema dentro de la industria textil, particularmente dentro de la empresa Maco-Mex Industrial S.A. de C.V. en el área de tendido y corte de tela, donde se busca el control y registro de yardas reales de tela tendidas. Incorporando un sistema capaz de medir los tendidos de tela, con tecnología que permitirá la automatización en el proceso de monitorear y capturar la medición de yardas de tela en las líneas de tendido y corte. Esto para poder proyectar el consumo real contra el consumo planeado de tela, mantener el alcance y eficiencia del material.

Índice

I. Agradecimientos	II
II. Resumen	II
III. Lista de tablas.....	V
IV. Lista de Figuras	V
CAPÍTULO 2: GENERALIDADES DEL PROYECTO	8
2.1 <i>Introducción</i>	8
2.2 <i>Descripción de la empresa y área de trabajo del residente.</i>	8
2.2.1 <i>Perfil de la empresa.</i>	8
2.2.2 <i>Antecedentes.</i>	9
2.2.3 <i>Caracterización de la empresa.</i>	10
2.2.4 <i>Estructura organizacional.</i>	11
2.2.5 <i>Principales clientes.</i>	11
2.3 <i>Problemas a resolver.</i>	11
2.4 <i>Objetivos.</i>	12
2.4.1 <i>Objetivo General</i>	12
2.2.4 <i>Objetivos específicos.</i>	12
2.3 <i>Justificación</i>	12
CAPÍTULO 3: MARCO TEÓRICO.....	13
3.1 <i>Industria textil.</i>	13
3.1.1 <i>Características de la industria textil</i>	13
3.1.2 <i>Historia de la industria textil.</i>	14
3.1.3 <i>Procesos de la industria textil.</i>	14
3.2 <i>Tendido.</i>	15
3.2.1 <i>Tendidos básicos</i>	15
3.2.1.1 <i>Cara arriba.</i>	15
3.2.1.2 <i>Cara a cara o zigzag.</i>	16
3.2.1.3 <i>Tendido en un sentido</i>	17
3.2.1.4 <i>Tendido de tejido de punto tubular</i>	18
3.2.2 <i>Tipos de tendidos</i>	18
3.2.1 <i>Tendido manual.</i>	18
3.2.2 <i>Tendido automático.</i>	18
3.3 <i>Controlador</i>	19

3.3.1	Sistema de control.....	19
3.3.2	Microcontrolador.....	19
3.3.2.1	Como se programa a un microcontrolador.....	20
3.3.3	Arduino.....	21
3.3.3.1	Arduino Mega.....	22
3.3.4	NodeMCU Esp8266	23
3.3.4.1	Kit o placa de desarrollo NodeMCU	24
3.3.5	Bus de comunicación I2C.....	25
3.4	Sensores.....	26
3.4.1	Sensores Digitales	27
3.4.1.1	Codificadores rotativos.....	27
3.4.1.2	Codificadores Incrementales	27
3.4.1.3	Encoder.....	28
3.4.1.4	Detección incremental.....	28
3.4.2	Sensores fotoeléctricos	28
3.4.2.1	Modelo retroreflectivo	29
3.4.3	Sensores de Vibración - Acelerómetros	29
3.4.3.1	Sensor mpu-6050.....	30
3.4.4	Push Button.....	31
3.5	Módulo RTC Ds3231	32
3.6	Etapa de Potencia.....	33
3.6.1	Módulo Mt3608.....	33
3.6.2	Módulo de Relevadores para Arduino	34
3.6.3	Driver L298N.....	35
3.7	Indicadores Luminosos	35
3.8	Interfaz con el usuario.....	36
3.8.1	Pantalla TFT 3.5”.....	36
3.8.2	Base de Datos.....	38
3.8.2.1	MySQL	38
3.9	Mecanismo de transformación de movimiento	39
3.9.1	Piñón-cremallera	40
3.10	Ventilador Fan	40
CAPÍTULO 4. DESARROLLO.....		42

4.1	<i>Análisis de proceso de tendido y corte de telas</i>	42
4.2	<i>Análisis de puntos críticos</i>	43
4.3	<i>Ideación</i>	44
4.4	<i>Desarrollo conceptual y básico</i>	45
4.5	<i>Desarrollo avanzado</i>	47
4.5.1	<i>Diseño de controlador:</i>	47
4.5.1.1	<i>Diseño circuito electrónico de control</i>	47
4.5.1.2	<i>Diseño Mecánico</i>	49
4.5.2	<i>Programación de prototipo de controlador</i>	49
4.5.3	<i>Ensamble de prototipo</i>	61
4.5.4	<i>Cronograma de actividades</i>	62
CAPÍTULO 5. RESULTADOS		62
CAPÍTULO 6. CONCLUSIONES.....		65
CAPÍTULO 7. COMPETENCIAS DESARROLLADAS.....		66
CAPÍTULO 8. FUENTES DE INFORMACIÓN.....		68
CAPÍTULO 9. ANEXOS		71

III. Lista de tablas

Tabla 1.	Objetivos y resultados	63
----------	------------------------------	----

IV. Lista de Figuras

Figura 1.	Organigrama de la empresa MACO-MEX.	11
Figura 2.	Conjunto ilustración de industria textil.	13
Figura 3.	Tendido de tela	15
Figura 3.1.	Tendido Cara Arriba.....	16
Figura 3.2	Tendido Cara A Cara.	17
Figura 3.3.	Tendido En Un Sentido	17
Figura 3.4	Tendido de tela manual.....	18
Figura 3.5	Tendido de tela automático	19
Figura 4.	Diagrama controlador.....	19

Figura 5. Microcontrolador	20
Figura 6. Ejemplo de programación	21
Figura 7. Diferentes tipos de placas Arduino	22
Figura 8 Placa Arduino Mega	22
Figura 9. Diagrama bloques funcionales esp8266 CPU.	23
Figura 10. ESP8266EX SOC	24
Figura 11. Placa NodeMCU Esp8266	25
Figura 12. Conexión bus I2C.....	25
Figura 13. Sensores con comunicación por bus.....	27
Figura 14. Ejemplos de sensores.....	27
Figura 15. Codificador rotatorio típico	27
Figura 16. Encoder Incremental	28
Figura 17. Sensor fotoeléctrico Típico.....	29
Figura 18. Descripción Sensor retroreflectivo.....	29
Figura 19. Sensor mpu-6050.....	31
Figura 20. Conexión Pulldown y Pullup de Push Button.....	32
Figura 21. Módulo RTC Ds3231 conexión con Arduino.....	33
Figura 22. Módulo Mt3608	33
Figura 23. Módulo de 4 relevadores para Arduino.....	34
Figura 24. Driver L298N.....	35
Figura 25. Indicadores luminosos.....	36
Figura 26. Interfaz de usuario	36
Figura 27. Pantalla tft 3.5".....	37
Figura 28. Base de datos	38
Figura 29. MySQL.....	39
Figura 30. Mecanismo de transformación de movimiento	40
Figura 31. Mecanismo de piñón-cremallera.	40
Figura 32. Ventilador Fan.....	41
Figura 33. Diagrama de flujo analisis del proceso.	42
Figura 34. Análisis de 3 M.....	43
Figura 35. Lluvia de ideas.....	44
Figura 36. Diagrama de flujo.....	45
Figura 37. Esquema de componentes y su interacción.	46

Figura 38. Circuito Electrónico.	47
Figura 39. Fuente de alimentación.	48
Figura 40. Diseño del mecanismo para montar encoder.	49
Figura 41. Programación conexión entre tarjeta Node MCU y base de datos.	60
Figura 42. Programación para guardar datos en tabla de base de datos	60
Figura 43. Ensamble de prototipo circuito interior.	61
Figura 44. Ensamble de prototipo cableado.	61
Figura 45. Ensamble total del sistema	61
Figura 46. Ensamble prototipo vista exterior.	62
Figura 47. Ensamble del mecanismo para montar encoder.....	62
Figura 48. Realización de pruebas en mesa de tendido.	64
Figura 49. Pruebas con sensores retroreflectivos (detección de tela).	64
Figura 50. Obtención de datos en los microcontroladores.....	64
Figura 51. De izquierda a derecha, se puede observar el proceso de tendido mostrado por el controlador.....	65
Figura 52. Registro de yardas y proceso de tendido en base de datos.	65

CAPÍTULO 2: GENERALIDADES DEL PROYECTO

2.1 Introducción

Dentro de la industria textil, la materia prima más importante es la tela ya que todos o la mayoría de sus productos son derivados de este material, en el proceso de la manufactura de productos textiles se encuentra la operación de tendido, el tendido de tela consiste en extender las capas de tela de manera uniforme a lo largo de la mesa de corte, para inmediatamente proceder a cortar. Pero para que este proceso se realice correctamente, es necesario saber qué tipo de tendido se debe realizar según la tela utilizada, cuáles son los equipos disponibles, las necesidades de la producción y otras cuestiones.

En Maco-Mex Industrial S.A. de C.V. se realiza el tendido de rollos de tela en las mesas de corte, el monitoreo y registro de tela tendida se realiza de una forma manual por parte de los operarios, el procedimiento de tendido es de las tareas de suma importancia porque se busca el mejor aprovechamiento en el consumo de tela, el consumo y cuidado de la tela puede significar un gasto importante dentro de los estándares de cada empresa textil.

En el presente reporte del proyecto, se describe el seguimiento y desarrollo de un controlador que será instalado en las líneas de tendido y podrá medir las yardas de tela que son tendidas y de esta forma saber con mayor precisión y exactitud el consumo real del material y proyectarlo contra el consumo planeado.

2.2 Descripción de la empresa y área de trabajo del residente.

2.2.1 Perfil de la empresa.

Nombre o razón social: Maco-Mex industrial s.a. de c.v.

Ramo: textil

Dirección: av. miguel hidalgo s/n, cosío, Aguascalientes México.

Teléfono: + 52 1 458- 987-0852

2.2.2 Antecedentes.

Maco-Mex Industrial S.A de C.V. Nace a finales de los años 80s, aun no teniendo este nombre, comenzó en la ciudad de Aguascalientes en una nave industrial, en donde se contaba con solamente cinco máquinas de costura recta, así como solo cinco operarios.

A partir de esos cinco operarios y los fundadores, se dedicaban a coser bolsas de polietileno para empacar cobertores, teniendo como cliente único San Gerardo Textil.

A inicios de los años 90 se cambiaron a Pablo Escaleras, Rincón de Romos, estableciéndose como empresa con el nombre de "Sábanas y Edredones" (cabe mencionar que nunca confeccionaron sábanas ni edredones), siguieron elaborando bolsas de polietileno y posteriormente adhirieron la serigrafía, como serigrafía elaboraban bolsas de plástico para Coca-Cola.

Después se establece en Tepezalá, Pabellón, Rincón de Romos, Emiliano Zapata y Cosío, llevando el nombre de Maco-Mex Industrial S.A de C.V., se puso este nombre porque se estableció plenamente en el Municipio de Cosío, Maco significa Maquiladora Cosío.

Comenzaron confeccionando overol y posteriormente en el Municipio de Cosío se inició a trabajar con uniformes escolares para exportación y a raíz de eso se ha ido avanzando como empresa consolidada.

En el año 2010 se comenzó con la negociación de tener en Cosío un Centro de Distribución, y así ir complementando a la fecha todo el proceso de distribución de productos Jostens de uniformes escolares.

Actualmente Maco-Mex tiene tres naves y cuenta con 1200 empleados.

Somos un grupo de empresas dedicadas a producir artículos de alto consumo que van desde uniformes para hospitales y clínicas, ropa de seguridad industrial, artículos

promocionales, hasta uniformes para universidades, que ha permanecido en el estado de Aguascalientes por más de 25 años.

En los 90`s, la industria textil estaba en pleno auge en el estado de Aguascalientes, había una enorme cantidad de empresas de la confección que demandaban un gran volumen de personas, generando con esto mucha competencia por mano de obra entre empresas del mismo ramo.

Viendo con anticipación la problemática que se presentaría, el grupo toma la decisión de sacar sus empresas a los municipios, y con esto, ir un paso delante de las empresas del sector.

2.2.3 Caracterización de la empresa.

Misión

“Fabricamos y distribuimos productos y servicios de alto consumo, con la mejor Calidad y Productividad, permitiendo a nuestros Clientes como a nuestra Empresa la permanencia en el mercado, siempre comprometidos con el desarrollo de nuestras comunidades”.

Visión

“Ser los mejores fabricantes y distribuidores de productos y servicios de alto consumo, con la mejor calidad, precio justo y servicio oportuno, satisfaciendo las expectativas de nuestros clientes”

Política de calidad

“En Maco-mex Industrial, estamos comprometidos a trabajar con los más altos estándares de calidad, cumpliendo con las expectativas de nuestros clientes, alineados a la estrategia marcada por la alta dirección buscando siempre la mejora continua.

2.2.4 Estructura organizacional.

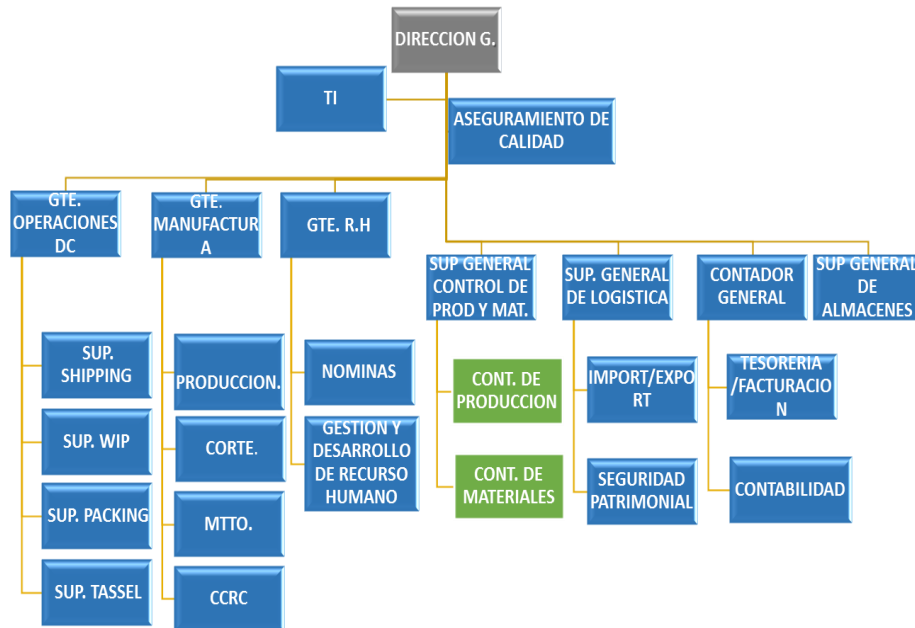


Figura 1. Organigrama de la empresa MACO-MEX.

2.2.5 Principales clientes.

Actualmente el principal cliente de Maco-Mex Industrial S.A. de C.V. es la manufactura de productos Jostens de uniformes escolares y productos para graduación escolar.

2.3 Problemas a resolver.

Dentro de la empresa Maco-Mex Industrial S.A. de C.V., el monitoreo y registro de yardas tendidas que se realiza en el área de tendido y corte se lleva a cabo de una forma manual por parte de los operarios y al ser así los datos pueden presentar variación respecto a las yardas reales tendidas, haciendo un análisis en este proceso y en el área se pudo observar que dentro de mano de obra no se sabe con exactitud total de yardas tendidas y por ende no se reporta totalmente yardas tendidas, en la máquina tendedora no cuenta con comunicación con base de datos para registro de mediciones, eventos y fallas, variación de yardas en rollos nuevos y uso de rollos con consumo medio y rollos nuevos.

2.4 Objetivos

2.4.1 Objetivo General

Desarrollar un controlador que mida tendido de tela en yardas y proyecte consumo.

2.2.4 Objetivos específicos

- I. Diseñar controlador.
- II. Manufacturar controlador.
- III. Programación de controlador.
- IV. Realizar pruebas, simulación y ajustes.

2.3 Justificación

La situación que se presenta en los tendidos de tela demanda un mayor control para la medición de yardas que se tienden en las mesas de corte, el proyecto Medidor de tendidos pretende dar solución y mejora en el área de tendido, ya que con un control más exacto en el medido de tela se podrán acaparar puntos secundarios como tiempos muertos y discrepancia de materia prima, la medición de las yardas tendidas traerá beneficios enfocados en mejorar el uso de la tela y disminución del scrap, cabe mencionar que la tela es la materia prima más importante para los productos manufacturados en Maco-Mex.

CAPÍTULO 3: MARCO TEÓRICO

3.1 Industria textil

La industria textil es la actividad económica dedicada a la manufactura de hilos, fibras, telas y otros materiales para obtener productos derivados como la ropa.

Esta industria abarca entonces la confección de prendas, calzados y otras piezas que usan las personas para vestirse.

La industria textil destaca, frente a otros sectores, por la gran cantidad de mano de obra que emplea a lo largo de todo el mundo. Además, al necesitar maquinaria liviana, puede instalarse en cualquier lugar del globo donde pueda aprovechar el menor coste por hora hombre de trabajo.

Esto último es motivo de preocupación por ciertos analistas que consideran que las grandes empresas textiles pueden explotar, pagando salarios bajos, a trabajadores en países en vías de desarrollo.



Figura 2. Conjunto ilustración de industria textil.

3.1.1 Características de la industria textil

Las principales características de la industria textil son:

- Se encuentra dentro de la categoría de la industria ligera. Esto, porque no requiere grandes cantidades de materiales, en comparación a otros sectores como el minero o metalúrgico.
- Utiliza tanto insumos naturales, como el algodón o la lana, como sintéticos, como el poliéster o el nailon.

- Sus productos suelen consumirse todo el año y de manera masiva.
- Pertenece al sector secundario de la economía, que transforma la materia prima en bienes destinados al consumidor final.
- Es un sector muy vinculado a la industria de la moda, desde donde se proponen las tendencias en el vestir. Estas, a su vez, varían según el país o región.
- La demanda cambia según la época del año, variando el tipo de prendas requeridas. En verano, por ejemplo, cae la venta de chaquetas.

3.1.2 Historia de la industria textil

La actividad textil, aunque no como industria como tal, prácticamente existe desde que el individuo comenzó a vestirse. Al principio, se desarrollaba en los hogares y de manera muy artesanal.

Posteriormente, con la Primera Revolución Industrial, comenzaron instalarse talleres para producir de forma constante. Esto, con procesos que aún necesitaban de una gran cantidad de mano de obra.

Sin embargo, a partir del siglo XIX, el sector textil comenzó a modernizarse con nuevas tecnologías. Así, se desarrolló una producción en mayores volúmenes sin requerir de un incremento significativo en el número de trabajadores.

Cabe señalar además que a partir del siglo XX las fibras sintéticas como el nailon o el poliéster empezaron a ganar terreno. Esto, frente a insumos naturales como el algodón y la lana.

3.1.3 Procesos de la industria textil

Los procesos de la industria textil principalmente son:

1. Obtención de la fibra, que puede ser de origen animal o sintético.
2. Hilandería, que consiste en el tratamiento de la fibra para fabricar hilos con los que se pueda confeccionar el producto final.
3. El tejido, que es el procedimiento mediante el cual se fabrican las telas entrelazando los hilos de manera mecánica o a mano.
4. La tintorería, que el proceso mediante el cual se pintan las telas.
5. La confección, que es el paso final donde se cortan y cosen las telas, en base a un diseño. De ese modo, se obtiene la mercancía que se llevará al mercado.

3.2 Tendido

Consiste en extender las capas de tela de manera uniforme a lo largo de la mesa de corte, para inmediatamente proceder a cortar.

El tendido debe realizarse tratando de maltratar la tela lo menos posible, sobre todo en cuanto a estirones, que en el tejido de punto deforman bastante la tela.

Una vez que la tela ha sido extendida hasta formar capas más o menos altas, de unos 10a 20cm aproximadamente, se la deja «descansar» sobre la mesa de corte, preferiblemente en la noche. Hacer vibrar o golpear la mesa de corte durante y después del tendido, contribuye también al relajamiento de la tela.

Deben evitarse las dobladuras y/o aglomeraciones de tela en cualquier punto de la mesa, puesto que éstas producen cortes defectuosos.

Además, la mesa de corte debe estar lisa y pulida, libre de asperezas y abolladuras, a fin de evitar agarres y roturas de tela al momento del tendido. Las dimensiones de la mesa de corte varían considerablemente, y dependen de la producción de cada empresa, así como de las necesidades específicas de uso.



Figura 3. Tendido de tela

3.2.1 Tendidos básicos

3.2.1.1 Cara arriba

El tendido se hace colocando la tela a lo largo de la mesa, y cuando se llega al extremo de ésta se corta el tejido, y se gira el rollo sujetándolo por la varilla que lo soporta, para iniciar nuevamente el tendido desde ese extremo.

Con este sistema el derecho o revés de la tela siempre quedan en el mismo sentido, hacia arriba o hacia abajo. Pero es preferible tender con el derecho de la tela hacia arriba, con el fin de poder controlar los defectos que se puedan presentar.

Este sistema permite disminuir los tiempos del tendido, pero tiene el inconveniente de que las columnas del tejido quedan con diferentes direcciones entre capa y capa.

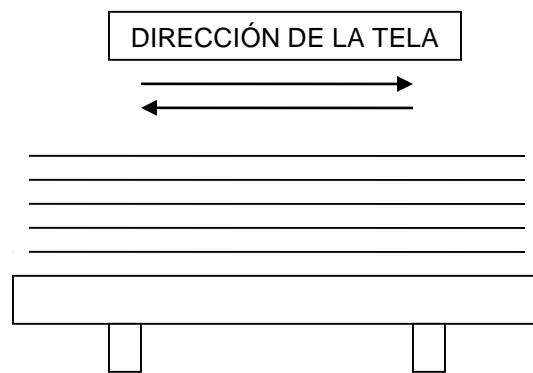


Figura 3.1. Tendido Cara Arriba

3.2.1.2 Cara a cara o zigzag

La tela se extiende hasta llegar al extremo del tendido, y se dobla, regresando el operario desde ese extremo colocando una nueva capa de tejido, quedando intercaladas cara con cara y revés con revés.



Figura 3.2 Tendido Cara A Cara.

3.2.1.3 Tendido en un sentido

Algunas telas presentan sombras, según el sentido del tejido –velvetines, telas satinadas– y para estos tejidos el sistema de tendido en un sentido es el ideal, pues al extender cada una de las capas siempre se comienza en la misma punta, y con la tela colocada en la misma dirección, con el derecho o el revés de la tela hacia arriba o hacia abajo.

Es preferible –como en el tendido de cara arriba–, tender con el derecho hacia arriba.

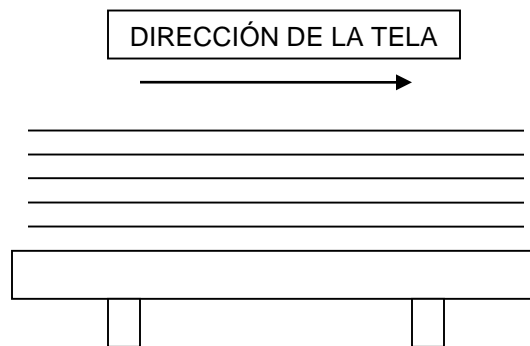


Figura 3.3. Tendido En Un Sentido

Este tendido es recomendado para los siguientes tipos de telas:

- Telas con pelos o satinadas como el velvetín.
- Estampados con el dibujo en una dirección.
- Telas a cuadros o rayadas.

3.2.1.4 Tendido de tejido de punto tubular

El principio es el mismo descrito anteriormente. Al extender la tela siempre queda cara a cara, pero considerando que la dirección de las mallas queda orientada, como en el extendido en un solo sentido, no así el método cara arriba ni el método zig-zag.

3.2.2 Tipos de tendidos

El extendido de las telas puede realizarse de dos maneras: manual o automáticamente.

3.2.1 Tendido manual

El extendido manual lo realizan los operarios, distribuyendo la tela por ambos lados de la mesa, dejándola lista para el corte.

Esta operación se realiza cuando no existe gran producción de prendas, y cuando las dimensiones de la mesa son pequeñas.



Figura 3.4 Tendido de tela manual

3.2.2 Tendido automático

El extendido automático se realiza con carros tendedores, capaces de realizar recorridos de vaivén, provistos de ruedas para rieles, e impulsados eléctricamente para realizar su trabajo.

En cada extremo de la mesa existe un tope final, que hace que la dirección de los rodillos alimentadores, colocados en su parte superior, sea la misma, de manera que el carro haga igual operación de tendido al regresar.

Cabe mencionar que estos carros tendedores también pueden ser operados manualmente por los operarios.



Figura 3.5 Tendido de tela automático

3.3 Controlador

3.3.1 Sistema de control

Un sistema de control manipula indirectamente los valores de un sistema controlado. Su objetivo es gobernar un sistema sin que el operador intervenga directamente sobre sus elementos. El operador manipula valores de referencia y el sistema de control se encarga de transmitirlos al sistema controlado a través de los accionamientos de sus salidas.

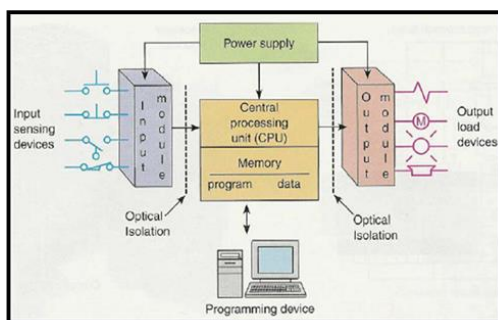


Figura 4. Diagrama controlador

3.3.2 Microcontrolador

El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida. También incluye a un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM). Funciona como una mini PC. Su función es la de automatizar procesos y procesar información.

El microcontrolador se aplica en toda clase de inventos y productos donde se requiere seguir un proceso automático dependiendo de las condiciones de distintas entradas.

Elementos de un Microcontrolador

Un microcontrolador al menos tendrá:

- Microprocesador.
- Periféricos (unidades de entrada/salida).
- Memoria.

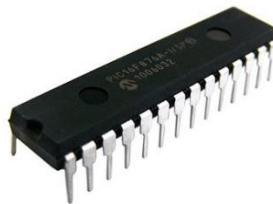


Figura 5. Microcontrolador

3.3.2.1 Como se programa a un microcontrolador

Un microcontrolador se programa en lenguaje ensamblador. Entonces cada microcontrolador, tiene su propio conjunto de instrucciones. Las instrucciones se dividen de acuerdo de las operaciones que realizan en:

- Aritméticas.
- Lógicas.
- Transferencia de bits.
- Misceláneas.

Para poder programar a un microcontrolador se requieren tres cosas:

- Compilador. Traduce lenguaje como el C y C++ al lenguaje ensamblador.
- IDE. Integrated Development Environment. Entorno de Desarrollo Integrado. Es donde mediante el compilador, se crea y compilan los programas.
- Programador. Cada fabricante y arquitectura tiene su propio programador. Existen programadores llamados Universales que pueden programar distintas arquitecturas.

```

Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}

```

Annotations in the image:

- introductory comments describe the program**: Points to the first three lines of the code.
- variable declaration section**: Points to the line `int led = 13;`.
- setup section**: Points to the `void setup() { ... }` block.
- loop section**: Points to the `void loop() { ... }` block.

Figura 6. Ejemplo de programación

3.3.3 Arduino

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo.

El Arduino es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE (Entorno de Desarrollo Integrado). Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.



Figura 7. Diferentes tipos de placas Arduino

3.3.3.1 Arduino Mega

El Arduino Mega es probablemente el microcontrolador más capaz de la familia Arduino. Posee 54 pines digitales que funcionan como entrada/salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un boton de reset y una entrada para la alimentación de la placa.



Figura 8 Placa Arduino Mega

Arduino Mega posee las siguientes especificaciones:

- Microcontrolador: ATmega2560
- Voltaje Operativo: 5V
- Voltaje de Entrada: 7-12V
- Voltaje de Entrada(límites): 6-20V
- Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)

- Pines análogos de entrada: 16
- Corriente DC por cada Pin Entrada/Salida: 40 mA
- Corriente DC entregada en el Pin 3.3V: 50 mA
- Memoria Flash: 256 KB (8KB usados por el bootloader)
- SRAM: 8KB
- EEPROM: 4KB
- Clock Speed: 16 MHz

3.3.4 NodeMCU Esp8266

NodeMCU es una placa de desarrollo totalmente abierta, a nivel de software y de hardware. Al igual que ocurre con Arduino, en NodeMCU todo está dispuesto para facilitar la programación de un microcontrolador o MCU (del inglés Microcontroller Unit). Son placas o kits de desarrollo que llevan incorporados un chip que se suele llamar SoC (System on a Chip) que dentro tiene un microcontrolador o MCU. En el NodeMCU este chip se integra dentro del SoC. a todo este conjunto (SoC) se le conoce como ESP8266. Por lo tanto, en términos estrictos el ESP8266 no es un microcontrolador. Dentro sí que lleva uno y se llama Tensilica L106 de 32-bit. La MCU se va a encargar de gestionar todas las entradas, salidas y cálculos necesarios para hacer funcionar el programa que hayamos cargado. Trabaja a una velocidad de 80MHz aunque puede llegar a los 160MHz. Esto nos indica la frecuencia con la que la MCU ejecuta las instrucciones.

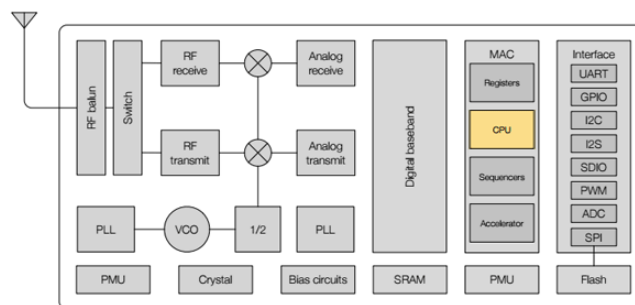


Figura 9. Diagrama bloques funcionales esp8266 CPU.

SoC ESP8266

Una vez que tenemos claro qué es una MCU o microcontrolador, ahora veremos que es en realidad el ESP8266. El nombre técnico es ESP8266EX.



Figura 10. ESP8266EX SOC

Se trata de un SoC o Sistema en Chip. Básicamente consiste en un chip que tiene todo integrado (o casi todo) para que pueda funcionar de forma autónoma como si fuera un ordenador. En el caso del ESP8266 lo único que no tiene es una memoria para almacenar los programas.

Esto supone un inconveniente ya que parte de los pines de entrada y salida, tendrán que ser utilizados para conectarse a una memoria Flash externa.

Las características principales son las siguientes:

- Incorpora una MCU de 32-bit de bajo consumo (Tensilica L106)
- Módulo WiFi de 2.4 GHz
- RAM de unos 50 kB
- 1 entrada analógica de 10-bit (ADC)
- 17 pines de entrada y salida GPIO (de propósito general)

Son el primer paso hacia el Internet de las Cosas o el IoT. Pueden enviar datos, recibir datos e incluso controlar los pines de entrada y salida de forma remota e inalámbrica.

3.3.4.1 Kit o placa de desarrollo NodeMCU

Existen diferentes modelos de varias marcas con características y funcionalidades diferentes dependiendo del SoC o microcontrolador que utilizan. Pero todos tienen el mismo objetivo, facilitarnos el prototipado y desarrollo de proyectos con microcontroladores.

NodeMCU es uno de ellos y sus características principales son:

- Conversor Serie-USB para poder programar y alimentar a través del USB
- Fácil acceso a los pines
- Pines de alimentación para sensores y componentes
- LEDs para indicar estado
- Botón de reset



Figura 11. Placa NodeMCU Esp8266

3.3.5 Bus de comunicación I2C

El estándar I2C (Inter-Integrated Circuit) fue desarrollado por Philips en 1982 para la comunicación interna de dispositivos electrónicos en sus artículos. Posteriormente fue adoptado progresivamente por otros fabricantes hasta convertirse en un estándar del mercado.

El bus I2C requiere únicamente dos cables para su funcionamiento, uno para la señal de reloj (CLK) y otro para el envío de datos (SDA), lo cual es una ventaja frente al bus SPI. Por contra, su funcionamiento es un poco más complejo, así como la electrónica necesaria para implementarla.

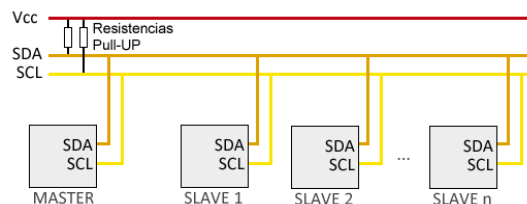


Figura 12. Conexión bus I2C.

En el bus Cada dispositivo dispone de una dirección, que se emplea para acceder a los dispositivos de forma individual. Esta dirección puede ser fijada por hardware (en cuyo caso, frecuentemente, se pueden modificar los últimos 3 bits mediante jumpers o interruptores) o totalmente por software.

El bus I2C tiene una arquitectura de tipo maestro-esclavo. El dispositivo maestro inicia la

comunicación con los esclavos, y puede mandar o recibir datos de los esclavos. Los esclavos no pueden iniciar la comunicación (el maestro tiene que preguntarles), ni hablar entre si directamente.

3.4 Sensores

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad o un sensor capacitivo), una tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc.

Los sensores se pueden clasificar en función de los datos de salida en:

- Digitales
- Analógicos

Y dentro de los sensores digitales, estos nos pueden dar una señal digital simple con dos estados como una salida de contacto libre de tensión o una salida en bus.

Sensores con comunicación por bus. Un bus (o canal) es un sistema digital que transfiere datos entre los componentes de un dispositivo electrónico o entre varios. Está formado por cables o pistas en un circuito impreso, dispositivos como resistencias y condensadores además de circuitos integrados.

Existen muchos tipos de buses de comunicaciones, algunos de ellos los implementa arduino mediante controladores HW integrados en la MCU (I2C) o mediante una librería (one wire) y en otros casos es necesario un hardware adicional para adaptar la señal con un transceiver y manejar el protocolo con un controlador, por ejemplo can bus o modbus.

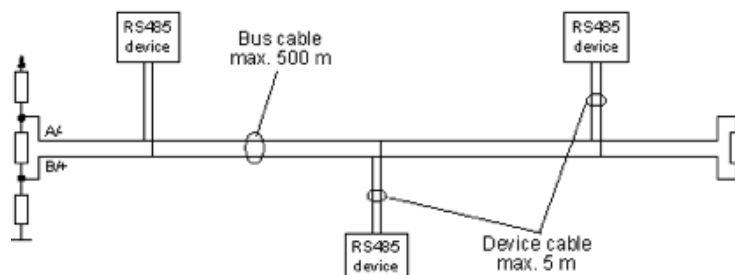


Figura 13. Sensores con comunicación por bus.

3.4.1 Sensores Digitales

Cuando hablamos de sensores digitales, nos referimos a aquellos sensores que emiten valores discretos, generalmente relacionados con la posición lineal o angular, así como aquellos sensores que se utilizan para detectar cuando un objeto está cerca.

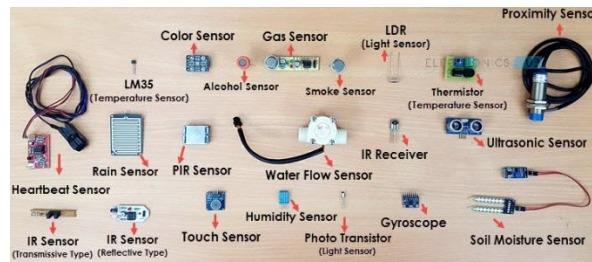


Figura 14. Ejemplos de sensores

3.4.1.1 Codificadores rotativos

Un codificador rotatorio generalmente proporciona una excelente resolución de ángulo, ya que están disponibles con hasta miles de pasos por revolución de 360 °, lo que permite pasos mucho más pequeños que 1 °. Muchos codificadores también pueden detectar la dirección de rotación, que es esencial en algunas aplicaciones.



Figura 15. Codificador rotatorio típico

3.4.1.2 Codificadores Incrementales

Los codificadores incrementales informan cambios relativos en la posición y la dirección; no siguen la posición absoluta (ángulo).

Los codificadores incrementales emiten señales A y B, que indican cambios en el movimiento y la dirección. Algunos de ellos son capaces de ser "dirigidos" o referenciados a una posición particular. Cuando se alcanza esta posición, se genera una señal de salida Z adicional. Los codificadores incrementales son los tipos de

codificadores más comunes y populares.

3.4.1.3 Encoder

El encoder es un dispositivo electromecánico que permite codificar el movimiento mecánico en distintos tipos de impulsos eléctricos: digitales binarios, analógicos en función de una onda, pulsos, etcétera. De este modo, un encoder es una interfaz entre un dispositivo mecánico móvil y un controlador.

Existen dos grandes tipos de encoder o generador de pulsos: los lineales y los rotatorios; en cada grupo a su vez hay distintos tipos de codificación (absolutos e incrementales) y principios electromecánicos.

3.4.1.4 Detección incremental

Este tipo de encoder detecta el movimiento —y la distancia recorrida en ese movimiento— gracias a la detección diferencial de dos valores codificados en la superficie detectable.

La ventaja de este sistema es que es más económico que el encoder absoluto, y que permite mantener la misma precisión independientemente de la longitud de la pieza móvil; como desventaja debemos señalar que, para inicializarse, el sistema necesita posicionar el encoder en un cero predeterminado.



Figura 16. Encoder Incremental

3.4.2 Sensores fotoeléctricos

Los sensores fotoeléctricos son considerados como sensores de movimiento activos los cuales trabajan al enviar destellos de ondas de sonido ultrasónicas, después de las cuales el sensor espera a que la energía se refleje de regreso.



Figura 17. Sensor fotoeléctrico Típico

3.4.2.1 Modelo retroreflectivo

Tanto el emisor de luz como los elementos receptores están contenidos en un mismo recinto. La luz del elemento emisor incide en el reflector y regresa al elemento receptor de luz. Cuando hay un objeto presente, se interrumpe la luz.

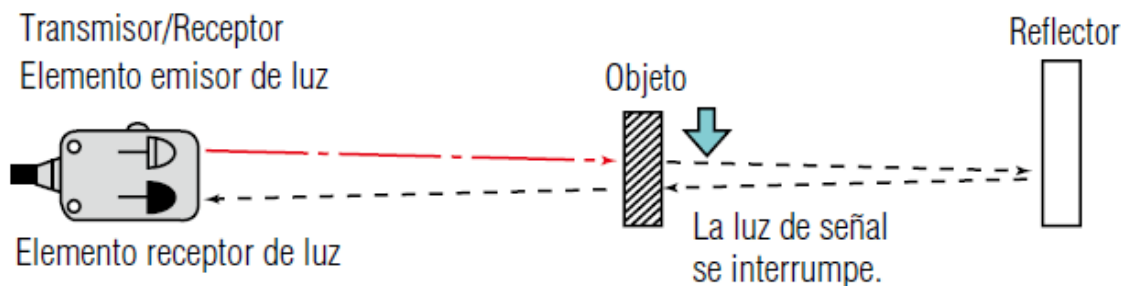


Figura 18. Descripción Sensor retroreflectivo

3.4.3 Sensores de Vibración - Acelerómetros

Los acelerómetros se usan para medir vibraciones y golpes en máquinas y básicamente cualquier cosa que se mueva. Sus salidas también se pueden integrar y doblemente integradas para calcular el desplazamiento y la velocidad.

Los acelerómetros para realizar mediciones dinámicas se basan normalmente en el principio piezoeléctrico: cuando un cristal de cuarzo se somete a tensión, libera una corriente de iones cargados proporcional a la tensión. Estos sensores de carga están conectados a un acondicionador de señal de tipo de carga. Un tipo aún más popular son los sensores IEPE (también conocidos como ICP®), que tienen un preamplificador integrado y que requieren un acondicionador de señal menos costoso.

También hay acelerómetros de tipo capacitivo que se basan en un principio diferente y que son populares en aplicaciones industriales menos exigentes.

Además, hay acelerómetros basados en MEMS que se utilizan mucho en aplicaciones de navegación, orientación de tabletas y teléfonos, pruebas automotrices y captura de movimiento.

3.4.3.1 Sensor mpu-6050.

El sensor MPU-6050 es una pequeña pieza tecnológica de procesamiento de movimiento. El cual mediante la combinación de un MEMS (Sistemas Micro electromecánicos) giroscopio de 3 ejes y un MEMS acelerómetro de 3 ejes en la misma pastilla de silicio junto con un DMP™ (Movimiento Digital Processor™), es capaz de procesar los algoritmos de movimientos complejos de 9 ejes (MotionFusion™) en una placa.

El MPU-6050 elimina los problemas de alineación del eje transversal que puede arrastrarse hacia arriba en porciones discretas. Las piezas integran el algoritmo MotionFusion para 9 ejes 'pueden incluso acceder a magnetómetros externos u otros sensores a través de un bus I2C auxiliar maestro, permitiendo reunir un conjunto completo de dispositivos sensores de datos, sin la intervención del procesador del sistema. El MPU-6050 es un 6 DOF (grados de libertad = Degrees of Freedom) o un sensor IMU de seis ejes, lo que significa que da seis valores de salida.

El procesador digital de movimiento (DMP) incorporado se encuentra dentro de la MPU-6050 y descarga el cálculo de los algoritmos de procesamiento de movimiento desde el procesador host. Los datos resultantes pueden ser leídos de los registros de la DMP, o pueden estar tamponados en un FIFO. El DMP tiene acceso a uno de los pines externos de la MPU, que pueden ser utilizados para la generación de interrupciones. El propósito del DMP es descargar los requisitos de temporización y la potencia de procesamiento del procesador anfitrión.

El sensor MPU-6050 es muy preciso, ya que contiene una conversión hardware de 16 bits de A/D por cada canal, para la digitalización de las salidas del acelerómetro. Para ello capta los canales x, y y z al mismo tiempo. Como se ha comentado, el sensor utiliza el I2C-bus para interconectar con el Arduino. Aunque algunas versiones (como la

mostrada) llevan un regulador que permite conectarla a 5V.



Figura 19. Sensor mpu-6050.

3.4.4 Push Button

Un push button es un componente eléctrico compuesto, en su mayoría, por 2 terminales o pines, estas terminales pueden estar o no en contacto eléctrico dependiendo de si el botón está siendo presionado o no, en este sentido los push button pueden clasificarse en:

- Normalmente abierto: Sus terminales no están en contacto eléctrico cuando el botón no es presionado y entran en contacto eléctrico cuando el botón es presionado.
- Normalmente cerrado: Sus terminales están en contacto eléctrico cuando el botón no es presionado y dejan de tener contacto eléctrico cuando el botón es presionado.

Un push button es un sensor digital de entrada, es decir, su funcionamiento consiste en entregar la información correspondiente si es presionado o no a través del voltaje en su pin de señal, la tarjeta Arduino a su vez recibe la señal y la lee con uno de sus pines digitales de entrada.

Para conectar un push button a una tarjeta de desarrollo es necesario una resistencia auxiliar para conseguir un arreglo en pull-up o pull-down.

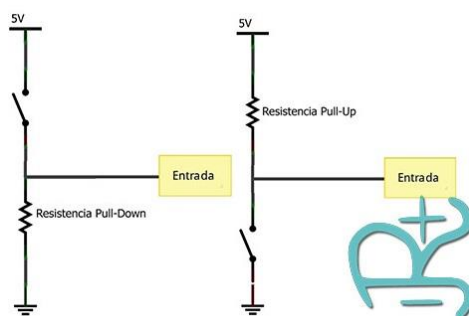


Figura 20. Conexión Pulldown y Pullup de Push Button.

3.5 Módulo RTC Ds3231

Un reloj de tiempo real (RTC) es un dispositivo electrónico que permite obtener mediciones de tiempo en las unidades temporales que empleamos de forma cotidiana.

El término RTC se creó para diferenciar este tipo de relojes de los relojes electrónicos habituales, que simplemente miden el tiempo contabilizando pulsos de una señal, sin existir relación directa con unidades temporales.

Por el contrario los RTC son más parecidos a los relojes y calendarios que usamos habitualmente, y que funcionan con segundos, minutos, horas, días, semanas, meses y años.

Los RTC normalmente están formados por un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo. La electrónica de los RTC tienen en cuenta las peculiaridades de nuestra forma de medir el tiempo, como por ejemplo el sistema sexagesimal, los meses con diferentes días, o los años bisiestos.

Los RTC aportan la ventaja de reducir el consumo de energía, aportar mayor precisión y liberar a Arduino de tener que realizar la contabilización del tiempo. Además, frecuentemente los RTC incorporan algún tipo de batería que permite mantener el valor del tiempo en caso de pérdida de alimentación.

En el mundo de la electrónica casera y Arduino existen dos RTC habituales el DS1307 y el DS3231. El DS3231 tiene una precisión muy superior y puede considerarse sustituto del DS1307.

La comunicación en ambos modelos se realiza a través del bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es 4.5 a 5.5V para el DS1307, y 2.3 a 5.5V para el DS3231.

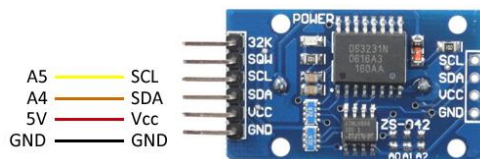


Figura 21. Módulo RTC Ds3231 conexión con Arduino.

3.6 Etapa de Potencia

Un circuito de potencia se basa en recibir señales digitales de un circuito de control, procesarlas y activar una salida de potencia de altos voltajes como pueden ser motores, bombillos, bobinas, etc.

Los circuitos de potencia son aquellos elementos que hacen de alguna manera el trabajo duro, puesto que son los encargados de ejecutar las órdenes dictaminadas por el circuito de mando.

3.6.1 Módulo Mt3608

El convertidor de voltaje CC-CC Step-Up (elevador) MT3608 tiene como función entregar un voltaje de salida regulado y constante superior al voltaje de entrada, sin importar las variaciones del voltaje de entrada o de carga. Soporta corrientes de salida de hasta 2A, voltaje de entrada entre 2V a 24V, y voltaje de salida entre 2V a 28V. El voltaje de salida se ajusta mediante un potenciómetro multivuelta o preset.

Cuando un sistema basado en Arduino debe ser móvil, es normal utilizar una pila de 9V para alimentar el Arduino y aislarlo de las variaciones de voltaje del conjunto de pilas que se utilice. Más aún cuando estas pilas alimentan elementos como motores, que generan pulsos de consumo en el momento del arranque y producen bajones momentáneos, que pueden reiniciar al Arduino, o bajones cuando aumenta el esfuerzo que deben hacer, y las pilas se van descargando.

El convertidor CC-CC MT3608 es un regulador de tipo conmutado elevador (Step-Up o Boost) con una alta eficiencia de conversión, excelente regulación de línea y bajo voltaje de alterna superpuesta a la salida.



Figura 22. Módulo Mt3608

ESPECIFICACIONES TÉCNICAS

- Convertidor CC-CC Boost: MT3608
- Voltaje de entrada: 2V a 24V CC
- Voltaje de salida: 5V a 28V CC
- Voltaje de salida ajustable (regulable por un potenciómetro)
- Corriente de salida: máxima 2A (usar disipador para corrientes mayores a 1,5A)
- Potencia de salida: 6W
- Eficiencia de conversión: 93% máx.
- Frecuencia de Trabajo: 1,2MHz
- Protección de sobre-temperatura: SI (apaga la salida)
- Protección de corto circuito: NO
- Protección limitadora de corriente: SI (4A)
- Protección frente a inversión de polaridad: NO
- Dimensiones: 36mm x 17mm x 7mm

3.6.2 Módulo de Relevadores para Arduino

Las salidas de las placas Arduino son perfectamente útiles para controlar cargas que no consuman demasiada corriente, como un Led, pero son insuficientes para cargas mayores.

Un módulo de relés (o relays) que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas, las que cuentan con leds individuales que sirven como indicadores de estado.

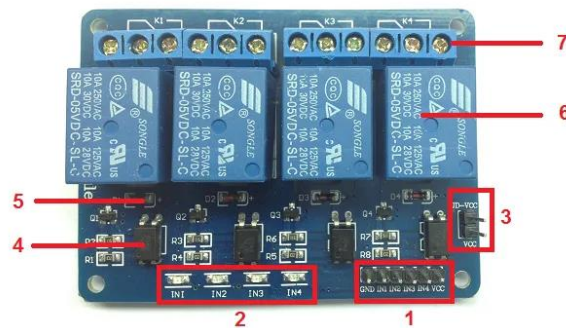


Figura 23. Módulo de 4 relevadores para Arduino.

La placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o

negativo y Vcc es el positivo) [1], cuatro leds que indican el estado de la entradas [2], un jumper selector para la alimentación de los relés [3], cuatro optoacopladores del tipo FL817C [4], cuatro diodos de protección [5], cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V [6] y cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés [7].

3.6.3 Driver L298N

Este módulo basado en el chip L298N te permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios.

El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N.

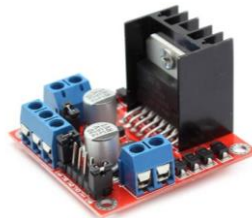


Figura 24. Driver L298N

3.7 Indicadores Luminosos

Las luces indicadoras, tienen como función principal, alertar o avisar a los trabajadores, de alguna falla técnica o mecánica que se presente en el momento.

Es imprescindible que se cuente con estas luces, puesto que indican y mejoran el rendimiento en la producción que se está realizando. Además, tienen la ventaja de que puedes personalizarlas según a la necesidad, es decir, se pueden armar tomando en cuenta los requerimientos que ocupe el área de producción.

Estas luces indicadoras, o indicadores luminosos, manejan diversos colores representativos para identificar las respectivas instrucciones. Por ejemplo, la luz verde, puede representar que el proceso se encuentra activo, la luz ámbar que está en proceso de detección y la luz roja, que el procedimiento se encuentra inactivo. Las indicaciones varían según al uso que se le dé al color.

Por su parte, manejan dos tipos de iluminación, a base de led, o de filamento. En esto es opcional el tipo de iluminación que mejor se adapte a la necesidad de la empresa.



Figura 25. Indicadores luminosos.

3.8 Interfaz con el usuario

La interfaz de usuario o UI (User Interface) es un concepto que abarca arquitectura de información, patrones y diferentes elementos visuales que nos permiten interactuar de forma eficaz con sistemas operativos y softwares de diversos dispositivos.

El medio a través del cual el usuario interactúa con un dispositivo tecnológico. Esto abarca todos los puntos de contacto entre la persona y el equipo.



Figura 26. Interfaz de usuario

3.8.1 Pantalla TFT 3.5"

Las pantallas o displays LCD permiten mostrar texto, gráficos e imágenes en los proyectos con microcontroladores como Arduino, Pic o Raspberry Pi. Ideal para visualización e ingreso de datos, implementar una interfaz hombre máquina (HMI) moderna y profesional o ayudarnos a realizar depuración de código (debugging) en nuestros proyectos.

El Shield display LCD TFT 3.5" es una pantalla táctil a colores con una resolución de 320x480 píxeles, controlador gráfico ILI9486, controlador táctil, comunicación LCD

paralela y puede mostrar hasta 262144 colores RGB distintos. Agregar el shield display LCD a nuestros proyectos con Arduino es muy sencillo, a nivel de hardware solo es necesario montar el shield sobre nuestro Arduino Uno o Mega, en cuanto a la programación con Arduino se recomienda utilizar la librería: MCFRIEND_kbv. El LCD se controla mediante 8 pines de datos (2-9) y 5 pines de control (A0-A4), la pantalla táctil comparte el bus de datos del LCD para enviar la posición. El shield display incluye además un socket para memorias micro SD, útil para almacenar imágenes en formato mapa de bits (bmp). La memoria SD se conecta por el puerto SPI al Arduino (10-13). Compatible con Arduino Uno, Mega y Due.



Figura 27. Pantalla tft 3.5”.

Especificaciones Técnicas

- Voltaje de operación: 5 VDC
- Regulador en tarjeta a 3.3 V
- Corriente de funcionamiento: 150 mA
- Controlador gráfico (driver): ILI9486
- Interface LCD: Data (8 pines) Control (5 pines)
- Interface micro SD: SPI (SS,DI,DO,SCK)
- Nivel lógico de SPI: 3.3 - 5V
- Tamaño de la pantalla (diagonal): 3.5" (89mm)
- Resolución: 320x480 píxeles
- Profundidad de color o bits por pixel: 16/18 bpp

- Cantidad de colores: 262144 colores (18-bit: R6G6B6)
- *Puede trabajar opcionalmente a RGB 16-bit: R5G6B5
- Dimensiones pantalla: 72*52mm (área visible)
- Dimensiones placa: 83*54mm
- Socket para memoria externa micro SD
- Pantalla táctil resistiva
- Controlador de pantalla con buffer de vídeo incluido

3.8.2 Base de Datos

Una base de datos es simplemente una colección de datos estructurados.

Una base de datos es un lugar en el que los datos son almacenados y organizados. La palabra «relacional» significa que los datos almacenados en el conjunto de datos son organizados en forma de tablas. Cada tabla se relaciona de alguna manera. Si el software no es compatible con el modelo de datos relacionales, simplemente se llama DBMS.

Las computadoras que tienen instalado y ejecutan el software RDBMS se llaman clientes. Siempre que necesitan acceder a los datos, se conectan al servidor RDBMS. Esa es la parte «cliente-servidor».

MySQL es una de las muchas opciones de software RDBMS. Suele pensarse que RDBMS y MySQL son lo mismo debido a la popularidad de MySQL. Para nombrar algunas aplicaciones web grandes como Facebook, Twitter, YouTube, Google y Yahoo!, todas usan MySQL para el almacenamiento de datos. Aunque inicialmente se creó para un uso limitado, ahora es compatible con muchas plataformas de computación importantes como Linux, macOS, Microsoft Windows y Ubuntu.

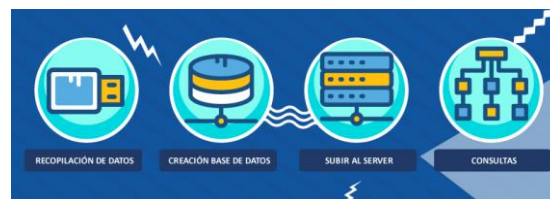


Figura 28. Base de datos

3.8.2.1 MySQL

MySQL y SQL no son lo mismo. Ten en cuenta que MySQL es una de las marcas más

populares de software RDBMS, que implementa un modelo cliente-servidor. Entonces, ¿cómo se comunican el cliente y el servidor en un entorno RDBMS? Utilizan un lenguaje específico del dominio: lenguaje de consulta estructurado (SQL, Structured Query Language). Cuando veas otros nombres que contienen SQL, como PostgreSQL y el servidor Microsoft SQL, es muy probable que sean marcas que también utilizan la sintaxis SQL. El software RDBMS a menudo se escribe en otros lenguajes de programación, pero siempre usa SQL como lenguaje principal para interactuar con la base de datos. MySQL como tal está escrito en C y C ++. Pasa como con los países sudamericanos, todos son geográficamente diferentes y tienen historias diferentes, pero todos hablan principalmente español.

- MySQL crea una base de datos para almacenar y manipular datos, definiendo la relación de cada tabla.
- Los clientes pueden realizar solicitudes escribiendo instrucciones SQL específicas en MySQL.
- La aplicación del servidor responderá con la información solicitada y esta aparecerá frente a los clientes.



Figura 29. MySQL

3.9 Mecanismo de transformación de movimiento

En estos mecanismos, el tipo de movimiento que tiene el elemento de entrada del mecanismo es diferente del tipo de movimiento que tenga el elemento de salida, es decir, el tipo de movimiento se transforma en otro distinto, de ahí el nombre de mecanismo de transformación.

Los mecanismos de transformación pueden ser, a su vez, agrupados en dos grandes grupos:

- Mecanismos de transformación circular-lineal: En este caso, el elemento de entrada tiene movimiento circular, mientras que el elemento de salida tiene

movimiento lineal.

- Mecanismos de transformación circular-alternativo: En este caso, el elemento de entrada tiene movimiento circular, mientras que el elemento de salida tiene movimiento alternativo.

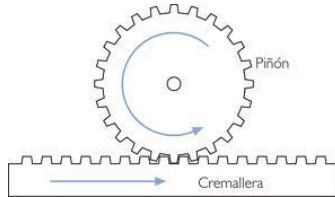


Figura 30. Mecanismo de transformación de movimiento

3.9.1 Piñón-cremallera

Este mecanismo convierte el movimiento circular de un piñón en uno lineal continuo por parte de la cremallera, que no es más que una barra rígida dentada. Este mecanismo es reversible, es decir, el movimiento rectilíneo de la cremallera se puede convertir en un movimiento circular por parte del piñón. En el primer caso, el piñón al girar y estar engranado a la cremallera, empuja a ésta, provocando su desplazamiento lineal.

Aunque el sistema es perfectamente reversible, su utilidad práctica suele centrarse solamente en la conversión de circular en lineal continuo, siendo muy apreciado para conseguir movimientos lineales de precisión (caso de microscopios u otros instrumentos ópticos como retroproyectors), desplazamiento del cabezal de los taladros sensitivos, movimiento de puertas automáticas de garaje, sacacorchos, regulación de altura de los trípodes, movimiento de estanterías móviles empleadas en archivos, farmacias o bibliotecas, cerraduras..

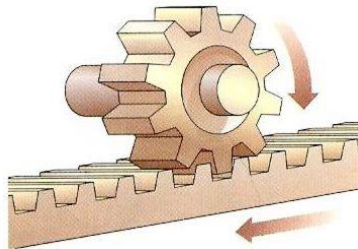


Figura 31. Mecanismo de piñón-cremallera.

3.10 Ventilador Fan

Los ventiladores de CPU o fan coolers son componentes para una única función: mantener refrigerado todo el sistema. Existen muchos elementos electrónicos dentro de una PC que generan altas temperaturas. Los fan coolers tienen la indispensable tarea de disipar el calor para mantener una temperatura ideal en el equipo y evitar un sobrecalentamiento.

Estos ventiladores tienen una gran vida útil. Hoy en día los puedes encontrar en diferentes tamaños, diseños y capacidades para que puedas escoger el que más se adapte a tu unidad.



Figura 32. Ventilador Fan.

Partes de un ventilador fan:

- Marco. Es la estructura cuadrada que envuelve al motor, por lo general es de color negro, hechos con un material plástico duro y ligero para evitar cualquier vibración que pueda generar ruidos molestos.
- Motor. Es el núcleo central que da vida al ventilador. La energía suministrada por la computadora es la que ayuda a lograr el movimiento de las aspas.
- Aspas. Son las aletas encargadas de realizar los movimientos necesarios para dispersar el aire caliente que se encuentra dentro de tu PC.

CAPÍTULO 4. DESARROLLO

4.1 Análisis de proceso de tendido y corte de telas.

Para comenzar con el desarrollo del proyecto, lo primero fue analizar y comprender el proceso que se realiza en esta área. Para llevar a cabo el análisis del proceso de tendido y corte de telas, se procedió a desarrollar un diagrama de flujo con las actividades que están involucradas para efectuar el tendido de telas, el cual se presenta a continuación:

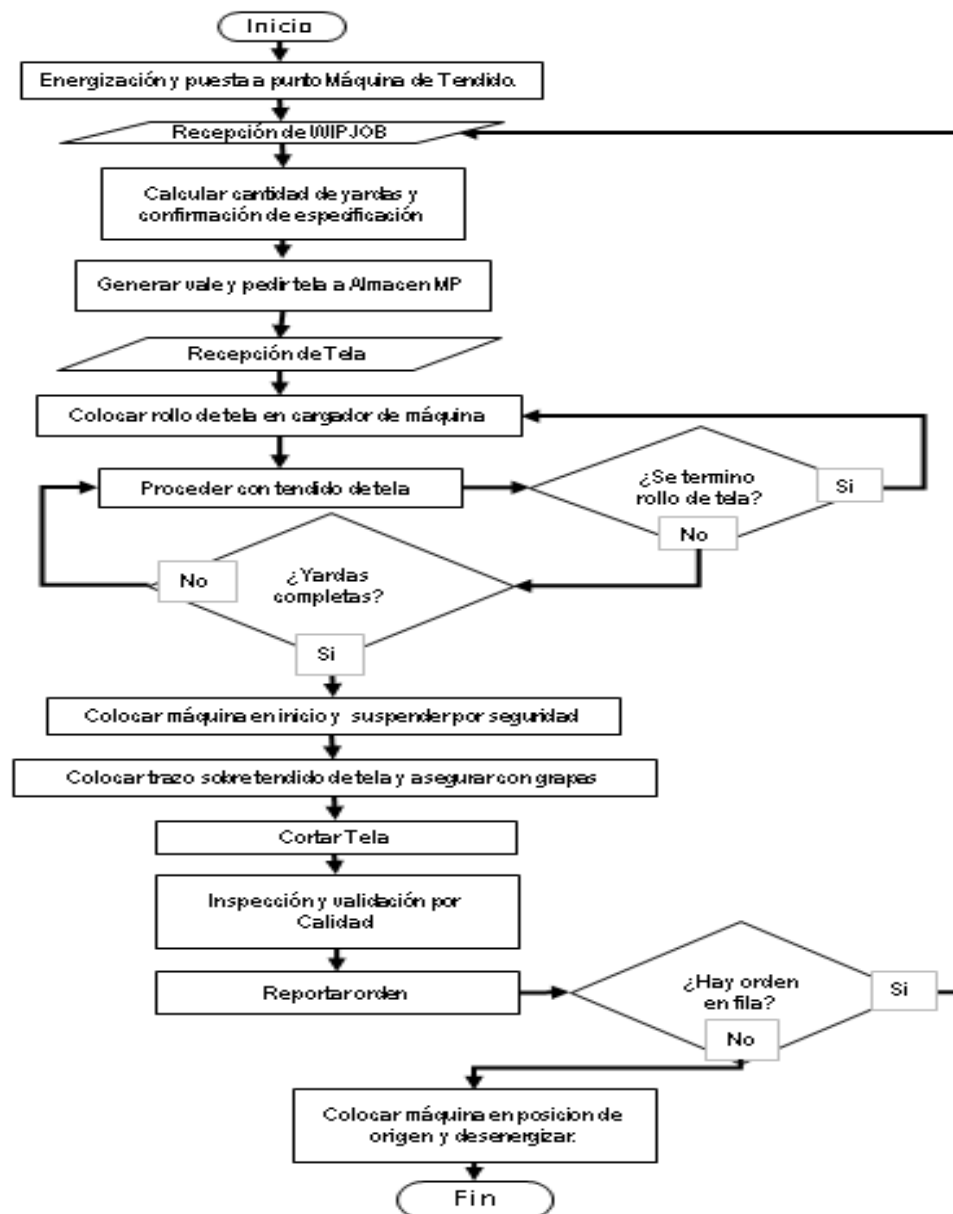


Figura 33. Diagrama de flujo analisis del proceso.

4.2 Análisis de puntos críticos

Dando seguimiento al análisis dentro del área de tendido y corte, se examinó y observo la condición en la que se presenta actualmente, de esta situación se desarrolló un análisis en la mano de obra, máquina y material, donde se logró observar algunas causas o puntos críticos dentro del área de trabajo, así mismo llegando a obtener su futura solución.

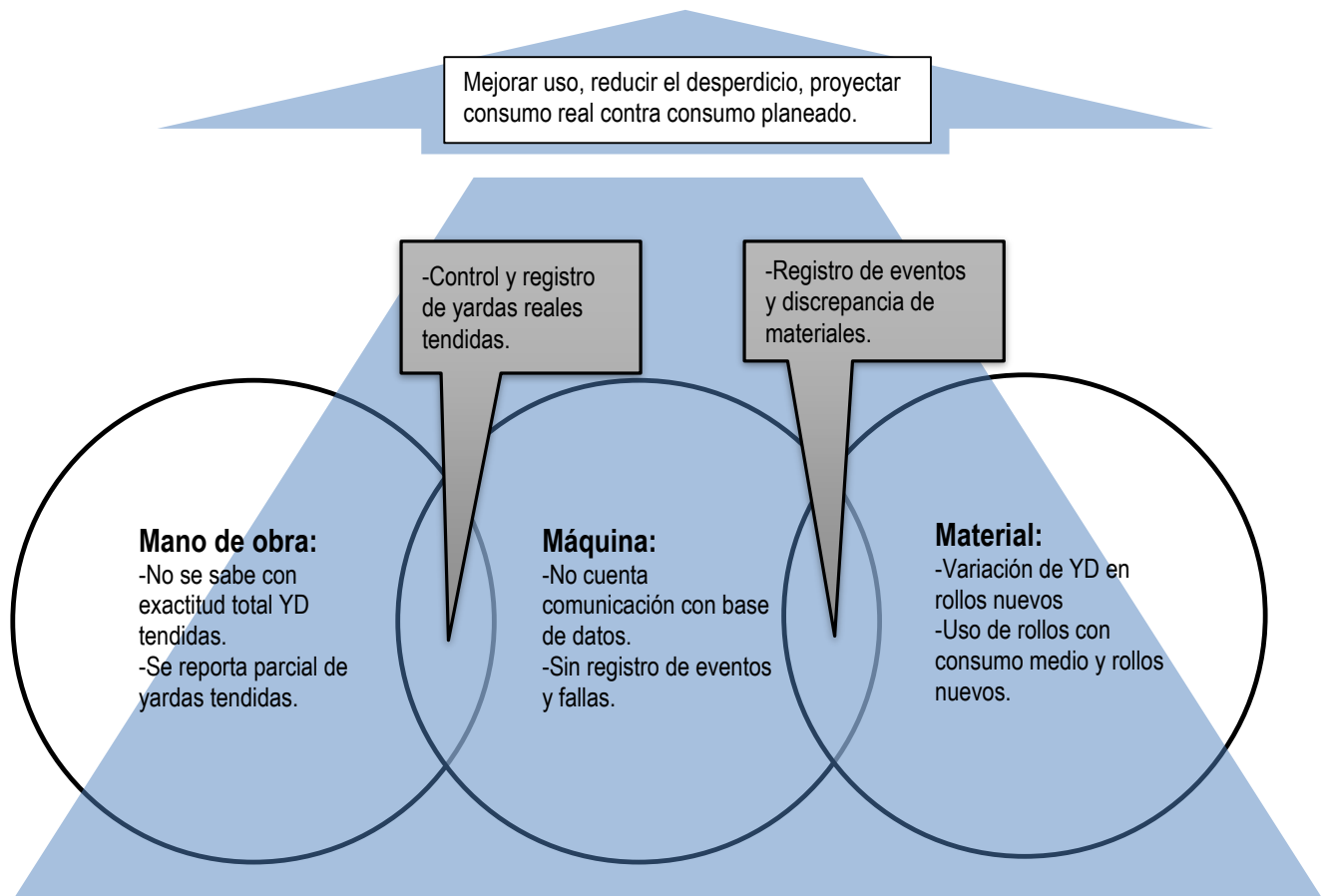


Figura 34. Análisis de 3 M.

4.3 Ideación

Para definir las necesidades del cliente y requerimientos del producto, se llevó a cabo una reunión donde se establecieron los detalles y especificaciones con las que debería de cumplir el controlador, para esto se elaboró una lluvia de ideas con parámetros que tendrían que ser implementados en el sistema.



Figura 35. Lluvia de ideas.

4.4 Desarrollo conceptual y básico

De acuerdo a los criterios establecidos en la lluvia de ideas, se generó una propuesta para el funcionamiento del controlador, diseñando un diagrama de flujo contemplando el proceso de tendido y corte.

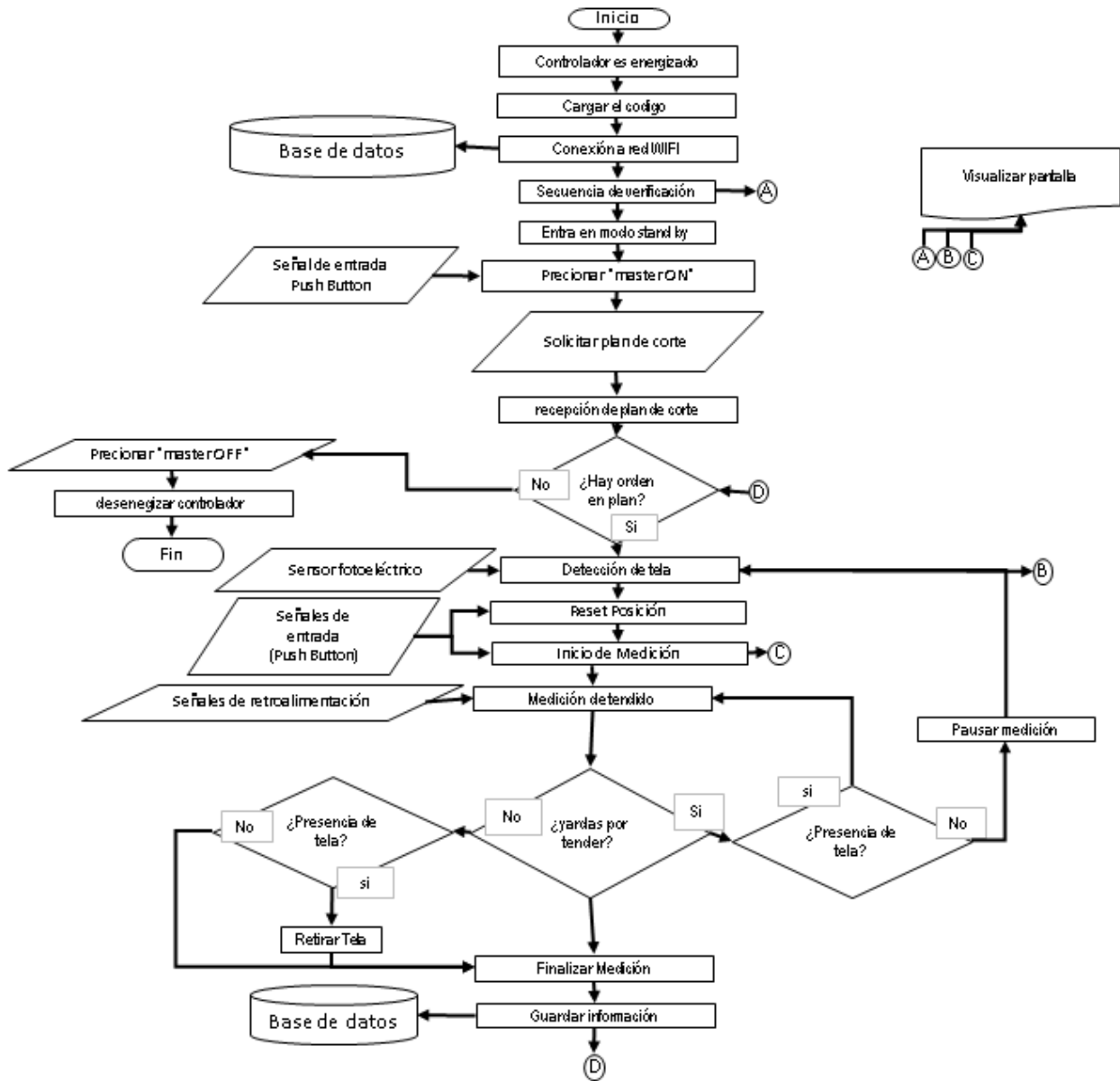


Figura 36. Diagrama de flujo.

Después de haber abordado el proceso del área, las ideas, el funcionamiento y demás aspectos, se realizó un esquema (Fig. 37) donde se seleccionaron varios componentes

los cuales teniendo como objetivo cumplir con los requerimientos y funcionamiento establecido.

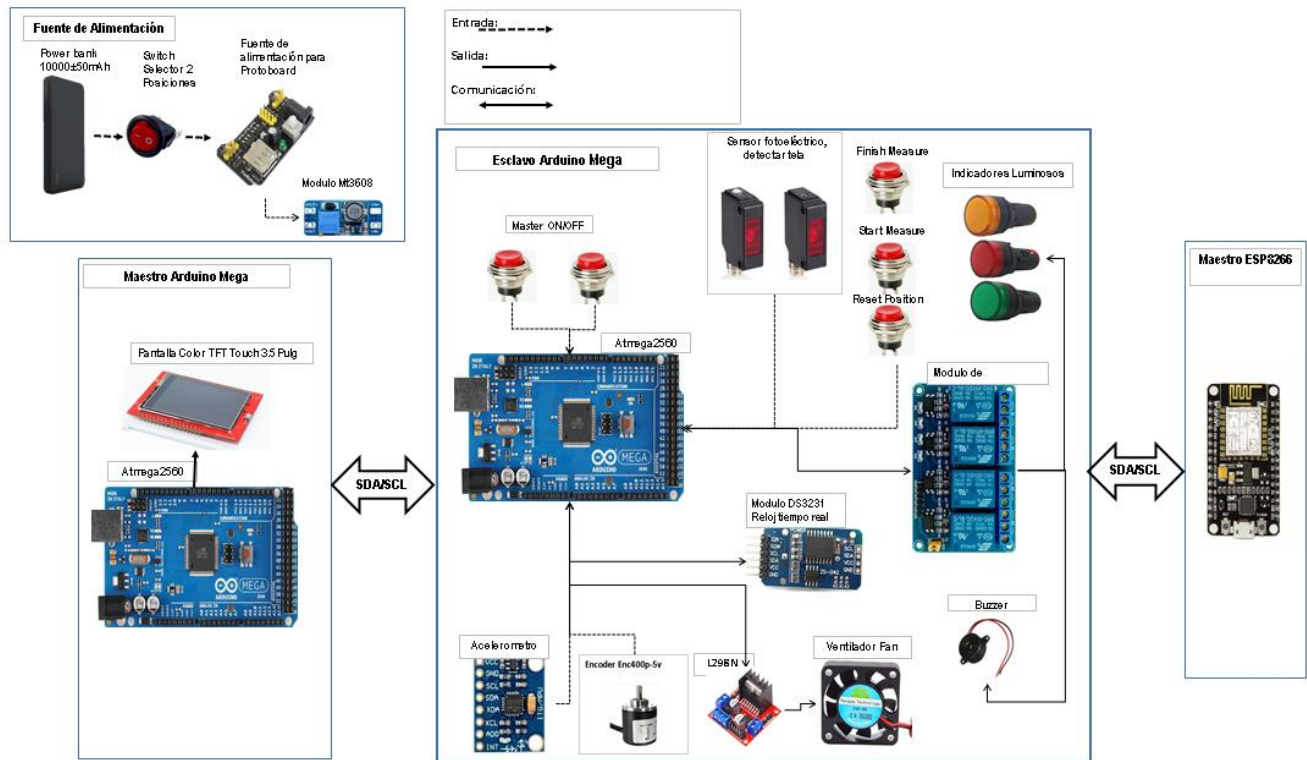


Figura 37. Esquema de componentes y su interacción.

En el esquema anterior se muestra las entradas y salidas que maneja el controlador, la fuente de alimentación usando una batería recargable, una etapa de potencia para alimentar los indicadores, los sensores para posición y detectar la presencia de tela, el encoder que será adaptado al mecanismo de piñón-cremallera y medir el recorrido del tendido de tela, diferentes push button que indicarán al controlador el estatus del proceso por parte del operador, la pantalla tft que brindara una interfaz al usuario, 2 microcontroladores Arduino Mega, una tarjeta Node MCU Esp8266 para realizar la conexión con base de datos, un módulo rtc ds3231 para la fecha y hora, un puente h l298n con ventilador para evitar calentamiento de sistema. Estos son los principales componentes y la función que desarrollaran dentro del controlador.

4.5 Desarrollo avanzado

Prototipo rápido

El prototipo del controlador se desarrolló en 3 etapas:

- Diseño de controlador.
- Programación de controlador.
- Ensamble de prototipo.
- Simulación y pruebas de prototipo.

4.5.1 Diseño de controlador:

De acuerdo al desarrollo conceptual y a la propuesta de funcionamiento establecida, se realizó el diseño del circuito electrónico de control y el diseño mecánico.

4.5.1.1 Diseño circuito electrónico de control

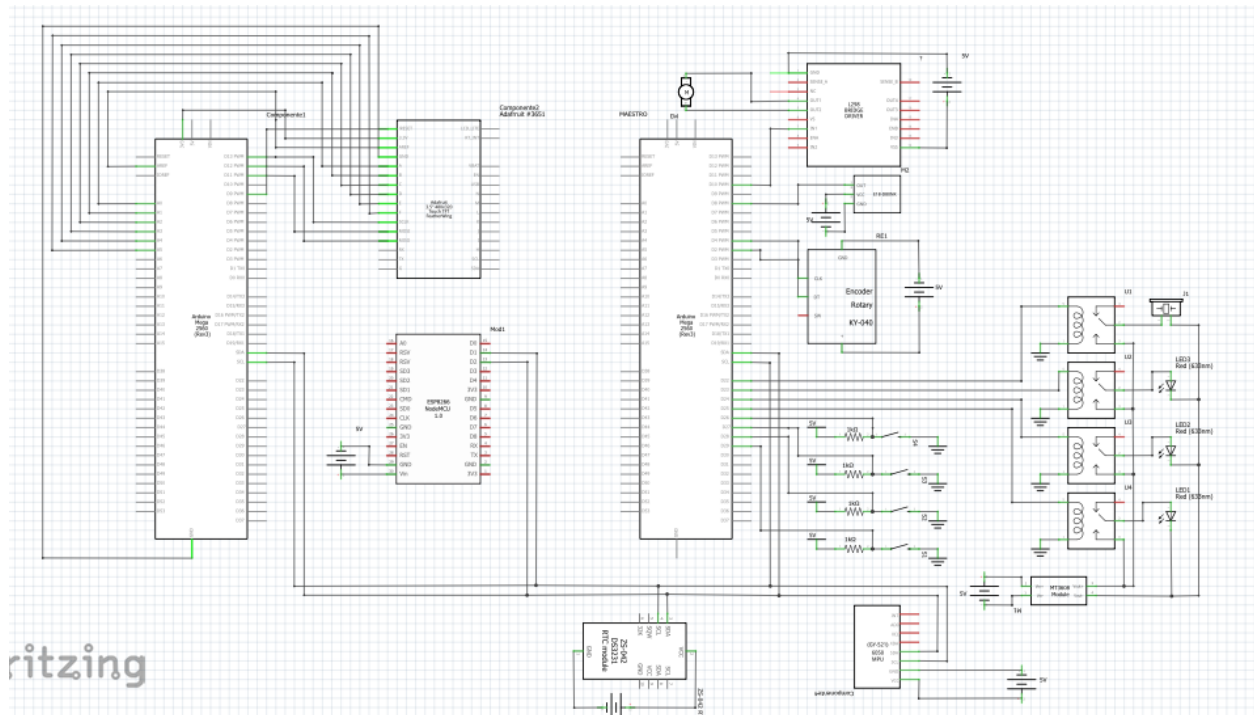


Figura 38. Circuito Electrónico.

Acorde a la idealización y requerimientos establecidos, se realizó un diseño del circuito electrónico de control que cumpliera con las funciones esperadas por el controlador,

tomando en cuenta un alto grado de eficiencia y lo más autónomo posible.

Principalmente para el diseño electrónico se buscó la manera de poder brindar una fuente de alimentación de voltaje totalmente remota y adecuada para los componentes que integraran el sistema. Optando por utilizar una batería recargable con una capacidad de 10000 mAh con un voltaje de salida de 5 volts y una corriente de 2.1 Amperios. Para que el circuito tuviera la alimentación correcta y no afectara en el funcionamiento, se implementó una fuente de protoboard reguladora de voltaje y corriente, para la etapa de potencia se seleccionó un módulo MT3608 que permite elevar el voltaje para los elementos que necesitan más de 5 volts.

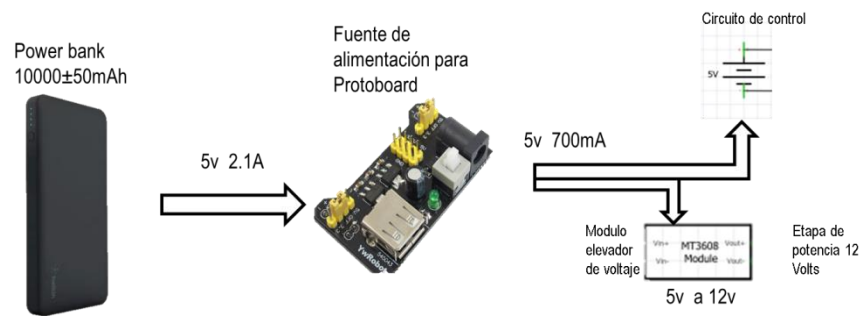


Figura 39. Fuente de alimentación.

Para el control y automatización del controlador, se seleccionaron 2 Arduino Mega 2560 y un Node MCU, estos microcontroladores estarán conectados por medio del bus I2C, siendo un Arduino 2560 y el Node MCU maestros y otro Arduino Mega 2560 esclavo.

Para realizar la medición de las yardas se optó por seleccionar un encoder que censara el desplazamiento del tendido de tela. Un sensor fotoeléctrico que reconocerá si hay presencia de tela. Un acelerómetro MPU6050 y conocer el estatus del tendido. Un RTC Ds3231 para obtener la fecha y hora en tiempo real. La implementación de Push Button que permitirán indicar al controlador el estatus del proceso de tendido por medio del operador, una pantalla TFT 3.5" haciendo la interfaz con el usuario y así mismo conocer que el controlador funciona correctamente. Los indicadores luminosos y auditivos que indicaran al usuario el estado en el que se encuentra el tendido y el controlador.

4.5.1.2 Diseño Mecánico

Para el diseño mecánico se realizó un mecanismo el cual será montado en la máquina tendedora, en donde se implementará el encoder con un sistema de banda y poleas dentadas, esto para transmitir el movimiento del piñón con la cremallera.

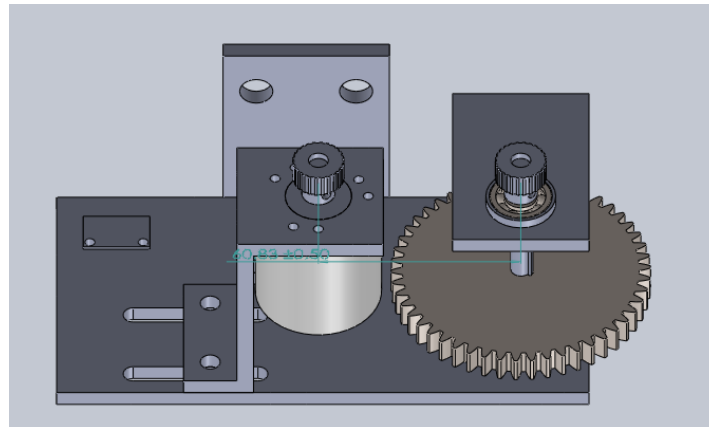


Figura 40. Diseño del mecanismo para montar encoder.

4.5.2 Programación de prototipo de controlador.

De acuerdo al funcionamiento propuesto y el circuito de control diseñado, se comenzó con la programación del controlador, traduciendo la lógica de funcionamiento del sistema completo a un lenguaje de programación (c++) dentro del entorno IDE de Arduino.

Para lograr un funcionamiento más eficiente y poder cumplir con las ideas planteadas, se realizaron 3 programas diferentes para cada microcontrolador.

Primera programación: Programación del Arduino Mega 2560, microcontrolador que funcionara como esclavo para censar las señales de los sensores utilizados y así mismo activara los indicadores, enviada los datos correspondientes a los demás microcontroladores.



```
//ESCLAVO
#include<Wire.h> //Libreria para comunicaci3n i2c
#include <RTClib.h> // incluye libreria para el manejo del modulo RTC
RTC_DS3231 rtc; // crea objeto del tipo RTC_DS3231

const int MPU = 0x69; // I2C address of the MPU-6050
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ; //Variables de acelerometro
////////////////////////////////////
float x_lat = 4000.00;
int p = 0; //variable si ha iniciado sistema
int t = 0; // variable presencia de tela
int c = 0; //variable si hay aceleracion en tendido
int i = 0; //variable si se ha iniciado el tendido
int r = 0; //variable si se ha reseteado la posici3n
int F = 0; //variable si se ha finalizado el tendido
char wy[2];
char ix[8];
char tl[2];
char ac[2];
char ic[2];
char re[2];
char fi[2];
////////////////////////////////////
int A = 2; //variable A a pin digital 2 (DT en modulo)
int B = 3; //variable B a pin digital 3 (CLK en modulo)

int ANTERIOR = 0; // almacena valor anterior de la variable POSICION

volatile int POSICION = 0; // variable POSICION con valor inicial de 0 y definida
// como global al ser usada en loop e ISR (encoder)

////////////////////////////////////Variables MASTER ON
int LED = 13;
int MasterON = 8;
int estado = 0;
int arranque = 0;
////////////////////////////////////Sensor INFRARROJO
int ledIF = 12;
int Sensor = 7;
int tela = 0;
int estadoIF = 0;
////////////////////////////////////Inicio de tendido
int ledInicio = 22;
int INICIO = 6;
int inicio = 0;
int estadoInicio = 0;
////////////////////////////////////RESET POSICION
int ResetP = 5;
int res = 0;
////////////////////////////////////Fin de tendido
int Fin = 9;
int fin = 0;
int f = 0;

int ledF = 24;
////////////////////////////////////Constantes para convertir los pulsos de encoder a yardas
const float pi = 3.1416; //pi
const float R = 3.9; //Radio de la rueda
const int N = 20; //Numero de pulsos por rueda

float distance = 0; //variable para guardar las yardas

//int buzzer=26;
//int alarm=28;

void setup() {
  Wire.begin(2); // Se inicia como el esclavo con la direcci3n 2
  Wire.onRequest(Request); // Void para para enviar datos solicitados
  //////////////////////////////////////
  pinMode(A, INPUT); // A como entrada
  pinMode(B, INPUT); // B como entrada

  Serial.begin(9600); // inicializacion de comunicacion serie a 9600 bps

  attachInterrupt(digitalPinToInterrupt(A), encoder, LOW);// interrupcion sobre pin A con
  // funcion ISR encoder y modo LOW
  Serial.println("Listo"); // imprime en monitor serial Listo
}
```

```

////////////////////////////////////// Verifica si el RTC esta conectado
if (! rtc.begin()) { // si falla la inicializacion del modulo
  Serial.println("Modulo RTC no encontrado !"); // muestra mensaje de error
  while (1); // bucle infinito que detiene ejecucion del programa
}
rtc.adjust(DateTime(__DATE__, __TIME__)); // funcion que permite establecer fecha y horario
// al momento de la compilacion. Comentar esta linea
// y volver a subir para normal operacion
//////////////////////////////////////Entradas y salidas
pinMode(MasterON, INPUT);
pinMode(LED, OUTPUT);
pinMode(ledIF, OUTPUT);
pinMode(Sensor, INPUT);
pinMode(ledInicio, OUTPUT);
pinMode(INICIO, INPUT);
pinMode(ResetP, INPUT);
pinMode(Fin, INPUT);
pinMode(ledF, INPUT);
// pinMode(buzzer, OUTPUT);
//pinMode(alarm, OUTPUT);

//////////////////////////////////////Establecer comunicaci3n con aceler3metro
Wire.beginTransmission(MPU);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);

}

void loop() {
  arranque = digitalRead(MasterON);
  estadoIF = digitalRead(Sensor);
  estadoInicio = digitalRead(INICIO);
  res = digitalRead(ResetP);
  fin = digitalRead(Fin);

  if (arranque == HIGH) {
    Serial.println("INICIO DE SISTEMA "); RTC();
    Serial.println("");
    estado = 1;
  }
  if (estado == 1) {
    p = 2;

    digitalWrite(LED, HIGH);
    if (estadoIF == LOW) {
      t = 1;
      digitalWrite(ledIF, HIGH);
      tela = 1; RTC();
      if (tela == 1) {
        Serial.println("PRESENCIA DE TELA");
        if (estadoInicio == HIGH) {
          i = 1;

          digitalWrite(ledInicio, HIGH);
          digitalWrite(ledF, LOW);
          Serial.println("");
          Serial.println("INICIO DE TENDIDO "); RTC();
          inicio = 1;
          f = 0;

          }
        }
      }
    }
  if (res == HIGH) {
    POSICION = 0;
    Serial.println("RESET POSICION"); RTC();
    i = 2;
  }
  if (estadoIF == HIGH) {
    digitalWrite(ledIF, LOW);
    Serial.println("NO HAY TELA");
    tela = 0;
    t = 0;
    alarma();
    buz();
  }
}

```

```

if (fin == HIGH) {
  f = 1;
  F = 1;
  i=3;
  Serial.println("FIN DE TENDIDO "); RTC();
  Serial.println("");
  Serial.println(distance);
}
//Lectura de Acelerómetro
Wire.beginTransmission(MPU);
Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU, 14, true); // request a total of 14 registers
AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)

Tmp = Wire.read() << 8 | Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
GyX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
GyY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
GyZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)

delay(100);

if (AcY < -300) {
  Serial.println("Tendiendo");
}

delay(100);
c = 1;
}
if (AcY > 300) {
  Serial.println("Tendiendo");
  delay(100);
  c = 1;
}
if (AcY < 300 && AcY > -300) {
  Serial.println("Tendido detenido");
  delay(500);
  c = 0;
}

}

if (f == 1) {
  digitalWrite(ledF, HIGH);
  digitalWrite(ledInicio, LOW);
  POSICION = 0;
}
delay(160);

if (POSICION != ANTERIOR) { // si el valor de POSICION es distinto de ANTERIOR
  Serial.println(POSICION); // imprime valor de POSICION
}

ANTERIOR = POSICION ;
distance = ((2 * pi * R) / N) * POSICION * 0.010936;
Serial.println(distance);
} // asigna a ANTERIOR el valor actualizado de POSICION

}

void Request() //Void enviar datos a maestros
{

//Converting float value to char
//The format (float, bytes, numbers of numbers after the decimal, char variable)
dtostrf(distance, 8, 2, ix); //Se envia las yardas tendidas
Wire.write(ix); // appx 8 bytes
Wire.write(",");

dtostrf(p, 2, 0, wy); //Se envia un 2 cuando inicia el sistema
Wire.write(wy); // appx 2 bytes
Wire.write("\n");
dtostrf(t, 2, 0, tl); //Se envia lectura del sensor de tela 1 o 0
delay(100);
}

```

```

Wire.write(tl); // appx 2 bytes
Wire.write("\n");
dtostrf(c, 2, 0, ac); //Se envia si hay aceleracion
Wire.write(ac); // appx 2 bytes
Wire.write("\n");
dtostrf(i, 2, 0, ic); //Se envia si se ha iniciado tendido
Wire.write(ic); // appx 2 bytes
Wire.write("\n");
dtostrf(r, 2, 0, re); //Si se reseteo la posicion
Wire.write(re); // appx 2 bytes
Wire.write("\n");
dtostrf(f, 2, 0, fi); //Se finalizo el tendido
Wire.write(fi); // appx 2 bytes
Wire.write("\n");
}

void encoder() //Void lectura del encoder
{
if (tela == 1 && inicio == 1 && f == 0) {
static unsigned long ultimaInterruccion = 0; // variable static con ultimo valor de
// tiempo de interrupcion
unsigned long tiempoInterruccion = millis(); // variable almacena valor de func. millis

if (tiempoInterruccion - ultimaInterruccion > 5) { // rutina antirebote desestima
// pulsos menores a 5 mseg.
if (digitalRead(B) == HIGH) // si B es HIGH, sentido horario
{
POSICION++; // incrementa POSICION en 1
}
else { // si B es LOW, senti anti horario
POSICION--; // decrementa POSICION en 1
}
}

ultimaInterruccion = tiempoInterruccion; // guarda valor actualizado del tiempo
// de la interrupcion en variable static
}
}

void RTC() //Void lectura de Reloj
{
DateTime fecha = rtc.now(); // funcion que devuelve fecha y horario en formato
// DateTime y asigna a variable fecha
Serial.print(fecha.day()); // funcion que obtiene el dia de la fecha completa
Serial.print("/"); // caracter barra como separador
Serial.print(fecha.month()); // funcion que obtiene el mes de la fecha completa
Serial.print("/"); // caracter barra como separador
Serial.print(fecha.year()); // funcion que obtiene el año de la fecha completa
Serial.print(" "); // caracter espacio en blanco como separador
Serial.print(fecha.hour()); // funcion que obtiene la hora de la fecha completa
Serial.print(":"); // caracter dos puntos como separador
Serial.print(fecha.minute()); // funcion que obtiene los minutos de la fecha completa
Serial.print(":"); // caracter dos puntos como separador
Serial.print(fecha.second()); // funcion que obtiene los segundos de la fecha completa

Serial.println(rtc.getTemperature());
}

void buz() {
digitalWrite(buzzer, LOW);
delay(500);
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
delay(500);
digitalWrite(buzzer, HIGH);
}

void alarma() {
digitalWrite(alarm, LOW);
delay(500);
digitalWrite(alarm, HIGH);
delay(500);
digitalWrite(alarm, LOW);
delay(500);
digitalWrite(alarm, HIGH);
delay(500);
}
}

```

Segunda programación: Programación del Arduino Mega 2560, microcontrolador que funcionara como maestro, en este microcontrolador se conecta y se programa la pantalla TFT 3.5", se solicitan datos censados por el microcontrolador esclavo.

```

//ARDUINO MEGA PANTALLA TFT 3.5"
//MAESTRO

#include <Wire.h>

#include <Adafruit_GFX.h> // Core graphics library
#include <Adafruit_TFTLCD.h> // Hardware-specific library
#include <TouchScreen.h>

#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;

#include "UTFT.h"
#include <UTFTGLUE.h>
UTFTGLUE myGLCD(0x9481, A2, A1, A3, A4, A0);
#include "indice.h"

// The control pins for the LCD can be assigned to any digital or
// analog pins...but we'll use the analog pins as this allows us to
// double up the pins with the touch screen (see the TFT paint example).
#define LCD_CS A3 // Chip Select goes to Analog 3
#define LCD_CD A2 // Command/Data goes to Analog 2
#define LCD_WR A1 // LCD Write goes to Analog 1
#define LCD_RD A0 // LCD Read goes to Analog 0

#define LCD_RESET A4 // Can alternately just connect to Arduino's reset pin

// Assign human-readable names to some common 16-bit color values:
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF900
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

/***** UI details */
#define BUTTON_X 52
#define BUTTON_Y 150
#define BUTTON_W 80
#define BUTTON_H 50
#define BUTTON_SPACING_X 26
#define BUTTON_SPACING_Y 30
#define BUTTON_TEXTSIZE 3

#define MINPRESSURE 10
#define MAXPRESSURE 1000

// text box where numbers go
#define TEXT_X 10
#define TEXT_Y 10
#define TEXT_W 300
#define TEXT_H 50
#define TEXT_TSIZE 2
#define TEXT_ICOLOR ILI9341_MAGENTA
// the data (phone #) we store in the textfield
#define TEXT_LEN 16
char textfield[TEXT_LEN + 1] = "";
uint8_t textfield_i = 0;

#define YP A2 // must be an analog pin, use "An" notation!
#define XM A3 // must be an analog pin, use "An" notation!
#define YM 8 // can be a digital pin
#define XP 9 // can be a digital pin

#define TS_MINX 130
#define TS_MAXX 905

#define TS_MINY 75
#define TS_MAXY 930
// We have a status line for like, is FONa working
#define STATUS_X 10
#define STATUS_Y 65

```

```

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);
// If using the shield, all control and data lines are fixed, and
// a simpler declaration can optionally be used:
// Adafruit_TFTLCD tft;

Adafruit_GFX_Button buttons[15];

int entrada; int aaa; int bbb;
int BarralEn;

String MissatgeVellP2;

void setup() {

  Wire.begin(); // join i2c bus (address optional for master)
  Serial.begin(9600); // start serial for output

  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.fillScr(VGA_WHITE); //FONDO DE PANTALLA BLANCO
  myGLCD.drawBitmap(150, 100, 200, 75, indice); //IMPRIMIR LOGO MACOMEX, DIMENSIONES, COORDENADAS
  delay(1000);
  myGLCD.clrScr();
  myGLCD.fillScr(VGA_WHITE);
  myGLCD.setTextColor(BLACK);
  myGLCD.setTextSize(4);

  myGLCD.print("Medidor de Tendidos", 15, 50);
  myGLCD.print("MACOMEX SA DE CV", 60, 130);
  myGLCD.setTextColor(RED);
  myGLCD.print("Press MASTER ON", 60, 230);
}

void loop() {
  Wire.requestFrom(2, 24); // request 20 bytes from slave device #2
  String string, string1, string2, string3, string4, string5, string6, string7;
  do // slave may send less than requested
  {
    char c = Wire.read(); // receive a byte as character

    string = string + c; //Keep saving whatever is coming

    string1 = string.substring(0, 8); //split String from 0 to 8
    string2 = string.substring(9, 11); // Split from 9 to the End of String
    string3 = string.substring(12, 14);
    string4 = string.substring(15, 17);
    string5 = string.substring(17, 19);
    string6 = string.substring(19, 21);
    string7 = string.substring(22, 24);

  } while (Wire.available());
  Serial.print("Receiving float as String...");
  Serial.println();
  Serial.println();

  Serial.print("Full String: ");
  Serial.print(string);

  Serial.println();
  Serial.print(" string 1: " );
  Serial.print(string1);
  Serial.println();

  Serial.print(" string 2: " );
  Serial.print(string2);
  Serial.println(" ");

  Serial.print(" string 3: " );
  Serial.print(string3);
  Serial.println(" ");

  Serial.print(" string 4: " );
  Serial.print(string4);
  Serial.println(" ");

  Serial.print(" string 5: " );
  Serial.print(string5);
  Serial.println(" ");

  Serial.print(" string 6: " );
  Serial.print(string6);

```

```

Serial.println(" ");

Serial.print(" string 7: ");
Serial.print(string7);
Serial.println(" ");

float d = string1.toFloat();

int p = string2.toInt(); //Inicio de sistema
Serial.println(p);

int t = string3.toInt(); //Si hay presencia de tela
Serial.println(t);

int a = string4.toInt(); //Si hay aceleración en tendido
Serial.println(a);

int r = string5.toInt(); //Si se inicio el tendido
Serial.println(r);

int i = string6.toInt(); //Si se inicio
Serial.println(i);

int f = string7.toInt(); //Si se finalizo el tendido
Serial.println(f);

switch (p)
{
  case 0 : Serial.println("");
    break;

  case 2 : Serial.println("Inicio de sistema");
    break;

}

if (p == 2) {

  if(t==1){
    tela2();
    delay(500);
    tendido();
  }else{
    tela();
  }
}

}

void tela() {
  myGLCD.clrScr();
  myGLCD.fillScr(VGA_RED);
  myGLCD.setTextColor(WHITE);
  myGLCD.setTextSize(5);
  myGLCD.print("NO", 100, 100);
  myGLCD.print("HAY", 140, 140);
  myGLCD.print("TELA", 180, 180);
}

void tela2() {
  myGLCD.clrScr();
  myGLCD.fillScr(VGA_WHITE);
  myGLCD.setTextColor(BLACK);
  myGLCD.setTextSize(5);
  myGLCD.print("HAY", 140, 140);
  myGLCD.print("TELA", 180, 180);
  delay(500);
}

void tendido(){
  myGLCD.clrScr();
  myGLCD.fillScr(VGA_WHITE);
  myGLCD.setTextColor(BLACK);
  myGLCD.setTextSize(5);
  myGLCD.print("Inicio de Tendido", 100, 100);
  myGLCD.print("Tendiendo", 140, 140);
  myGLCD.printNumF(d, 2, 180, 180); //Imprimir Yards
}

```

Tercera programación: Programación del Node MCU Esp8266, microcontrolador que funcionará como maestro, se solicitarán datos al microcontrolador esclavo, con este microcontrolador se transmiten los datos a una base de datos dentro de un servidor local.

ESP8266

```
//ESP8266 VER LOS DATOS LLEGAR
//MAESTRO

#include <ESP8266WiFi.h>
#include <Wire.h>

// Replace with your network credentials
const char* ssid = "linksys";
const char* password = "XT1986kasLOOP**";

const char* host = "172.28.8.155"; // cmd -> ipconfig
const uint16_t port = 80;

// Set your Static IP address
IPAddress local_IP(172, 28, 9, 23);
// Set your Gateway IP address
IPAddress gateway(172, 28, 8, 248);

IPAddress subnet(255, 255, 254, 0);
//IPAddress primaryDNS(8, 8, 8, 8); //optional
//IPAddress secondaryDNS(8, 8, 4, 4); //optional

String estatus;

float yard;

void setup() {

  Wire.begin(); // join i2c bus (address optional for master)
  Serial.begin(9600); // start serial for output

  // Configures static IP address
  if (!WiFi.config(local_IP, gateway, subnet)) {
    Serial.println("STA Failed to configure");
  }

  // Connect to Wi-Fi network with SSID and password
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Print local IP address and start web server
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  Wire.requestFrom(2, 24); // request 20 bytes from slave device #2
  String string, string1, string2, string3, string4, string5, string6, string7;
  do // slave may send less than requested
  {
    char c = Wire.read(); // receive a byte as character

    string = string + c; //Keep saving whatever is coming

    string1 = string.substring(0, 8); //split String from 0 to 8
    string2 = string.substring(9, 11); // Split from 9 to the End of String
    string3 = string.substring(12, 14);
    string4 = string.substring(15, 17);
    string5 = string.substring(17, 19);
    string6 = string.substring(19, 21);
    string7 = string.substring(22, 24);

  } while (Wire.available());
  Serial.print("Receiving float as String...");
  Serial.println();
  Serial.println();
  Serial.print("Full String: ");
  Serial.print(string);
}
```

```

Serial.println();
Serial.print(" string 1: " );
Serial.print(string1);
Serial.println();

Serial.print(" string 2: " );
Serial.print(string2);
Serial.println(" ");

Serial.print(" string 3: " );
Serial.print(string3);
Serial.println(" ");

Serial.print(" string 4: " );
Serial.print(string4);
Serial.println(" ");

Serial.print(" string 5: " );
Serial.print(string5);
Serial.println(" ");

Serial.print(" string 6: " );
Serial.print(string6);
Serial.println(" ");

Serial.print(" string 7: " );
Serial.print(string7);
Serial.println(" ");

yard = string1.toFloat();

int p = string2.toInt(); //Inicio de sistema
Serial.println(p);

int t = string3.toInt(); //Si hay presencia de tela
Serial.println(t);

int a = string4.toInt(); //Si hay aceleración en tendido
Serial.println(a);

int r = string5.toInt(); //Si se inicio el tendido
Serial.println(r);

int i = string6.toInt(); //Si se inicio
Serial.println(i);

int f = string7.toInt(); //Si se finalizo el tendido
Serial.println(f);

switch (p)
{
  case 0 : Serial.println("");
    break;

  case 2 : Serial.println("Inicio de sistema");
    break;
}

if (p == 2) {
  switch (t)
  {
    case 0 : Serial.println("No hay tela");
      break;

    case 1 : Serial.println("Hay tela");
      break;
  }

  switch (a)
  {
    case 0 : Serial.println("Tendido detenido");
      break;
    case 1 : Serial.println("Tendiendo");
      break;
  }
}

```

```

switch (i)
{
  case 1 : Serial.println("Inicio de Tendido");
           break;

  case 2 : Serial.println("Reset Posicion");
           break;

  case 3 : Serial.println("Fin de Tendido");
           break;
}

if (i == 1) {
  estatus = "inicio";
}
if (i == 2) {
  estatus = "Reset";
}
if (i == 3) {
  estatus = "finalizado";
}

if (a == 0) {
  estatus = "detenido";
}
if (a == 1) {
  estatus = "tendiendo";
}
}

SendDataToHost();
delay(10000);
}

void SendDataToHost(){
  Serial.print(">>>>>> connecting to ");
  Serial.print(host);
  Serial.print(':');
  Serial.println(port);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;
  if (!client.connect(host, port)) {
    Serial.println("connection failed");
    delay(5000);
    return;
  }

  String query = "esp8266/save.php?estatus=";
  query += estatus;
  query += "&y=yard=";
  query += yard;

  Serial.println(query);
  Serial.println("[Sending a request]");
  client.print(String("GET /") + query + " HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n" +
              "\r\n"
              );

  // wait for data to be available
  unsigned long timeout = millis();
  while (client.available() == 0) {
    if (millis() - timeout > 5000) {
      Serial.println(">>>> Client Timeout !");
      client.stop();
      delay(3000);
      return;
    }
  }
}

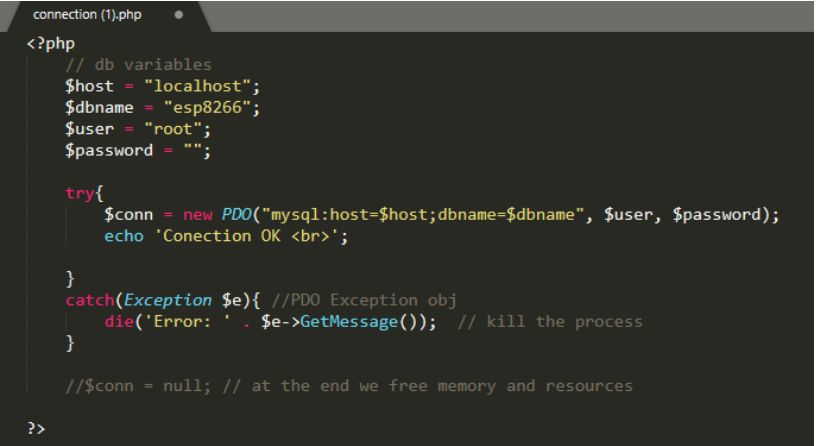
```

```
// Read all the lines of the reply from server
Serial.println("receiving from remote server");
String msg;
while (client.available()) {
  char ch = static_cast<char>(client.read());
  Serial.print(ch); // print the reply
  msg += ch;
}

Serial.println();
if (msg.indexOf("Guardado correcto!") != -1){
  Serial.println("Data Saved");
}
else{
  Serial.println("Could not save data");
}

// Close the connection
Serial.println();
Serial.println(")>>>>>>>>> closing connection");
Serial.println();
client.stop();
} // EOF SendDataToHOST
```

Usando Xampp para crear un servidor local, dentro de este servidor local y usando MySQL como base de datos se creó una tabla para guardar el registro de datos. Para lograr hacer la conexión entre la tarjeta Node MCU Esp8266 se realizaron las siguientes programaciones usando lenguaje php.



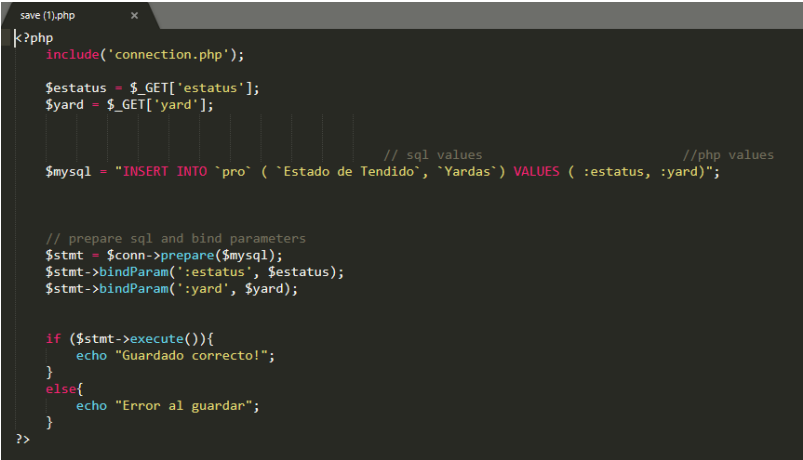
```
connection (1).php
<?php
// db variables
$host = "localhost";
$dbname = "esp8266";
$user = "root";
$password = "";

try{
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    echo 'Conection OK <br>';
}
catch(Exception $e){ //PDO Exception obj
    die('Error: ' . $e->getMessage()); // kill the process
}

//$conn = null; // at the end we free memory and resources

?>
```

Figura 41. Programación conexión entre tarjeta Node MCU y base de datos.



```
save (1).php
<?php
include('connection.php');

$status = $_GET['estatus'];
$yard = $_GET['yard'];

// sql values //php values
$query = "INSERT INTO `pro` ( `Estado de Tendido`, `Yardas`) VALUES ( :estatus, :yard)";

// prepare sql and bind parameters
$stmt = $conn->prepare($query);
$stmt->bindParam(':estatus', $status);
$stmt->bindParam(':yard', $yard);

if ($stmt->execute()){
    echo "Guardado correcto!";
}
else{
    echo "Error al guardar";
}

?>
```

Figura 42. Programación para guardar datos en tabla de base de datos

4.5.3 Ensamble de prototipo

La compra de los materiales para el ensamble del prototipo se realizó de acuerdo al presupuesto presentado por la empresa. Teniendo en cuenta el presupuesto se pudo comprar algunos materiales necesarios para el ensamble del prototipo y proceder a hacer algunas pruebas y ajustes tanto en el prototipo como en la programación.

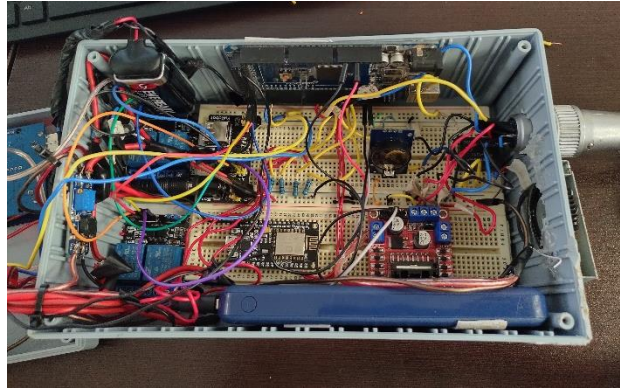


Figura 43. Ensamble de prototipo circuito interior.

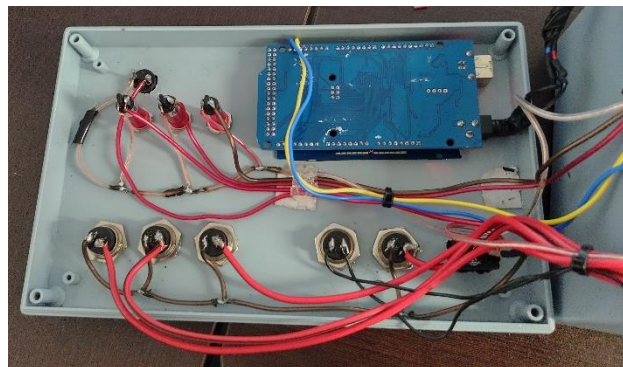


Figura 44. Ensamble de prototipo cableado.

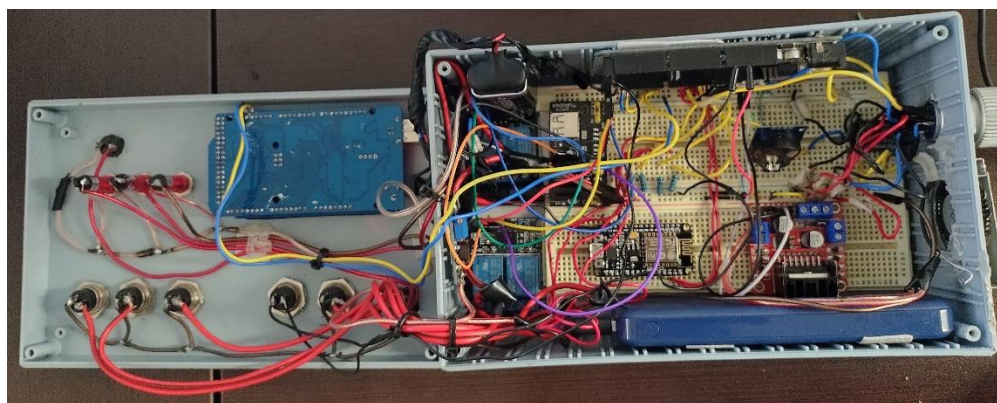


Figura 45. Ensamble total del sistema



Figura 46. Ensamble prototipo vista exterior.

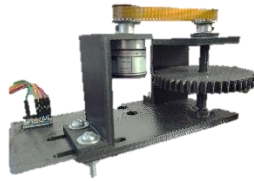


Figura 47. Ensamble del mecanismo para montar encoder.

4.5.4 Cronograma de actividades

Actividades	Mes	Febrero				Marzo				Abril				Mayo				
	Sem	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Introducción a MACOMEX. (Inducción e integración al área)		■	■															
Análisis de proceso de tendido y corte de telas			■	■	■													
Análisis de puntos críticos de proyecto.				■	■	■												
Desarrollar especificación de controlador.						■	■											
Diseñar controlador. (Diseño eléctrico, diseño mecánico, etc.)									■	■								
Revisión y liberación de diseño.										■	■							
Cotización y compra de materiales/dispositivos.											■	■						
Ensamble de prototipo de controlador.													■	■				
Programación de controlador.															■	■		
Simulación, pruebas y ajustes internos.																■	■	

CAPÍTULO 5. RESULTADOS

Dando seguimiento del capítulo anterior, se realizaron pruebas de funcionamiento del sistema, obteniendo resultados satisfactorios durante las pruebas realizadas. El funcionamiento de los sensores fue el adecuado, el registro de datos fue el correcto. Los objetivos planteados y resultados obtenidos se muestran en la siguiente tabla:

Tabla 1. Objetivos y resultados

Objetivo propuesto	Resultado esperado
Diseñar controlador	El diseño del sistema completo se desarrolló de una manera satisfactoria, cumpliendo y respetando las dimensiones de la maquina tendedora y de la mesa de tendido, para una implementación adecuada.
Manufacturar controlador	La manufactura y ensamble del prototipo del controlador, se realizó de acuerdo al diseño planteado.
Programación de controlador	La programación realizada para los microcontroladores que constituyen el sistema fue efectiva, cada microcontrolador cumplió con el objetivo esperado.
Realizar pruebas, simulación y ajustes	Las pruebas y simulaciones se elaboraron de una forma acertada, obteniendo un correcto funcionamiento del sistema.
Obtención y registro de datos.	El registro de datos fue el apropiado durante las pruebas realizadas, se logró realizar el sensado de las yardas y proceso del tendido en una base de datos.

Las pruebas de funcionamiento realizadas se hicieron simulando el proceso de tendido, obteniendo resultados esperados y apropiados respecto a la medición del tendido de la tela.

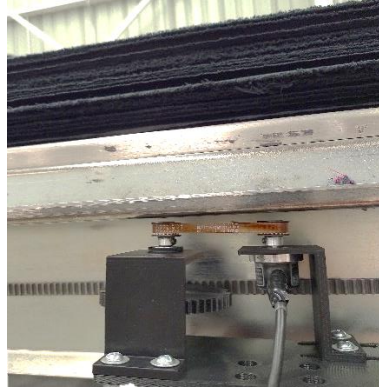


Figura 48. Realización de pruebas en mesa de tendido.



Figura 49. Pruebas con sensores retroreflectivos (detección de tela).

Los resultados de las pruebas realizadas fue la obtención de datos dentro de los mismos microcontroladores, el proceso de tendido mostrado por el controlador, el registro de yardas y proceso de tendido dentro de una tabla en la base de datos.

```

COM6
Listo
Inicio de sistema 2021-05-04 16:23:11
No hay Tela 2021-05-04 16:23:13
Hay Tela 2021-05-04 16:24:01
Inicio de Tendido 2021-05-04 16:29:07
0.00
0.00
0.01
0.01
0.01
0.01
0.01
0.01
0.01
0.01
0.02
 Autoscroll  Mostrar marca temporal
Nueva línea 9600 baudio Limpiar salida
  
```

Figura 50. Obtención de datos en los microcontroladores.



Figura 51. De izquierda a derecha, se puede observar el proceso de tendido mostrado por el controlador.

ID	Estado de Tendido	Yardas	Fecha y hora
33	inicio	0	2021-05-04 16:23:11
34	Tendido	1.09	2021-05-04 16:29:17
35	Tendido	1.09	2021-05-04 16:29:28
36	Tendido	1.09	2021-05-04 16:29:38
37	Tendido	1.09	2021-05-04 16:29:49
38	Tendido	1.09	2021-05-04 16:29:59
39	Tendido	1.09	2021-05-04 16:30:10
40	Tendido	1.09	2021-05-04 16:30:21
41	Tendido	1.09	2021-05-04 16:30:31
42	Tendido	1.3	2021-05-04 16:30:42
43	Tendido	1.96	2021-05-04 16:30:52
44	Tendido	2.22	2021-05-04 16:31:03
45	Tendido	2.22	2021-05-04 16:31:13
46	Tendido	2.22	2021-05-04 16:31:24
47	Tendido	2.22	2021-05-04 16:31:35
48	Tendido	2.22	2021-05-04 16:31:45
49	Tendido	2.22	2021-05-04 16:31:56
50	Tendido	2.22	2021-05-04 16:32:06
51	Tendido	2.22	2021-05-04 16:32:17
52	Tendido	2.22	2021-05-04 16:32:28
53	Tendido	2.22	2021-05-04 16:32:38
54	Tendido	2.22	2021-05-04 16:32:49
55	Tendido	2.22	2021-05-04 16:32:59
56	Tendido	2.48	2021-05-04 16:33:10
57	Tendido	3.35	2021-05-04 16:33:20

Figura 52. Registro de yardas y proceso de tendido en base de datos.

CAPÍTULO 6. CONCLUSIONES

En el presente documento se describe la necesidad de implementar y desarrollar procesos con mayor automatización, la aplicación de nuevas tecnologías que incrementan el avance tecnológico, facilitan el monitoreo y control de los procesos dentro de la industria que busca nuevas técnicas de supervisión y adquisición de datos en sus procesos.

La inversión e incursión en nuevas tecnologías dentro de los procesos en las industrias en general, permite lograr cambios que destinan a mejorar e innovar los procedimientos para así identificar los problemas o cuellos de botella que no dejan a las empresas avanzar en su objetivo, idear diferentes soluciones que darán solución a la problemática y así reflejar una mejora constante, controlar los procesos y todas sus variables. Así mismo al implementar la automatización da desarrollo a nuevas herramientas que favorecen a la industria a seguir progresando hacia el futuro.

Dentro de la industria textil, específicamente dentro de Maco-Mex S.A. de C.V., se identificó la oportunidad y la necesidad de implementar un sistema de monitoreo automático dentro del área donde se comienza a manufacturar su materia prima más importante que es la tela, así tener un control más eficiente en el proceso de tendido y una medición de yardas más precisa.

A causa del tiempo de estadía y tiempo para manipular la máquina tendedora, debido a la temporada alta de trabajo en la empresa, no se dio la oportunidad de finalizar la implementación del proyecto dentro del área de trabajo, aun así, con este proyecto se logró simular y observar cómo sería el monitoreo del proceso de tendido, la medición más precisa de las yardas de tela, el registro automático del estado en el que se encuentra el tendido y la medición de tela utilizada, y verificar dicho registro desde una computadora remota en una base de datos en tiempo real.

La idealización, desarrollo e implantación de estas tecnologías dentro de la industria es totalmente lo que se espera al involucrarse en este tipo de proyectos, aplicar la ingeniería usando las herramientas aprendidas y desarrolladas a lo largo de la carrera de ingeniería mecatrónica, desenvolverse en un ambiente donde se puede aplicar conocimientos como programación, diseño mecánico y eléctrico, el manejo de sensores y actuadores, todo en conjunto para formar un sistema capaz de solucionar problemas.

CAPÍTULO 7. COMPETENCIAS DESARROLLADAS

Durante el desarrollo del presente proyecto se me presentaron algunos inconvenientes y obstáculos, donde pude desarrollar diferentes competencias y poner en practica diversas habilidades que aprendí durante la carrera. Al momento de que se me presentaba algún problema siempre busque en todos los medios posibles la forma de

darle solución, y al darle solución al mismo tiempo obtenía nuevos conocimientos y aprendizajes, desarrollando aún más mis habilidades y capacidades para aplicar herramientas y métodos que involucran la ingeniería mecatrónica.

La primera competencia que aplique y logre desarrollar aún más, fue el diseño electrónico, donde tuve que diseñar el circuito de control para que el controlador cumpliera con las especificaciones planteadas y lograr el funcionamiento esperado. Otra competencia que aplique fue el diseño mecánico, donde tuve que desarrollar y adquirir más conocimientos en el tema para poder hacer un diseño eficiente y capaz de cumplir con la necesidad que se requería, principalmente al momento que diseñe el engranaje que embonara con la cremallera que se presentaba en la mesa de tendido. La competencia que más aplique y desarrolle fue en el entorno de la programación, ya que la programación es lo más importante en un proyecto y de ahí despende el correcto funcionamiento, un inconveniente que se me presento fue la comunicación que utilice para enlazar los microcontroladores y compartieran los datos en tiempo real, así mismo otro obstáculo fue al momento de hacer la comunicación entre el controlador y la base de datos ya que se requería que fuera una comunicación por red wifi pero si pude hacerlo y al mismo tiempo obtuve más aprendizajes que los que ya tenía.

Dentro del análisis de circuitos electrónicos, se derivaron varios inconvenientes al momento de hacer las conexiones entre los componentes y que no se me presentaran problemas como cortos circuitos, y así mismo también cuando adapte el circuito y cableado al espacio del gabinete presentado, ya que era más pequeño del planeado.

Al desarrollar este proyecto, se me presentaron varias situaciones donde tuve que profundizar en varios temas que aprendí durante la carrera, tuve que utilizar herramientas que había aprendido pero esta vez las tuve que poner en práctica, desenvolviéndome totalmente en mis habilidades, conocimientos y capacidades para llevar a cabo el proyecto desde un principio hasta el avance que logré obtener.

CAPÍTULO 8. FUENTES DE INFORMACIÓN

- [1] Westreicher, G.. (2020). Industria textil. marzo 28, 2021, de Economipedia Sitio web: <https://economipedia.com/definiciones/industria-textil.html>
- [2] FIUBA. (*). AUTOMATIZACION. marzo 28, 2021, de Facultad de Ingenieria - UBA Sitio web: <http://materias.fi.uba.ar/7566/Automatizacion.pdf>
- [3] Dr. E-Marmolejo, R.. (*). Microcontrolador – qué es y para que sirve. marzo 28, 2021, de HETPRO Sitio web: <https://hetpro-store.com/TUTORIALES/microcontrolador/>
- [4] Fernández, Y.. (2020). Qué es Arduino, cómo funciona y qué puedes hacer con uno. marzo 29, 2021, de XATAKA Sitio web: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- [5] García, A.. (2013). Arduino Mega: Características, Capacidades y donde conseguirlo en Panamá. marzo 29, 2021, de PANAMAHITEK Sitio web: <http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>
- [6] del Valle, L.. (2017). NodeMCU tutorial paso a paso desde cero. marzo 29, 2021, de ProgramarFacil Sitio web: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>
- [7] Llamas, L.. (2016). El bus I2C en Arduino. marzo 29, 2021, de Luis Llamas Ingeniería, informática y diseño Sitio web: <https://www.luisllamas.es/arduino-i2c/>
- [8] Crespo, E.. (2015). Aprendiendo Arduino. marzo 29, 2021, de wordpress Sitio web: <https://aprendiendoarduino.wordpress.com/2015/03/30/sensores/>
- [9] DEWESoft. (2020). ¿Qué es un Sensor y Qué Hace?. marzo 29, 2021, de DEWESoft Sitio web: <https://dewesoft.com/es/daq/que-es-un-sensor#digital-sensors>
- [10] Demaquinasyherramientas. (2017). ¿Qué es un encoder, cuáles son sus tipos y para qué sirven?. marzo 30, 2021, de de maquinas y herramientas Sitio web: <https://www.demaquinasyherramientas.com/mecanizado/encoder-tipos>
- [11] Cruz, D.. (2021). ¿Cómo funcionan los sensores fotoeléctricos/fotoceldas/fotobeams?. mayo 4, 2021, de tecnosinergia Sitio web: <https://tecnosinergia.zendesk.com/hc/es/articles/360001380312--Como-funcionan-los-sensores-fotoelectricos-fotoceldas-fotobeams->
- [12] DEWESoft. (2020). ¿Qué es un Sensor y Qué Hace?. marzo 30, 2021, de DEWESoft Sitio web: <https://dewesoft.com/es/daq/que-es-un-sensor#digital-sensors>

- [13] García, V.. (2015). Sensor MPU6050.. marzo 30, 2021, de Electrónica Práctica Aplicada Sitio web: <https://www.diarioelectronico hoy.com/blog/sensor-mpu6050>
- [14] García, S.. (2020). Cómo conectar un Push Button a Arduino. marzo 30, 2021, de 330ohms Sitio web: <https://blog.330ohms.com/2020/06/06/como-conectar-un-push-button-a-arduino/>
- [15] Llamas, L.. (2016). Reloj y calendario en Arduino con los RTC DS1307 y DS3231. marzo 30, 2021, de LuisLlamas Sitio web: <https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/>
- [16] Fernandez, O.. (2016). Arduino circuito de portencia. marzo 31, 2021, de Código Electrónica Sitio web: <http://codigoelectronica.com/blog/arduino-circuito-de-portencia>
- [17] Robots Didácticos. (2019). Convertidor de voltaje MT3608. marzo 30, 2021, de Robots Didácticos Sitio web: <http://robots-argentina.com.ar/didactica/convertidor-de-voltaje-mt3608/>
- [18] Tolocka, E.. (2015). Módulo de 4 relés para Arduino. marzo 30, 2021, de Profe Tolocka Sitio web: <https://www.profetolocka.com.ar/2015/05/09/modulo-de-4-reles-para-arduino/>
- [19] Cruz, A.. (2014). Tutorial: Uso de Driver L298N para motores DC y paso a paso con Arduino . marzo 30, 2021, de ELECTRONILAB Sitio web: <https://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>
- [20] Alarmas Acústicas y Visuales. (2016). La importancia de las luces indicadoras en los procesos industriales . marzo 30, 2021, de Alarmas Acústicas y Visuales S.A. de C.V. Sitio web: http://alarmasacusticas.blogspot.com/2014/01/la-importancia-de-las-luces-indicadoras_30.html
- [21] Corrales, J.. (2019). Interfaz de usuario o UI: ¿qué es y cuáles son sus características?. marzo 31, 2021, de rockcontent Sitio web: <https://rockcontent.com/es/blog/interfaz-de-usuario/>
- [22] Naylamp Mechatronics. (*). SHIELD DISPLAY LCD TFT 3.5" TÁCTIL.

marzo 31, 2021, de Naylamp Mechatronics SAC Sitio web: <https://naylampmechatronics.com/ardusystem-shields/360-shield-display-lcd-tft-35-tactil.html#:~:text=El%20Shield%20display%20LCD%20TFT,hasta%20262144%20colores%20RGB%20distintos.>

[23] Gustavo B.. (2020). ¿Qué es MySQL? Explicación detallada para principiantes. marzo 31, 2021, de HOSTINGER TUTORIALES Sitio web: <https://www.hostinger.mx/tutoriales/que-es-mysql>

[24] Fajardo, B., Cabrera, G., Pulido, A., & Zamorin, M.. (2008). Mecanismos de transformación del movimiento. Marzo 31, 2021, de APRENDAMOS TECNOLOGIA Sitio web: <https://aprendemostecnologia.org/maquinas-y-mecanismos/mecanismos-de-transformacion-del-movimiento/>

[25] Fajardo, B., Cabrera, G., Pulido, A., & Zamorin, M.. (2008). Mecanismos de transformación del movimiento. Marzo 31, 2021, de APRENDAMOS TECNOLOGIA Sitio web: <https://aprendemostecnologia.org/maquinas-y-mecanismos/mecanismos-de-transformacion-del-movimiento/>

[26] Amer, F.. (2019). Definición de ventiladores para CPU. Abril 01, 2021, de Techlandia Sitio web: https://techlandia.com/definicion-ventiladores-cpu-hechos_100466/

CAPÍTULO 9. ANEXOS

Carta de termino de residencias profesionales.



COSIO, AGS; A 02 DE JUNIO DEL 2021.

TECNOLOGICO NACIONAL DE MEXICO,
INSTITUTO TECNOLOGICO DE PABELLON DE ARTEAGA,
DEPTO. DE GESTION TECNOLOGICA Y VINCULACION,
AT'N LIC. MA. MAGDALENA CUEVAS MARTINEZ,
P R E S E N T E :

Por medio de la presente una vez más manifestamos nuestro apoyo a los programas educativos encaminados a la formación del estudiante, así como en el reflejo que se da hacia las Empresas por el servicio que prestan al desarrollar proyectos de interés y de aprovechamiento para las mismas, ante ello tenemos a bien comunicar que el estudiante **ISIDRO SANTILLAN ZAMARRIPA** con número de control 161050128, del Programa Educativo de "INGENIERIA MECATRONICA", ha cumplido satisfactoriamente con el desarrollo del proyecto MEDIDOR DE TENDIDOS de residencias profesionales dentro del periodo proyectado de CUATRO A SEIS MESES, cumpliendo con las 500 horas que conformaron el requisito para su liberación.

Así mismo manifestamos que fue desarrollado el proyecto "MEDIDOR DE TENDIDOS", para el desarrollo de la RESIDENCIA PROFESIONAL correspondiente por el estudiante residente de referencia.

En espera de haber facilitado las actividades para el desarrollo y aprovechamiento del proyecto de residencia para el alumno, quedamos de ustedes como sus atentos y seguros servidores.

ATENTAMENTE

L.A.E. MANUEL DONATO DE LUNA.
GERENTE DE RECURSOS HUMANOS.

MACO-MEX INDUSTRIAL S.A. DE C.V.
AV. MIGUEL HIDALGO S/N
R.F.C. MMI 910227JY8
COSIO, AGS.
TELS. 456 96 79062 • 456 96 70917