UNMDP-FI

Carrera: Ingeniería Electrónica

Informe de Proyecto Final

EROSIS

Sistema de pulido automático con interfaz IoT

Gerino, Franco Gabriel

Revelli, Esteban Nahuel

Marzo 2025

RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.
Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.
A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios

Esta obra está bajo una Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional.

UNMDP-FI

Carrera: Ingeniería Electrónica

Informe de Proyecto Final

EROSIS

Sistema de pulido automático con interfaz IoT

Gerino, Franco Gabriel

Revelli, Esteban Nahuel

Marzo 2025

Agradecimiento

Quisiéramos expresar nuestro más sincero agradecimiento a todas las personas que, con su apoyo, orientación y dedicación, hicieron posible la culminación de este trabajo final de grado.

En primer lugar, agradecemos profundamente a nuestros directores de tesis, Ing. Melisa Gisele Kuzman y Ing. Juan Manuel Lopez, por su guía, paciencia y consejos durante todo el proceso. Su compromiso y conocimiento nos orientaron en cada etapa, ayudándonos a enfrentar desafíos y a alcanzar nuestras metas con éxito.

De igual manera, queremos expresar nuestra gratitud a nuestras familias, quienes nos brindaron su comprensión y aliento incondicional. Gracias por su apoyo y por su confianza en nosotros en cada paso de este arduo camino. Su constante presencia fue fundamental para mantenernos motivados y enfocados.

Finalmente, extendemos nuestro agradecimiento al LIVRA (Laboratorio de Instrumentación Virtual y Robótica Aplicada) de la Universidad Nacional de Mar del Plata, cuyo ambiente de investigación y trabajo fue clave en el desarrollo de nuestro trabajo. Gracias a su disposición y recursos, pudimos llevar a cabo las pruebas y análisis necesarios para la realización de este proyecto.

A todos, muchas gracias por hacer posible este logro.

Contenido

1. *Introducción* **4**

1.1. Propósito del documento 4
1.2. Alcance del proyecto 4
1.3. Personal involucrado 5

2. *Anteproyecto* **6**

2.1. Requerimientos 6
2.2. Plan de proyecto y cronograma 6

3. *Proyecto* **8**

3.1. Descripción del sistema 8
3.2. Interfaces de usuario 9
3.3. Interfaz Física 9
3.3.1. Modelo de Menú 9
3.3.2. Esquema de Botones 10
3.3.3. Indicadores Visuales 11
3.3.4. Estructura de Estados y Tareas 12
3.4. Interfaz Web 13
3.5. Desarrollo de Software 15
3.5.1. Microcontrolador de comunicación 15
3.5.2. Microcontrolador de control 15
3.6. Desarrollo de Hardware 28
3.6.1. Detector de Cruces por Cero 28
3.6.2. Regulación de tensión por ángulo de conducción 30
3.6.3. Tacómetro 33
3.6.4. Sensado de Corriente 37
3.6.5. Regulador de Tensión 39

4. *Resultados* **41**

5. *Conclusiones* **44**

5.1. Conclusiones del Proyecto 44
5.2. Gestión del proyecto 44

6. ***Bibliografía*** **46**

7. ***Anexos*** **48**

7.1. Plan de Proyecto 48
7.2. Especificación de Requerimientos 48
7.3. Especificación Funcional 48
7.4. Especificación Técnica 48
7.5. Plan de Pruebas 48
7.6. Códigos de programación 48
7.7. Kicad 48
7.8. Manual de Usuario 48

1. *Introducción*

Este documento constituye el Informe Final del Proyecto de Grado titulado "Sistema de Pulido Automático con Interfaz IoT". El informe contiene de manera detallada el proceso de resolución del proyecto, los objetivos planteados, los desafíos identificados a lo largo de su desarrollo y las soluciones implementadas para superarlos.

1.1. Propósito del documento

El presente documento tiene como objetivo proporcionar información detallada sobre la resolución del proyecto, los objetivos alcanzados, los requerimientos planteados, las especificaciones funcionales y técnicas del mismo, así como las conclusiones obtenidas a lo largo de su desarrollo. Está dirigido a los directores del proyecto, a la cátedra de “Trabajo Final de Ingeniería Electrónica” y a la empresa “Señal de jugar”, quien es la proveedora de la idea del proyecto y espera la solución para poder disminuir el problema que presenta a la hora de lijar las piezas de madera para sus juguetes.

1.2. Alcance del proyecto

Se busca desarrollar el prototipo de una máquina para pulir piezas de madera. La idea del dispositivo es que sea capaz realizar la tarea de pulido de forma automática. Para ello, el prototipo debe ser capaz de pulir madera del tipo MDF de diferentes tamaños, por lo que debe contar con la posibilidad de elegir entre diferentes velocidades y tiempos de trabajo. El dispositivo debe incluir un panel físico en la máquina, y un panel virtual accesible desde la red interna de la empresa, que mostrarán los parámetros más importantes de funcionamiento. Ambos elementos forman parte de los requerimientos del proyecto, permitiendo al usuario programar la velocidad y el tiempo de trabajo, y consultar el estado del dispositivo desde cualquier lugar dentro de la empresa.

1.3. Personal involucrado

Nombre	Gerino, Franco Gabriel
Rol	Encargado del proyecto
Categoría Profesional	Estudiante
Responsabilidad	Diseño, implementación y pruebas de funcionamiento
Información de contacto	fgerinog@gmail.com

Nombre	Revelli, Esteban Nahuel
Rol	Encargado del proyecto
Categoría Profesional	Estudiante
Responsabilidad	Diseño, implementación y pruebas de funcionamiento
Información de contacto	estebannrevelli@gmail.com

Nombre	Kuzman, Melisa Gisele
Rol	Directora del proyecto
Categoría Profesional	Ingeniera Electrónica (UNMdP, 2014)
Responsabilidad	Asistencia técnica en desarrollo e implementación
Información de contacto	melisakuzman@fi.mdp.edu.ar

Nombre	Lopez, Juan Manuel
Rol	Co-Director
Categoría Profesional	Ingeniero Electrónico (UNMdP, 2015)
Responsabilidad	Asistencia técnica en desarrollo e implementación
Información de contacto	juanml@fi.mdp.edu.ar

2. *Anteproyecto*

El presente proyecto tiene como objetivo el desarrollo de un prototipo de una máquina capaz de pulir piezas de madera del tipo MDF. El sistema propuesto contempla la creación de un prototipo con dos interfaces: una física, integrada en el dispositivo, y otra virtual, accesible a través de una aplicación web. Esto incluye el diseño e implementación del hardware y software necesarios para garantizar el correcto funcionamiento del dispositivo, así como las pruebas correspondientes para validar su desempeño.

En esta sección se presentan los requerimientos y especificaciones del proyecto, junto con una descripción del plan de trabajo. Para obtener detalles más específicos sobre estos aspectos, se recomienda consultar los documentos anexos al presente informe, titulados "Especificación de Requerimientos" y "Plan de Proyectos".

2.1. Requerimientos

La función principal del prototipo es el pulido de piezas de madera del tipo MDF, las cuales pueden variar en tamaño y forma. Para adaptarse a estas variaciones, el dispositivo debe ofrecer distintas velocidades de operación y la posibilidad de programar el tiempo de trabajo. El dispositivo debe comunicar a través de una aplicación web y un panel físico información de la velocidad de operación, tiempo transcurrido y restante de trabajo.
El prototipo debe contar con dos interfaces, una física integrada en el dispositivo y otra remota implementada en una aplicación web, las cuales permitan programar la velocidad y el tiempo de trabajo, modificar el programa durante su ejecución, guardar y utilizar programas predefinidos.
Además, el dispositivo debe estar equipado con un botón de emergencia que permita detener su funcionamiento de manera inmediata en caso de necesidad.

2.2. Plan de proyecto y cronograma

La idea de este proyecto surgió a finales de 2022, a partir de una problemática planteada por la empresa "Señal de Jugar", dedicada a la fabricación de juguetes de madera para niños de la primera infancia. Esta empresa realiza el pulido de las piezas de manera manual, lo que representaba un proceso lento y laborioso, por lo que se planteó la necesidad de automatizar dicha tarea.

El proyecto fue presentado como una posible propuesta para el Trabajo Final de Grado

en el Laboratorio de Instrumentación Virtual y Robótica Aplicada (LIVRA) de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. Tras varias reuniones con los directores del laboratorio, se comenzó a trabajar en el desarrollo del dispositivo.

El desarrollo del trabajo comenzó en paralelo con las cursadas y los exámenes finales de las últimas materias de la carrera de Ingeniería Electrónica durante el año de 2023. A lo largo de ese año se desarrollaron los primeros documentos necesarios para el proyecto, en el cual uno de los destacados fue el “Plan de Proyecto”, que identifica las fechas mas importantes del mismo. Este documento se encuentra anexado al presente informe, describiendo etapas, tareas, recursos y tiempos estimados.

3. *Proyecto*

Durante el desarrollo del proyecto se realizaron investigaciones, diseños, mejoras y pruebas para las distintas etapas. En este capitulo se mencionarán los aspectos mas relevantes del trabajo. En caso de necesitar mas detalles se recomienda leer los documentos “Especificación de Requerimientos”, “Especificación Funcional”, “Especificación Técnica” y “Plan de Pruebas”.

3.1. Descripción del sistema

En la Figura 1, que se muestra a continuación se puede observar un esquemático general de la solución propuesta y cómo se comunica cada etapa del dispositivo.

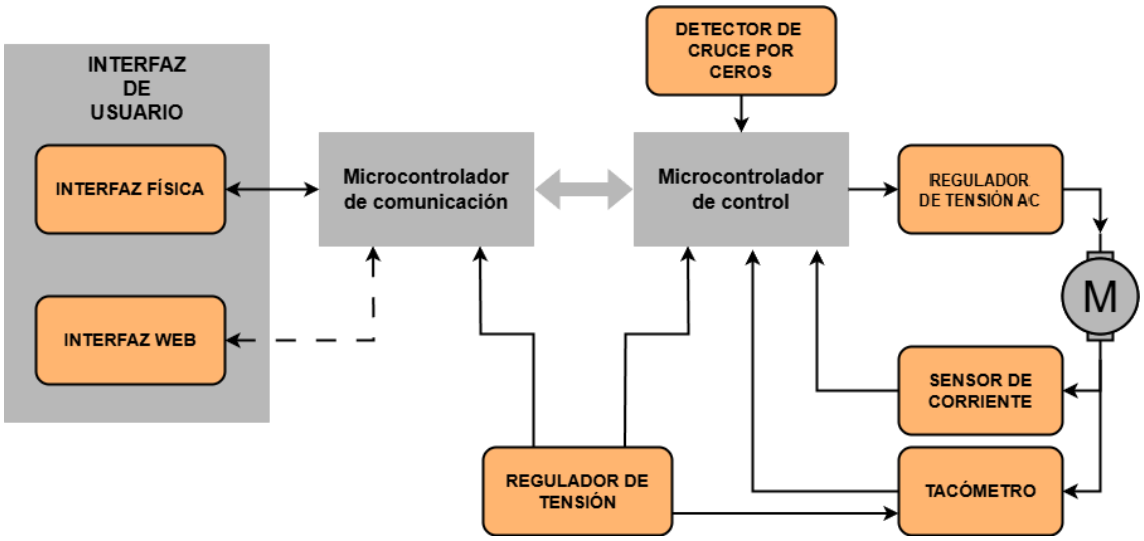

Figura 1: Esquemático general de la solución

Como se puede observar en la Figura 1, la primera etapa está compuesta por las *Interfaces de Usuario*, que incluyen tanto la interfaz física como la interfaz web. Estas interfaces permiten programar la velocidad y el tiempo de funcionamiento del dispositivo, así como ofrecer opciones como detener el proceso, continuar su funcionamiento, finalizar el programa actual y utilizar programas predefinidos.

La siguiente etapa corresponde al Microcontrolador de Comunicación, el cual se encarga de gestionar la comunicación entre las dos interfaces y la etapa de control. Posteriormente, tenemos el Microcontrolador de Control, que recibe la información de velocidad y tiempo de trabajo. Utilizando estos datos junto con un control PI y las señales generadas por el hardware, permite que la máquina funcione a la velocidad programada. Además,

se incluye un sistema de sensado de corriente que permite monitorear la corriente que circula por el motor, con el fin de prevenir posibles excesos en el consumo que podrían provocar una falla en el motor.

3.2. Interfaces de usuario

El control del dispositivo, integra una interfaz física compuesta por botones y un conjunto de indicadores visuales, así como una interfaz remota basada en una aplicación web. A través de estas herramientas, el usuario puede configurar parámetros como velocidad y tiempo de trabajo, observar el estado de la máquina, y gestionar programas predefinidos de forma intuitiva y eficiente.

3.3. Interfaz Física

La interfaz física incluye botones dedicados a la navegación, ajuste de parámetros, confirmación de acciones y manejo de emergencias, permitiendo un control preciso y seguro del dispositivo. Adicionalmente, los indicadores visuales (LEDs y pantalla LCD) proporcionan realimentación en tiempo real sobre el estado del sistema, ayudando al usuario a identificar conexiones activas, condiciones de operación y posibles fallos. Esta interfaz es completamente autónoma, permite manejar el prototipo en el caso que las comunicaciones inalámbricas fallen.

3.3.1. Modelo de Menú

La interfaz física sigue el esquema de un menú de opciones para que el manejo del prototipo sea lo mas simple e intuitivo posible. El menú cuenta con opciones que permiten modificar parámetros de trabajo y envían comandos de funcionamiento hacia el siguiente microcontrolador.

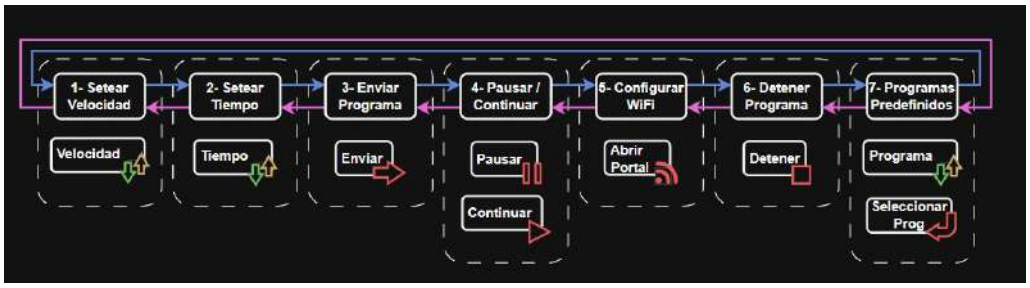

Figura 2: Menú de Opciones

Opción del Menú	Descripción de Funcionalidades
1. Setear Velocidad	Permite ajustar la velocidad para un programa nuevo.
2. Setear Tiempo	Permite ajustar el tiempo de ejecución para un programa nuevo.
3. Enviar Programa	Envía los parámetros seleccionados al motor para iniciar el programa.
4. Pausa/Play	Pausa el programa en ejecución o lo reanuda si está pausado.
5. Configuración WiFi	Permite cambiar las credenciales de red WiFi del dispositivo.
6. Detener Programa	Finaliza el programa actual, incluso si está pausado.
7. Predefinidos	Permite seleccionar y cargar programas almacenados en memoria.

Cuadro 1: Resumen de funcionalidades de las opciones del menú de control.

3.3.2. Esquema de Botones

Cada botón responde a un tipo de funcionalidad para el manejo de un sistema de menú de opciones. Se incluye un botón de emergencia no relacionado al menú, que corta el suministro de red como medida de seguridad.

Figura 3: Esquema de Botones e Indicadores visuales

Botón	Descripción de Funcionalidades
Navegación Arriba	Permite moverse hacia la opción anterior dentro del menú.
Navegación Abajo	Permite moverse hacia la siguiente opción del menú.
Incrementar	Incrementa el valor de una opción seleccionada.
Decrementar	Decrementa el valor de una opción seleccionada.
Confirmar	Confirma la selección o aplica cambios en el menú. Presionar para validar la opción actual.
Emergencia	Botón con retención. Corta el suministro de tensión completo de la máquina. Presionar para desactivar el prototipo de manera inmediata.

Cuadro 2: Descripción de los botones principales del panel de control.

3.3.3. Indicadores Visuales

- **Indicador LED Wi-Fi**: Proporciona retroalimentación visual sobre el estado de la conexión Wi-Fi. Encendido indica que hay conexión WiFi, apagado indica que no

tiene conexión establecida.

- **Indicador LED Neopixel**: Proporciona realimentación visual sobre el estado y la velocidad de la máquina.
- **Pantalla LCD**: Es el principal indicador visual. Encargado de dar información del menú y estado de la maquina, las modificaciones sobre los programas, recepción de mensajes MQTT y presentación de errores.

3.3.4. Estructura de Estados y Tareas

El diseño del menú se basa en la estructura de estados que puede adoptar el prototipo: **Preparado, Funcionando y Pausado**. En el estado "Preparado", el dispositivo está listo para recibir instrucciones y configurar un programa; en "Funcionando", se encuentra ejecutando activamente un programa; y en "Pausado", permanece a la espera de órdenes para reanudar o modificar su operación.

Cada estado delimita y habilita el acceso a diferentes tareas, las cuales determinan el comportamiento actual o futuro de la máquina. Estas tareas permiten transiciones entre estados y se clasifican en dos tipos: pasivas, que no afectan el estado de inmediato, y activas, que implican cambios directos en el funcionamiento del dispositivo.

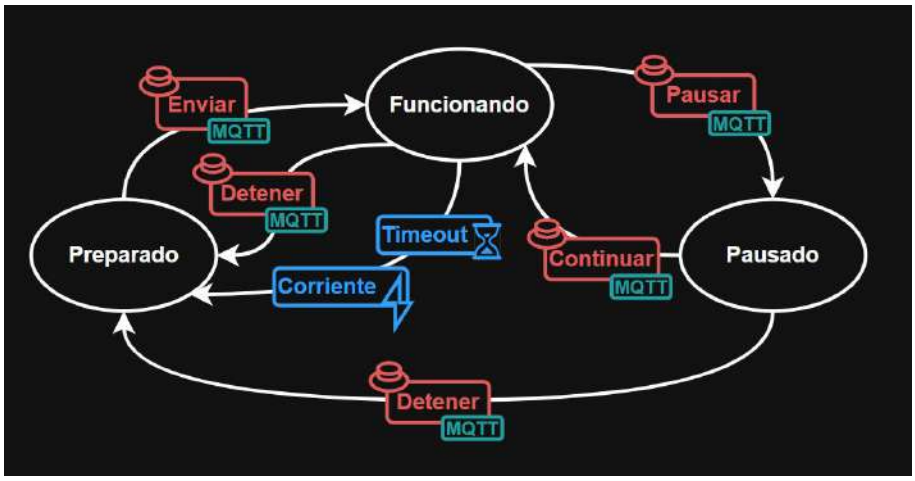

Figura 4: Transicion de estados

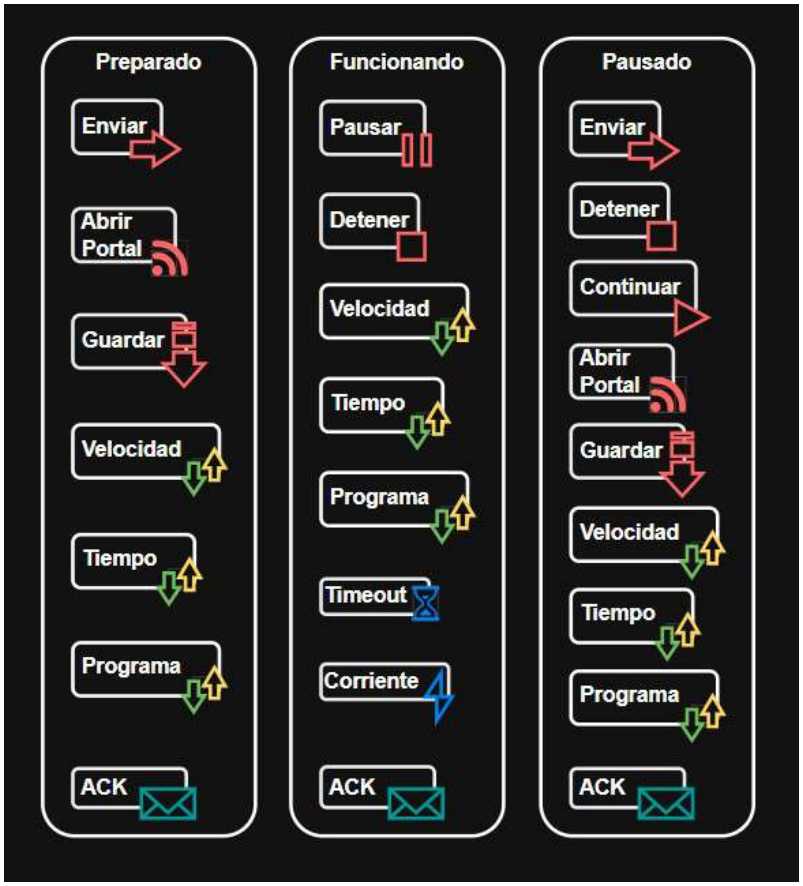

Figura 5: Tareas de cada estado

3.4. Interfaz Web

La Interfaz Web está compuesta por un dashboard implementado en *Node-RED*. *Node-RED* es una herramienta de desarrollo basada en flujos para la programación visual que permite crear aplicaciones de manera sencilla. En este caso, el dashboard está diseñado para interactuar con dispositivos y sistemas externos a través de mensajes de tipo MQTT.

MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicación ligero, ideal para aplicaciones de Internet de las Cosas (IoT) que requieren baja latencia y bajo consumo de ancho de banda.

La comunicación entre *Node-RED* y los dispositivos se realiza a través de un **Broker MQTT**, en este caso *Mosquitto*, que actúa como intermediario para gestionar el flujo de mensajes entre los dispositivos y el dashboard.

En la figura 6 se observa la Interfaz Web desarrollada.

Figura 6: Interfaz Web desarrollada

A continuación se describe el funcionamiento de cada sección de la Interfaz Web.

Programación manual del dispositivo: En este apartado, el usuario tiene la capacidad de programar el dispositivo de forma manual, eligiendo tanto la velocidad como el tiempo de trabajo. Además, permite detener y continuar el funcionamiento del dispositivo en cualquier momento, así como terminar el programa actual cuando sea necesario.

Funcionamiento del dispositivo: En este apartado, el usuario puede observar los parámetros de funcionamiento del dispositivo, tales como velocidad de referencia y velocidad de giro del motor, el tiempo programado y el tiempo transcurrido y restante, la corriente del motor y el estado del dispositivo, que puede incluir estados como funcionando, esperando programa y programa detenido.

Estado de la Interfaz Web: En este apartado, se observa la conexión de la Interfaz Web con el servidor y con el dispositivo en forma de led, si está en verde significa que esta conectado y si esta en rojo significa que no lo esta.

Configurar programas predefinidos: En este apartado, el usuario puede modificar alguno de los programas predefinidos y guardarlos.

Activar programas predefinidos: En este apartado, el usuario puede activar directamente algunos de los programas predefinidos guardados con antelación.

3.5. Desarrollo de Software

A continuación se desarrolla la etapa de software que involucra la programación de los dos microcontroladores ESP32. En la figura 1 se pueden observar como “Microcontrolador de comunicación” y “Microcontrolador de control”.

Estos microcontroladores se comunican entre sí utilizando una comunicación UART.

3.5.1. Microcontrolador de comunicación

Este microcontrolador actúa como mediador entre el usuario desde el panel de control fisico o virtual, y el microcontrolador de control. Se encarga de interpretar y mantener el flujo de datos. Esto incluye actualizar los estados de la maquina, realizar las tareas comandadas, proporcionar realimentación visual, y mantener la comunicación con los demás periféricos.

Es el responsable directo del panel de control físico, procesa las señales de los botones y de las comunicaciones para comandar el microcontrolador de control. Gestiona las indicaciones luminosas de los LEDs y muestra al usuario los mensajes correspondientes. Además, realiza un monitoreo continuo para detectar cualquier tipo de falla y notificarla de manera inmediata a través de la pantalla.

El software creado para este microcontrolador se puede consultar dentro de la sección de anexos.

3.5.2. Microcontrolador de control

Este microcontrolador tiene la función principal de controlar la velocidad de la máquina. Utiliza los valores de tiempo y velocidad que recibe del “Microcontrolador de comunicación” para determinar cuánto tiempo debe operar y a qué velocidad.

Además, gestiona las señales provenientes del hardware. Como entradas, recibe las señales del detector de cruces por cero, la señal del tacómetro y el sensado de corriente. Dentro del microcontrolador, se genera una señal pulsada que se envía al circuito que regulador de tensión del motor.

El software para el detector de cruces por cero se encarga de sincronizar el ESP32 con la tensión de linea de la red, el del tacómetro permite calcular la velocidad de giro del motor y el del sensado de corriente se encarga de muestrear la señal a su entrada para calcular la corriente RMS que circula por el motor.

El control de velocidad de giro del motor implementado es un PI digital. Para su implementación se utilizó la siguiente librería externa:

- Librería PID de Brett Beauregard

  Ver librería

Si bien la librería facilitó los cálculos realizados por el control, fue necesario obtener los parámetros del PI (K_p y K_i), los cuales indican la ganancia proporcional e integral del control, respectivamente, y determinan la posición del cero y el polo. Para obtener estos valores se realizaron los siguientes pasos:

1. Descripción del sistema
2. Modelado del sistema
3. Análisis del sistema
4. Diseño del controlador PI
5. Implementación

1) Descripción del sistema:

Se busca controlar la velocidad de giro de un motor universal a través de un microcontrolador. Además, el usuario debe poder elegir entre tres velocidades de funcionamiento del dispositivo por lo que el control debe funcionar en estas velocidades.

La velocidad que se utilizará como referencia es la velocidad del eje del motor en RPM.

La señal de salida del microcontrolador es un pulso de disparo sincronizado con la tensión de alimentación, el cual ajusta su inicio con un ángulo de disparo. El pulso de disparo activa el circuito "Regulador de tensión por ángulo de conducción", el cual actúa como llave, recortando la señal de alimentación del motor.

Se debe tener en cuenta que la relación del ángulo de disparo y la tensión de alimentación no es lineal por lo que se va a trabajar en torno a un punto de operación.

Para obtener información sobre la velocidad del motor, se utilizó la señal proveniente del tacómetro, el cual está incorporado en el motor universal. Esta señal es de tipo senoidal y presenta un aumento tanto en su amplitud como en su frecuencia a medida que se incrementa la alimentación del motor. Para una descripción más detallada del proceso de caracterización de esta señal, véase la sección 3.6.3.

De la señal proveniente del tacómetro, la información de interés es la frecuencia. Para extraerla, se desarrolló el circuito denominado Tacómetro, también descrito en la sección 3.6.3, cuya función es convertir la señal senoidal proveniente del tacómetro del motor en una señal pulsada. Posteriormente, utilizando esta señal pulsada, el microcontrolador calcula la velocidad mediante un contador que suma los flancos ascendentes detectados durante un intervalo de 100 mseg.

A continuación, se muestra un esquemático que da una idea de las conexiones que se realizaron y que circuitos se utilizarán.

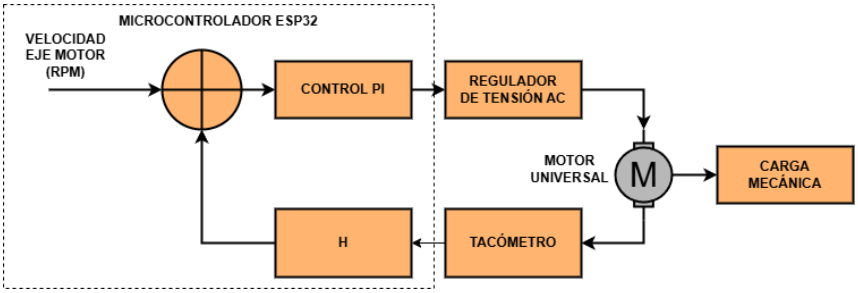

Figura 7: Esquemático de conexiones

En la siguiente imagen, se observa el modelo completo del sistema.

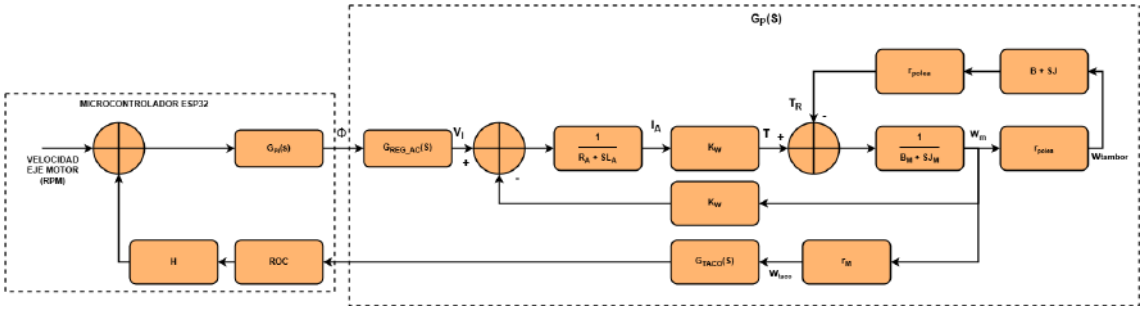

Figura 8: Modelo del sistema

Como se puede observar de la Figura 8, es un sistema complejo de caracterizar. Además, se debe tener en cuenta que la transferencia de $G_{REG_AC}(s)$ presenta una alinealidad. Por cuestiones de simplicidad, el modelado del sistema que se realizó se hizo a través de un método de identificación de sistemas aproximando gran parte del modelo a través de $G_p(s)$.

2) Modelado del sistema:

Como se mencionó en la descripción del sistema el ángulo de disparo, que es la variable que vamos ajustar con el control, no es lineal con la alimentación del motor. Debido a esta alinealidad se va a modelar el sistema entorno a un punto de operación. Para obtener el punto de operación se debe definir las velocidades a las cuales va a funcionar el dispositivo.

A continuación, se muestra una tabla con las velocidades del dispositivo.

Velocidad	Tambor [RPM]	Eje del motor [RPM]	Tacómetro [RPM]
Baja	15	225	1800
Media	30	450	3600
Alta	45	675	5400

La tabla anterior indica, primero la velocidad a la cual va a trabajar el tambor del dispositivo. Luego, se indica la velocidad en el eje del motor. Entre la velocidad del tambor y la velocidad del eje del motor existe una relación de 15:1 respectivamente debido a la relación de poleas utilizadas. Por último, se indica la velocidad del tacómetro del motor, que es la velocidad con la que va a trabajar el control. Entre la velocidad del eje del motor y la que se obtiene del tacómetro hay una relación de 8:1 respectivamente.

El modelado realizado cuenta con un ángulo de disparo fijo de 125°. Se eligió este ángulo ya que le permite al dispositivo trabajar con la velocidad media aproximadamente (35 RPM en el tambor y 525 RPM en el eje del motor).

La carga mecánica utilizada para el modelado es de 3,5 kg, esta carga es el peso máximo en piezas de madera con que se contaba para modelar el sistema.

Con el ángulo de disparo fijo se busca obtener la respuesta de velocidad del dispositivo en función del tiempo y luego compararla con un método de identificación de plantas. En este caso, se utilizó el método **Stark**, explicado en [1]. Debido a que el banco de medición utilizado es el mismo que se desarrolló en el proyecto para obtener la velocidad del dispositivo, la gráfica obtenida es de velocidad en función de muestras y cada muestra se realizó con un tiempo de muestreo de 100 mseg.

A continuación, se observa la respuesta obtenida luego de obtener 400 muestras, las cuales representan 40 segundos.

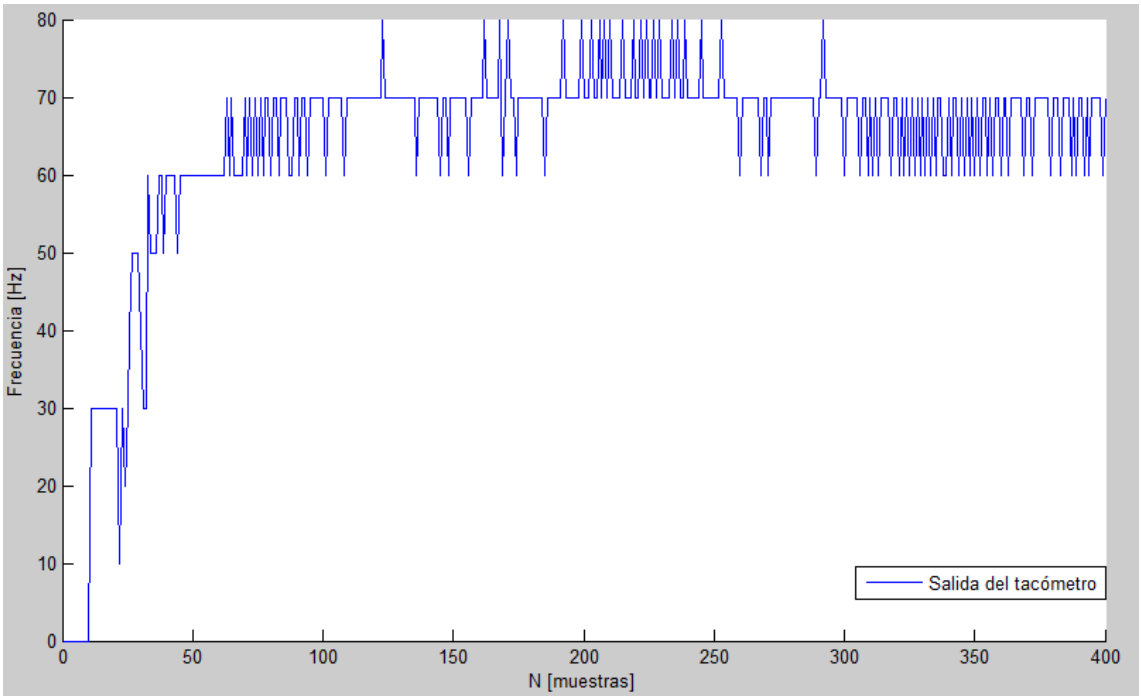

Figura 9: Respuesta del sistema

A causa de que la resolución de la curva obtenida era baja se decidió obtener la medición de la velocidad de forma diferente. Se optó por realizar mediciones del periodo de la señal y luego obtener la frecuencia de la señal. Para ello se observó que la Figura 9 muestra que el dispositivo va a trabajar con una velocidad final de aproximadamente 70 Hz en el tacómetro, el periodo es de $\frac{1}{70}\ segundos = 0{,}014\ segundos$ por lo que se decidió obtener 2800 muestras, de forma de que el tiempo que se esta observando la salida sea de 40 segundos como en la Figura 9.

A continuación, se observa la gráfica obtenida con la medición del periodo.

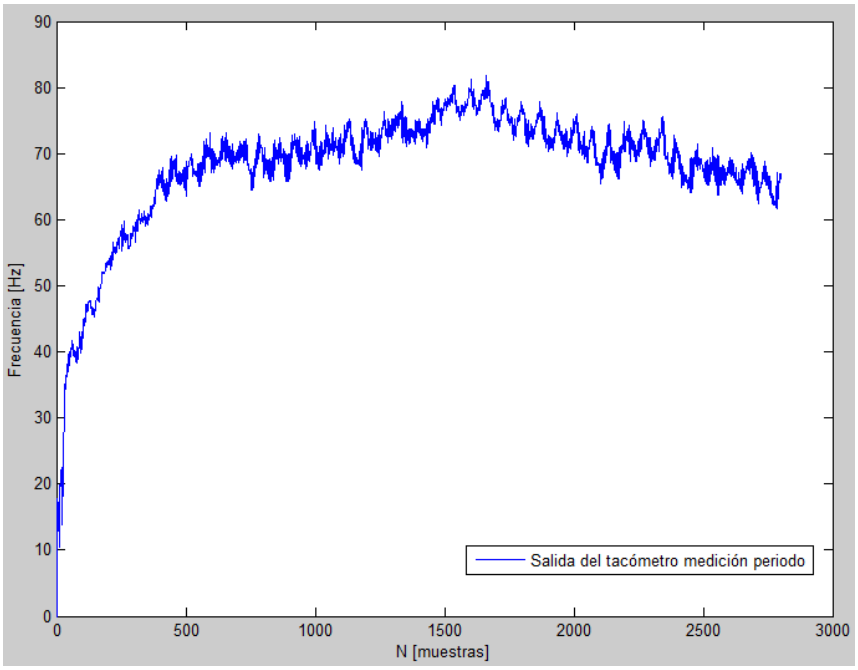

Figura 10: Respuesta del sistema calculando la frecuencia a través del periodo

Al aplicar el método de identificación Stark a la curva obtenida en la Figura 10 se obtuvo la siguiente transferencia.

$$Gp(s) = \frac{69{,}678 * e^{-0{,}91*s}}{(\frac{s}{0{,}52}+1)(\frac{s}{10{,}83}+1)}[\frac{Hz}{seg}] \tag{1}$$

Se puede destacar, a partir de la Figura 10 y de la transferencia obtenida, la presencia de un retardo en el sistema. En dicha figura, este retardo no se encuentra representado gráficamente. Esto se debe a las limitaciones del banco de medición empleado para registrar el período. Para obtener una muestra, es necesario medir el tiempo entre dos flancos ascendentes de la señal pulsada generada por el circuito Tacómetro. Por lo tanto, si inicialmente no hay una señal disponible, no es posible realizar la medición.

Como consecuencia de esta condición, la curva no refleja el retardo presente en el sistema. Para validar la existencia del retardo y confirmar que la transferencia obtenida representa adecuadamente el comportamiento del sistema, es necesario comparar los resultados con los de la Figura 9, en cuya medición sí se puede contemplar el retardo.

A continuación se muestran dos gráficos comparando la transferencia obtenida con las Figuras 9 y 10, para verificar si es representativa de nuestro sistema. En la Figura 10 se ajustó el retardo de forma que pueda observarse si la transferencia se ajusta a la curva.

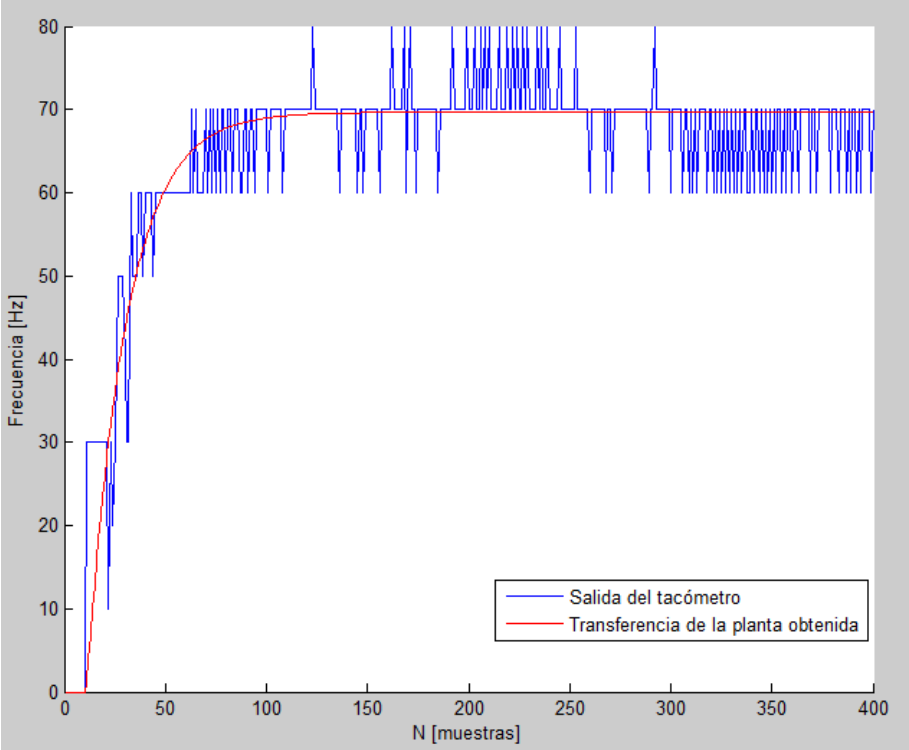

Figura 11: Respuesta del sistema con la medición y el modelo obtenido de la planta

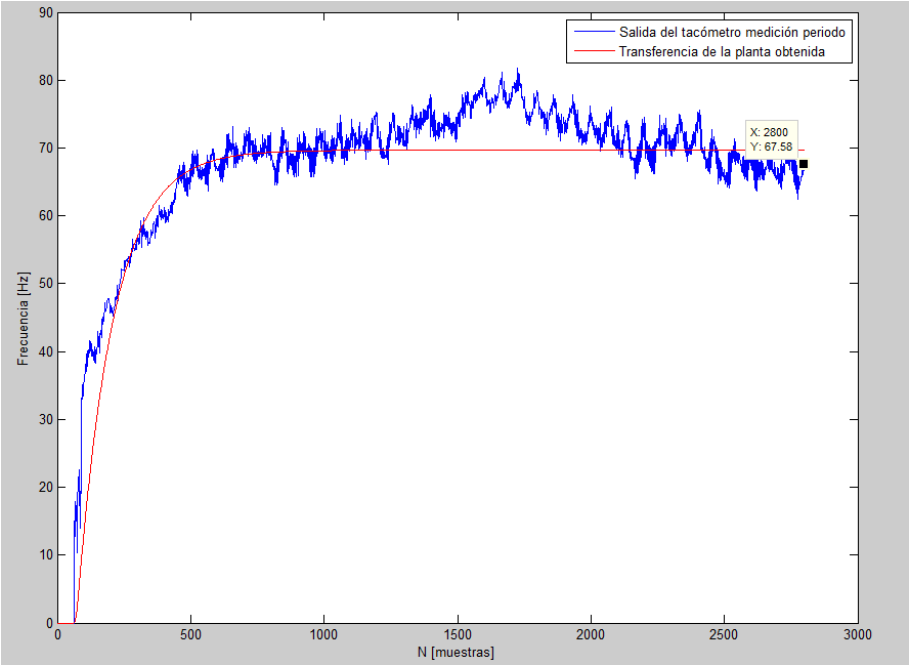

Figura 12: Respuesta del sistema con la medición y el modelo obtenido de la planta

Se debe destacar que el eje de abscisas de la Figura 11 representa muestras separadas con $Ts = 100\ mseg$ y el eje de abscisas de la Figura 12 representa muestras separadas

con $Ts \approx 14{,}28\ mseg$.

3)Análisis del sistema:

Como es un sistema con retardo, lo primero que debemos analizar en la Ecuación 1 es a qué frecuencia este aporta 45° de fase. Esto es importante porque existe una relación lineal entre el retardo y la frecuencia. Si analizamos la estabilidad de un sistema con retardo utilizando un diagrama de Bode, podemos identificar a qué frecuencia el retardo contribuye con 45° de fase. Luego, una década por encima de esta frecuencia, el retardo estaría contribuyendo con 450° de fase, lo que haría que el sistema sea inestable. Por eso, se analiza el aporte de 45° de fase del retardo y se utiliza como criterio para el cruce por $0dB$. A continuación, se muestra la relación entre la fase y el retardo.

$$Fase = -W * T_d * \frac{180}{\pi} \tag{2}$$

Si despejamos W, obtenemos la siguiente relación:

$$W = -\frac{Fase}{T_d} * \frac{\pi}{180} \tag{3}$$

Reemplazando la Fase por 45° y el retardo por 0.91 segundos, obtenemos la frecuencia.

$$W_{45} = \frac{45}{0{,}91}\frac{\pi}{180} = 0{,}8631[\frac{rad}{seg}] \tag{4}$$

Observando la posición a la cual el retardo aporta los 45°, se puede descartar el polo de alta frecuencia de la Ecuación 1 ya que está una década por encima. Además, para este análisis se descartó el retardo aportado por el tiempo de muestreo, debido a que su aporte con respecto al retardo de la planta es mucho menor. Luego, mantenemos la siguiente ecuación como transferencia de la planta:

$$Gp(s) = \frac{69{,}678 * e^{-0{,}91*s}}{(\frac{s}{0{,}52} + 1)}[\frac{Hz}{seg}] \tag{5}$$

A continuación, observamos que la transferencia obtenida sigue siendo representativa del sistema.

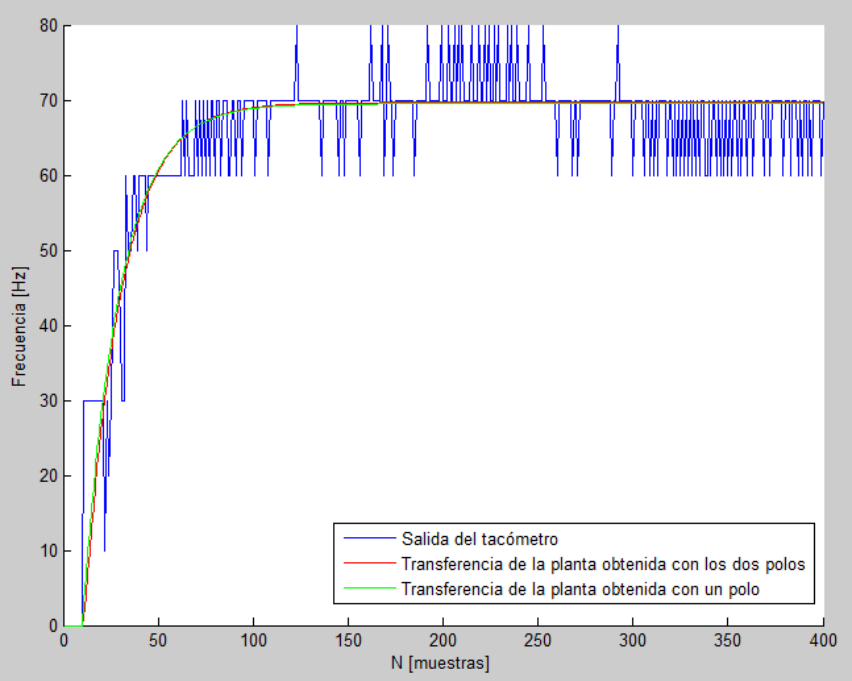

Figura 13: Respuesta del sistema, modelo de la planta con 2 polos y con 1 polo

4) Diseño del controlador PI:

El control que se quiere implementar es un PI, el cual tiene la siguiente transferencia de Laplace:

$$G_c(s) = \frac{K_i}{s} * (1 + \frac{s}{\frac{K_i}{K_p}})[\frac{^\circ}{RPM}] \tag{6}$$

Notar que la transferencia del controlador tiene unidades. El microcontrolador a la hora de hacer los cálculos no emplea unidades, esto es una interpretación de los valores de entrada y salida del controlador ya que estos no dejan de ser unidades físicas fuera del mismo.

Internamente se traducen los valores de entrada a un rango de números que se ingresan al controlador PI. Posteriormente, el resultado se reinterpreta a un valor físico de salida que representa la respuesta de control al error de entrada.

La transferencia de la planta obtenida en la Ecuación 5, es la transferencia obtenida utilizando como entrada un ángulo de disparo de 125°. Para trabajar con esta transferencia

y obtener el controlador, esta transferencia se debe normalizar, quedando de la siguiente manera:

$$G_{pn}(s) = \frac{69,678}{125}\frac{e^{-0,91*s}}{(\frac{s}{0,52}+1)} = \frac{0,557*e^{-0,91*s}}{(\frac{s}{0,52}+1)}[\frac{Hz}{^\circ}] \tag{7}$$

Sin embargo, se debe tener en cuenta que el punto de operación estará entorno a los 125°.

La realimentación H, es la relación de vueltas entre el eje del motor y el tacómetro multiplicado por la conversión de frecuencia a RPM.

$$H = \frac{1}{8}*60 = 7,5[\frac{RPM}{Hz}] \tag{8}$$

A continuación se observa el diagrama de bode de la transferencia de la planta (7) por H (8).

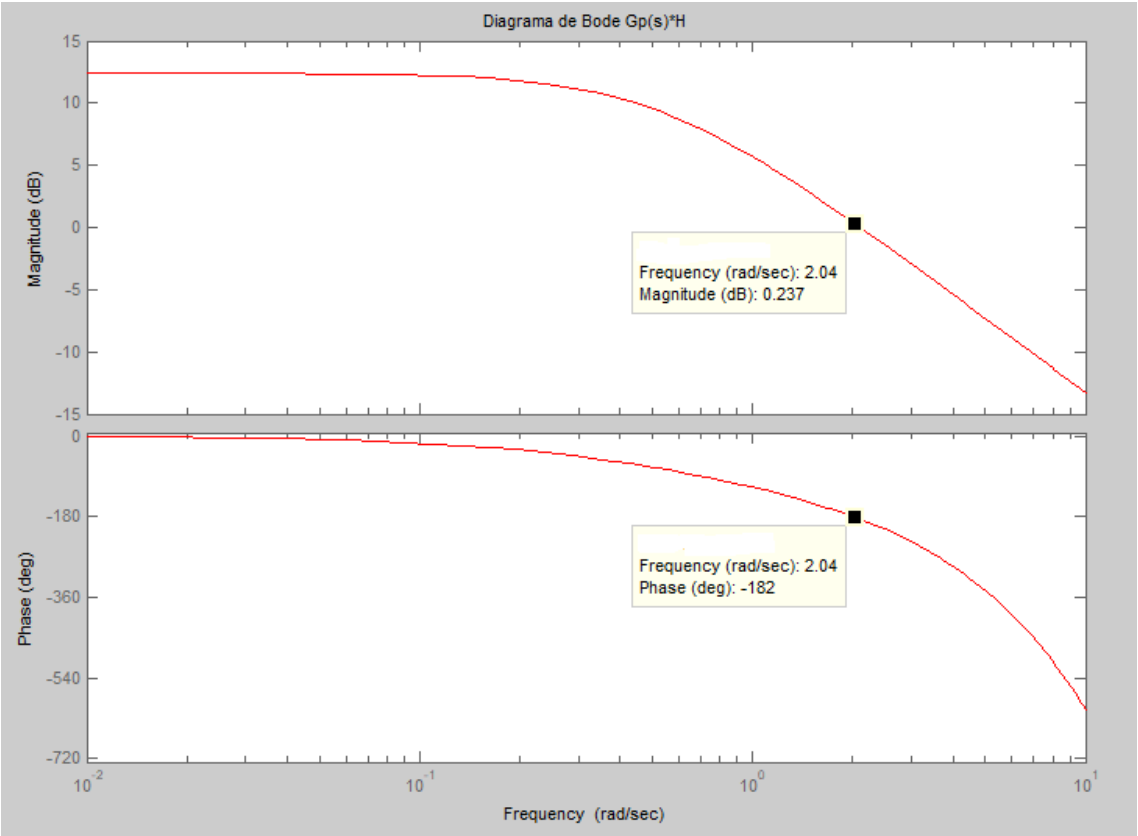

Figura 14: Bode de $G_p(s) * H$

El gráfico anterior muestra magnitudes en dB, debido a que se realizó en Matlab y coloca el nombre a los ejes por defecto. Sin embargo, este eje representa dBRPM.

Como se puede observar del bode de la Figura 14, el cruce por $0[dBRPM]$ se encuentra en $W = 2{,}04\ [\frac{rad}{seg}]$ y el $MF \approx 0$, lo que indica que el sistema se encuentra al borde de la inestabilidad. Se espera que con el controlador la estabilidad mejore. Para ello se utiliza como criterio que el cruce por $0[dBRPM]$ sea en $W = 0{,}8631\ [\frac{r}{s}]$ de forma que el retardo solo aporte 45°de fase.

A continuación, se muestra un gráfico que tiene el diagrama de bode de la planta sin el controlador (en rojo) y con el controlador (en naranja).

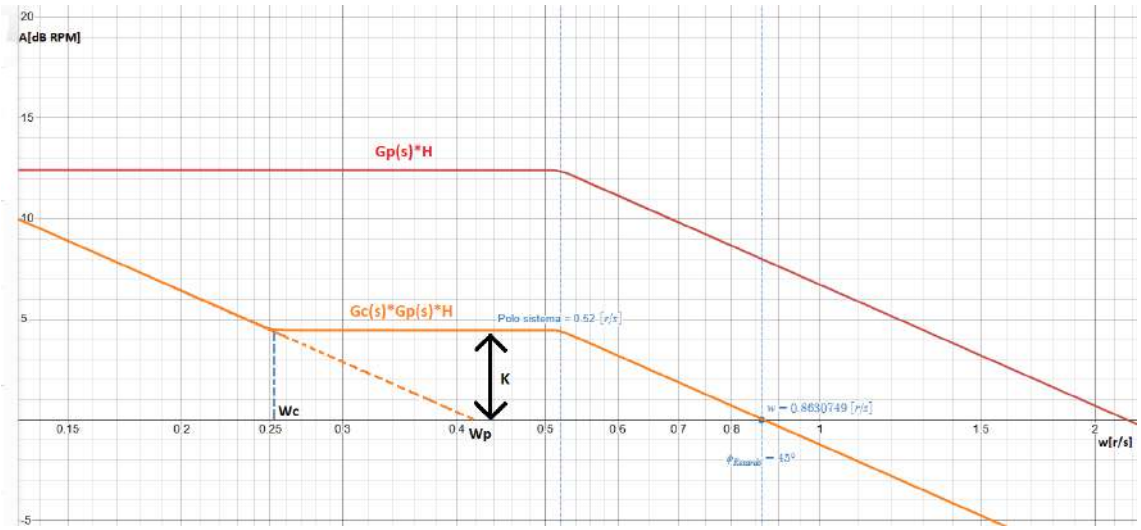

Figura 15: Transferencia de la planta con el controlador esperado

A partir de la Figura 15 se describirá como se determinó el controlador.

En principio, se debe tener en cuenta que se decidió implementar un controlador PI, por lo que el ajuste del controlador es a través de un polo y un cero. También, se utilizó como criterio que el cruce por cero sea $W = 0{,}8631\ [\frac{rad}{s}]$ para que el aporte de fase del retardo sea $45°$. A partir de estas características se obtuvo el diagrama de bode de la transferencia naranja de la Figura 15.

1. En el punto de cruce por cero dBRPM y $W = 0{,}8631\ [\frac{rad}{s}]$, se trazo una recta con una pendiente de $-20[\frac{dBRPM}{\frac{r}{s}}]$. Esta pendiente se trazo hasta alcanzar el polo del sistema en $W = 0{,}52\ [\frac{rad}{s}]$.

2. Con la recta del paso anterior se pudo determinar el valor de K, que se calcula como:

$$K = \frac{0{,}86}{0{,}52} = 1{,}654[RPM] \tag{9}$$

3. Como la carga utilizada en el tambor para el modelado es de 3,5 kg y se espera que el dispositivo funcione con cargas de hasta 6 kg se decidió colocar el cero del controlador media década antes del polo del sistema. Esta decisión contempla que la carga mecánica afecta la respuesta del sistema debido a su impacto en el momento de inercia. Bajo esta consideración, se colocó el cero del controlador en $W_c = 0{,}26[\frac{r}{s}]$.

4. Para determinar W_p se utilizó W_c y K.

$$W_p = K * W_C = 0{,}43[\frac{r}{seg}] \tag{10}$$

5. Con los valores de W_p y W_c se puede obtener la transferencia de la curva $G_c(s) * G_{pn}(s) * H$ de la Figura 15, la cual queda.

$$G_c(s) * G_{pn}(s) * H = \frac{W_p}{s} * (\frac{(1 + \frac{s}{W_c})}{(1 + \frac{s}{0{,}52})}) \tag{11}$$

Si reemplazamos $G_c(s) * G_{pn}(s) * H$ con las ecuaciones 6, 7, 8, utilizando estas ecuaciones en módulo y comparándolas con la ecuación 11 podemos obtener los valores de K_p y K_i.

$$(\frac{K_i}{s} * (1 + \frac{s}{\frac{K_i}{K_p}})) * (\frac{0{,}557}{(\frac{s}{0{,}52} + 1)}) * 7{,}5 = \frac{0{,}43}{s} * (\frac{(1 + \frac{s}{0{,}26})}{(1 + \frac{s}{0{,}52})}) \tag{12}$$

Comparando estas ecuaciones, obtenemos los siguiente valores para K_i y K_p

$$K_i = 0{,}43 * \frac{1}{0{,}557 * 7{,}5} = 0{,}1 \tag{13}$$

$$K_p = \frac{K_i}{0{,}26} = 0{,}4 \tag{14}$$

Con K_p y K_i podemos obtener la transferencia del controlador:

$$G_c(s) = \frac{0{,}1}{s} * (1 + \frac{s}{0{,}26})[\frac{^\circ}{RPM}] \tag{15}$$

5) Implementación:

El controlador implementado se realizó de forma digital en un microcontrolador. Como se mencionó anteriormente y se utilizo una librería externa para su desarrollo.

La librería permite modificar el tiempo de muestreo del control; en el caso de este proyecto, el muestreo es de $100ms$. También, es necesario invertir la salida del controlador. Para el control del sistema, cuando el error entre la velocidad de referencia y la velocidad realimentada aumente, el ángulo de disparo debe disminuir permitiendo aumentar la potencia entregada al motor incrementando su velocidad.

Por último, los controladores PI discretos tienen dos formas para la expresión de la integral, por rectángulo o por trapecios. La forma utilizada para el desarrollo de este proyecto es la integral por rectángulos.

Luego del modelado, y más avanzados en el desarrollo del proyecto fue necesario un cambio en las velocidades de funcionamiento del dispositivo. Esto parte de que la velocidad de 15 RPM era demasiado baja y no sería utilizada. Por ello se decidió cambiar las velocidades. En cuanto al control implementado, se verificó que funcionara de forma satisfactoria con diferentes niveles de cargas en el tambor para estas nuevas velocidades, llegando a un límite de 6 kg.

A continuación, se observa una tabla con las velocidades finales.

Velocidad	Tambor [RPM]	Eje del motor [RPM]	Tacómetro [RPM]
Baja	30	450	3600
Media	45	675	5400
Alta	60	900	7200

3.6. Desarrollo de Hardware

A continuación se desarrolla el hardware implementado en el proyecto. Los circuitos son:

- **Detector de cruces por ceros**
- **Regulador de tensión por ángulo de conducción**
- **Tacómetro**
- **Sensor de corriente**
- **Regulador de Tensión**

Estos circuitos pueden observarse dentro del diagrama en bloques general de la figura 1.

3.6.1. Detector de Cruces por Cero

El circuito de detección de cruces por cero tiene como objetivo sincronizar el pulso de disparo generado por el microcontrolador (un ESP32), con la señal de alimentación, en este caso, la tensión de línea de 220V de alterna. Este tipo de circuito es fundamental en aplicaciones donde se requiere coordinar una señal alterna (como la de la red eléctrica) con los procesos de un microcontrolador.

A continuación se muestra el esquemático del circuito implementado.

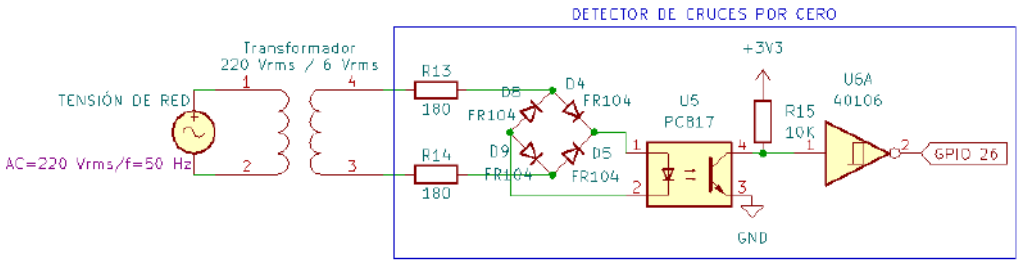

Figura 16: Esquemático del circuito implementado

Principalmente, el circuito convierte la señal alterna en una señal cuadrada, que es más fácil de manejar y procesar por el microcontrolador. Para lograrlo, la señal de la red pasa primero a través de un transformador que reduce la tensión de 220 Vrms a 6 Vrms. Esto disminuye significativamente la potencia que los componentes del sistema necesitarán.

Luego, la señal atraviesa un rectificador de onda completa y un optoacoplador. Juntos, estos componentes se encargan de convertir la señal alterna en una señal cuadrada, que se

utiliza como señal lógica en donde 0V se puede ver como el estado bajo y 3.3V como el estado alto.

Inicialmente, se planeaba enviar esta señal directamente al microcontrolador y realizar el sincronismo mediante software. Sin embargo, en la practica esto no fue posible, ya que los flancos de la señal cuadrada generada no eran lo suficientemente rápidos para que el microcontrolador los detectara con precisión. Para solucionar este problema, se incorporó un Schmitt Trigger, que tiene la capacidad de limpiar la señal y asegurar transiciones más rápidas y confiables entre los niveles lógicos. Este componente permite que el microcontrolador pueda detectar correctamente los cruces por cero de la señal de alimentación.

Además, en los primeros diseños del detector de cruces por cero, se contemplaba que el circuito funcionara directamente con la tensión de la red, sin utilizar un transformador. Por esta razón, el diseño incluía un optoacoplador, cuya función era aislar eléctricamente la red del microcontrolador. Sin embargo, con la incorporación del transformador, y dado que el circuito ya se encontraba implementado y probado, se optó por conservar esta etapa, aunque la aislación del optoacoplador ya no fuera necesaria.

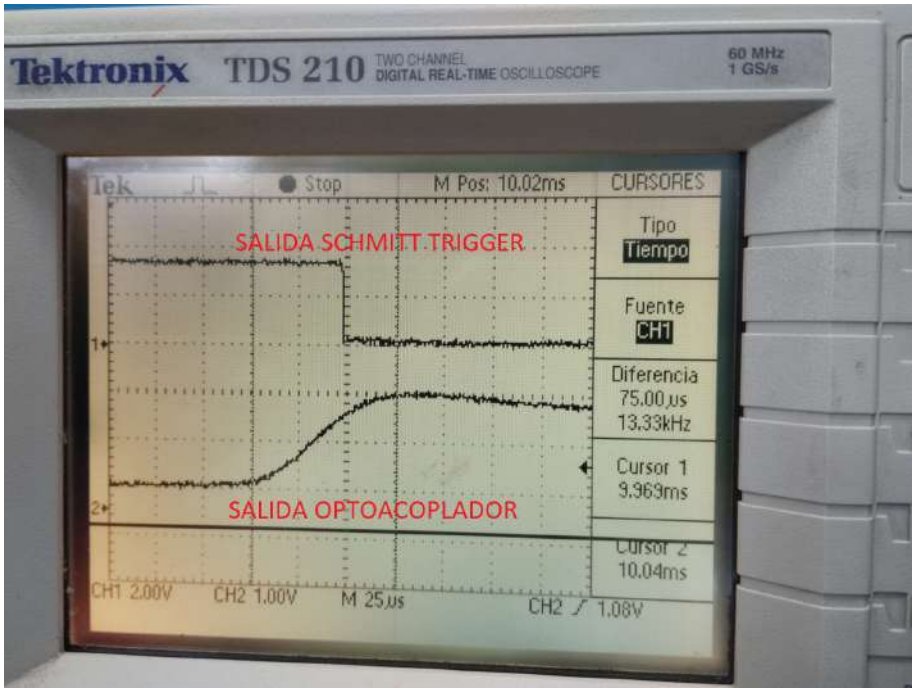

Figura 17: Transición a la salida del optoacoplador y del Schmitt Trigger

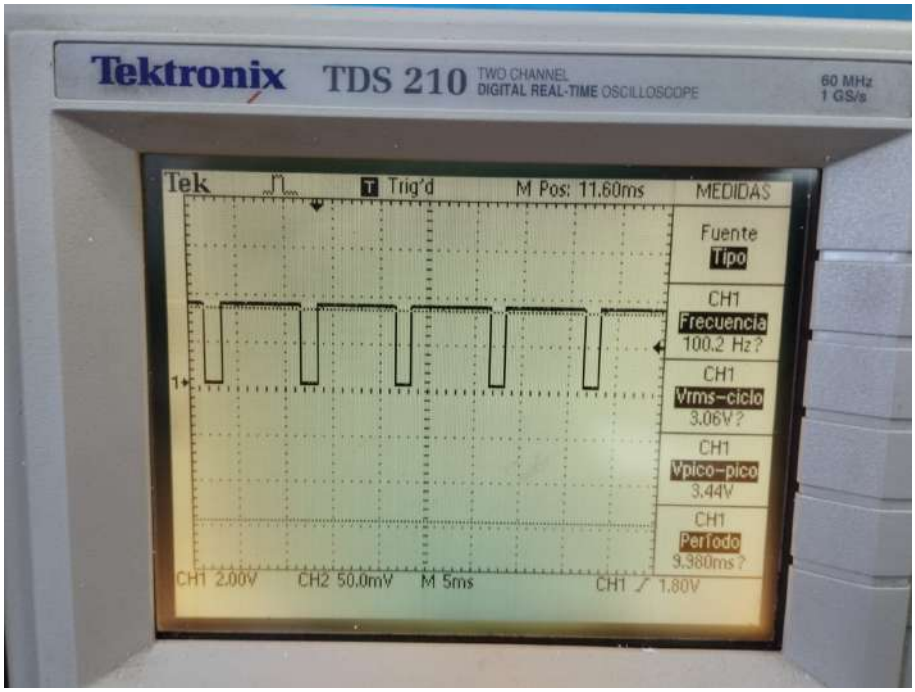

Figura 18: Señal a la salida del circuito del detector de cruces por cero

3.6.2. Regulación de tensión por ángulo de conducción

Este circuito está encargado de controlar el encendido del motor mediante un sistema de regulación basado en el control del ángulo de conducción.

El método consiste en generar una señal de disparo que habilita el paso de corriente hacia el motor únicamente durante una parte del semiciclo de la señal de línea, regulando así la potencia entregada.

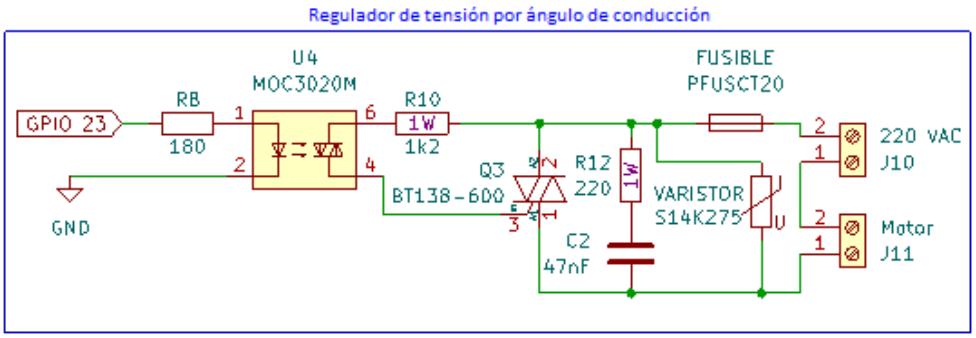

Figura 19: Esquemático del circuito implementado

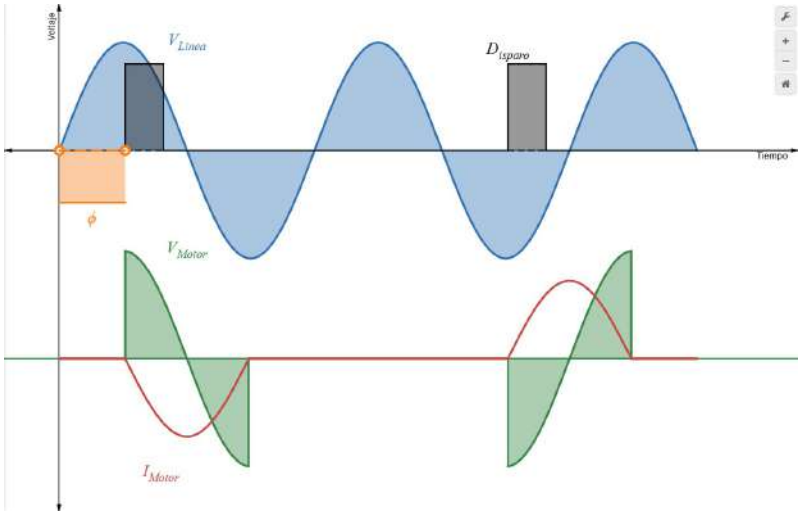

Figura 20: Señales de onda deseadas

Por la entrada GPIO23 ingresa una señal pulsada de disparo (provenientes del ESP32) que activa un optotriac, cuya salida energiza el gate de un TRIAC de mayor potencia, provocando su conducción y permitiendo el paso de la corriente sobre el motor. Una vez activado, el TRIAC permanece en conducción mientras que la corriente que fluye a través de él sea mayor que su corriente de mantenimiento.

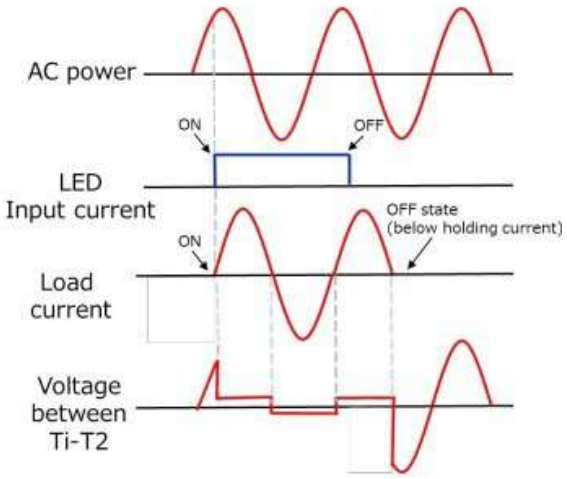

Figura 21: Comportamiento con Carga Inductiva

Al trabajar con cargas inductivas, la corriente se retrasa con respecto al voltaje. Debido a este desfasaje, el TRIAC se apaga (en el siguiente cruce por cero de la corriente) mientras todavía hay un voltaje significativo presente entre sus terminales. En este momento, el voltaje aplicado puede variar rápidamente debido a la naturaleza de la carga inductiva. La

energía almacenada en la inductancia se descarga cuando el TRIAC se apaga, causando picos transitorios de voltaje.

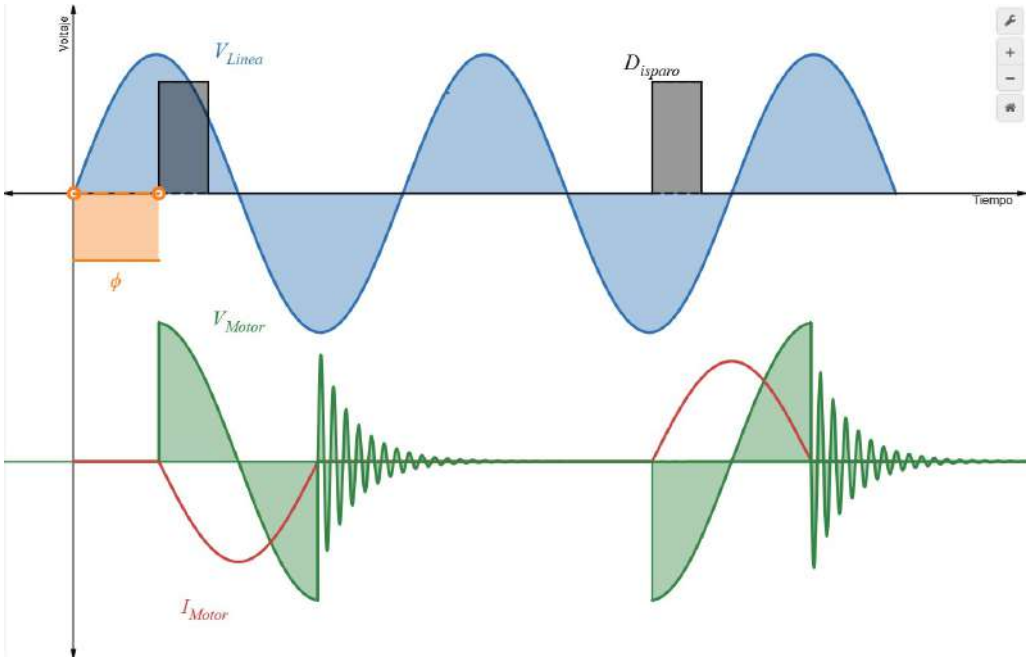

Figura 22: Distorsión generada por conmutación

El objetivo de una Red Snubber es mitigar estos cambios bruscos de voltaje para prevenir disparos no deseados. Principalmente, funciona como una vía de evacuación y disipación de energía. Utiliza un capacitor que actúa como un corto ante componentes de alta frecuencia y una resistencia que disipa la energía.

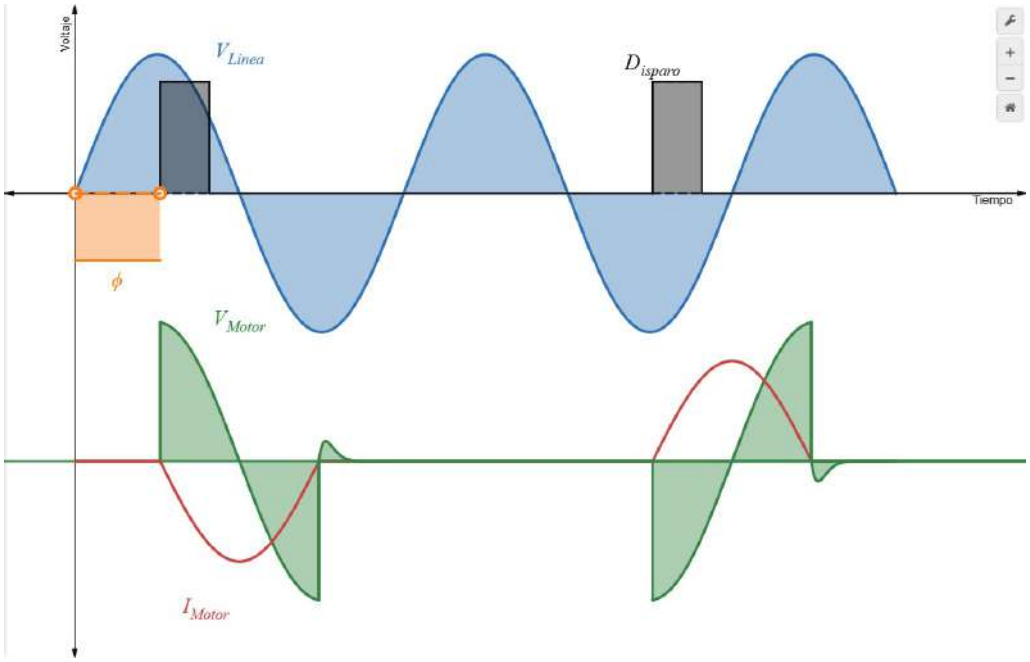

Figura 23: Efecto de la red Snubber

Como medida de seguridad extra, el circuito cuenta con un Fusible y un Varistor adaptados a la potencia de trabajo del motor.

3.6.3. Tacómetro

Una parámetro importante a medir en el motor universal es la velocidad de giro. Para ello se aprovechó el tacómetro integrado en el motor.

Para caracterizar la señal que proviene del tacómetro del motor lo primero que se hizo fue alimentar el motor con un variador de tensión. La idea con esta prueba era observar la señal de salida a medida que se variaba la señal de alimentación. El variador de tensión ofrece una señal senoidal de 50 Hz cuya amplitud puede modificarse manualmente. A medida que la amplitud del variador fue aumentando se observó que la salida del tacómetro también aumentaba y entregaba una señal senoidal que aumentaba su amplitud y su frecuencia. Sin embargo, la información importante se encuentra en la frecuencia, por lo que el circuito implementado se encarga de extraer solo esta información de la señal.

El circuito conectado al tacómetro del motor se encarga de convertir dicha señal en una señal pulsada dentro de los rangos compatibles con el ESP32, permitiendo que, mediante software, se pueda medir la frecuencia de la señal a través de la detección de los flancos de la señal generada.

A continuación, se muestra el esquemático del circuito.

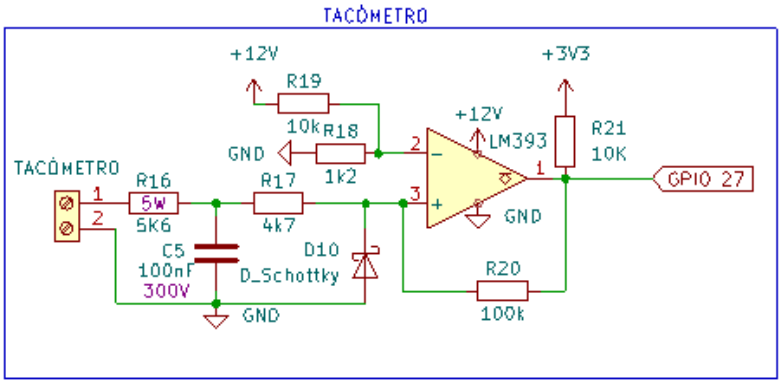

Figura 24: Esquemático del circuito implementado

Inicialmente, el circuito cuenta con un filtro pasa bajos para eliminar las componentes de alta frecuencia. Se debe destacar del filtro la utilización de la resistencia de $5W$, la cual se debe a que cuando se caracterizó la señal del tacómetro los valores de tensión a su salida superaban los $100V$ cuando el variador de tensión llegaba a $220V$, es por esto que se utilizó una resistencia de potencia disponible en el laboratorio.

Luego de ser filtrada, la señal pasa por un diodo schottky de modo que se elimina el semi-ciclo negativo de la señal. La señal resultante ingresa a la entrada no inversora del comparador LM393, donde se compara con una tensión de 1.2V aplicada en la entrada inversora. Este valor de referencia se establece como umbral de comparación, de modo que, si la señal en la entrada no inversora es superior a 1.2V, la salida del comparador será alta (3.3V), y si es inferior, la salida será 0V. El valor de 1.2V fue seleccionado para evitar que el circuito opere en un rango cercano al de la tensión de conducción directa del diodo, que es de aproximadamente 0.6V.

La salida del comparador es del tipo “Open Collector”, lo que brinda mayor flexibilidad al seleccionar el valor de la salida en estado alto. Dado que se desea utilizar esta señal como entrada para el ESP32, y este maneja 3.3V como nivel lógico alto, se utiliza el mismo microcontrolador como fuente tensión, el cual entrega una tensión de este valor, conectado con una resistencia de pull-up de $10k\Omega$.

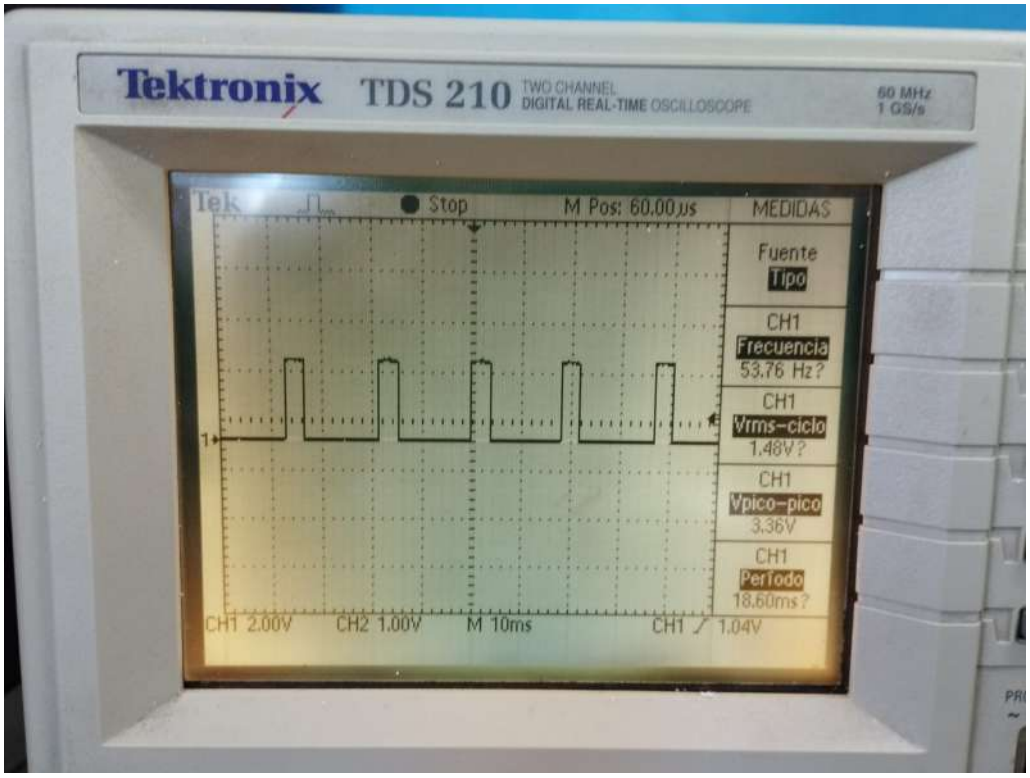

Figura 25: Señal a la salida del circuito del tacómetro

Una vez medida la frecuencia en el tacómetro, se debe determinar la relación que hay entre esta señal y la del eje del motor, ya que se quiere medir la velocidad angular a la cual esta girando el eje del motor.

La medición se basa en colocar un micrófono cercano al eje del motor y conectado a un analizador de espectros analógico. Con ello se logra determinar la frecuencia fundamental del tono que emite el motor siendo esta frecuencia su velocidad de giro. La frecuencia obtenida se comparó con la frecuencia obtenida a la salida del circuito tacómetro y se determinó que la relación entre el eje del motor y el tacómetro es de 8:1, es decir, que por cada vuelta del eje del motor, el tacómetro entrega ocho pulsos.

A continuación, se presentan dos imágenes. En la primera imagen se observa el micrófono, el motor y el analizador de espectros empleados durante el proceso de medición. La segunda imagen muestra el espectro correspondiente al tono generado por el eje del motor, así como el espectro asociado a la señal cuadrada obtenida a la salida del circuito del tacómetro. Se puede observar con la segunda imagen que la relación entre el eje del motor y el tacómetro es de 8:1.

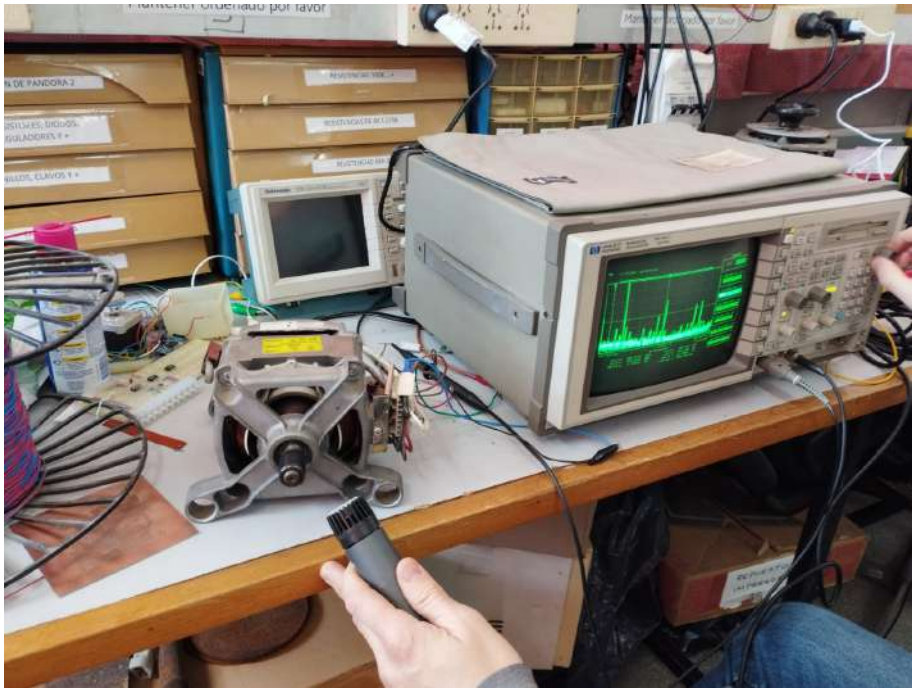

Figura 26: Medición de la velocidad del eje del motor

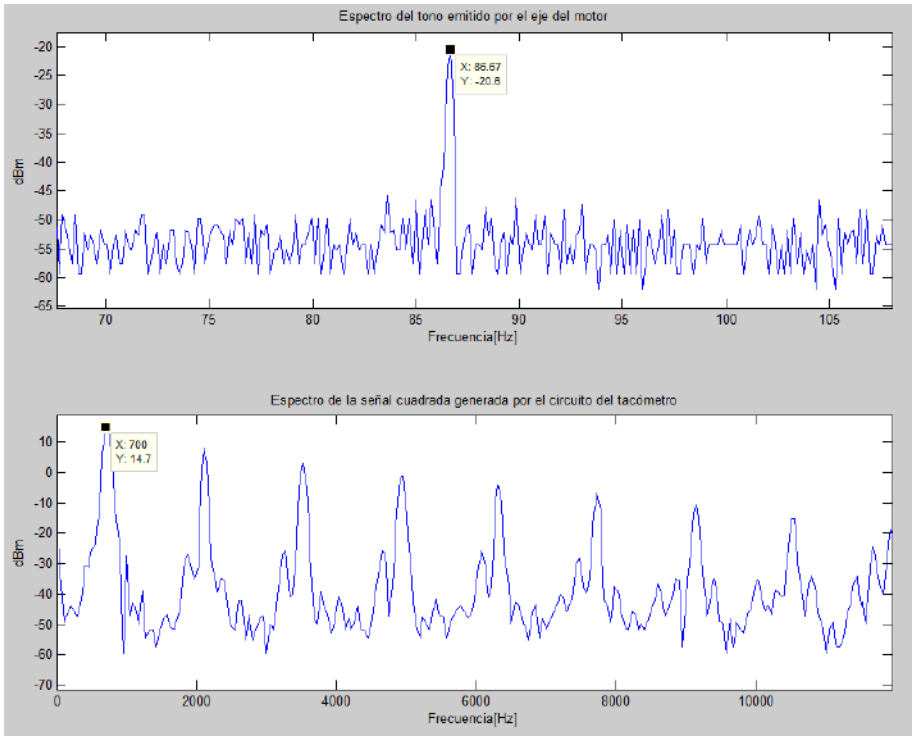

Figura 27: Medición del tono generado por el eje del motor y de la señal cuadrada producida por el circuito del tacómetro

3.6.4. Sensado de Corriente

Una medición importante al trabajar con un motor es la de corriente. La medición realizada se realiza con varios objetivos.

1. Determinar el consumo de energía del motor.
2. Detectar posibles falla. Por ejemplo, que el motor se trabe.
3. Proteger el motor en caso de exceso de corriente.

Para medir la corriente, se utilizó un transformador de corriente. Este sensor permite una medición no invasiva de la corriente del motor. Genera una señal de tensión que sigue la forma de onda de la corriente, con una relación de 10A/1Vrms.

Figura 28: Sensor de corriente

El circuito diseñado e implementado para obtener el valor de la corriente es el siguiente.

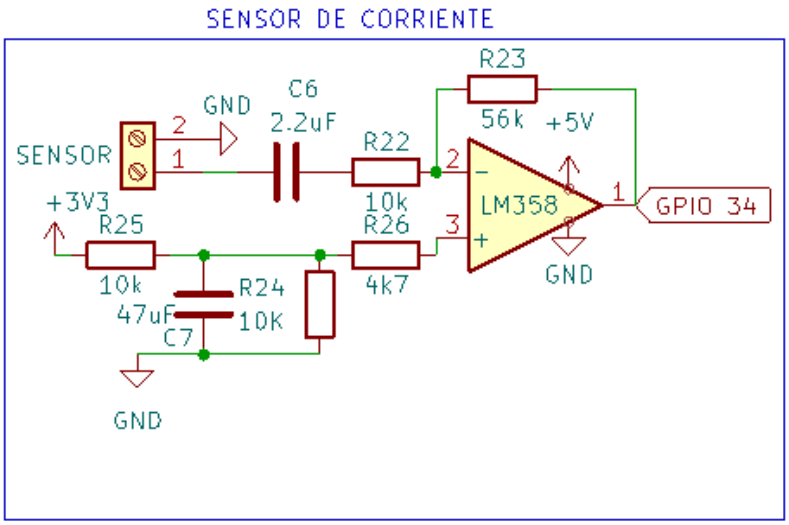

Figura 29: Esquemático del circuito para el sensor de corriente

Se basa en amplificar la señal obtenida del sensor y agregarle una continua de forma de poder operar en el rango de tensión de entrada del ESP32. Luego, el microcontrolador se encarga de calcular el valor de corriente RMS muestreando la señal con una frecuencia de $1200\ Hz$. Para obtener el valor de corriente RMS se siguen los siguientes pasos dentro del microcontrolador:

1. Cálculo de la sumatoria de las muestra:

$$Sum(i) = \sum V_{muestra} \tag{16}$$

2. Cálculo del valor cuadrático de las muestras:

$$Prod(i) = \sum {V_{muestra}}^2 \tag{17}$$

3. Luego de tomar $N = 720$ muestras para trabajar con ciclos enteros de la señal muestreada dividimos las sumatorias obtenidas por la cantidad de muestras.

$$Sum = \frac{Sum_{total}}{N_{muestras}} \tag{18}$$

$$Prod = \frac{Prod_{total}}{N_{muestras}} \tag{19}$$

4. Cálculo de la tensión rms de la señal que entrega el sensor.

$$V_{rms} = \sqrt{(Prod) - (Sum)^2} \tag{20}$$

5. Cálculo de la corriente rms del motor.

$$I_{rms} = V_{rms} * \frac{G_{sensor}}{G_{amplificador}} \tag{21}$$

Para verificar la medición obtenida se compararon los valores de corriente con el de una Sonda Rigol RP100C.

3.6.5. Regulador de Tensión

El regulador de tensión diseñado se utiliza para convertir un voltaje de entrada de 12 V a una salida estable de 5 V, necesario para alimentar los dos microcontroladores ESP32. También se encarga de suministrar 12 V a la referencia de voltaje y el comparador LM393 del circuito del tacómetro. El voltaje de entrada al regulador proviene de una fuente comercial de 12V, 2A.

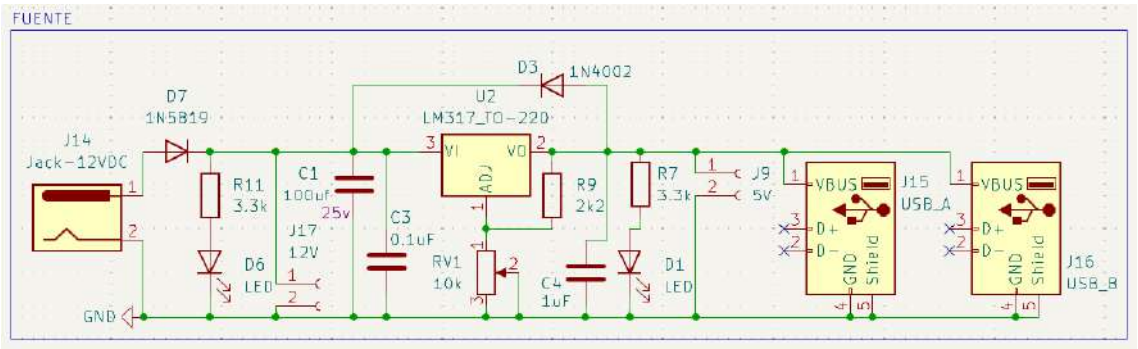

Figura 30: Esquemático Regulador de Tensión

Se utilizó un LM317, ya que es un regulador de voltaje ajustable de tres terminales, capaz de suministrar más de 1,5 A en un rango de voltaje de salida de 1,25 V a 37 V. Solo requiere dos resistencias externas para ajustar el voltaje de salida. Además, incluye limitación de corriente, protección contra sobrecargas térmicas y protección de área de operación segura. Se eligió este componente por sobre otros debido a su disponibilidad en el laboratorio y a las experiencias previas de uso en otros proyectos.

El conector Molex que toma de la entrada de 12V se utiliza para alimentar el comparador LM393 del circuito del tacómetro, mientras que aquel que toma la salida de 5V se

mantiene en reserva para futuras aplicaciones.

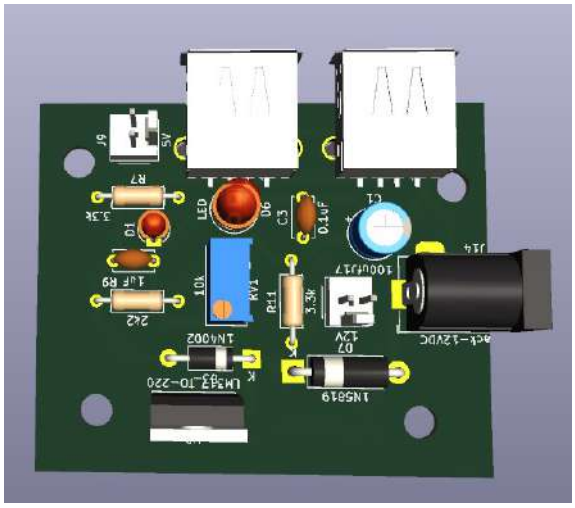

Figura 31: Modelo de Regulador de Tensión

Figura 32: Regulador implementado

4. *Resultados*

El dispositivo tiene como función principal el pulido de piezas de madera del tipo MDF, las cuales pueden variar en tamaño y forma. Para adaptarse a estas variaciones, el dispositivo ofrece trabajar con tres velocidades de operación diferente, las cuales son 30 RPM, 45 RPM y 60 RPM. Además, ofrece programar tiempos de trabajo de hasta 4 horas.

Las interfaces también informan al usuario sobre el estado de la máquina, proporcionando: el tiempo de trabajo transcurrido y restante, la velocidad de trabajo programada, la velocidad a la que está operando y el consumo de corriente. La Figura 6 muestra la Interfaz Web.

La interfaz física se desarrolló en una caja estanca, en la cual se agregaron dentro de ella los circuitos diseñados e implementados a lo largo del proyecto. Además, en ella se colocó el botón de emergencia, por lo que se espera que este cercana al dispositivo.

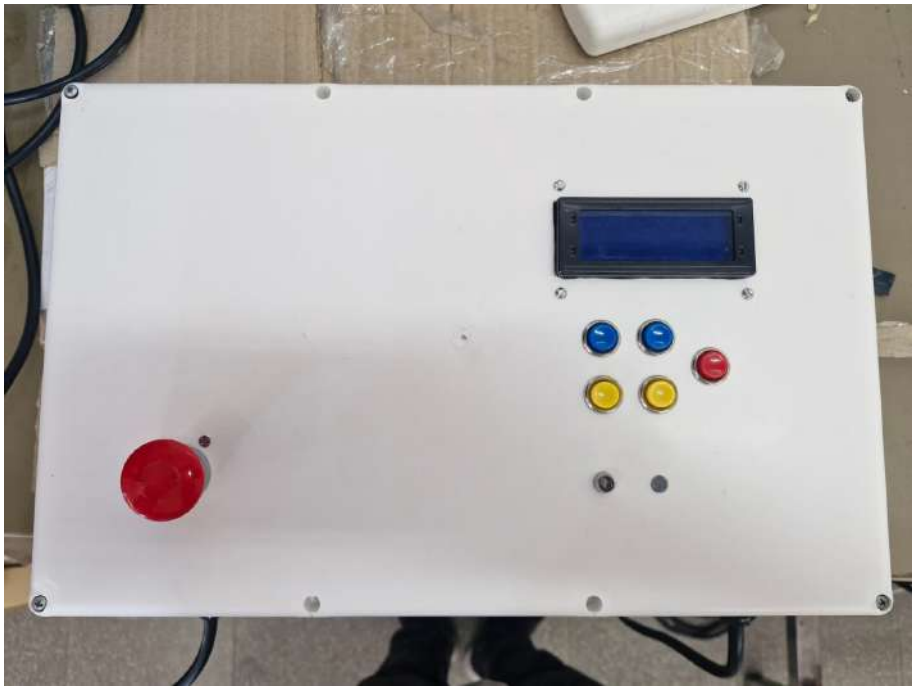

Figura 33: Botón de emergencia e interfaz física

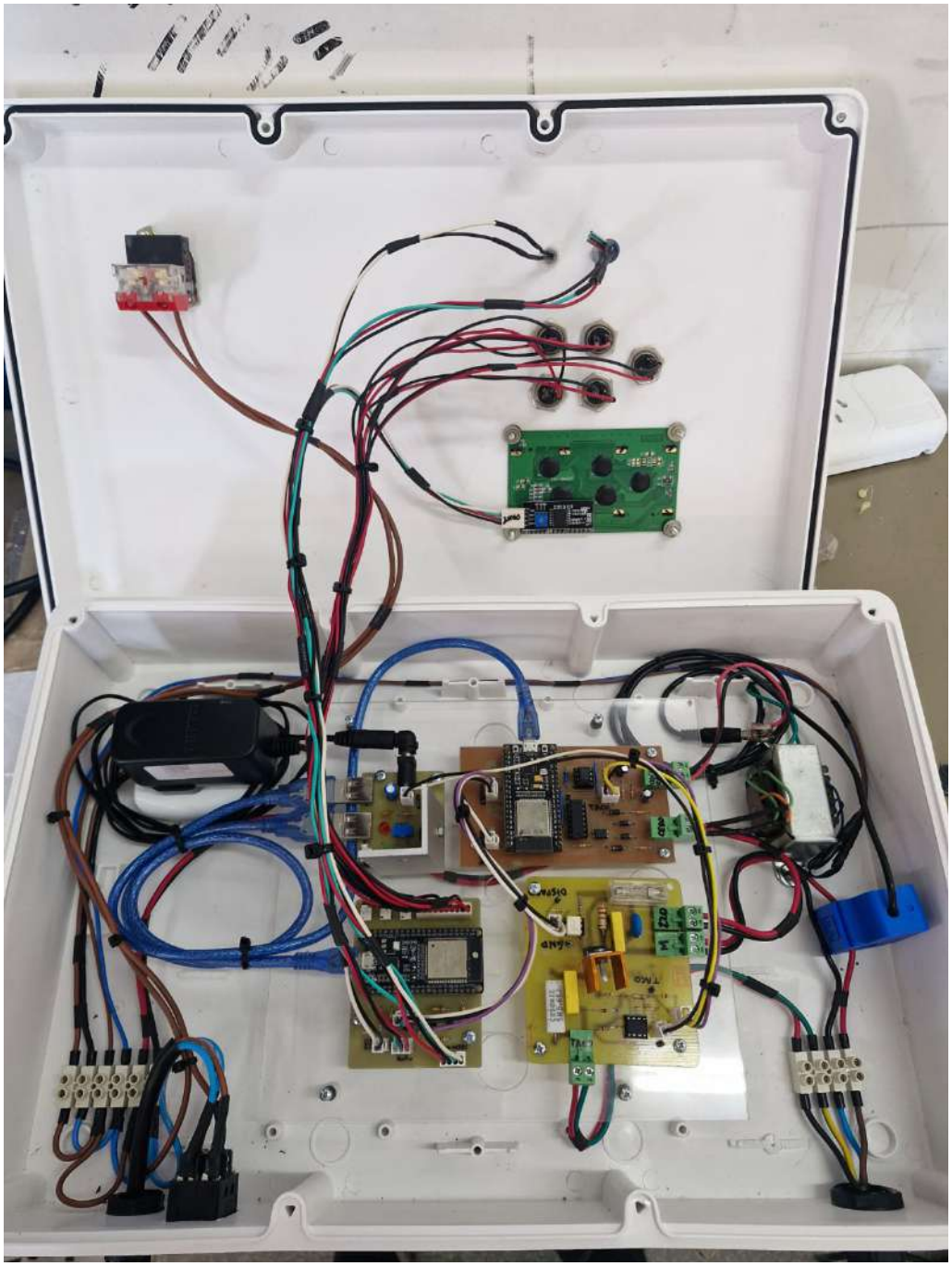

Figura 34: Circuitos implementados en la caja estanca

También, el dispositivo ofrece trabajar con cargas de hasta 6 kg. Inicialmente se había previsto trabajar con cargas de hasta 10 kg, esta carga final no se pudo alcanzar porque el motor no estaba preparado para funcionar con ese peso y la corriente superaba los valores nominales para los cuales el motor estaba preparado. La máquina podría trabajar con mayor carga modificando arreglo de poleas y un cambio de estructura.

La parte mecánica cuenta con una estructura que soporta el motor y el tambor giratorio. Dado que el motor era de un lavarropas ya contaba con poleas y correas comerciales, por lo cual se optó por utilizar algunas de ellas. La polea utilizada es la más grande para en este tipo de motores y es de unos 28 cm de diámetro, lo que resulta en una relación de 15:1 con respecto a la polea que tiene el motor que es de aproximadamente 2 cm de diámetro.

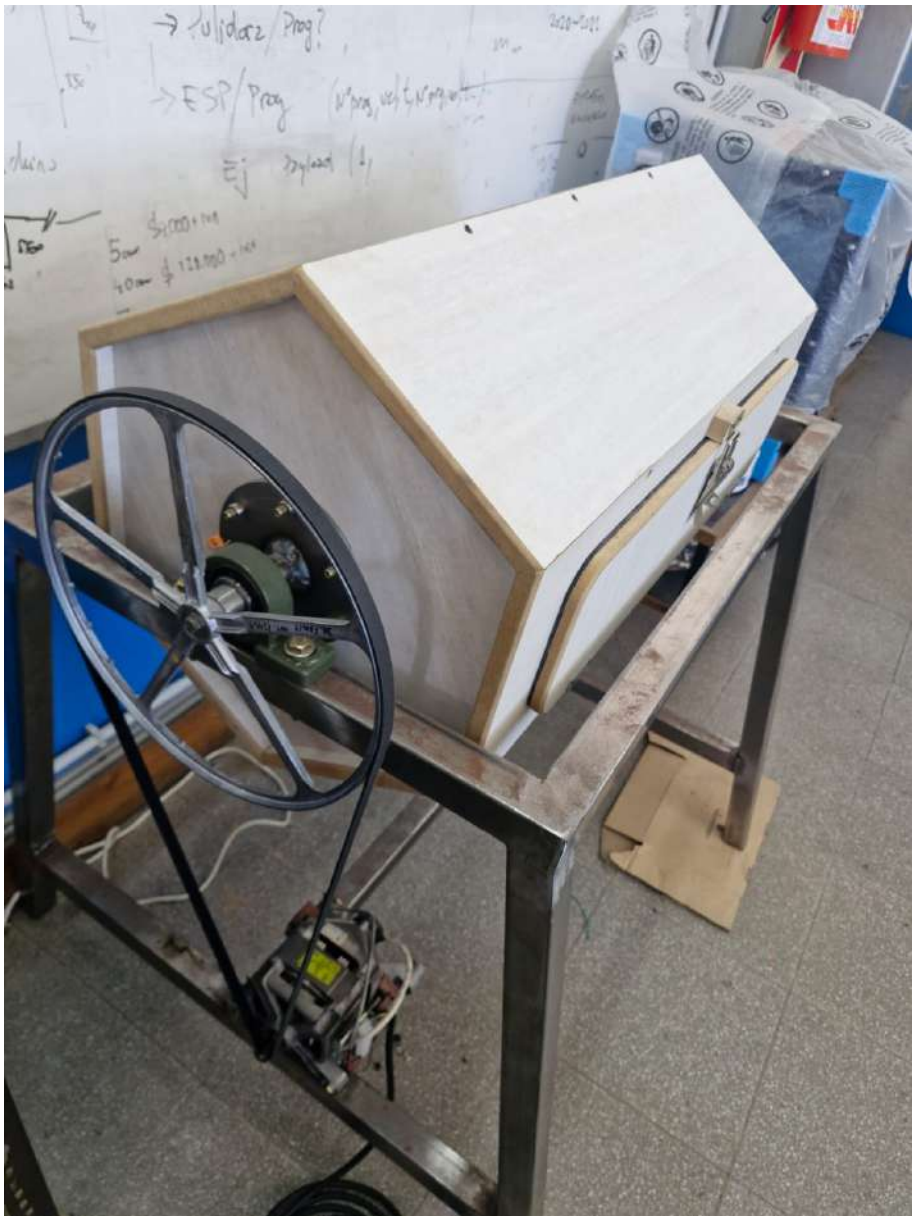

Figura 35: Estructura mecánica

5. *Conclusiones*

5.1. Conclusiones del Proyecto

Se logró cumplir con el objetivo principal del proyecto que era el desarrollo de una máquina para pulir piezas de madera del tipo MDF. El dispositivo funciona correctamente y cumple con los requerimientos planteados en el documento "Especificación de Requerimientos". Adicionalmente se desarrolló un manual de uso de la máquina para los usuarios de la empresa.

Durante el desarrollo del proyecto se presentaron varias dificultades, tanto en la parte de simulaciones, la evaluación de opciones de diseño, implementaciones de software y hardware, y al momento de realizar la interconexión entre las etapas. Esto permitió adquirir experiencias muy enriquecedoras en la resolución de problemas que se fueron presentando.

El desarrollo de la Interfaz Web junto con el sensado y accionamiento de circuitos periféricos con microcontroladores programados por los alumnos permitió adquirir conocimientos y experiencias orientadas en tecnologías de IoT.

Durante el diseño de los circuitos se adquirió un gran conocimiento en las áreas de diseño, implementación, simulación y pruebas de los circuitos. Se utilizaron en gran medida componentes del Laboratorio de Instrumentación Virtual y Robótica Aplicada, por lo que los diseños se ajustaban a los componentes que se poseían. Se analizaron hojas de datos de componentes y notas de aplicación para llevar a cabo la funcionalidad de cada circuito.

5.2. Gestión del proyecto

Si bien se esperaba finalizar con el proyecto en Junio de 2024, se presentaron varias situaciones que provocaron un retraso en la culminación del mismo.

El desarrollo del proyecto comenzó en paralelo con cursadas y exámenes finales de las últimas materias de la carrera de Ingeniería Electrónica, lo que resultó en una dedicación menor a la esperada durante ese año. Como consecuencia, los avances más significativos en ese periodo fueron principalmente en documentación e investigación.

Durante el 2024 se anticipó un aumento en la carga horaria dedicada al proyecto debido a la finalización de las cursadas y exámenes finales. Sin embargo, esto no se pudo cumplir debido a la necesidad de que uno de los miembros del equipo comenzara a traba-

jar.

Por último se subestimaron las etapas de investigación, implementación y pruebas del proyecto. Si bien se tuvo en cuenta el tiempo que podría llevar a cabo desarrollar el software y hardware del proyecto, no se considero el tiempo que llevaría hacer funcionar estas dos etapas separadas en conjunto.

Debido a los acontecimientos mencionados anteriormente, no se pudo cumplir con el Plan de Proyecto originalmente planteado. El proyecto fue finalmente aprobado a finales de 2024. Lo que dejó para principios de 2025 la finalización de la documentación necesaria y su presentación.

6. *Bibliografía*

[1] R. A. Mollenkamp, Introduction to Automatic Process Control. 1984, Instrument Society of America.

[2] Espressif. "ESP32 Series." Datasheet. (2024), dirección: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[3] E. D. Portal. "ESP32's family Memory Map 101." (2024), dirección: https://developer.espressif.com/blog/esp32-memory-map-101/.

[4] E. D. Portal. "ESP32 Programmers' Memory Model." (2020), dirección: https://developer.espressif.com/blog/esp32-programmers-memory-model/.

[5] Espressif. "API Reference." (2024), dirección: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/index.html.

[6] K. S. Reza Sarwo Widagdo Balok Hariadi. "Modelling and Analysis of Ziegler-Nichols and ChienHrones-Reswick Tuning PID on DC Motor Speed Control." Jurnal Teknologi Elektro, Vol. 14. No. 01. (2023), dirección: https://media.neliti.com/media/publications/565244-modelling-and-analysis-of-ziegler-nichol-48062d3d.pdf.

[7] Toshiba. "Triac couplers—basic properties and application design." Application Note. (2021), dirección: https://toshiba.semicon-storage.com/info/TLP360J_application_note_en_20210607_AKX00767.pdf?did=63136&prodName=TLP360J.

[8] STMicroelectronics. "Controlling a Triac with a phototriac." Application Note. (2020), dirección: https://www.st.com/resource/en/application_note/an5114-controlling-a-triac-with-a-phototriac-stmicroelectronics.pdf.

[9] STMicroelectronics. “RC snubber circuit design for triacs.” Application Note. (2008), dirección: https://www.st.com/resource/en/application_note/an308-triac-analog-control-circuits-for-inductive-loads-stmicroelectronics.pdf.

[10] STMicroelectronics. “TRIAC analog control circuits for inductive loads.” Application Note. (2024), dirección: https://www.st.com/resource/en/application_note/an437-rc-snubber-circuit-design-for-triacs-stmicroelectronics.pdf.

[11] F. Semiconductor. “RC Snubber Networks for Thyristor Power Control and Transient Suppression.” Application Note. (2002), dirección: https://www.onsemi.cn/pub/Collateral/AN-3008JP.pdf.

[12] J. Williams. “High Efficiency Linear Regulators.” Application Note. (1989), dirección: https://www.analog.com/media/en/technical-documentation/application-notes/an32f.pdf.

[13] T. Instruments. “LM317 3-Terminal Adjustable Regulator.” Datasheet and Application Note. (2020), dirección: https://www.ti.com/lit/ds/symlink/lm317.pdf?ts=1739827411932.

7. *Anexos*

7.1. Plan de Proyecto

Plan de Proyecto

7.2. Especificación de Requerimientos

Especificación de Requerimientos

7.3. Especificación Funcional

Especificación Funcional

7.4. Especificación Técnica

Especificación Técnica

7.5. Plan de Pruebas

Plan de Pruebas

7.6. Códigos de programación

Microcontrolador de Comunicación
Microcontrolador de Control

7.7. Kicad

Proyecto.zip
3D
PCB
Esquemático

7.8. Manual de Usuario

Manual de Usuario