



Informe Técnico

Circuito electrónico para el viscosímetro de cilindros concéntricos

Version 2

**Benjamín Valera Orozco
Gabriel Ascanio Gazca
Andrea Aguilar Valera**

Junio 2024

Proyecto Interno

Financiamiento Interno

Circuito electrónico para el viscosímetro de cilindros concéntricos

Benjamín Valera Orozco, Gabriel Ascanio Gazca, Andrea Aguilar Valera

Instituto de Ciencias Aplicadas y Tecnología
Universidad Nacional Autónoma de México, Circuito Exterior S/N, Ciudad
Universitaria AP 70-186, C.P. 04510, México D.F. México

Resumen

La viscosidad es una característica de algunos materiales de gran importancia en el estudio del comportamiento de los fluidos. Además, en la industria, la viscosidad es un parámetro de control en diversos procesos de producción. En este sentido, la viscosidad se determina experimentalmente mediante equipos llamados viscosímetros. Existen diversas configuraciones de viscosímetros que se utilizan en diversas circunstancias. En este informe desarrollamos un circuito electrónico para una configuración de viscosímetro de cilindros concéntricos enfocado hacia experimentos académicos en el nivel de licenciatura. El circuito está desarrollado sobre la base de un microcontrolador ESP32 e incorpora dispositivos electrónicos modernos como lo son: un despliegue TFT táctil, adecuación de señal de galga extensiométrica con 24 bits de resolución, impulsor de motor a pasos silencioso, comunicación Bluetooth y control de velocidad con codificador efecto Hall. Con estas características, se consiguió desarrollar un aparato de medición con alta precisión, confiabilidad y tecnológicamente moderno. Se reporta el diseño del circuito y su construcción en placa de circuito impreso. Como resultados se incluyen las evaluaciones aisladas acerca del comportamiento cualitativo de sus diversos componentes y un experimento de evaluación cuantitativa en el transductor de fuerza. Se incluye, como trabajo a futuro, el diseño mecánico del arreglo de cilindros concéntricos y una carcasa con elevado nivel estético y funcional.

Índice

Nomenclatura	3
Introducción	4
Definición del problema	4
Entorno actual	4
Descripción del problema a resolver y motivación	4
Relevancia y limitaciones	5
Relación con otras áreas	5
Método	5
Objetivos y resultados esperados	6
Organización del informe	6
1 Circuito para el viscosímetro	7
1.1 Descripción	7
1.2 Algoritmo de control y medición	8
1.3 Procedimiento de operación	10
1.4 Construcción en máquina enrutadora	18
2 Resultados	19
2.1 Resultados	19
2.1.1 Cualitativos	20
2.1.2 Cuantitativos	22
2.2 Discusión y trabajo a futuro	23
2.3 Guía para la detección de fallas	23
2.4 Especificaciones	24
Conclusiones	25
Anexo A Diagrama esquemático	26
Anexo B Circuito impreso	27
Anexo C Lista de partes	30
Anexo D Código fuente	31
Referencias	42

Nomenclatura

Símbolo, término o abreviatura	Significado
CD	Corriente Directa.
CA	Corriente Alterna.
IDE	<i>Integrated Development Enviroment</i> , ambiente de desarrollo integrado.
CNC	Control Numérico por Computadora.
TFT	<i>Thin Film Transistor</i> , Transistor de Película Delgada, tecnología para el despliegue de imágenes.
PCB	<i>Printed Circuit Board</i> , tarjeta de circuito impreso, es una tarjeta de conexiones en donde se montan componentes electrónicos.
GPIO	<i>General Purpose Input/Output</i> , Entrada/Salida de Propósito General.
I ² C	<i>Inter-Integrated Circuit</i> , protocolo serial para la comunicación entre dispositivos electrónicos.
SetPoint	Punto específico. Magnitud deseable de la variable en cuestión.
SPI	<i>Serial Peripheral Interface</i> . Interfaz periférica serial. Es una interfaz para transmisión y recepción de datos en serie.
Bluetooth	Método de conexión electrónica inalámbrica.
PWM	<i>Pulse Width Modulation</i> . Modulación por ancho del pulso. Es una señal de voltaje con componente de voltaje en directa variable.
rpm	Revoluciones Por Minuto. Medida de velocidad angular.
Gerber	Es un formato de archivos que contienen la información para construir un circuito impreso.
CAD	<i>Computer Aided Design</i> , Diseño asistido por computadora.
Excellon	Es un formato de archivo que contiene la información de taladros para una máquina de control numérico.
CAM	<i>Computer Aided Manufacturing</i> , Maquinado asistido por computadora.
Offset	Desplazamiento.
Software	Conjunto de reglas que ejecutan los procesadores electrónicos.
ISO	<i>International Organization for Standardization</i> . Organización Internacional de Estandarización. Se dedica a la creación de normas o estándares para asegurar la calidad, seguridad y eficiencia de productos y servicios.
Código G	Es un código de programación que se utiliza para ordenar acciones específicas a máquinas CNC.
Efecto Hall	Es un fenómeno que permite detectar campos magnéticos que se utiliza para la implementación de codificadores lectores de posición angular.

Nomenclatura

Símbolo, término o abreviatura	Significado
CD	Corriente Directa.
CA	Corriente Alterna.
IDE	<i>Integrated Development Enviroment</i> , ambiente de desarrollo integrado.
CNC	Control Numérico por Computadora.
TFT	<i>Thin Film Transistor</i> , Transistor de Película Delgada, tecnología para el despliegue de imágenes.
PCB	<i>Printed Circuit Board</i> , tarjeta de circuito impreso, es una tarjeta de conexiones en donde se montan componentes electrónicos.
GPIO	<i>General Purpose Input/Output</i> , Entrada/Salida de Propósito General.
I ² C	<i>Inter-Integrated Circuit</i> , protocolo serial para la comunicación entre dispositivos electrónicos.
SetPoint	Punto específico. Magnitud deseable de la variable en cuestión.
SPI	<i>Serial Peripheral Interface</i> . Interfaz periférica serial. Es una interfaz para transmisión y recepción de datos en serie.
Bluetooth	Método de conexión electrónica inalámbrica.
PWM	<i>Pulse Width Modulation</i> . Modulación por ancho del pulso. Es una señal de voltaje con componente de voltaje en directa variable.
rpm	Revoluciones Por Minuto. Medida de velocidad angular.
Gerber	Es un formato de archivos que contienen la información para construir un circuito impreso.
CAD	<i>Computer Aided Design</i> , Diseño asistido por computadora.
Excellon	Es un formato de archivo que contiene la información de taladros para una máquina de control numérico.
CAM	<i>Computer Aided Manufacturing</i> , Maquinado asistido por computadora.
Offset	Desplazamiento.
Software	Conjunto de reglas que ejecutan los procesadores electrónicos.
ISO	<i>International Organization for Standardization</i> . Organización Internacional de Estandarización. Se dedica a la creación de normas o estándares para asegurar la calidad, seguridad y eficiencia de productos y servicios.
Código G	Es un código de programación que se utiliza para ordenar acciones específicas a máquinas CNC.
Efecto Hall	Es un fenómeno que permite detectar campos magnéticos que se utiliza para la implementación de codificadores lectores de posición angular.

Introducción

El circuito electrónico para el viscosímetro descrito en este informe constituye la segunda versión desarrollada en años recientes. La idea principal es mejorar la primera versión al incorporar elementos electrónicos modernos que incrementan la confiabilidad en las mediciones.

Definición del problema

Analizando en retrospectiva la capacidad de los componentes electrónicos, en el año de 2013 no se contaba con la gran cantidad de características que hoy se ofrecen en un empaquetado sencillo de circuito integrado. Un ejemplo es la comparación entre los microcontroladores PIC18F4550 del año 2013 y el microcontrolador ESP32 del año 2024. Definitivamente, al diseñar aplicaciones con un ESP32 se consiguen resultados muy superiores a los que se podían obtener con un PIC18F4550. Estas ventajas tienen que ver, entre otras, con la velocidad de procesamiento, la resolución en los procesos de conversión analógica a digital, las tecnologías de comunicación inalámbrica, etc. En este sentido, una línea de trabajo puede ser la actualización de los trabajos previos para incorporar las tecnologías modernas.

Entorno actual

En la actualidad, y en comparación con el año de 2014, hoy se cuentan con dispositivos electrónicos como pantallas gráficas a colores, conexión inalámbrica y módulos simplificados de adquisición de variables con gran resolución. Sumado a lo anterior, se han desarrollado ambientes integrados de desarrollo para microcontroladores que simplifican y agilizan las labores de implementación de aplicaciones.

Descripción del problema a resolver y motivación

Con respecto a la primera versión del circuito electrónico para el viscosímetro (Valera et al, 2013), y de acuerdo con lo expuesto anteriormente, se identificaron las siguientes oportunidades para mejorar el desempeño:

1. Incrementar la resolución y exactitud en la medición.
2. Habilitar al instrumento con pantalla gráfica táctil.
3. Desarrollar una interfase amigable y moderna con el operador.
4. Dotar al instrumento con conexión inalámbrica para su operación mediante teléfono inteligente.
5. Disminuir el ruido acústico que genera el equipo.
6. Simplificar la placa de circuito impreso.

Entonces el problema a resolver consiste en diseñar y construir un circuito electrónico para el viscosímetro seleccionando dispositivos electrónicos actuales que permitan mejorar el desempeño mencionado.

Relevancia y limitaciones

La principal relevancia del proyecto radica en la creación y utilización de infraestructura con renovada funcionalidad con respecto al desarrollo previo mencionado anteriormente. De esta manera se proporciona una herramienta confiable para ser utilizada en las prácticas de laboratorio de reología, en donde los alumnos pueden manipular directamente este equipo de bajo costo, evitando el uso de equipo comercial costoso que eventualmente se puede exponer a un daño accidental por parte de los alumnos.

La principal limitación está relacionada con el principio de medición en el equipo que, debido a su propia naturaleza, tiende a limitar el alcance, resolución y exactitud de las lecturas, en comparación con otro tipo de viscosímetros.

Relación con otras áreas

Las actividades realizadas en el presente trabajo están estrechamente relacionadas con las áreas de sistemas dinámicos, instrumentación, control digital y microcontroladores.

Método

El método empleado para diseñar y construir el circuito electrónico para el viscosímetro de cilindros concéntricos se muestra esquemáticamente en la figura 1.

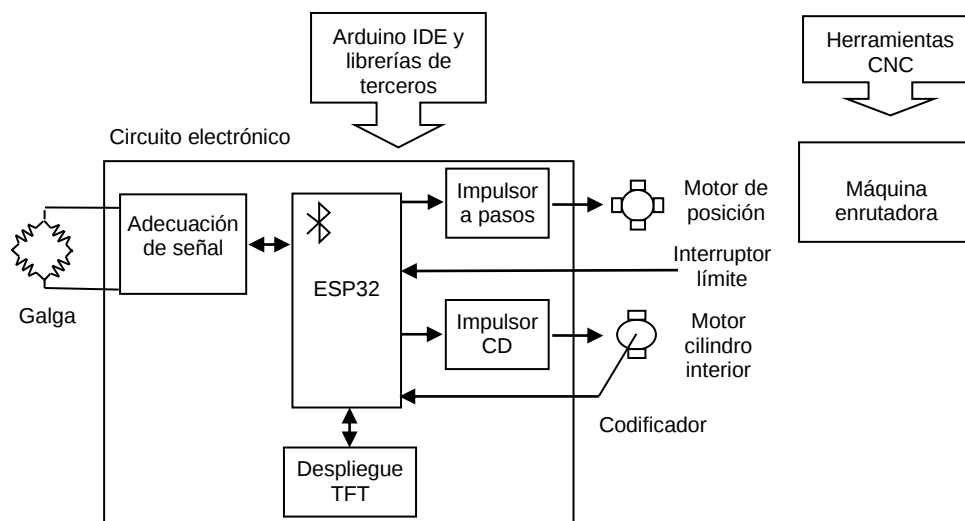


Figura 1. Método por emplear (Elaboración propia).

En la figura 1 se muestran dos procesos independientes:

- Diseño electrónico. Se utiliza el IDE de Arduino y las librerías provistas por los dispositivos electrónicos seleccionados para programar el algoritmo de medición e interfaz con el usuario incrustado en el microcontrolador ESP32. El circuito electrónico manipula los siguientes dispositivos: Un circuito de adecuación de

señal para determinar la fuerza en una galga extensiométrica, un despliegue gráfico táctil, un impulsor silencioso para motor a pasos, un impulsor de CD, conexión con codificador de efecto Hall para determinar la velocidad en el motor del cilindro interior y conexión inalámbrica Bluetooth.

- Construcción. Se utiliza una máquina enrutadora CNC para trazar el diseño del circuito en una placa PCB mediante herramientas CNC de uso libre.

Objetivos y resultados esperados

El objetivo del presente informe es mantener documentado el desarrollo tecnológico que aquí se describe, de forma que pueda ser de utilidad al tomarlo como referencia para desarrollos posteriores.

El objetivo del proyecto es diseñar y construir un circuito para el viscosímetro de cilindros concéntricos que posea características mejoradas y modernas con relación a su primera versión desarrollada en el año de 2013.

Organización del informe

La sección de introducción plantea el problema, el método empleado para resolverlo y expone los resultados esperados. En el capítulo uno se describe el diseño detallado del circuito electrónico para el viscosímetro. Se incluye la descripción hardware, software, procedimiento de operación y su construcción en placa PCB. En el capítulo dos se describen cualitativa y cuantitativamente los resultados obtenidos. Se discute el trabajo desarrollado y se proporciona una breve guía de posibles fallas. Se incluyen las conclusiones, referencias y anexos conteniendo las herramientas que permiten la reproducibilidad de este trabajo.

1 Circuito para el viscosímetro

El circuito electrónico está construido sobre la base del microcontrolador ESP32, un despliegue gráfico táctil TFT, un acondicionador de señal de fuerza, un impulsor silencioso para motor a paso y control de velocidad para motor de CD que incluye circuito impulsor y adquisición de posición con codificador de efecto Hall.

1.1 Descripción

La figura 2 muestra el esquema general del circuito electrónico para el viscosímetro.

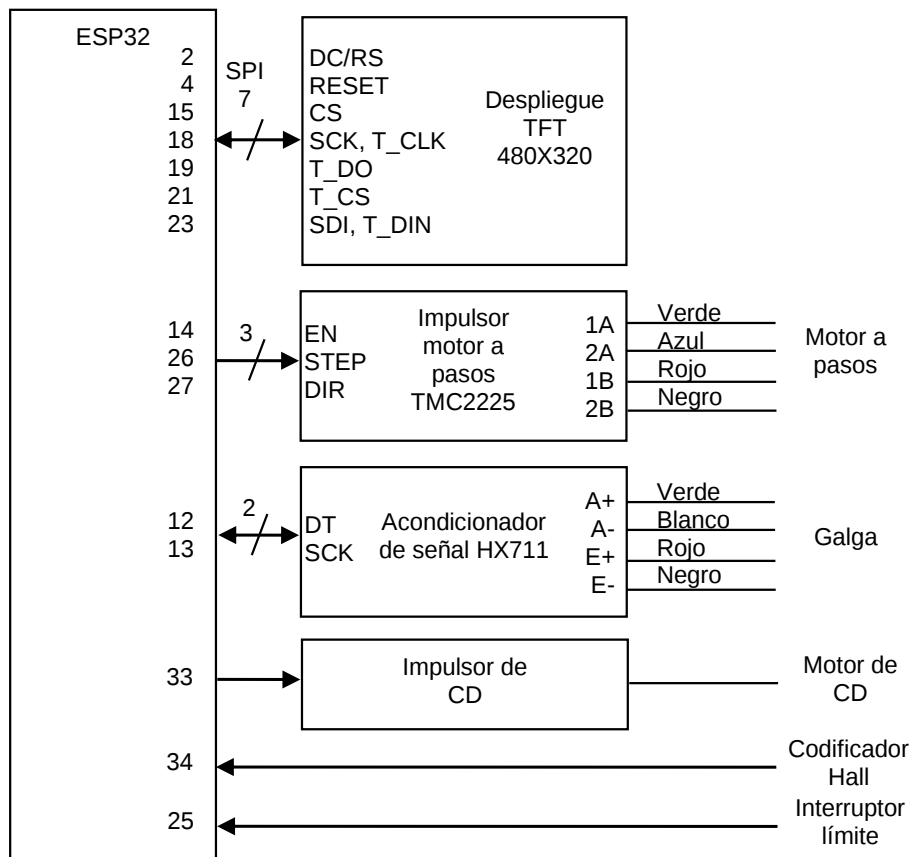


Figura 2. Esquema general para el circuito electrónico del viscosímetro (Elaboración propia).

La figura 2 muestra la conexión de un microcontrolador ESP32 a los dispositivos electrónicos seleccionados para medir la viscosidad de acuerdo con el principio de cilindros concéntricos (Guerra, 2024). Describiendo cada uno de los bloques funcionales de la figura 2:

- El despliegue TFT se conecta mediante su interfaz SPI a las 7 líneas GPIO del microcontrolador ESP32. El despliegue requiere dos canales SPI, uno para las funciones gráficas y otro para las funciones táctiles. Las dos líneas compartidas del estándar SPI entre los dos canales son T_DIN con SDI y T_CLK con SCK,

datos y reloj respectivamente. Las líneas independientes a cada canal corresponden a los selectores CS y T_CS, el reinicio RESET, la línea de datos del canal táctil T_DO y la línea de datos del despliegue DC/RS. Con este esquema de conexión, se envía información para ser mostrada en el canal gráfico y se reciben pulsaciones provenientes del canal táctil (GitHub TFT_eSPI, 2024). Con estos dos elementos es posible construir una interfaz amigable y moderna con el usuario del viscosímetro.

- El impulsor silencioso para motor a pasos TMC2225 requiere como entrada tres señales de datos: Una señal de habilitación EN, una señal que indique el avance de paso por pulso STEP y una señal que indique el sentido horario o antihorario de avance DIR (Llamas, 2024). Estas señales son generadas por el microcontrolador ESP32 de acuerdo con el algoritmo de medición y control. El motor a pasos impulsado en este esquema permite ingresar el cilindro interior en el cilindro exterior del viscosímetro. El límite en el recorrido lineal que realiza el cilindro interior es determinado por el interruptor límite.
- El acondicionador de señal de la galga HX711 se comunica con una modalidad muy parecida al estándar serial I²C, es decir, una línea de datos DT y una línea de reloj SCK. Con este esquema bidireccional de comunicación es posible transferir mediciones de fuerza en la galga hacia el algoritmo de medición en el microcontrolador ESP32 (Random Nerd Tutorials, 2024). Posteriormente, y desarrollando un ejercicio de calibración, la fuerza en la galga se puede correlacionar con la viscosidad del fluido entre los cilindros concéntricos.
- El impulsor de CD en conjunto con el motor de CD y el codificador Hall, conforman un esquema de control de velocidad en lazo cerrado (Ogata, 1996). Mediante este esquema se consigue que el cilindro interior del viscosímetro gire a velocidad angular constante y predeterminada. La rotación del cilindro interior es importante ya que es el principio de medición en el tipo de viscosímetros de cilindros concéntricos (Valera et al, 2013).

Los anexos A, B y C contienen los detalles de la implementación final del esquema de la figura 2.

1.2 Algoritmo de control y medición

Las dos variables de interés que deben ser procesadas con el esquema de la figura 2 son la velocidad angular del cilindro interior, *frecuencia*, y la fuerza en la galga, *fuerza*. Derivado de estas variables y de algunas constantes geométricas es posible calcular la viscosidad de algún fluido que se desplace entre los dos cilindros concéntricos.

Entonces en una actividad regular de medición de viscosidad primero se debe introducir el fluido por analizar entre los cilindros concéntricos y después especificar la velocidad angular deseada entre cilindros. A esta velocidad deseada se le denomina comúnmente *SetPoint*.

La figura 3 muestra el diagrama de flujo que desarrolla el microcontrolador ESP32 como parte del algoritmo de control y medición.

El algoritmo de la figura 3 básicamente es un encuestador del estado de la pantalla táctil y del dato recibido por Bluetooth para ejecutar la actividad indicada. La variable *flag* se utiliza para evitar que la lectura de la cuenta del codificador Hall sea errónea durante el proceso de muestreo (jgustavoam, 2020). La variable *flag* es puesta a *false* en el proceso de lectura y devuelta a *true* en el servicio de atención a la interrupción por pulsos digitales.

El código fuente completo que implementa el algoritmo de la figura 3 se muestra en el Anexo D.

1.3 Procedimiento de operación

La figura 4 muestra el despliegue del circuito electrónico para el viscosímetro justamente al inicio de su operación (al aplicar energía).

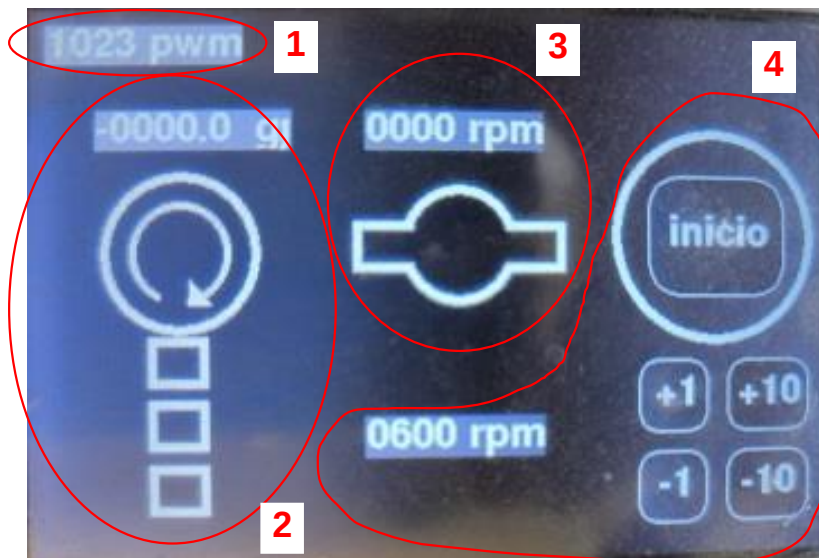


Figura 4. Pantalla inicial del circuito electrónico (Elaboración propia).

La pantalla se divide en las siguientes cuatro secciones:

1. Sección de depuración. Muestra en texto el valor de PWM que esta generando el algoritmo de control de velocidad para el motor del cilindro interior. Esta información no es útil para el operador del dispositivo.
2. Lectura de fuerza en la galga. Representada por un icono de torque y el texto numérico que indica la fuerza en gramos aplicada sobre la galga y que es almacenada en la variable *fuerza*.
3. Lectura de velocidad en el cilindro interior. Representada por un icono de motor y el texto numérico que indica la velocidad angular en rpm's del cilindro interior y que es almacenda en la variable *frecuencia*.

4. Configuración de la medición. Representada por los siguientes elementos: Un botón redondo grande que alterna entre el inicio y el paro de la medición, cuatro botones chicos ovalados que sirven para incrementar o decrementar el valor deseado de la velocidad o *SetPoint*. El texto numérico que muestra el valor del *SetPoint* en rpm's.

En un proceso típico de medición de viscosidad se siguen los siguientes pasos:

1. Se coloca el fluido por analizar en el cilindro exterior.
2. Se energiza el circuito electrónico y se observa que la pantalla muestre el despliegue de la figura 4.
3. Se presionan los botones "+1", "+10", "-1" o "-10" para especificar el valor deseado de la velocidad para la medición. Se observa que el valor deseado o *SetPoint* se despliegue adecuadamente en la sección 4.
4. Se presiona el botón "inicio". En este momento el botón cambia por "parar" y el motor a pasos acciona el dispositivo mecánico para introducir el cilindro interior en el cilindro exterior.
5. Una vez alcanzada la posición adecuada entre los cilindros interior y exterior, la velocidad del motor del cilindro interior se tendrá que ajustar automáticamente al valor de *SetPoint* introducido y la fuerza en la galga se verá afectada debido a la viscosidad del fluido bajo análisis.
6. Cuando el operador considere la medición lo suficientemente estable, deberán capturarse las lecturas de velocidad y fuerza para obtener la viscosidad mediante cálculos analíticos.
7. Una vez obtenidos los resultados del punto 6, el operador puede presionar el botón "parar" para finalizar la medición. En este momento el cilindro interior se desplaza hacia afuera del cilindro exterior con fines de limpieza, cambio de fluido o finalización de la medición.

En forma alternativa, este mismo proceso realizado desde la pantalla TFT se puede realizar desde alguna aplicación para teléfono inteligente como Serial Bluetooth (prometec, 2023). El procedimiento sería el siguiente:

1. Energizar el sistema electrónico.
2. En la sección Configuración/Bluetooth del teléfono buscar los dispositivos disponibles para su conexión, como se muestra en la figura 5.



Figura 5. Sección Bluetooth en el teléfono inteligente (Elaboración propia).

3. El sistema electrónico tiene como nombre “ViscolCAT”, como se muestra en la figura 6.

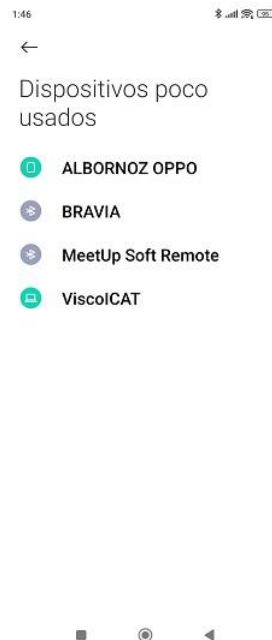


Figura 6. Sistema electrónico Bluetooth “ViscolCAT” (Elaboración propia).

4. Al presionar sobre el nombre “ViscolCAT”, se deberá establecer la vinculación como se muestra en la figura 7.

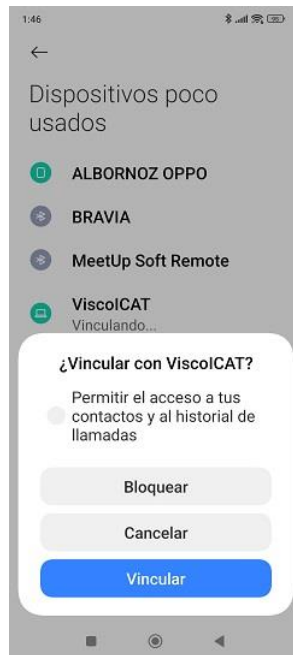


Figura 7. Vinculación entre “ViscoICAT” y teléfono inteligente (Elaboración propia).

5. Una vez vinculado el dispositivo se deberá desplegar una pantalla como la de la figura 8.

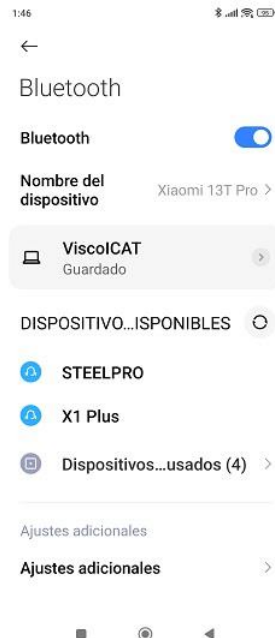


Figura 8. Vinculación exitosa (Elaboración propia).

6. Por otra parte, se deberá ingresar a la tienda de aplicaciones para descargar la aplicación Serial Bluetooth de la figura 9.



Figura 9. Aplicación Serial Bluetooth para teléfono inteligente (Elaboración propia).

7. Al ejecutar la aplicación se desplegará la pantalla de la figura 10.

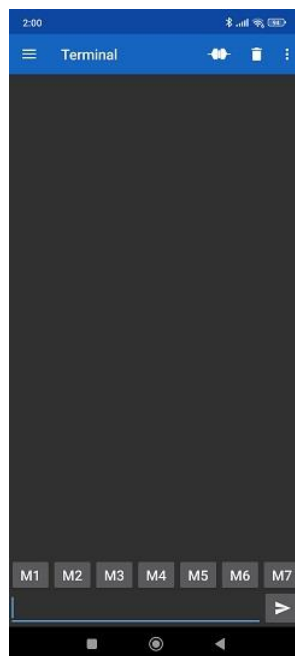


Figura 10. Pantalla principal de Serial Bluetooth (Elaboración propia).

8. Al presionar sobre las tres líneas horizontales de la parte superior derecha de la aplicación se deberá elegir la opción “Devices” como se muestra en la figura 11.

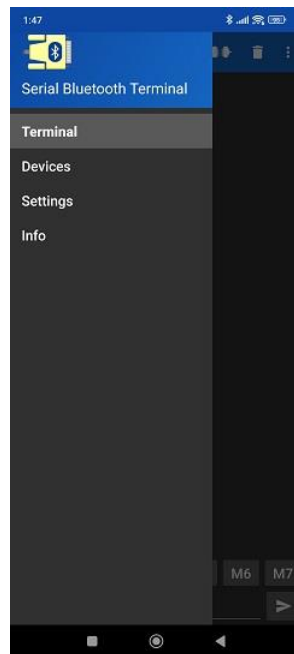


Figura 11. Opción “Devices” de Serial Bluetooth (Elaboración propia).

9. En la opción “Devices” se deberá seleccionar el dispositivo “ViscolCAT” de la figura 12.

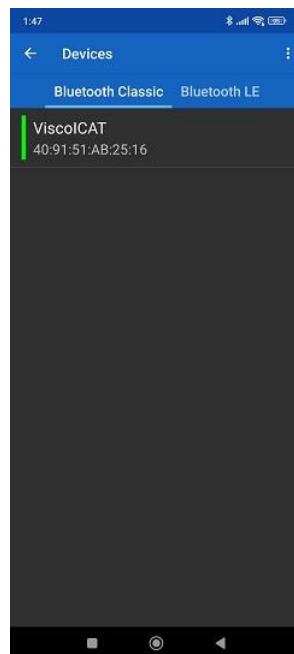


Figura 12. ViscolCAT en Serial Bluetooth (Elaboración propia).

10. Para que la conexión entre el teléfono y el sistema electrónico sea exitosa se deberá desplegar la pantalla de la figura 13.

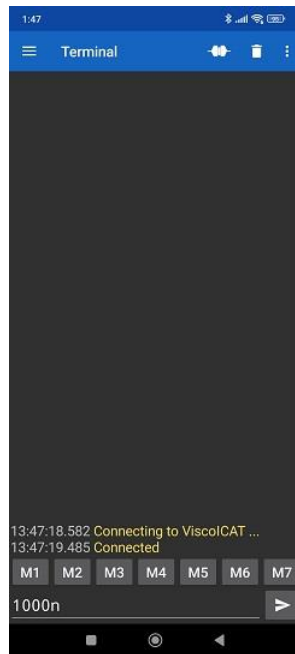


Figura 13. Conexión exitosa de ViscolCAT (Elaboración propia).

11. En esta situación se podrán enviar los comandos de operación de la tabla 1. La figura 14 ejemplifica el comando para establecer el SetPoint a 400rpm. Es importante mencionar que el comando se envía sin terminador de línea.

Tabla 1. Comandos Bluetooth del sistema electrónico (Elaboración propia).

Comando	Descripción
fn	El sistema electrónico devuelve el valor de la fuerza en gramos.
in	El sistema electrónico inicia el proceso de medición de viscosidad.
sn	El sistema electrónico finaliza el proceso de medición de viscosidad.
Daton	El sistema electrónico recibe el numero entero decimal en Dato para establecer el SetPoint
pn	El sistema electrónico devuelve el valor del SetPoint en rpm's.
vn	El sistema electrónico devuelve la velocidad del motor del cilindro interior en rpm's.

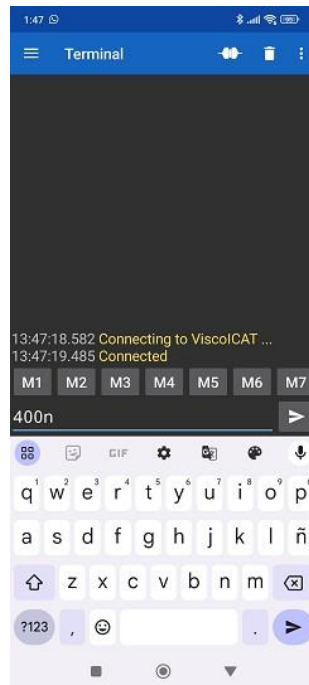


Figura 14. Comando Bluetooth para establecer el SetPoint (Elaboración propia).

12. Finalmente, la figura 15 muestra el comando enviado en texto de color gris y la respuesta del sistema electrónico en texto de color verde lo que indica que el comando se ha ejecutado exitosamente.

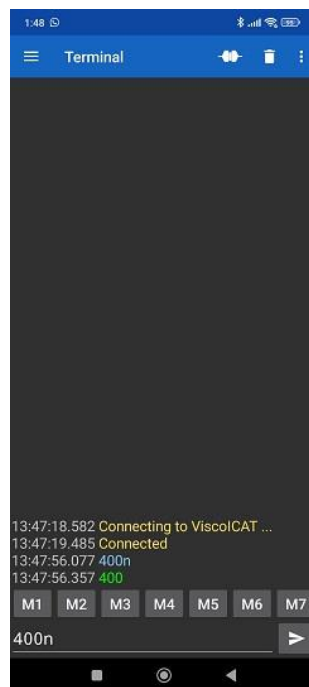


Figura 15. Comando Bluetooth ejecutado exitosamente (Elaboración propia).

1.4 Construcción en máquina enrutadora

El proceso de construcción de la placa PCB que sirve de soporte al circuito electrónico de la figura 2, tiene como base el método desarrollado en la referencia bibliográfica (Valera, Maturano, 2024). El método citado utiliza una máquina enrutadora construida anteriormente y herramientas CAD CAM de uso libre. El proceso se describe a continuación:

- El concepto mostrado en la figura 2 se materializa con el dibujo esquemático del anexo A y circuito impreso del anexo B, elaborados en KiCad (KiCad, 2024). A partir del circuito impreso se generan los archivos Gerber y Excellon para construir la placa PCB.
- Los archivos Gerber y Excellon se transforman en código G mediante el software de uso libre denominado FlatCam (FlatCam, 2024). Los parámetros en FlatCam para el archivo Gerber son los de la tabla 2 y la tabla 3 muestra los parámetros para el archivo Excellon.

Tabla 2. Parámetros FlatCam Gerber (Elaboración propia).

	Parámetro	Valor
	Offset;	(-0.3, 0.65)
Para generar la geometría ISO	Tool día:	0.0053
	Width (# pases):	1
	Pass overlap:	0
Para generar el objeto ISO CNC	Cut Z:	-0.0028
	Travel Z:	0.1
	Feed Rate:	3
	Tool día:	0.0053
Para exportar el código G:	Tool día:	0.0053

Tabla 3. Parámetros FlatCam Excellon (Elaboración propia).

	Parámetro	Valor
	Offset;	(-0.3, 0.65)
Para generar el objeto ISO CNC	Tools:	1-6
	Cut Z:	-0.0026
	Travel Z:	0.1
Para exportar el código G:	Tool día:	0.0053

- Finalmente, el código G se carga en Candle (GitHub Candle, 2024) para su maquinado mediante enrutado.

2 Resultados

Se realizaron ejercicios de evaluación de manera independiente para cada uno de los módulos que conforman el circuito electrónico para el viscosímetro. Esto garantiza la confiabilidad del instrumento para la medición de viscosidad una vez que el circuito sea integrado en un sistema electromecánico de medición.

2.1 Resultados

La figura 16 muestra el circuito desarrollado para el viscosímetro de cilindros concéntricos.

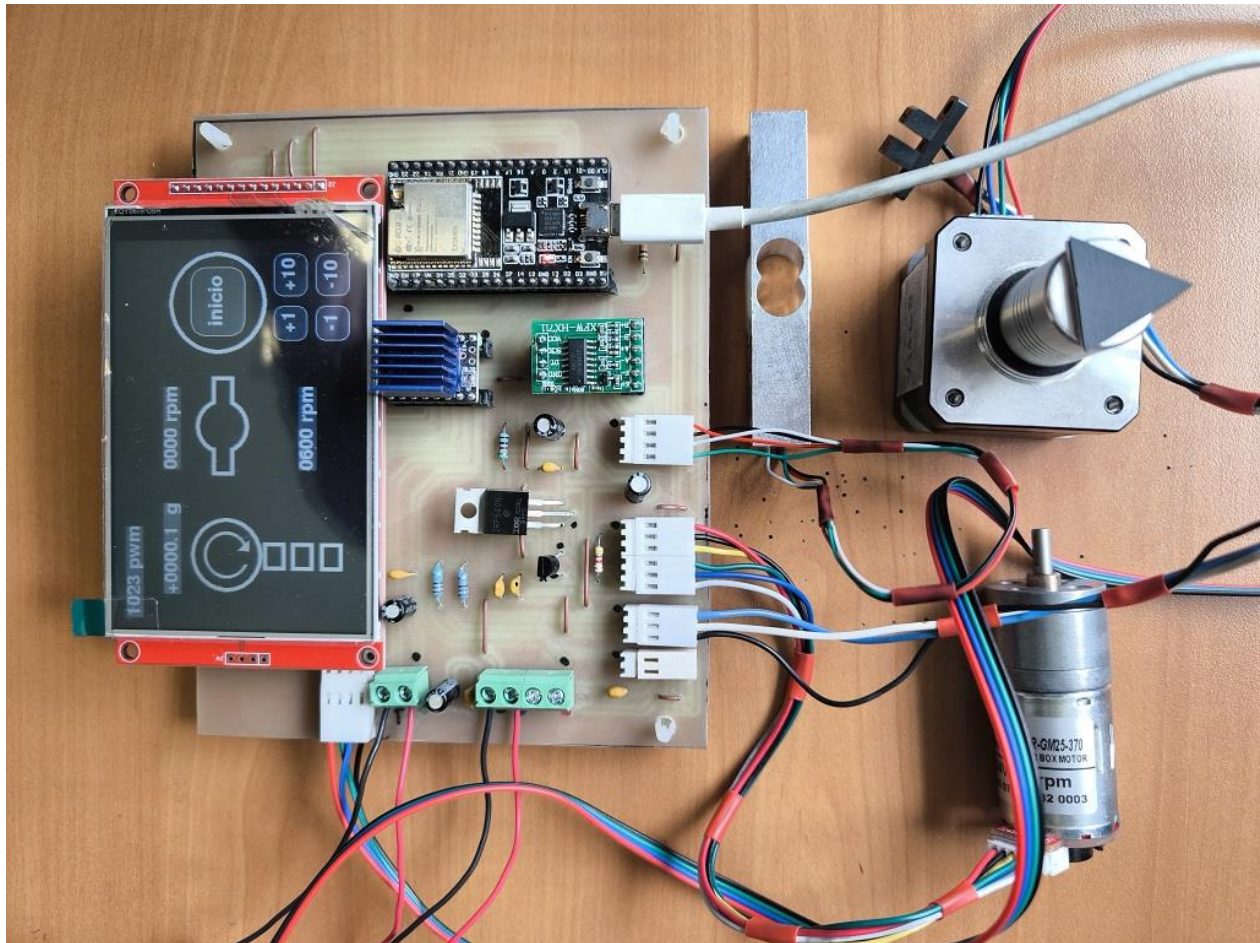


Figura 16. Circuito electrónico (Elaboración propia).

Los módulos que podemos identificar para su evaluación independiente de la figura 2 son los siguientes:

- Interfase con el usuario.
- Motor a pasos e interruptor límite.
- Galga extensiométrica.

- Motor CD y codificador Hall.

Los dos primeros módulos son evaluados cualitativamente mientras que los dos últimos se evalúan cuantitativamente como se describe a continuación.

2.1.1 Cualitativos

Se realizó la simulación de una medición de viscosidad con 200 gr de fuerza a una velocidad angular o Setpoint de 1000 rpm. La figura 17 muestra este proceso realizado de manera local mediante la manipulación del despliegue TFT y la figura 18 muestra el mismo proceso realizado de manera remota a través de la conexión Bluetooth.



Figura 17. Medición local (Elaboración propia).

Como resultado de la medición local, se observa de la figura 17 que la medida de fuerza es de 200.1 gr y la medida de velocidad es de 1012 rpm, muy cercana a la velocidad deseada de 1000 rpm. Con estos datos es posible calcular la medida de viscosidad de forma indirecta.

Por otra parte, y de la figura 18, el resultado de la medición remota mediante Bluetooth arroja como resultados 200.27 gr de fuerza y 1015 rpm de velocidad, valores muy cercanos a 200 gr y 1000 rpm.

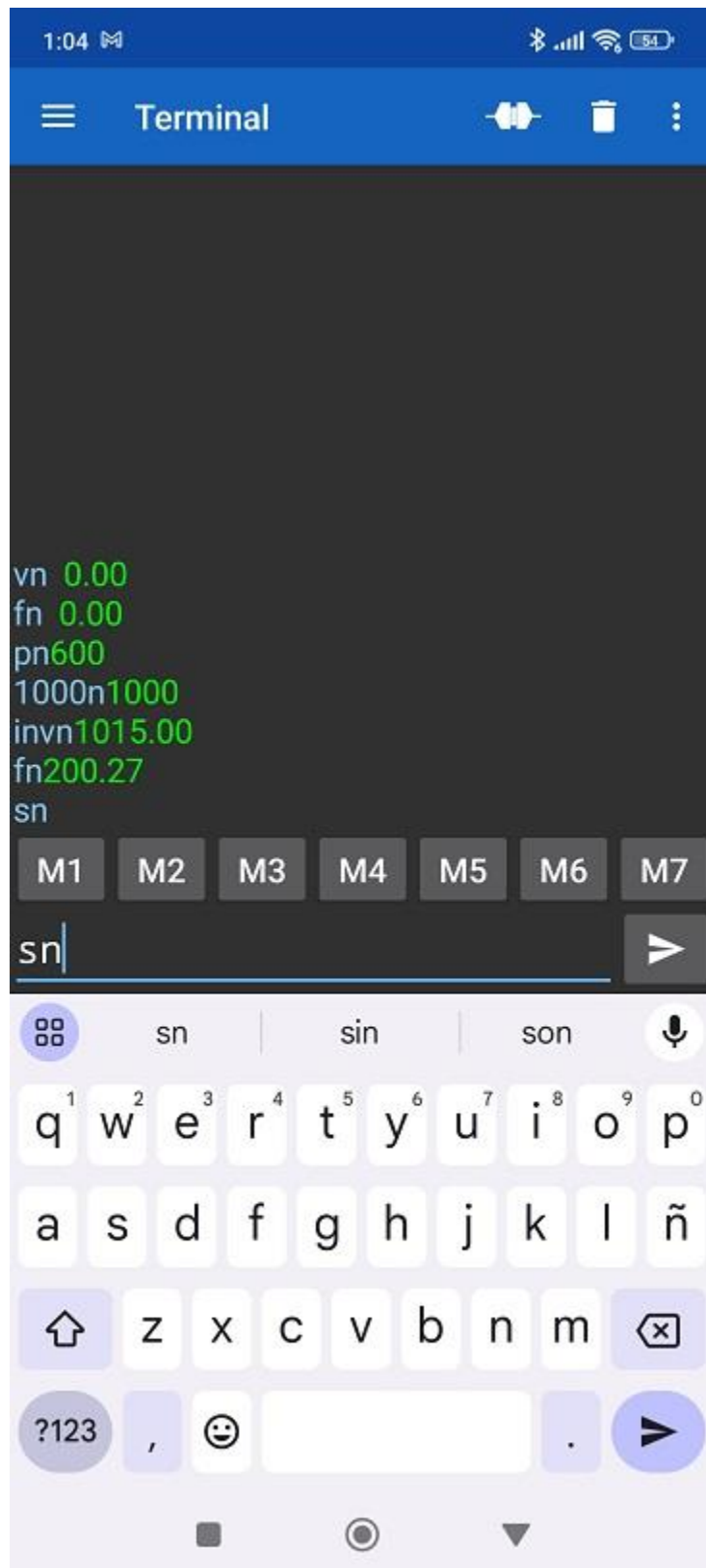


Figura 18. Medición remota (Elaboración propia).

En las pruebas de medición local y remota el desempeño del motor a pasos se demostró al girar en sentido horario para el inicio de la medición y sentido antihorario al finalizar la medición.

2.1.2 Cuantitativos

Se realizaron dos experimentos cuantitativos, uno para evaluar la medida de fuerza y el otro para evaluar el control de velocidad.

La tabla 4 muestra los resultados del experimento que consistió en colocar pesas patrón de 200 gr en la galga y observar la lectura reportada por el circuito electrónico. El experimento se realizó cinco veces consecutivas iniciando en cero y finalizando en cero pasando por los valores de 200 y 400 gr.

Tabla 4. Medición de fuerza (Elaboración propia).

1	2	3	4	5
0.0	0.0	0.0	0.0	0.0
199.9	200.1	200.0	199.9	199.9
400.2	400.3	400.1	400.1	400.2
0.0	0.0	0.0	0.0	0.0

La tabla 4 muestra una desviación máxima de 0.3 gr y una excelente repetibilidad del cero.

La tabla 5 muestra los resultados del experimento que consistió en especificar diversas velocidades objetivo o Setpoint y medir la velocidad real alcanzada por el algoritmo de control de velocidad, tres veces consecutivas.

Tabla 5. Control de velocidad (Elaboración propia).

Setpoint rpm	Velocidad real rpm		
	1	2	3
250	252	246	242
500	501	497	508
750	745	758	748
1000	1008	991	1003
1250	1231	1260	1252
1500	1459	1515	1502
1750	1757	1749	1748
2000	1992	1996	1980

La tabla 5 muestra un desempeño adecuado para el control de velocidad con algunas oportunidades de mejorarlo.

2.2 Discusión y trabajo a futuro

El sistema electrónico desarrollado manipula los módulos funcionales suficientes para implementar un viscosímetro que utiliza el principio de cilindros concéntricos. Si bien no se ha probado en la medición de viscosidad, según nuestra experiencia en el desarrollo de la primera versión del viscosímetro, las pruebas realizadas en este trabajo son suficientes y aseguran la confiabilidad de la medición de viscosidad.

Adicionalmente, se pueden plantear dos aspectos que introducen incertidumbre en la medición de viscosidad que pueden ser aspectos por evaluar en un futuro.

El primero consiste en la configuración del soporte de la galga extensiométrica comúnmente llamado celda de carga. En el caso del presente proyecto se utiliza una celda de carga específica para medir peso, es decir la fuerza con la que la gravedad atrae a los objetos. La geometría de la celda entonces es la específica para sujetarse de manera que la fuerza normal actúe sobre ella. En el caso del presente trabajo la celda es girada 90 grad de manera que el par de torsión horizontal entre cilindros exterior e interior sea determinado por la celda de carga. No se sabe exactamente cuánto influye el cambio de posición de la celda, pero se puede cuantificar a futuro realizando una calibración utilizando torquímetros patrón.

El segundo consiste en que las ligeras variaciones en el control de velocidad pueden introducir incertidumbre en la medición de viscosidad, sobre todo para fluidos con baja viscosidad. En el momento de la elaboración del presente informe no se sabe cuál sería la mencionada incertidumbre. Sin embargo, a futuro se puede realizar una inter-comparación entre viscosímetros utilizando fluidos patrón para determinar la mencionada incertidumbre.

Otro trabajo a futuro es, evidentemente, desarrollar el mecanismo de cilindros concéntricos e integrarle el presente sistema electrónico.

2.3 Guía para detección de fallas

Las posibles fallas que se pueden presentar al operar el circuito electrónico para el viscosímetro y sus posibles soluciones se describen en la tabla 6.

Tabla 6. Guía de fallas y soluciones (Elaboración propia).

Falla	Solución
El despliegue TFT se muestra estático y no responde a las pulsaciones táctiles.	Apagar y volver a encender el circuito electrónico. Revisar que las conexiones sean correctas
El circuito no responde cuando se pulsa el botón "inicio"	Revisar que el interruptor limite no este ocluido por algún obstáculo o suciedad.
El circuito no responde como yo deseo a	Pruebe pulsando con un tiempo un poco

pulsaciones con el dedo.	más prolongado y con una presión los suficientemente fuerte o pulse con la pluma suministrada con el equipo.
La lectura de fuerza durante la medición no es la esperada.	Espere a que el circuito estabilice la velocidad angular alrededor de la velocidad deseada o Setpoint.
Le velocidad angular no se ajusta al Setpoint.	Esto sucede principalmente cuando se establece un Setpoint elevado y súbitamente se cambia a un Setpoint demasiado bajo. Espere algunos segundos a que el algoritmo de control se estabilice o inicie una nueva medición.

2.4 Especificaciones

Alcance en velocidad angular: 250 a 2000 rpm.

Resolución en velocidad angular: 1 rpm.

Exactitud en velocidad angular: ± 20 rpm.

Alcance en fuerza: 0 a 5000 gr.

Resolución en fuerza: 0.1 gr.

Exactitud en fuerza: ± 0.3 gr.

Voltaje: 127 VCA.

Conclusiones

Se presentó el desarrollo y construcción de un circuito electrónico para el viscosímetro de cilindros concéntricos en una versión mejorada respecto a la desarrollada anteriormente. Las mejoras incluyen la calidad en la medición de fuerza, interfaz moderna de operación que incluye despliegue táctil gráfico TFT, operación más silenciosa o menos ruidosa y conexión Bluetooth.

En el circuito electrónico se utilizaron herramientas hardware y software de uso libre para su desarrollo y construcción.

El desempeño del circuito fue evaluado cualitativa y cuantitativamente, lo que asegura su buen funcionamiento una vez que se integre el instrumento electromecánico completo para la medición de viscosidad.

Uno de los objetivos consiste en incorporar este instrumento de medición a la enseñanza de la reología, en donde los alumnos puedan manipular este equipo de bajo costo y mantenimiento accesible sin el riesgo de ocasionar costosos daños en otros equipos de laboratorio.

Al contar con el presente informe se facilita la tarea de replicación del circuito electrónico para el viscosímetro, situación que eventualmente ocurriría si el éxito y confiabilidad de este primer prototipo así lo permite.

26

Anexo B Circuito impreso

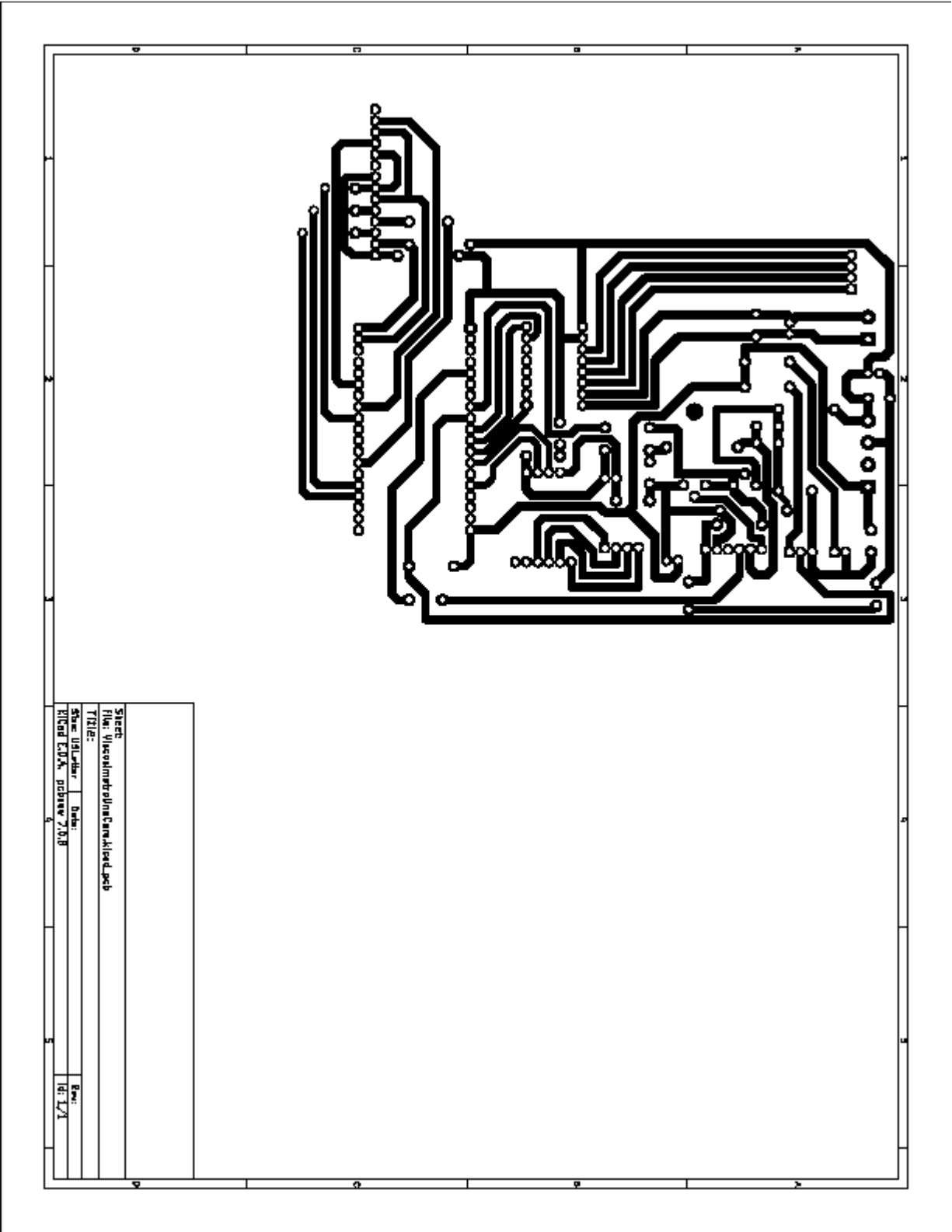


Figura 20. Circuito impreso, soldadura (Elaboración propia).

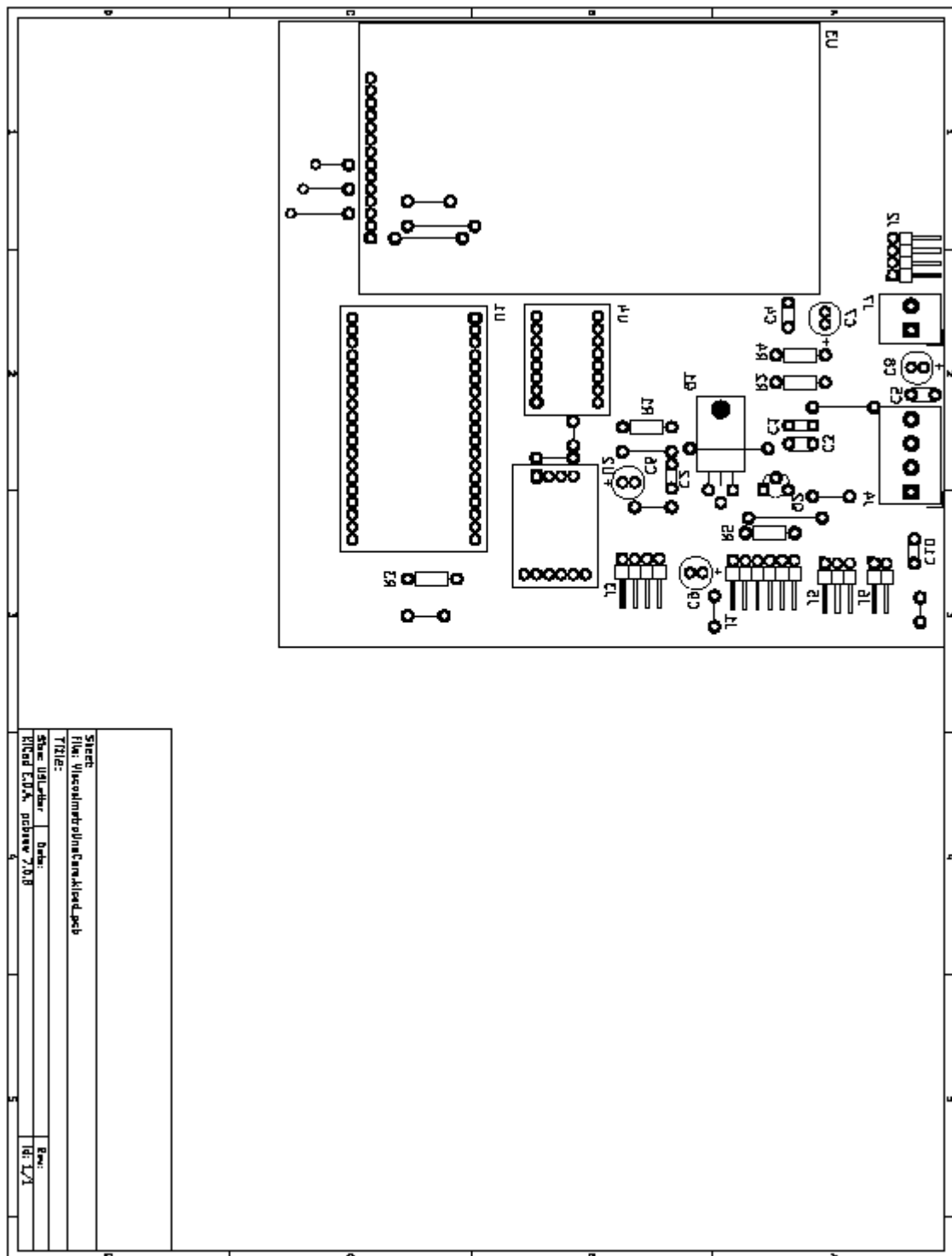


Figura 21. Circuito impreso, componentes (Elaboración propia).

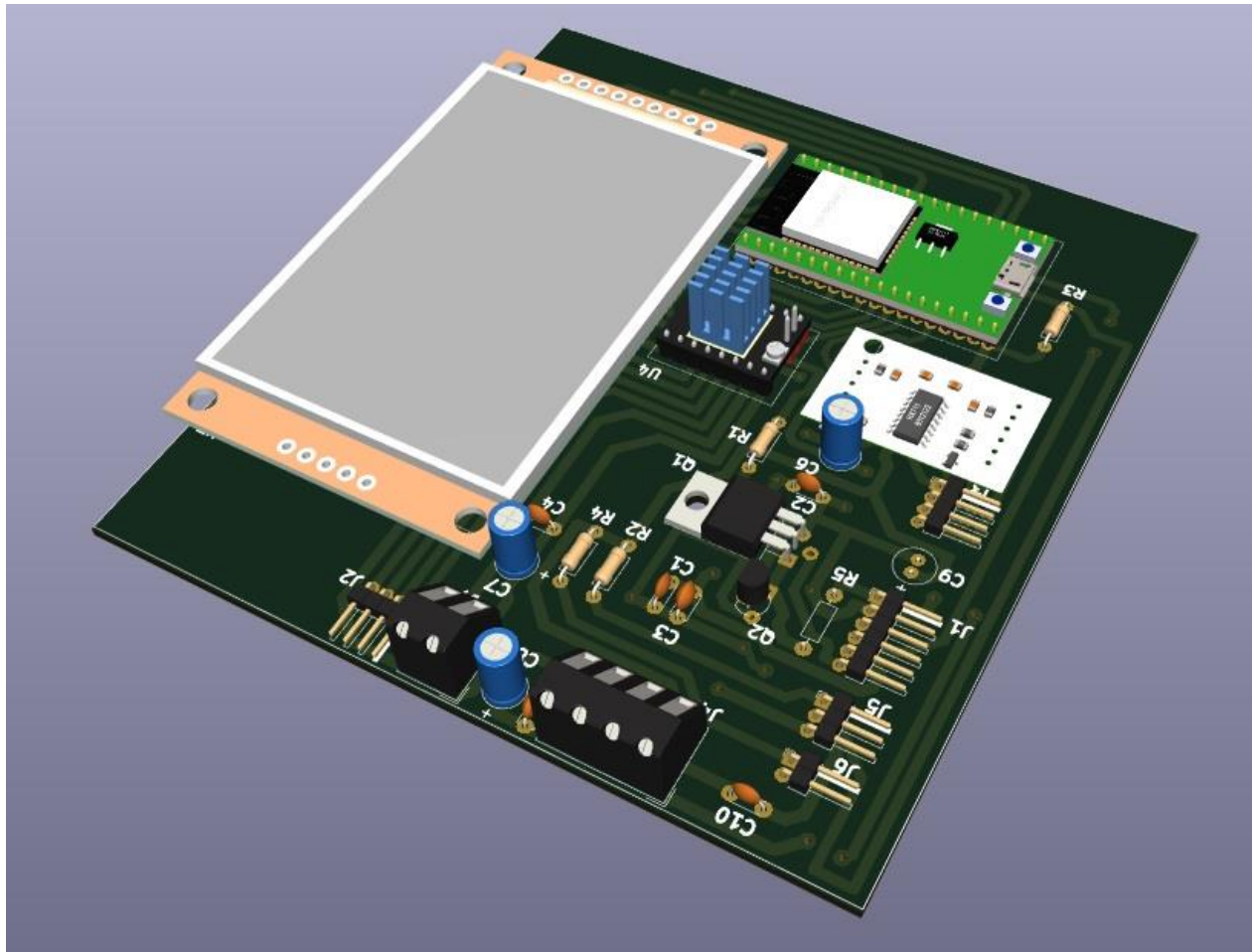


Figura 22. Circuito impreso, vista 3D (Elaboración propia).

Anexo C Lista de partes

Tabla 7. Lista de partes (Elaboración propia).

Id	Identificador	Huella	Cantidad	Identificación
1	R5	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	1	4k7
2	C9, C8, C7, C6	C_Radial_D6.3mm_H11.0mm_P2.50mm	4	10u
3	R4, R2	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	2	330
4	J6	PinHeader_1x02_P2.54mm_Horizontal	1	Luz
5	J5	PinHeader_1x03_P2.54mm_Horizontal	1	Switch
6	J1	PinHeader_1x06_P2.54mm_Horizontal	1	Motor DC Encoder
7	J2	PinHeader_1x04_P2.54mm_Horizontal	1	Motor pasos
8	C1, C4, C2, C10	C_Disc_D4.3mm_W1.9mm_P5.00mm	4	0.01u
9	U4	TMC2225	1	TMC2225
10	R1	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	1	1k
11	Q1	TO-220-3_Horizontal_TabDown	1	IRF540N
12	U1	ESP32 DevKitC V4 32U	1	ESP32 DevKit C 32U
13	U2	HX711	1	HX711
14	J4	Altech_AK300_1x04_P5.00mm_45-Degree	1	Fuentes
15	J3	PinHeader_1x04_P2.54mm_Horizontal	1	Galga
16	R3	R_Axial_DIN0207_L6.3mm_D2.5mm_P10.16mm_Horizontal	1	10k
17	U3	TFT 480X320	1	TFT 480X320
18	C3, C5	C_Disc_D4.7mm_W2.5mm_P5.00mm	2	0.01u
19	J7	Altech_AK300_1x02_P5.00mm_45-Degree	1	Fuente
20	Q2	TO-92_Inline	1	BC547

Anexo D Código fuente

```
// ESP32 Dev Kit C V4 32 U

#include <SPI.h>
#include <TFT_eSPI.h> // Hardware-specific library
#include <Arduino.h>
#include "HX711.h"
#include "soc/rtc.h"

#include "stdio.h" //
Biblioteca STDIO
#include "driver/ledc.h" //
Biblioteca ESP32 LEDC
#include "driver/pcnt.h" //
Biblioteca ESP32 PCNT
#include "soc/pcnt_struct.h"

// Bluetooth
#include "BluetoothSerial.h"
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif
BluetoothSerial BT; // Objeto Bluetooth
char cstr[16];
char result[8]; // Buffer big enough for 7-character float
String message = "";
char incomingChar;

// Motor a pasos
int PUL = 26; //define Pulse pin
int DIR = 27; //define Direction pin
int ENA = 14; //define Enable Pin
const int buttonPin = 25; // the number of the pushbutton pin

// TFT 320X480
TFT_eSPI tft = TFT_eSPI(); // Invoke custom library
TFT_eSPI_Button mas, menos, mas10, menos10, iniciar; // Botones
char buffer[30];
bool inicio="true";

// HX711 circuit wiring load cel
const int LOADCELL_DOUT_PIN = 12;
const int LOADCELL_SCK_PIN = 13;
float fuerza=0.0;
HX711 scale;

// PWM
const int ledPin = 33; // PWM
int SetPoint=600;
int error;
int PWM=1023;
float PWMF=1023;
// setting PWM properties
const int freq = 5000;
const int ledChannel = 0;
```

```

const int resolution = 10;

// Encoder
#define PCNT_COUNT_UNIT          PCNT_UNIT_0 //
Unidade 0 do Contador de pulso PCNT do ESP32
#define PCNT_COUNT_CHANNEL      PCNT_CHANNEL_0 //
Canal 0 do Contador de pulso PCNT do ESP32
#define PCNT_INPUT_SIG_IO       GPIO_NUM_34 //
Entrada do Frequencimetro - GPIO 34
#define PCNT_INPUT_CTRL_IO      GPIO_NUM_35 //
Pino de controle do PCNT - HIGH = count up, LOW = count down
#define OUTPUT_CONTROL_GPIO     GPIO_NUM_32 //
Saida do timer - Controla a contagem - GPIO_32
#define PCNT_H_LIM_VAL          overflow //
Limite superior de contagem
bool          flag              = true; //
Indicador de fim de contagem - libera impressão
uint32_t      overflow          = 20000; //
Valor maximo para overflow do contador PCNT
int16_t       pulses            = 0; //
Quantidade de pulsos contados
uint32_t      multPulses        = 0; //
Quantidade de overflows do contador PCNT
uint32_t      janela            = 1000000; //
Tempo de amostragem de 1 segundo para a contagem de pulsos 999990
uint32_t      mDuty              = 0; //
Valor calculado do ciclo de carga
uint32_t      resolucao          = 0; //
Valor calculado da resolucao
float          frequencia        = 0; //
Variavel para calculo de frequencia
esp_timer_create_args_t create_args; //
Argumentos do ESP-Timer
esp_timer_handle_t timer_handle; //
Instancia de ESP-Timer
portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED; //
variavel tipo portMUX_TYPE para sincronismo

static void IRAM_ATTR pcnt_intr_handler(void *arg) //
Contagem do contador de Overflow
{
    portENTER_CRITICAL_ISR(&timerMux); //
    Bloqueia nova interrupção
    multPulses++; //
    Incrementa contador de overflow
    PCNT.int_clr.val = BIT(PCNT_COUNT_UNIT); //
    Limpa indicador de interrupção
    portEXIT_CRITICAL_ISR(&timerMux); //
    Libera nova interrupção
}

void inicializa_contador(void) //
Inicializacao do contador de pulsos
{
    pcnt_config_t pcnt_config = { }; //
    Instancia PCNT config

```

```

    pcnt_config.pulse_gpio_num = PCNT_INPUT_SIG_IO; //
Configura GPIO para entrada dos pulsos
    pcnt_config.ctrl_gpio_num = PCNT_INPUT_CTRL_IO; //
Configura GPIO para controle da contagem
    pcnt_config.unit = PCNT_COUNT_UNIT; //
Unidade de contagem PCNT - 0
    pcnt_config.channel = PCNT_COUNT_CHANNEL; //
Canal de contagem PCNT - 0
    pcnt_config.counter_h_lim = PCNT_H_LIM_VAL; //
Limite maximo de contagem - 20000
    pcnt_config.pos_mode = PCNT_COUNT_INC; //
Incrementa contagem na subida do pulso
    pcnt_config.neg_mode = PCNT_COUNT_INC; //
Incrementa contagem na descida do pulso
    pcnt_config.lctrl_mode = PCNT_MODE_DISABLE; //
PCNT - modo lctrl desabilitado
    pcnt_config.hctrl_mode = PCNT_MODE_KEEP; //
PCNT - modo hctrl - se HIGH conta incrementando
    pcnt_unit_config(&pcnt_config); //
Configura o contador PCNT

    pcnt_counter_pause(PCNT_COUNT_UNIT); //
Pausa o contador PCNT
    pcnt_counter_clear(PCNT_COUNT_UNIT); //
Zera o contador PCNT

    pcnt_event_enable(PCNT_COUNT_UNIT, PCNT_EVT_H_LIM); //
Configura limite superior de contagem
    pcnt_isr_register(pcnt_intr_handler, NULL, 0, NULL); //
Configura rotina de interrupção do PCNT
    pcnt_intr_enable(PCNT_COUNT_UNIT); //
Habilita interrupções do PCNT

    pcnt_counter_resume(PCNT_COUNT_UNIT); //
Reinicia a contagem no contador PCNT
}

void tempo_controle(void *p) //
Fim de tempo de leitura de pulsos
{
    gpio_set_level(OUTPUT_CONTROL_GPIO, 0); //
Controle do PCNT - para o contador
    pcnt_get_counter_value(PCNT_COUNT_UNIT, &pulses); //
Obtem o valor contado no PCNT
    flag = true; //
Informa que ocorreu interrupção de controle
}

void inicializa_frequencimetro()
{
    inicializa_contador(); //
Inicializa o contador de pulsos PCNT

    gpio_pad_select_gpio(OUTPUT_CONTROL_GPIO); //
Define o port de controle
    gpio_set_direction(OUTPUT_CONTROL_GPIO, GPIO_MODE_OUTPUT); //
Define o port de controle como saída

```

```

    create_args.callback = tempo_controle; //
Instancia o tempo de controle
    esp_timer_create(&create_args, &timer_handle); //
Cria parametros do timer

    gpio_matrix_in(PCNT_INPUT_SIG_IO, SIG_IN_FUNC226_IDX, false); //
Direciona a entrada de pulsos
}

void setup(void)
{
// Inicia motor a pasos
    pinMode (PUL, OUTPUT);
    pinMode (DIR, OUTPUT);
    pinMode (ENA, OUTPUT);
    pinMode(buttonPin, INPUT_PULLUP);
    delay(10);
    digitalWrite(ENA, LOW);

// Inicia tft
    tft.init();
    tft.setRotation(1);
    drawPanel();

// Inicia celda de carga
    scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
    scale.set_scale(-500.52); // this value is obtained by
calibrating the scale with known weights; see the README for details
    scale.tare(); // reset the scale to 0

// Inicia encoder y pwm
    // configure LED PWM functionalites
    ledcSetup(ledChannel, freq, resolution);
    // attach the channel to the GPIO to be controlled
    ledcAttachPin(ledPin, ledChannel);
    ledcWrite(ledChannel, PWM);
    inicializa_frequencimetro(); //
Inicializa o frequencimetro

// Inicia BlueTooth
    BT.begin("ViscoICAT"); // Nombre de su dispositivo Bluetooth y en modo
esclavo
}

void loop()
{
    uint16_t t_x = 0, t_y = 0; // To store the touch coordinates

    // Pressed will be set true is there is a valid touch on the screen
    bool pressed = tft.getTouch(&t_x, &t_y);
    t_y=320-t_y; // Transforma coordenadas solo para el touch

    if(pressed)
    {
        snprintf(buffer , sizeof buffer, "%03d, %03d", t_x, t_y);
        statusxy(buffer);
    }
}

```

```

}

// / Check if any key coordinate boxes contain the touch coordinates
if (pressed && iniciar.contains(t_x, t_y))
{
    iniciar.press(true); // tell the button it is pressed
} else
{
    iniciar.press(false); // tell the button it is NOT pressed
}

if (pressed && mas.contains(t_x, t_y))
{
    mas.press(true); // tell the button it is pressed
} else
{
    mas.press(false); // tell the button it is NOT pressed
}

if (pressed && menos.contains(t_x, t_y))
{
    menos.press(true); // tell the button it is pressed
} else
{
    menos.press(false); // tell the button it is NOT pressed
}

if (pressed && mas10.contains(t_x, t_y))
{
    mas10.press(true); // tell the button it is pressed
} else
{
    mas10.press(false); // tell the button it is NOT pressed
}

if (pressed && menos10.contains(t_x, t_y))
{
    menos10.press(true); // tell the button it is pressed
} else
{
    menos10.press(false); // tell the button it is NOT pressed
}

// Ejecutar aqui las acciones asignadas al boton
if (iniciar.justReleased())
{
    iniciar.drawButton(); // draw normal
    if(inicio==true)
    {
        inicio=false;
        iniciar.initButtonUL(&tft, 364, 96, // x, y, w, h, outline, fill, text
            86, 71, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
            "parar", 1);
        iniciar.drawButton();
        digitalWrite(DIR, LOW);
        do
        {

```

```

        digitalWrite(PUL, HIGH);
        delayMicroseconds(800);
        digitalWrite(PUL, LOW);
        delayMicroseconds(800);
    } while (digitalRead(buttonPin)==0);
    PWMF=(0.0000001*(float)SetPoint*(float)SetPoint*(float)SetPoint)-
(0.0005*(float)SetPoint*(float)SetPoint)+(0.3627*(float)SetPoint)+(float)563.6
5;

    PWM=(int)PWMF;
    ledcWrite(ledChannel, 10);
    delay(10);
    ledcWrite(ledChannel, PWM);
}
else
{
    inicio=true;
    iniciar.initButtonUL(&tft, 364, 96, // x, y, w, h, outline, fill, text
        86, 71, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
        "inicio", 1);
    iniciar.drawButton();
    PWM=1023;
    ledcWrite(ledChannel, PWM);
    digitalWrite(DIR, HIGH);
    for (int i = 0; i < 1000; i++) //Backward 5000 steps
    {
        digitalWrite(PUL, HIGH);
        delayMicroseconds(800);
        digitalWrite(PUL, LOW);
        delayMicroseconds(800);
    }
}
}
if (mas.justReleased())
{
    mas.drawButton(); // draw normal
    if((SetPoint>=0) && (SetPoint<2200)) SetPoint++;
    sprintf(buffer , sizeof buffer, "%04d rpm", SetPoint);
    tft.drawString(buffer, 197, 236);
}
if (menos.justReleased())
{
    menos.drawButton(); // draw normal
    if((SetPoint>0) && (SetPoint<=2200)) SetPoint--;
    sprintf(buffer , sizeof buffer, "%04d rpm", SetPoint);
    tft.drawString(buffer, 197, 236);
}
if (mas10.justReleased())
{
    mas10.drawButton(); // draw normal
    if((SetPoint>=0) && (SetPoint<2190)) SetPoint=SetPoint+10;
    sprintf(buffer , sizeof buffer, "%04d rpm", SetPoint);
    tft.drawString(buffer, 197, 236);
}
if (menos10.justReleased())
{
    menos10.drawButton(); // draw normal
    if((SetPoint>=10) && (SetPoint<=2200)) SetPoint=SetPoint-10;
}

```

```

    snprintf(buffer , sizeof buffer, "%04d rpm", SetPoint);
    tft.drawString(buffer, 197, 236);
}

// Dibujar aqui el estado invertido de los botones
if (iniciar.justPressed())
{
    iniciar.drawButton(true); // draw invert
}
if (mas.justPressed())
{
    mas.drawButton(true); // draw invert
}
if (menos.justPressed())
{
    menos.drawButton(true); // draw invert
}
if (mas10.justPressed())
{
    mas10.drawButton(true); // draw invert
}
if (menos10.justPressed())
{
    menos10.drawButton(true); // draw invert
}

fuerza=scale.get_units(5), 2;
if(fuerza<0.0) fuerza=0.0;
snprintf(buffer , sizeof buffer, "%+07.1f g ", fuerza);
tft.drawString(buffer, 40, 60);

if (flag == true) //
Se a contagem tiver terminado
{
    flag = false; //
Impede nova impressao
    frecuencia = (pulses + (multPulses * overflow)) / 2 ; //
Calcula a soma dos pulsos contados no PCNT
    multPulses = 0; //
Zera contador de overflow
    if (inicio==false)
    {
        if(SetPoint==0)
        {
            PWM=1023;
            ledcWrite(ledChannel, PWM);
        }
        else
        {
            if(frecuencia>SetPoint)
            {
                error=abs(frecuencia-SetPoint);
                if(error>5)
                {
                    if((PWM>=0)&&(PWM<1018)) PWM=PWM+5;
                }
                else if(error>0)

```

```

        {
            if ((PWM>=0) && (PWM<1023)) PWM++;
        }
    }
    else
    {
        error=abs(frequencia-SetPoint);
        if(error>10)
        {
            if ((PWM>5) && (PWM<=1023)) PWM=PWM-5;
        }
        else if(error>0)
        {
            if ((PWM>0) && (PWM<=1023)) PWM--;
        }
    }
    ledcWrite(ledChannel, PWM);
}

}
snprintf(buffer , sizeof buffer, "%04.0f rpm", frequencia);
tft.drawString(buffer, 197, 60);
snprintf(buffer , sizeof buffer, "%04d pwm", PWM);
tft.drawString(buffer, 10, 10);

    delay (100); //
Delay 100 ms
    pcnt_counter_clear(PCNT_COUNT_UNIT); //
Zera o contador PCNT
    esp_timer_start_once(timer_handle, janela); //
Inicia contador de tempo de 1 segundo
    gpio_set_level(OUTPUT_CONTROL_GPIO, 1); //
Porta de controle - habilita contagem dos pulsos
}

if (BT.available())
{
    char incomingChar = BT.read();
    if (incomingChar != 'n')
    {
        message += String(incomingChar);
    }
    else
    {
        if(message=="v")
        {
            dtostrf(frequencia, 6, 2, result); // Leave room for too large
numbers!
            BT.println(result); // Envía el mensaje de texto a través de BT Serial
            message="";
        }
        else if(message=="p")
        {
            BT.println(SetPoint); // Envía el mensaje de texto a través de BT
Serial
            message="";
        }
        else if(message=="f")

```

```

{
    dtostrf(fuerza, 6, 2, result); // Leave room for too large numbers!
    BT.println(result); // Envía el mensaje de texto a través de BT Serial
    message="";
}
else if(message=="i")
{
    if(inicio==true)
    {
        inicio=false;
        iniciar.initButtonUL(&tft, 364, 96, // x, y, w, h, outline, fill,
text
            86, 71, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
            "parar", 1);
        iniciar.drawButton();
        digitalWrite(DIR, LOW);
        do
        {
            digitalWrite(PUL, HIGH);
            delayMicroseconds(800);
            digitalWrite(PUL, LOW);
            delayMicroseconds(800);
        } while (digitalRead(buttonPin)==0);
        PWMF=(0.0000001*(float)SetPoint*(float)SetPoint*(float)SetPoint)-
(0.0005*(float)SetPoint*(float)SetPoint)+(0.3627*(float)SetPoint)+(float)563.6
5;
        PWM=(int)PWMF;
        ledcWrite(ledChannel, 10);
        delay(10);
        ledcWrite(ledChannel, PWM);
        message="";
    }
}
else if(message=="s")
{
    if (inicio==false)
    {
        inicio=true;
        iniciar.initButtonUL(&tft, 364, 96, // x, y, w, h, outline, fill,
text
            86, 71, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
            "inicio", 1);
        iniciar.drawButton();
        PWM=1023;
        ledcWrite(ledChannel, PWM);
        digitalWrite(DIR, HIGH);
        for (int i = 0; i < 1000; i++) //Backward 5000 steps
        {
            digitalWrite(PUL, HIGH);
            delayMicroseconds(800);
            digitalWrite(PUL, LOW);
            delayMicroseconds(800);
        }
        message="";
    }
}
}

```

```

        else
if((message!="f")||(message!="i")||(message!="s")||(message!="p")||(message!="
v"))
{
    SetPoint=message.toInt();
    BT.println(SetPoint); // Envía el mensaje de texto a través de BT
Serial
    message="";
    snprintf(buffer , sizeof buffer, "%04d rpm", SetPoint);
    tft.drawString(buffer, 197, 236);
}
}
}

void drawPanel()
{
    tft.fillScreen(TFT_BLACK);
    tft.setFreeFont(&FreeSansBold12pt7b);

    // Draw torquimetro
    tft.fillRect(72, 263, 35, 29, TFT_WHITE);
    tft.fillRect(77, 268, 25, 19, TFT_BLACK);

    tft.fillRect(72, 226, 35, 29, TFT_WHITE);
    tft.fillRect(77, 231, 25, 19, TFT_BLACK);

    tft.fillRect(72, 190, 35, 29, TFT_WHITE);
    tft.fillRect(77, 195, 25, 19, TFT_BLACK);

    tft.fillCircle(90, 142, 46, TFT_WHITE);
    tft.fillCircle(90, 142, 41, TFT_BLACK);
    tft.drawCircle(90, 142, 26, TFT_WHITE);
    tft.drawCircle(90, 142, 27, TFT_WHITE);
    tft.drawCircle(90, 142, 28, TFT_WHITE);
    tft.drawCircle(90, 142, 29, TFT_WHITE);
    tft.fillRect(79, 156, 16, 16, TFT_BLACK);
    tft.fillTriangle(96, 157, 108, 171, 95, 171, TFT_WHITE);

    // Draw motor
    tft.fillCircle(251, 137, 34, TFT_WHITE);
    tft.fillCircle(251, 137, 29, TFT_BLACK);

    tft.fillRect(190, 124, 35, 29, TFT_WHITE);
    tft.fillRect(195, 129, 33, 19, TFT_BLACK);

    tft.fillRect(279, 124, 35, 29, TFT_WHITE);
    tft.fillRect(277, 129, 32, 19, TFT_BLACK);

    // Draw start button
    tft.fillCircle(405, 131, 63, TFT_WHITE);
    tft.fillCircle(405, 131, 58, TFT_BLACK);

    // Draw buttons
    iniciar.initButtonUL(&tft, 364, 96, // x, y, w, h, outline, fill, text
                        86, 71, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
                        "inicio", 1);

```

```

iniciar.drawButton();

mas.initButtonUL(&tft, 358, 204, // x, y, w, h, outline, fill, text
                43, 43, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
                "+1", 1);
mas.drawButton();

menos.initButtonUL(&tft, 358, 258, // x, y, w, h, outline, fill, text
                  43, 43, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
                  "-1", 1);
menos.drawButton();

mas10.initButtonUL(&tft, 413, 204, // x, y, w, h, outline, fill, text
                  53, 43, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
                  "+10", 1);
mas10.drawButton();

menos10.initButtonUL(&tft, 413, 258, // x, y, w, h, outline, fill, text
                    53, 43, TFT_WHITE, TFT_DARKGREY, TFT_WHITE,
                    "-10", 1);
menos10.drawButton();

// Draw text
snprintf(buffer , sizeof buffer, "%+07.1f g ", fuerza);
tft.drawString(buffer, 40, 60);

snprintf(buffer , sizeof buffer, "%04.0f rpm", frecuencia);
tft.drawString(buffer, 197, 60);

snprintf(buffer , sizeof buffer, "%04d rpm", SetPoint);
tft.drawString(buffer, 197, 236);
}

void statusxy(const char *msg)
{
    tft.drawString(msg, 350, 10);
}

```

Referencias bibliográficas

FlatCam (2024) FlatCAM: Free and Open-source PCB CAM [Internet] Disponible desde <<http://flatcam.org/>> [Acceso 9 de mayo de 2024].

GitHub Candle (2024) Candle [Internet] Disponible desde <<https://github.com/Denvi/Candle>> [Acceso 9 de mayo de 2024].

GitHub TFT_eSPI (2024) TFT_eSPI [Internet] Disponible desde <https://github.com/Bodmer/TFT_eSPI> [Acceso 29 de julio de 2024].

Guerra JC (2024) Programar fácil [Internet] Disponible desde <<https://programarfacil.com/esp8266/esp32/>> [Acceso 29 de julio de 2024].

jpgustavoam (2020) Best Frequency Meter ever made with ESP32 - awesome! [Internet] Disponible desde <<https://esp32.com/viewtopic.php?t=17018#p64156>> [Acceso 29 de julio de 2024].

KiCad (2024) A Cross Platform and Open Source Electronics Design Automation Suite [Internet] Disponible desde <<https://www.kicad.org/>> [Acceso 9 de mayo de 2024].

Llamas L (2024) Motores paso a paso en silencio con Arduino y los driver TMC2100, TMC2130 y TMC2208 [Internet] Disponible desde <<https://www.luisllamas.es/motores-paso-a-paso-en-silencio-con-arduino-y-los-driver-tmc2100-tmc2130-y-tmc2208/>> [Acceso 29 de julio de 2024].

Ogata (1996) **Sistemas de control en tiempo discreto**. Pearson, pp 756.

prometec (2023) ESP32 Serial Terminal Bluetooth [Internet] Disponible desde <<https://www.prometec.net/esp32-serial-bluetooth/>> [Acceso 29 de julio de 2024].

Random Nerd Tutorials (2024) Arduino with Load Cell and HX711 Amplifier (Digital Scale) [Internet] Disponible desde <<https://randomnerdtutorials.com/arduino-load-cell-hx711/>> [Acceso 29 de julio de 2024].

Valera B et al. (2013) **Construcción de un viscosímetro de cilindros concéntricos con rodamiento mecánico**. Informe técnico ICAT UNAM, pp 54.

Valera B Maturano V. (2024) **Circuito para el control de temperatura y su construcción en enrutador CNC**. Informe técnico ICAT UNAM, pp 43.