Universidad de Costa Rica
Escuela de Ingeniería Eléctrica
IE0624 - Laboratorio de Microcontroladores

Laboratorio 4

Sebastián Vargas Quesada, C18295

30 de octubre del 2024, Ciclo II

1. Resumen

En el presente laboratorio, se desarrolló un sismógrafo y un lector del nivel de batería utilizando el giroscopio L3GD20 y los pines ADC integrados en el microcontrolador STM32F429. Para su implementación, se emplearon componentes electrónicos adicionales, como resistencias, una batería de 9V y un conector para dicha batería. Además, se utilizaron protocolos de comunicación como SPI para interactuar con el giroscopio y la pantalla LCD. Todos los datos obtenidos pueden transmitirse a un script de Python a través de USART, donde son procesados y enviados a la plataforma de IoT Thingsboard, que visualiza la información mediante distintos widgets. La lógica del programa fue codificada en lenguaje C, utilizando los ejemplos proporcionados por la biblioteca libopencm3. Por otro lado, se utilizó Python para procesar los datos y enviarlos a la web. Durante el desarrollo, la configuración del giroscopio representó un desafío, ya que debía realizarse mediante el protocolo SPI. Sin embargo, se logró configurar correctamente el giroscopio y obtener datos de los diferentes ejes. El repositorio del proyecto se puede consultar en el siguiente enlace: https://github.com/sebasvq106/microcontroladores.

2. Nota Teorica

2.1. Información General del Microcontrolador

Para la realización de este laboratorio, se utilizará el STM32F429, un microcontrolador de 32 bits de la familia STM32, fabricado por STMicroelectronics. Este microcontrolador se basa en la arquitectura ARM Cortex-M4, conocida por su alto rendimiento y capacidades de procesamiento, además, este microcontrolador opera a una frecuencia de 180 MHz lo que lo convierte en una opción ideal para aplicaciones que requieren eficiencia y velocidad en el manejo de datos.

En este laboratorio, nos enfocaremos en el uso de los pines de entrada/salida (GPIO), el conversor analógico a digital (ADC) y las funcionalidades de comunicación que ofrece el STM32F429. Este microcontrolador cuenta con hasta 168 pines de GPIO, que se pueden configurar de manera flexible para funcionar como entradas o salidas digitales, así como hasta 3 conversores ADC de 12 bits,

que permiten la lectura precisa de señales analógicas. Además, el STM32F429 soporta múltiples interfaces de comunicación, como USART, I2C y SPI, facilitando la transmisión de datos entre el microcontrolador y otros dispositivos, como sensores y módulos de comunicación.

2.2. Diagrama de bloques

A continuación se muestra el diagrama de bloques del STM32F429:

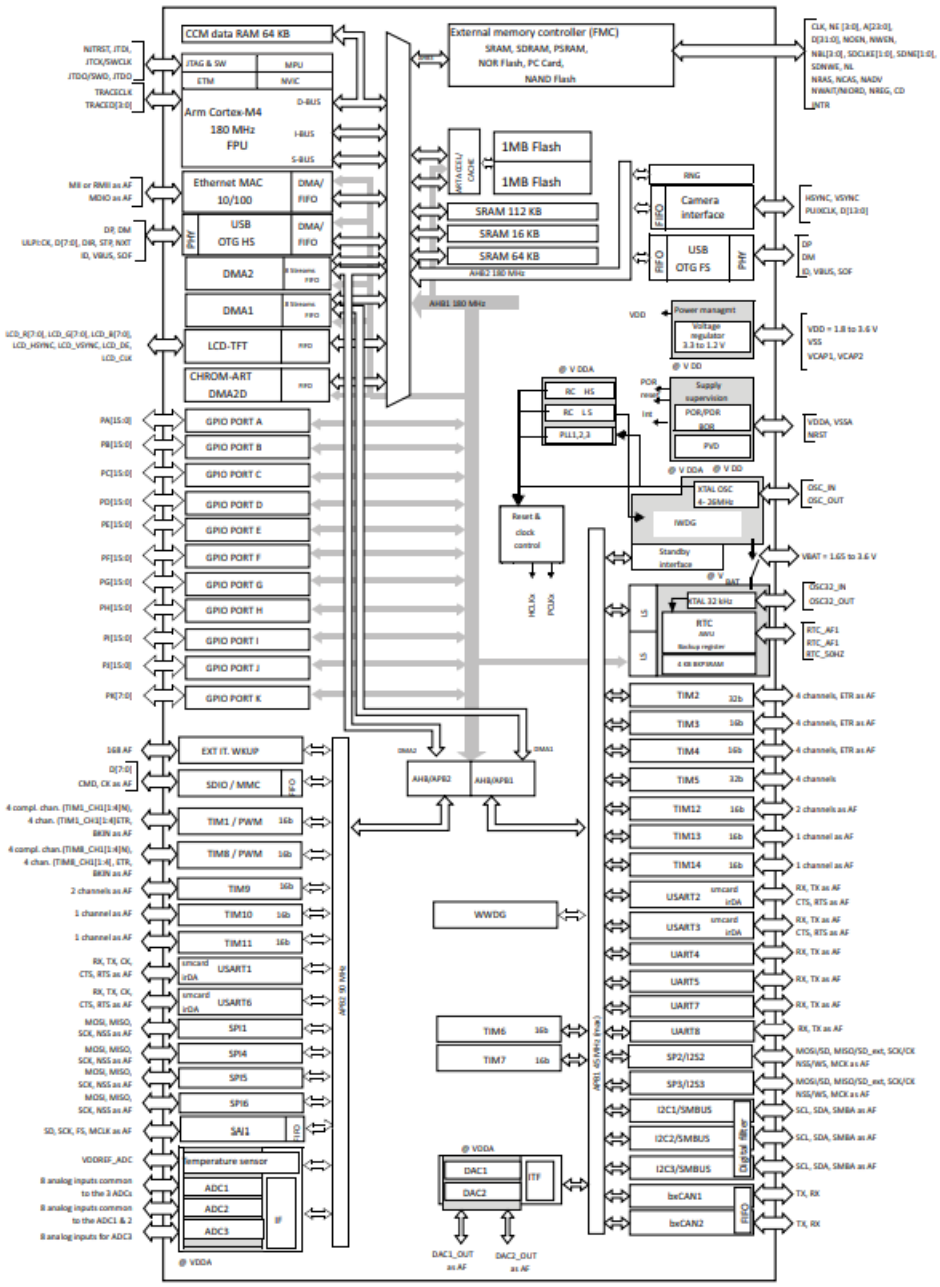

Figura 1: Diagrama de bloques del STM32F429 [1]

2.3. Características Eléctricas

A continuación se muestran las características eléctricas del STM32F429:

Symbol	Ratings	Min	Max	Unit
V_{DD}–V_{SS}	External main supply voltage (including V_{DDA}, V_{DD} and VBAT)(1)	– 0.3	4.0	V
V_{IN}	Input voltage on FT pins(2)	V_{SS} – 0.3	V_{DD}+4.0	
	Input voltage on TTa pins	V_{SS} – 0.3	4.0	
	Input voltage on any other pin	V_{SS} – 0.3	4.0	
	Input voltage on BOOT0 pin	V_{SS}	9.0	
$\lvert\Delta V_{DDx}\rvert$	Variations between different V_{DD} power pins	-	50	mV
$\lvert V_{SSX} - V_{SS}\rvert$	Variations between all the different ground pins including V_{REF-}	-	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see *Section 6.3.15: Absolute maximum ratings (electrical sensitivity)*		

Figura 2: Características de voltaje del STM32F429 [1]

Symbol	Ratings	Max.	Unit
ΣI_{VDD}	Total current into sum of all V_{DD_x} power lines (source)(1)	270	mA
ΣI_{VSS}	Total current out of sum of all V_{SS_x} ground lines (sink)(1)	– 270	
I_{VDD}	Maximum current into each V_{DD_x} power line (source)(1)	100	
I_{VSS}	Maximum current out of each V_{SS_x} ground line (sink)(1)	– 100	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current sourced by any I/Os and control pin	– 25	
ΣI_{IO}	Total output current sunk by sum of all I/O and control pins (2)	120	
	Total output current sourced by sum of all I/Os and control pins(2)	– 120	
$I_{INJ(PIN)}$ (3)	Injected current on FT pins (4)	– 5/+0	
	Injected current on NRST and BOOT0 pins (4)		
	Injected current on TTa pins(5)	±5	
$\Sigma I_{INJ(PIN)}$ (5)	Total injected current (sum of all I/O and control pins)(6)	±25	

Figura 3: Características de corriente del STM32F429 [1]

2.4. Diagrama de Pines

A continuación se muestra el diagrama de pines del STM32F429:

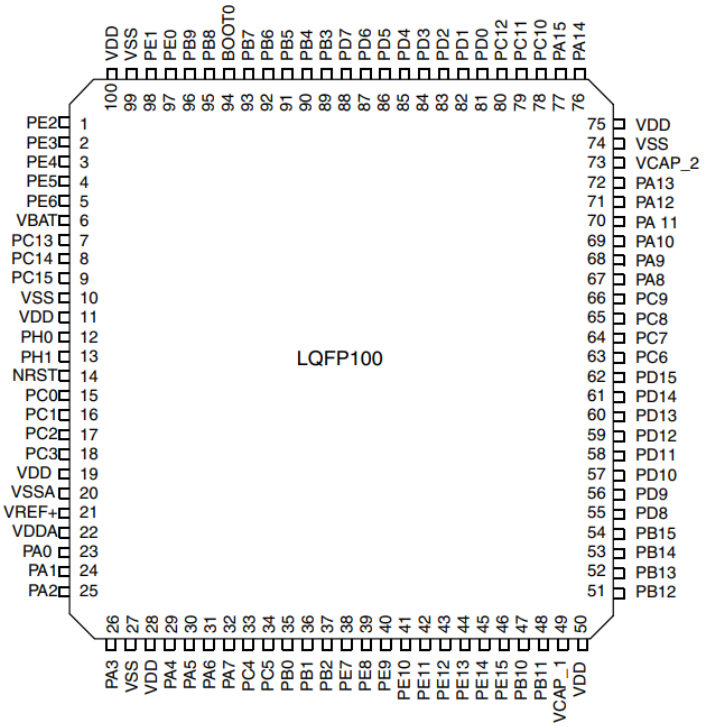

Figura 4: Diagrama de pines del STM32F429 [1]

2.5. Periféricos utilizados

Como se menciono anteriormente ara este laboratorio se utilizarán los diferentes pines GPIO de entrada y salida que posee el microcontrolador, los pines ADC y la conexión serial.

2.5.1. GPIO

Los pines de GPIO pueden configurarse tanto como entradas como salidas, según las necesidades del proyecto. En este laboratorio, se configuraron dos pines para encender los LEDs, mientras que el otro pin GPIO se utilizó para el botón que transmite información.

2.5.2. ADC

El microcontrolador STM32F429 cuenta con pines ADC (Convertidor Analógico-Digital) que son esenciales para medir señales analógicas y convertirlas en valores digitales. El ADC tiene una resolución de 12 bits, lo que permite representar un rango de valores de 0 a 4095. Además, su voltaje de referencia es de 3.3 V, lo cual debe tenerse en cuenta para adaptar el voltaje de entrada si es necesario.

2.5.3. Comunicación Serial

La comunicación serial se establece a través de USART la cual es fundamental para enviar y recibir datos entre el microcontrolador y otros dispositivos. En este laboratorio, se utilizará esta comunica-

ción serial para transmitir los datos del giroscopio y de la batería capturados por el microcontrolador a un script de Python para luego enviarlos a la plataforma Thingsboard.

2.6. Componentes Electrónicos Complementarios

2.6.1. Sensor L3GD20

El L3GD20 es un sensor giroscópico de tres ejes que ofrece alta precisión y bajo consumo de energía. Este sensor cuenta con una resolución de hasta 16 bits y puede medir velocidades de rotación en un rango de ±250, ±500 y ±2000 grados por segundo. Para la comunicación con el microcontrolador, se utiliza el protocolo SPI, que permite una transferencia de datos rápida y eficiente. En este laboratorio, se configuró el L3GD20 para operar en modo SPI, donde el microcontrolador envía comandos y recibe datos de manera sincronizada a través de pines específicos. Algunos de los registros importantes de este sensor son:

- WHOAMI: Es el indentificador del sensor.
- CTRLREG1: Se encarga del control de ejes.
- CTRLREG2: Se encarga de la configuración del filtro paso alto.
- CTRLREG3: Se encarga de la configuración de dps y modo SPI.

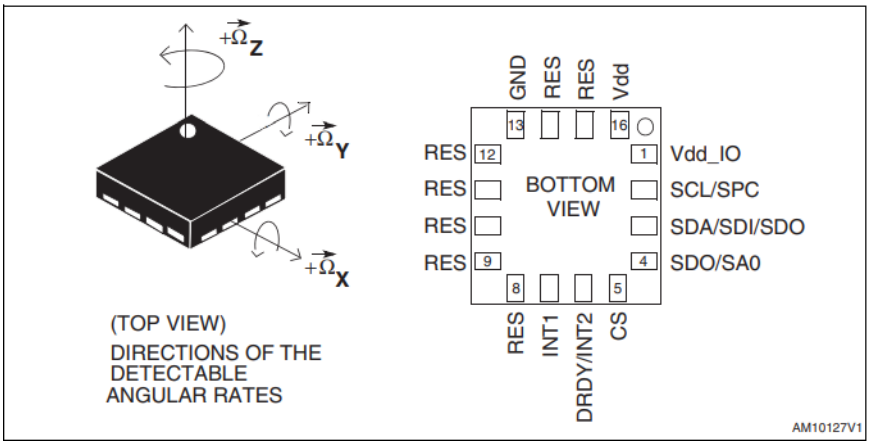

Figura 5: Diagrama de pines del L3GD20 [2]

2.6.2. Pantalla LCD

La pantalla LCD/TFT ILI9341 es un display a color de 2.8 pulgadas que utiliza la tecnología TFT (Transistor de Película Fina) para ofrecer imágenes nítidas y brillantes con una resolución de 240x320 píxeles. La ILI9341 admite una interfaz de comunicación SPI, lo que facilita su conexión con el microcontrolador STM32f429. Esta pantalla va a resultar sumamente útil para mostrar la información del giroscopio y del nivel de batería en tiempo real.

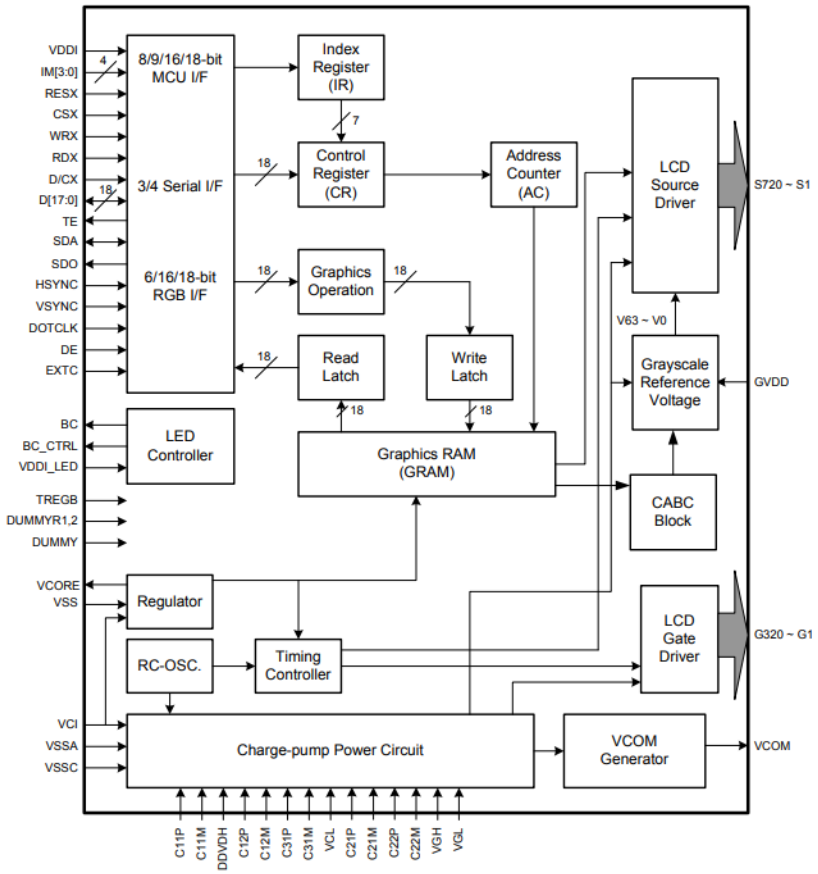

Figura 6: Diagrama de bloques del ILI9341 [3]

2.6.3. Plataforma Thingsboard

Para el apartado de IoT, se utilizará la plataforma Thingsboard, proporcionada por la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica. Esta plataforma ofrece una interfaz sencilla y amigable que facilita la configuración de los dispositivos y la creación de dashboards personalizados. Además, cuenta con una variedad considerable de widgets para desplegar la información de manera efectiva. Para comunicarse con esta plataforma web, se utilizó un script en Python que procesa los datos y emplea el protocolo MQTT (Message Queuing Telemetry Transport) para la transmisión eficiente de mensajes. Esto permite una comunicación ligera y en tiempo real entre los dispositivos IoT y la plataforma Thingsboard, optimizando la gestión y visualización de la información recopilada.

2.6.4. Presupuesto

Otro apartado importante contar con un presupuesto para realizar el laboratorio, por lo que en la siguiente tabla se las los precios aproximados de los componentes a utilizar:

Componentes	Precio aproximado
STM32F429	$50 US
4 x resistores	$2 US
Batería 9 V	$5 US
Conector batería	$1 US

3. Desarrollo

3.1. Implementación del giroscopio

Para implementar el sismógrafo, se utilizó el giroscopio L3GD20 integrado en el microcontrolador. La comunicación con este sensor se realizó a través del protocolo SPI, para lo cual fue necesario configurar el STM32F429 como maestro y el sensor como esclavo. Una vez completada la configuración, se leyeron los registros que contenían los datos del sensor, los cuales se desplegaron en la pantalla LCD del microcontrolador. A continuación, se presentan los resultados:

Figura 7: Visualización de los ejes en la pantalla LCD

Figura 8: Prueba del giroscopio

3.2. Implemetación del botón para la comunicación USART

Una vez verificado el correcto funcionamiento del giroscopio, se procedió a implementar un botón para transmitir los datos mediante USART. Para ello, se utilizaron ejemplos de la biblioteca libopencm3, que ya incluía la funcionalidad del botón. A continuación, se presenta una imagen que muestra el momento en que se presiona el botón para encender un LED:

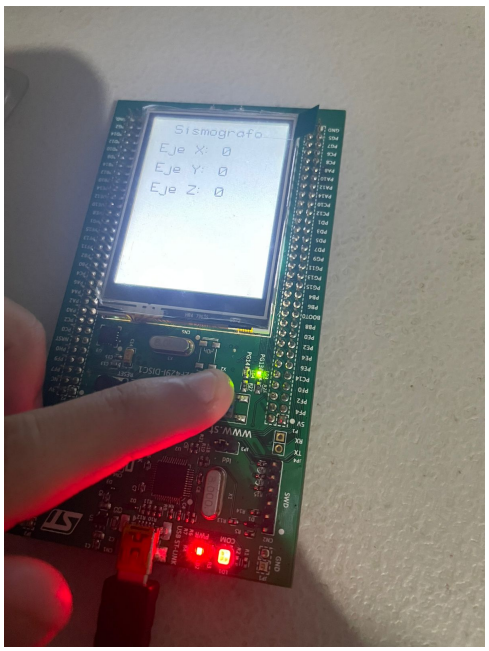

Figura 9: Prueba del botón

Al confirmar que el botón estaba correctamente configurado, se continuó con la implementación del envío de datos a través de USART. Se utilizó un ejemplo de la biblioteca para la configuración y se creó un pequeño script en Python para visualizar los datos de los ejes en tiempo real. Además, se configuró el LED verde para que parpadee cuando se transmitan datos, y que la pantalla LCD muestre un mensaje. En las siguientes imágenes se puede ver el comportamiento:

Figura 10: Comportamiento al precionar el botón para enviar datos

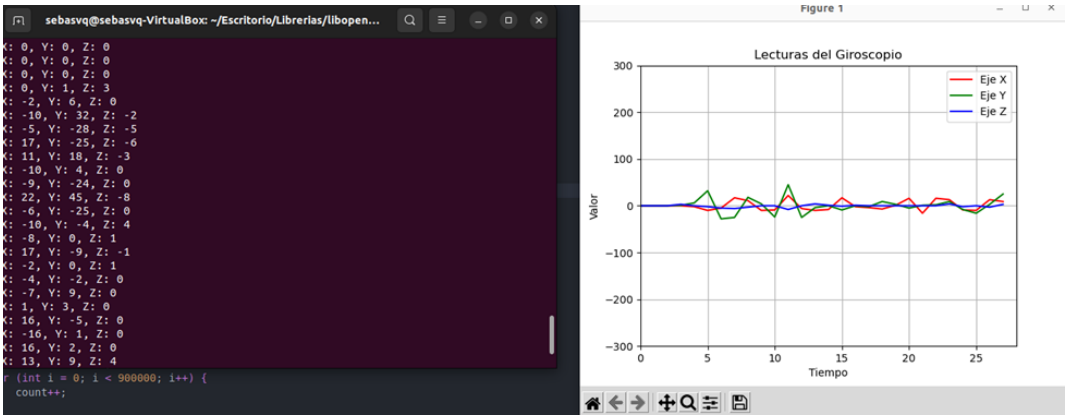

Figura 11: Datos recibidos y procesados

3.3. Implementación del ADC para la batería

Los puertos ADC del STM32F429 tienen un voltaje de referencia de 3.3 V, por lo que es necesario construir un divisor de tensión para reducir un voltaje de entrada de 9 V a un nivel lo más cercano posible a 3.3 V. Debido a las limitaciones en las resistencias disponibles, se encontró una combinación que se acercaba bastante a los 3.3 V, y se decidió continuar con esa configuración. A continuación, se presenta una simulación realizada en TINA y la construcción en una protoboard:

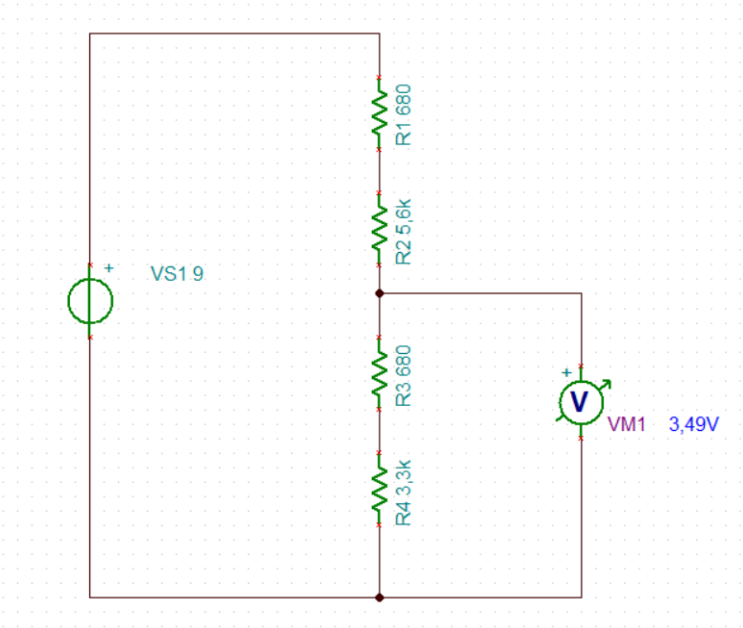

Figura 12: Simulación del divisor de tensiones

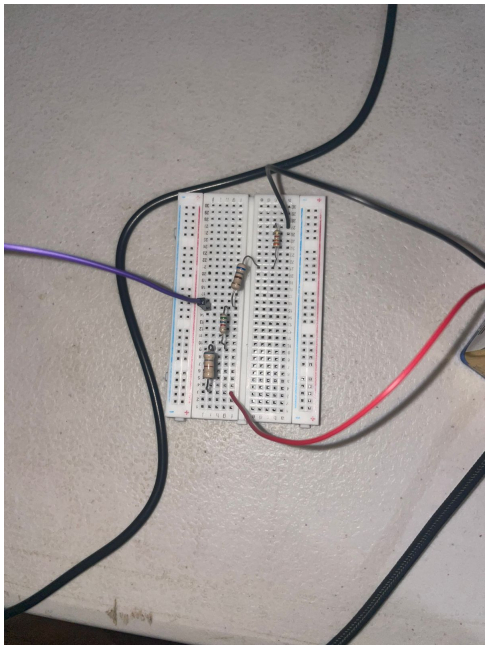

Figura 13: Divisor de tensiones construido

Se configuró el pin PA3 utilizando un ejemplo de ADC de la biblioteca libopencm3. Además, se logró mostrar el valor de la batería en la pantalla LCD, y cuando este se encuentra por debajo de 7 V, el LED rojo comienza a parpadear. Todo este comportamiento se puede observar en las siguientes imágenes:

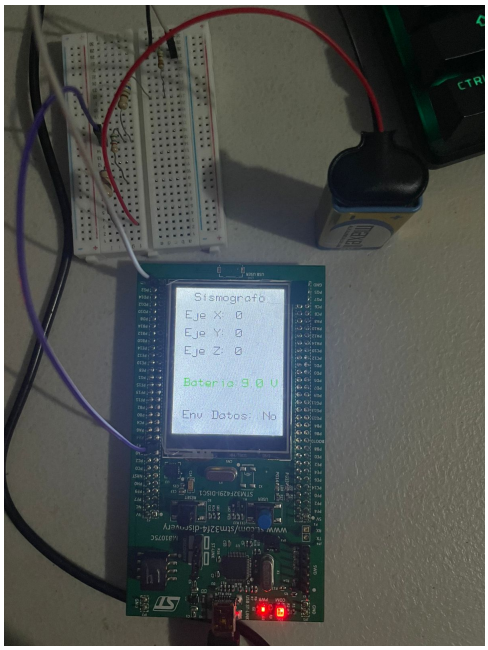

Figura 14: Lectura de batería de un valor optimo

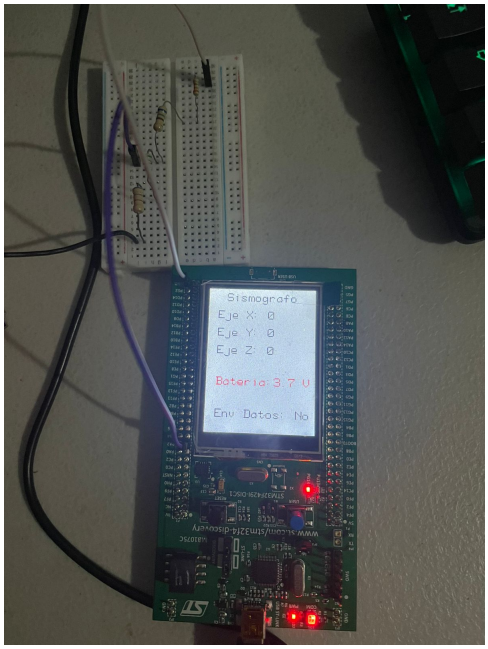

Figura 15: Lectura de batería de un valor bajo

Por último, se creó un script en Python para establecer la conexión con la plataforma web de Thingsboard. Para ello, se utilizó la comunicación MQTT y se emplearon diversos widgets para mostrar la información solicitada.

3.4. Resultados y Análisis

A continuación se muestran los resultados obtenidos en el dashboard de Thingsboard:

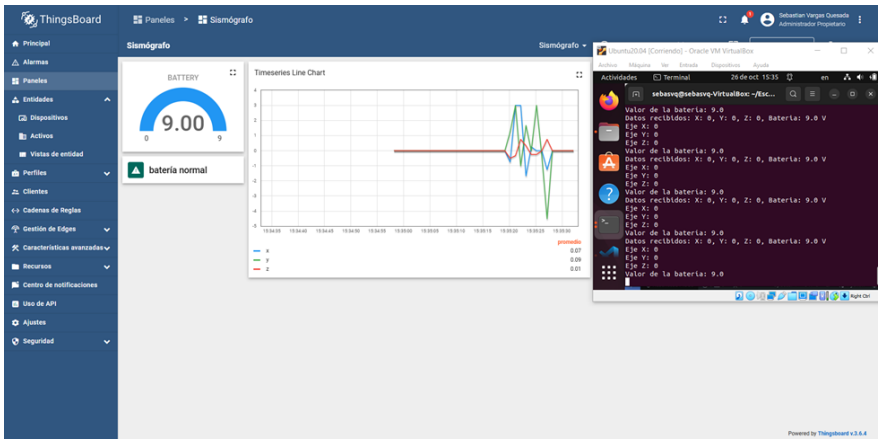

Figura 16: Datos procesados en el dashboard (Batería normal)

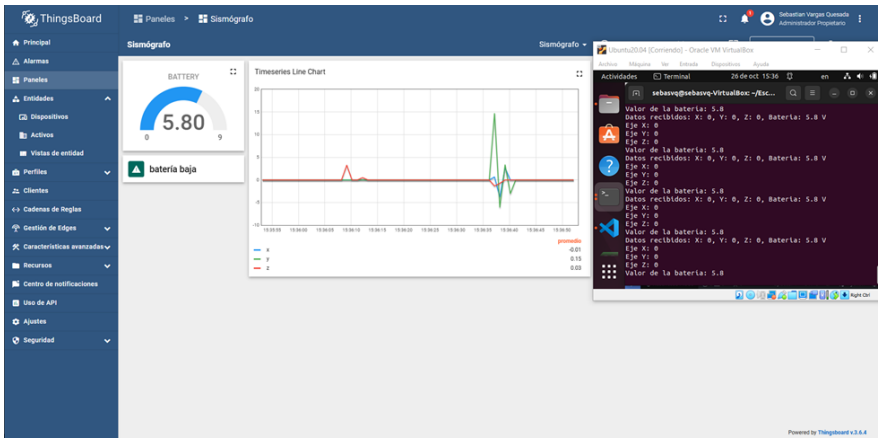

Figura 17: Datos procesados en el dashboard (Batería baja)

Tras analizar los resultados obtenidos, se confirma que el circuito funciona según lo esperado. En la figura 16, se observa que el nivel de batería es adecuado, por lo que se utilizó un widget de notificación para informar al usuario. Además, se probó el funcionamiento del sismógrafo realizando movimientos leves en el microcontrolador, los cuales se pudieron visualizar en la gráfica. Por otro lado, en la figura 17, se muestra que el voltaje de la batería está por debajo del valor aceptable, lo que activa una notificación al usuario indicando que la batería está baja. De manera similar, se realizaron movimientos en el microcontrolador para observar su comportamiento en la gráfica.

4. Conclusiones y Recomendaciones

- Se logró diseñar un circuito capaz de leer un giroscopio utilizando el protocolo SPI y medir el nivel de voltaje de una batería a través de un pin ADC.
- Se incorporaron varios LEDs de aviso para indicar cuándo se están transmitiendo los datos y cuándo el nivel de la batería es bajo.
- Se utilizó un puerto serial para transmitir los datos del giroscopio y de la batería a un script en Python, lo que permitió la implementación del IoT al comunicar directamente con la plataforma Thingsboard.
- Se recomienda utilizar los ejemplos de la biblioteca libopencm3 para lograr una implementación más rápida.
- Se recomienda experimentar con diferentes combinaciones de resistencias al construir el divisor de tensiones, con el fin de lograr un voltaje de salida lo más cercano posible a 3.3 V.

Referencias

[1] ST. (2018). *STM32F427xx STM32F429xx*.Recuperado el 29 de octubre del 2024, de https://www.st.com/resource/en/datasheet/stm32f427vg.pdf.

[2] ST. (2013). *L3GD20*.Recuperado el 29 de octubre del 2024, de https://www.st.com/en/mems-and-sensors/l3gd20.html.

[3] ILITEK. *ILI9341*.Recuperado el 29 de octubre del 2024, de http://www.ilitek.com.

5. Apéndices

Se presentán los datasheets de los componetes utilizados