

IE-0624

Laboratorio de Microcontroladores

Laboratorio #4: STM32: GPIO, ADC, comunicaciones, Iot

Fernando Jiménez Ureña B74020
Kristel Herrera Rodríguez C13769

II ciclo
Octubre 2023

1. Resumen

En el presente reporte y laboratorio, se llevó a cabo el diseño y la implementación de un sismógrafo digital utilizando el microcontrolador STM32F429 y la biblioteca libopencm3, con el fin de registrar y estudiar las oscilaciones en el edificio de la Escuela de Ingeniería Eléctrica. El diseño integra la lectura de los ejes (X, Y, Z) del giroscopio y presenta la funcionalidad de habilitar o deshabilitar las comunicaciones por USART/USB.

Además el sismógrafo tiene la capacidad de monitorear el nivel de la batería, cuyo rango se extiende de 0 a 9 V. En situaciones donde el nivel de batería se aproxima al límite mínimo de operación del microcontrolador (7V), el sistema activa un LED de alarma y envía una notificación de batería baja al dashboard de ThingsBoard.

Los resultados del monitoreo, incluyendo los valores del giroscopio y el nivel de batería, así como el estado de las comunicaciones, son desplegados en una pantalla LCD. Además, se desarrolló un script en Python capaz de leer y escribir al puerto serial/USB, facilitando la transmisión de estos datos a un dashboard en ThingsBoard. Este último se configuró para visualizar la información relevante, proporcionando una interfaz intuitiva y accesible para el análisis de los datos recopilados.

Este laboratorio permitió además la oportunidad de estudiar sobre el internet de las cosas mediante la plataforma de ThingsBoard, logrando así aprender sobre las aplicaciones de monitoreo y análisis de datos en tiempo real.

Dirección del repositorio: <https://tinyurl.com/bdfb978w>

2. Nota Teórica

2.1. Microcontrolador STM32F429

2.1.1. Características Generales

El microcontrolador STM32F429ZIT6 es una unidad potente y versátil basada en el núcleo ARM 32-bits Cortex-M4 con FPU (unidad de punto flotante). Esta unidad RISC opera a una frecuencia de hasta 180 MHz.

2.1.2. Memoria y Almacenamiento

El STM32F429ZIT6 posee una memoria flash de 2 MB y una SRAM de 256 KB. Además, cuenta con un SDRAM adicional de 64-Mbit.

2.1.3. Comunicación y Conectividad

- USB OTG con conector Micro-AB.
- Seis LEDs para diferentes propósitos, incluidos USB Comms y Power On.
- Dos push-buttons (Usuario y reset).
- Header para LQFP144 I/Os.
- Funciones USB: Debug, virtual COM y almacenamiento.
- 21 interfaces de comunicaciones que incluyen I2C, USART, SPI, SAI y CAN.
- Conectividad avanzada USB 2.0.
- Sensor de movimiento I3G4250D y giroscopio STM EMS de 3-ejes.

2.1.4. Alimentación y Energía

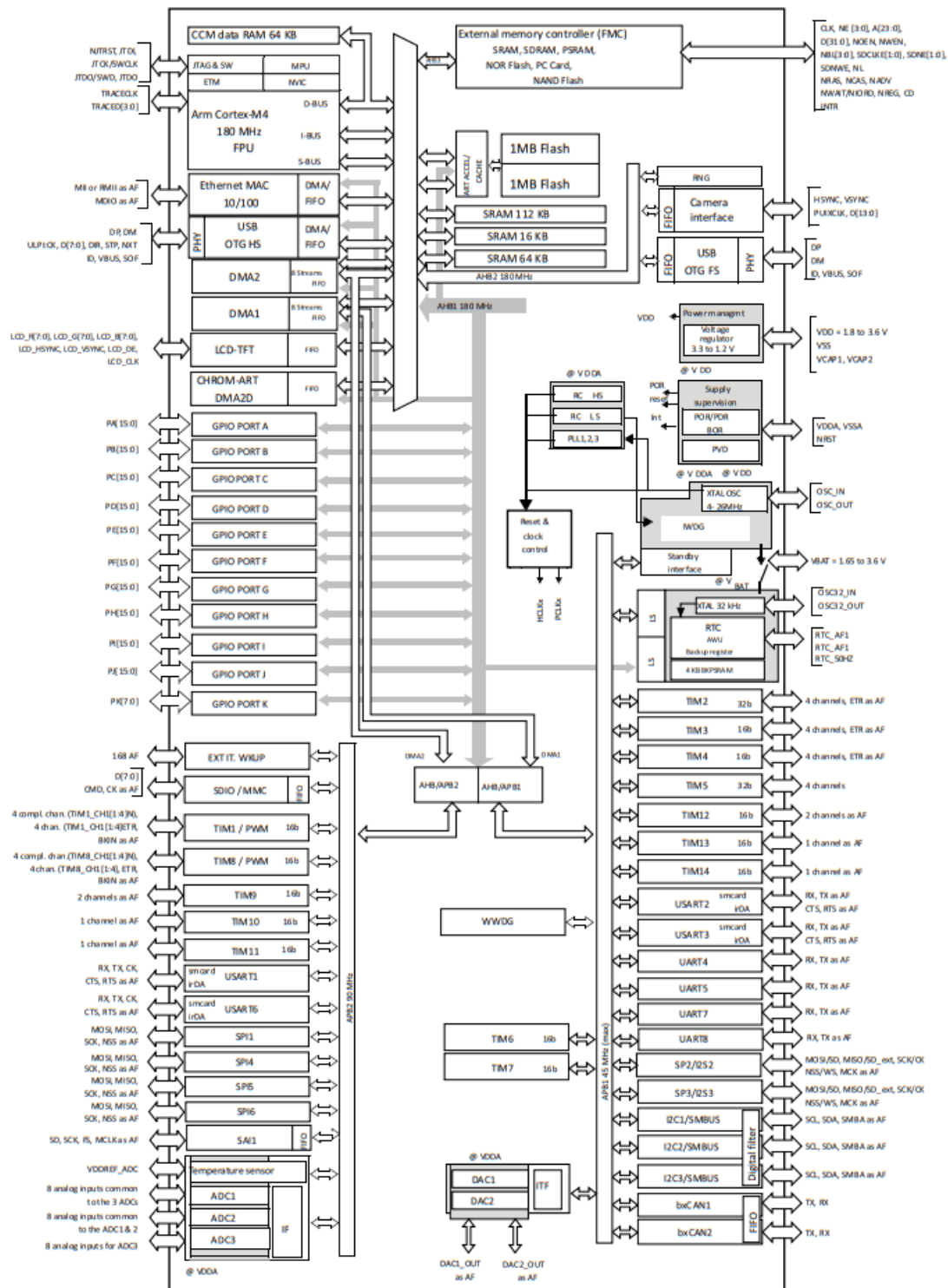
El microcontrolador puede ser alimentado a través de USB o una fuente externa de 3V o 5V. Posee características de bajo consumo para aplicaciones de bajo poder.

2.1.5. Otras Características

- Controlador LCD-TFT.
- 3 ADCs de 12-bit.
- 2 convertidores D/A de 12-bit.

- 17 timers: 12 de 16-bit y 2 de 32-bit que operan a hasta 180 MHz.
- Capacidades de debug: SWD y JTAG.
- 168 I/Os con capacidad de interrupción.
- Interfaz de cámara.
- True RNG (generador de números aleatorios).
- CRC (verificación de redundancia cíclica).
- Controladores DMA.

2.1.6. Diagrama de bloques



MSv30420V5.svg

2.1.7. Características eléctricas

Table 14. Voltage characteristics

Symbol	Ratings	Min	Max	Unit
$V_{DD}-V_{SS}$	External main supply voltage (including V_{DDA} , V_{DD} and V_{BAT}) ⁽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	
$ \Delta V_{DDx} $	Variations between different V_{DD} power pins	-	50	mV
$ V_{SSx}-V_{SS} $	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. [1]

Table 15. Current characteristics

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. [1]

2.1.8. Diagrama de pines

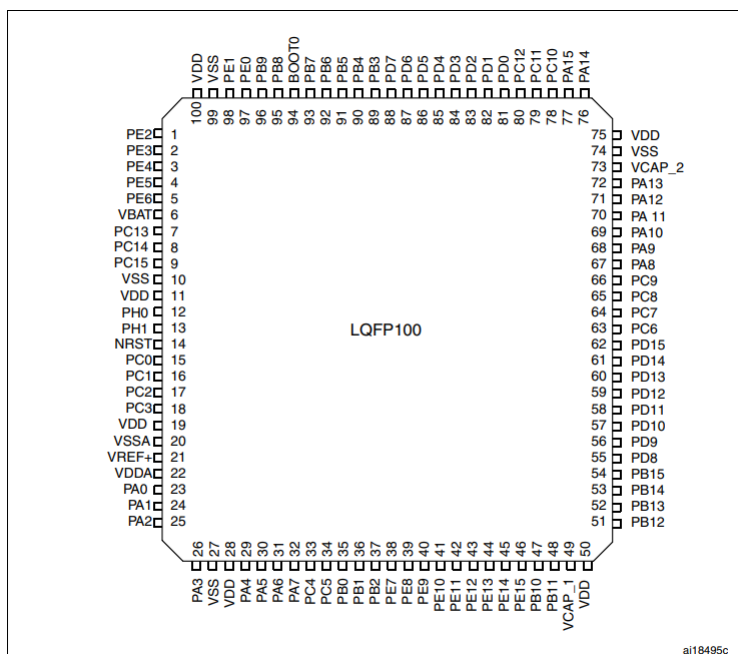


Figura 4: Diagrama de pines. [1]

2.1.9. Sensor L3GD20

El sensor **MEMS L3GD20** es un giroscopio de tres ejes desarrollado por STMicroelectronics. Este sensor es particularmente destacado por su bajo consumo y capacidad para medir la tasa angular con respecto a un entorno externo. Es decir, puede detectar movimientos de rotación a lo largo de sus tres ejes ortogonales. Para la comunicación con microcontroladores u otros dispositivos, el L3GD20 ofrece interfaces tanto I2C como SPI.

En el contexto del **STM32F429 Discovery Kit**, este sensor se comunica específicamente a través del protocolo SPI, conectado en la interfaz SPI5 del microcontrolador. La configuración y comunicación con el L3GD20 se realiza mediante instrucciones SPI. Existen varios registros esenciales para su operación:

- **WHOAMI**: Un identificador único que se puede usar para validar la configuración al leer su valor.
- **CTRLREG1**: Se encarga del control de ejes y del modo de energía.
- **CTRLREG2**: Es responsable de la configuración del filtro de paso alto.
- **CTRLREG4**: Define la configuración de los dps y el modo SPI.

Para operar el L3GD20 con el STM32F429, es necesario realizar una serie de pasos de configuración, que incluyen habilitar el reloj para el SPI, configurar pines, inicializar y configurar el protocolo SPI y, finalmente, configurar el L3GD20 a través de la interfaz SPI.

2.1.10. Pantalla y Gráficos

El microcontrolador tiene un controlador LCD-TFT integrado y está equipado con una pantalla TFT LCD de 2.4" QVGA.

La **Pantalla LCD/TFT ILI9341** es una pantalla táctil a color con una resolución de 240x320 píxeles. Esta pantalla utiliza el controlador gráfico ILI9341 y un controlador táctil XPT2046 para la detección de toques. A nivel de comunicación, esta pantalla ofrece interfaces I2C y SPI, y en el contexto del **STM32F429 Discovery Kit**, se comunica específicamente a través de la interfaz SPI5. Como con otros dispositivos periféricos, la configuración y comunicación con el ILI9341 se realiza mediante instrucciones SPI.

El controlador STM32F429 incluye características específicas para el manejo de pantallas, como[1]:

- Interfaz paralela LCD en modos 8080/6800.
- Controlador LCD-TFT con resolución completamente programable, capaz de manejar hasta un ancho de 4096 píxeles, una altura de 2048 líneas y un reloj de píxeles de hasta 83 MHz.
- Capacidades para interfaz con la mayoría de los controladores gráficos LCD, soportando los modos Intel 8080 y Motorola 6800.
- Controlador de pantalla LCD-TFT que proporciona una salida RGB paralela digital de 24 bits, compatible con una amplia gama de paneles LCD y TFT hasta resolución XGA (1024x768) con características como:
 - Dos capas de visualización con FIFO dedicado (64x32-bit).
 - Tabla de búsqueda de colores (CLUT) de hasta 256 colores (256x24-bit) por capa.
 - Hasta 8 formatos de color de entrada seleccionables por capa.
 - Fusión flexible entre dos capas usando un valor alfa (por píxel o constante).
 - Parámetros programables flexibles para cada capa.
 - Clave de color (color de transparencia).
 - Hasta 4 eventos de interrupción programables.

2.2. Precios de los componentes

A continuación se muestran la lista de precios de los componentes utilizados

Componente	Precio (Colones)
STM32F429	24890.09
Resistencias 680Ω	75
Resistencia $3,3K\Omega$	35
Resistencia $5,1K\Omega$	55
Batería $9V$	2395

Tabla 1: Precio de los componentes

3. Desarrollo

3.1. Diseño

En la figura a continuación se muestra el diseño del circuito implementado.

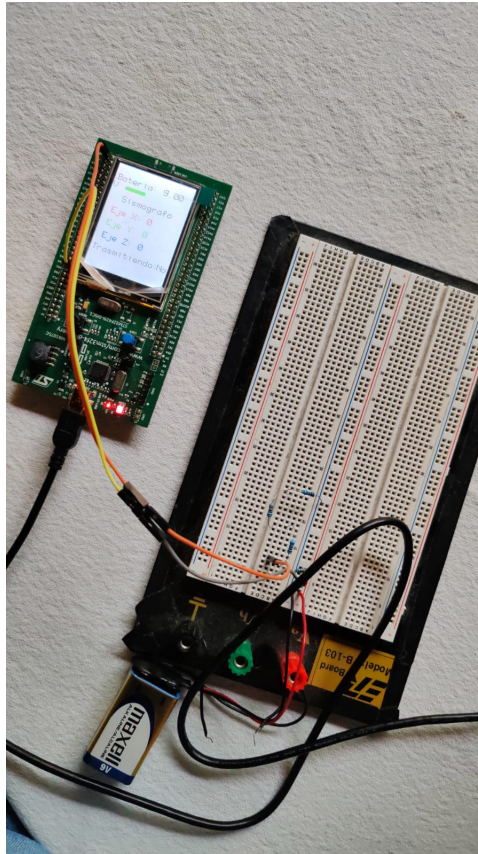


Figura 5: Diseño del circuito

3.2. Funcionalidad de Hardware

Para el diseño del circuito implementado se diseñaron los siguientes valores de las resistencias.

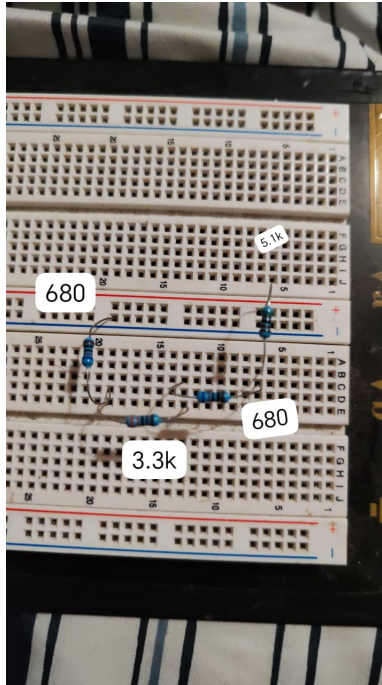


Figura 6: Diseño del circuito

En la figura previamente presentada, se observa la implementación de varios componentes. Específicamente, se han utilizado dos resistencias de $680\ \Omega$, una resistencia de $3,3K\Omega$ y otra resistencia de $5,1K\Omega$.

Estos valores de las resistencias y su configuración consisten en una división de tensión con el fin de reducir la tensión de entrada de $9V$ proveniente de la batería externa a un nivel seguro para el microcontrolador que en este caso puede soportar que es hasta $5\ V$.

Por lo tanto aplicando la ecuación para una división de tensión se obtiene que:

$$V_{out} = V_{in} * \left(\frac{R2}{R1 + R2} \right) \quad (1)$$

$$V_{out} = 9V * \left(\frac{5,1k\Omega}{3,3k\Omega + 2 \cdot 680\Omega + 5,1k\Omega} \right) \quad (2)$$

$$V_{out} = 4,7V \quad (3)$$

Es importante notar que se hizo este arreglo de resistencias, porque luego de varias pruebas de prueba y error, con estos valores se obtuvo un valor de tensión funcional para el microcontrolador. A continuación se aprecia el valor obtenido midiéndolo con el multímetro.

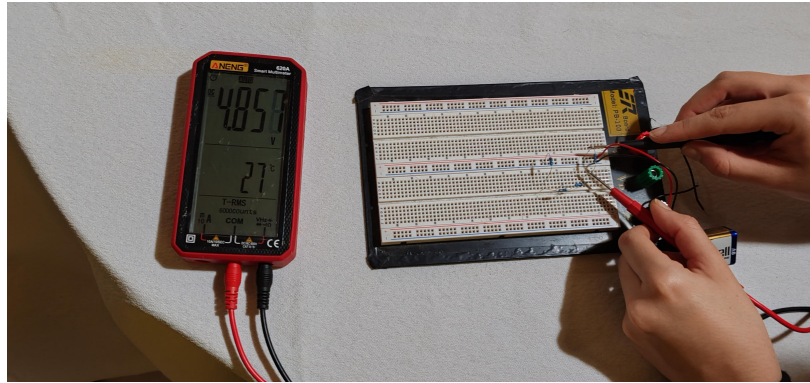


Figura 7: Medición del valor de tensión luego de la división de tensión

Se puede apreciar que se obtuvo un valor de **4.857 V**, el cuál se encuentra dentro del rango funcional del microcontrolador y muy cercano al calculado teóricamente con la fórmula de división de tensión.

Funcionamiento de la Pantalla

Nivel de batería Se representa tanto en formato de texto como gráficamente mediante una barra que cambia de color según la carga.

Datos del sismógrafo Los valores de los ejes se muestran con colores distintos: rojo, verde y azul para los ejes X, Y y Z respectivamente.

Estado de transmisión Indica si los datos se están transmitiendo o no.

El funcionamiento de la pantalla se puede observar en la siguiente imagen:

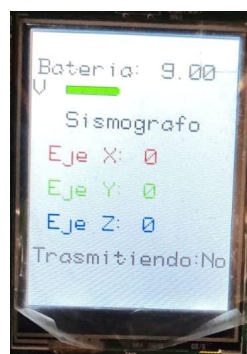


Figura 8: Datos impresos en pantalla.

Interacción con LEDs

Adicionalmente, el sistema utiliza LEDs para ofrecer retroalimentación sobre ciertos estados, complementando así la información mostrada en la pantalla.

3.3. Funcionalidad de Software

Para la funcionalidad del Software se implementó un código en el lenguaje de programación C donde se cubren múltiples funcionalidades relacionadas con la interacción de los periféricos y sensores como un giroscopio basándose en lógica que fue implementada en la librería de ejemplos del microcontrolador STM32: *libopencm3-examples*.

Es importante señalar nuevamente que gran parte de la configuración realizada fue para configurar un giroscopio que es la base del funcionamiento de un sismógrafo, esto debido a que gracias al giroscopio se puede interactuar con la posición del dispositivo en las coordenadas X,Y y Z. Entre las definiciones relacionadas con el giroscopio se encuentran la identificación del dispositivo, registros de control, registros de datos, entre otros. Esto es con el fin de configurar y obtener datos del giroscopio. Se hizo uso de una *struct* en C llamada *gyro* que tiene tres enteros de 16 bits para almacenar los valores de los ejes X, Y y Z del giroscopio.

Es importante notar además que se definen funciones para realizar transacciones SPI con el giroscopio y configurar la interfaz SPI. El SPI es un protocolo de comunicación para que pueda existir la comunicación entre periféricos. Además se implementó otro protocolo de comunicación llamado USART (Universal Synchronous Asynchronous Receiver Transmitter) para la transmisión de datos que genera el giroscopio. En este caso, se utiliza para transmitir datos a través del pin GPIO9 del puerto A con una velocidad de transmisión de 115200 bps.

La pantalla requiere un proceso de inicialización y configuración. Este proceso incluye la activación de ciertos periféricos, configuración de pines y comunicación inicial con la pantalla para establecer parámetros operativos.

El sistema utiliza una biblioteca gráfica que ofrece funciones para diferentes operaciones gráficas como:

- Limpiar la pantalla.
- Configurar el tamaño y color del texto.
- Dibujar y rellenar rectángulos.

Para la configuración de los pines, se conectó de la siguiente manera:

GPIOC, GPIO1:

Utilizado para controlar el Chip Select (CS) del giroscopio durante las transacciones SPI.

GPIOF, GPIO7 — GPIO8 — GPIO9:

Estos pines se configuran para el modo de función alternativa (AF), específicamente para la interfaz SPI con el giroscopio. La función exacta de cada uno no se detalla, pero generalmente en SPI estos pines corresponden a MISO (Master In, Slave Out), MOSI (Master Out, Slave In) y SCLK (Serial Clock).

GPIOA, GPIO9:

Configurado para transmitir datos desde USART1. Por este medio es donde se transmiten los datos y son interpretados por un script de *Python*.

GPIOA, GPIO0:

Se configuran como entradas.

GPIOG, GPIO13 y GPIO14:

Se configuran como salidas.

GPIOA, GPIO3:

Utilizado para leer el nivel de batería mediante el ADC. Es una entrada analógica que recoge los valores de voltaje de la batería.

A continuación se muestra un diagrama de bloques del funcionamiento básico del sismógrafo

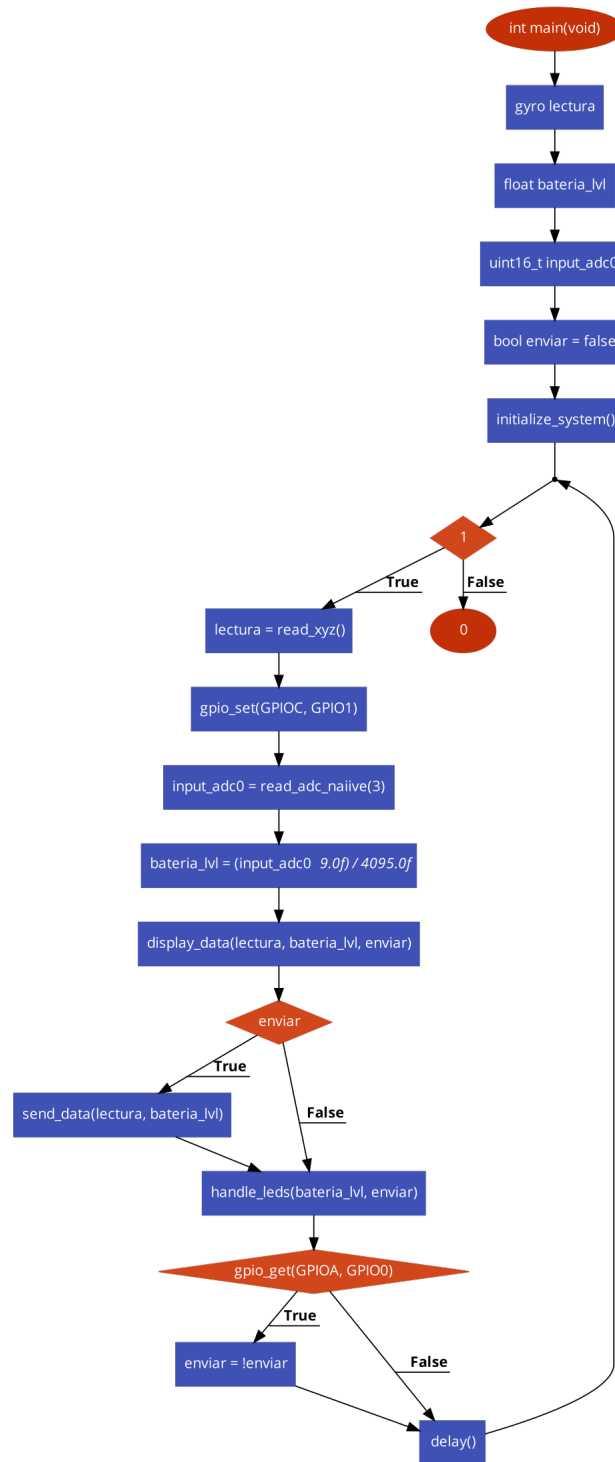


Figura 9: Diagrama de bloques del código implementado

Código en Python

Además es importante destacar que también se implementó otro código en el lenguaje de programación *Python*. Este otro código, tiene la finalidad de leer/escribir el puerto USB y que pueda enviar la información del giroscopio y nivel de batería para ser desplegados en un dashboard de la plataforma Iot thingsboard.

El código se estructura en una clase principal denominada *IOTClient*, la cual encapsula todas las funciones necesarias para establecer las conexiones, leer los datos del sensor y enviarlos al servidor. Hay 3 elementos importantes en el código para su funcionamiento:

- `time`: maneja las pausas entre envíos de datos
- `json`: convierte los datos a formato JSON
- `serial`: permite la comunicación a través del puerto serial
- `paho.mqtt.client`: logra la interacción con el protocolo MQTT.

La clase *IOTClient* mencionada anteriormente, maneja la configuración de las conexiones, la lectura de datos del sensor, y su envío al servidor. Al iniciar, establece las conexiones serial y MQTT, e inicia un bucle que lee los datos del sensor, los convierte a formato JSON, y los publica en el servidor MQTT con una pausa entre envíos. Si se ejecuta el script directamente, se crea una instancia de *IOTClient* y se inicia este proceso, permitiendo la transmisión continua de datos del sensor al servidor MQTT.

Iot Thingsboard

Luego del correcto funcionamiento del sismógrafo y la implementación del código en Python para leer/escribir el puerto USB, se configuró la plataforma de Iot Thingsboard para poder manejar y visualizar los datos enviados y recibidos. Para la configuración en la plataforma, se utilizó la cuenta que fue creada por el profesor con el correo institucional.

A continuación se puede apreciar la data visualizada del sismógrafo mediante dicha plataforma:

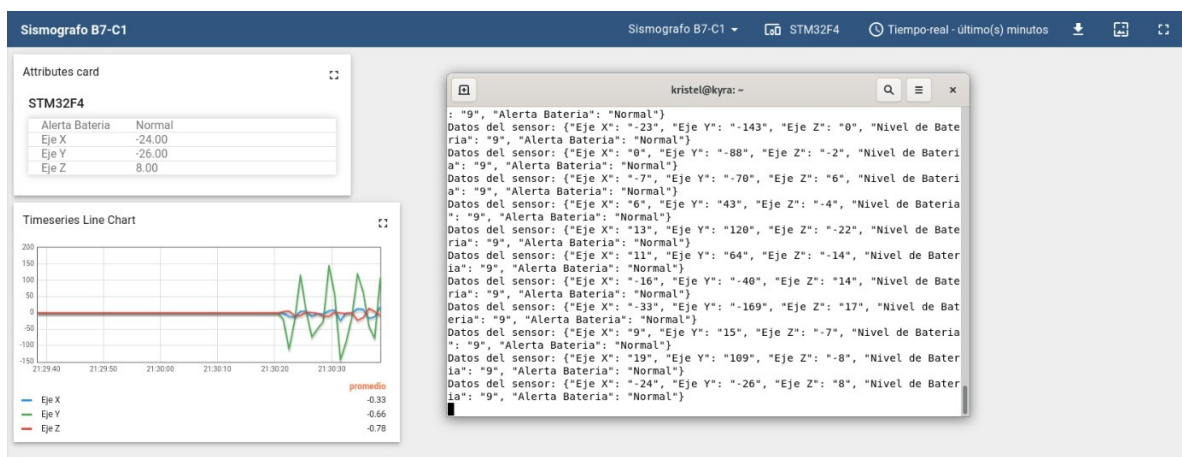


Figura 10: Sismógrafo en plataforma ThingsBoard

4. Conclusiones y Recomendaciones

4.1. Conclusiones

- El desarrollo y puesta en marcha del sismógrafo se completó con éxito mediante la integración del Microcontrolador STM32F429 Discovery kit y el apoyo de la biblioteca *libopencm3*.
- La conexión entre el sismógrafo y la plataforma de IoT Thingsboard se estableció exitosamente usando un script de *Python*.
- La implementación del sistema demostró la versatilidad y potencial de los microcontroladores STM32F429 en aplicaciones de monitoreo y recolección de datos en tiempo real.
- El uso de la plataforma de IoT Thingsboard permitió una visualización efectiva y en tiempo real de los datos recogidos, mostrando la importancia de las plataformas de IoT en la era digital.
- La combinación de hardware y software utilizado en este proyecto evidencia que es posible desarrollar sistemas de monitoreo eficientes y asequibles con tecnología disponible en el mercado.

4.2. Recomendaciones

- Es recomendable experimentar con diversas combinaciones de resistencias hasta alcanzar la tensión adecuada que garantice el óptimo desempeño del microcontrolador.
- Para asegurar la correcta lectura de todas las bibliotecas requeridas, se sugiere trabajar con la biblioteca *libopencm3*. Además, es aconsejable implementar y ejecutar el código del sismógrafo dentro de las carpetas de *libopencm3-examples*.
- Para facilitar la implementación y acelerar el proceso de desarrollo, es beneficioso tomar como referencia los ejemplos previamente establecidos en *libopencm3-examples*.

Referencias

- [1] STMicroelectronics, *STM32F427xx STM32F429xx Datasheet: 32b Arm® Cortex®-M4 MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 com. interfaces, camera & LCD-TFT*, STMicroelectronics, 1 2018, available at www.st.com. [Online]. Available: www.st.com

5. Apéndices