



UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624 LABORATORIO DE MICROPROCESADORES

Laboratorio 4

José Antonio Ramos Pereira, B86485

Grupo 01

Prof. MSc. Marco Villalta Fallas.

https://github.com/JAR1224/Laboratorio_4_Jose_Ramos

Índice de contenidos

1.	Introducción	1
1.1.	Resumen	1
1.2.	Conclusiones	1
2.	Nota Teórica	2
2.1.	Microcontrolador	2
2.2.	Componentes Electrónicos Complementarios	7
2.3.	Precios	11
2.4.	Diseño del circuito	12
2.5.	Temas de laboratorio	18
3.	Desarrollo y análisis	20
3.1.	Programa	20
3.2.	Componentes y funcionamiento	22
4.	Conclusiones	30
4.1.	Recomendaciones	30
5.	GIT	32

Índice de diagramas

2.1.	Características generales stm32f429	3
2.2.	Diagrama stm32f429	4
2.3.	Pines stm32f429	5
2.4.	Características eléctricas stm32f429	6
2.5.	Características eléctricas stm32f429	6
2.6.	Diagrama ILI9341	7
2.7.	Características eléctricas ILI9341	8
2.8.	Características generales L3GD20	8
2.9.	Diagrama L3GD20	9
2.10.	Pines L3GD20	9
2.11.	Características eléctricas L3GD20	10
2.12.	Esquemático stm-discovery	12
2.13.	Esquemático breadboard	13
2.14.	Conexión con stm32	14
2.15.	Circuito completo	15
2.16.	Batería	16
3.1.	Diagrama de bloques programa	21
3.2.	LED Usart	22
3.3.	LED alarma	23
3.4.	mqtt python	24
3.5.	Dashboard thingsboard	25

3.6. Batería nivel alto	25
3.7. Batería nivel bajo	26
3.8. Ejes MEMS	26
3.9. Parar comunicación USART	27
3.10. Caso 1	28
3.11. Caso 2	29
3.12. Caso 3	30

1. Introducción

1.1. Resumen

La elaboración del proyecto consistió en plantear una solución utilizando una tarjeta de desarrollo STM32F429I-Discovery y otros componentes electrónicos para simular un sismógrafo. La aplicación debía realizar las siguientes funciones:

- Leer los ejes del giroscopio (X,Y,Z)
- Incluir un switch/boton para habilitar/deshabilitar comunicaciones por USART/USB
- Un led debe parpadear indicando la habilitación de la transmisión/recepción de datos por el periférico USART/USB
- Debe leer el nivel de la batería cuyo rango es de [0,9]V, en caso de estar cerca del límite mínimo de operación del microcontrolador (7 V) debería encender un led de alarma parpadeante y enviar la notificación de batería baja al dashboard de thingsboard.
- Debe desplegar en la pantalla LCD el nivel de batería, los valores de los ejes X,Y,Z y si la comunicación serial/USB está habilitada.
- Crear un script de python que lea/escriba al puerto serial/USB y que envíe la información del giroscopio y nivel de batería para ser desplegados en un dashboard de una plataforma IoT thingsboard.

1.2. Conclusiones

Se concluyó que por medio de las metodologías mencionadas anteriormente, se pudo implementar una solución al problema planteado que cumple con todos los requisitos de diseño requeridos por el enunciado.

2. Nota Teórica

2.1. Microcontrolador

Características generales

- 2 Mbytes memoria flash
- 256 Kbytes memoria RAM
- 6 LEDS
- 5 SPI
- UART y I2C
- MEMS

- ◆ Display resolution: [240xRGB](H) x 320(V)
- ◆ Output:
 - 720 source outputs
 - 320 gate outputs
 - Common electrode output (VCOM)
- ◆ a-TFT LCD driver with on-chip full display RAM: 172,800 bytes
- ◆ System Interface
 - 8-bits, 9-bits, 16-bits, 18-bits interface with 8080- I /8080- II series MCU
 - 6-bits, 16-bits, 18-bits RGB interface with graphic controller
 - 3-line / 4-line serial interface
- ◆ Display mode:
 - Full color mode (Idle mode OFF): 262K-color (selectable color depth mode by software)
 - Reduce color mode (Idle mode ON): 8-color
- ◆ Power saving mode:
 - Sleep mode
- ◆ On chip functions:
 - VCOM generator and adjustment
 - Timing generator
 - Oscillator
 - DC/DC converter
 - Line/frame inversion
 - 1 preset Gamma curve with separate RGB Gamma correction
- ◆ Content Adaptive Brightness Control
- ◆ MTP (3 times):
 - 8-bits for ID1, ID2, ID3
 - 7-bits for VCOM adjustment

Características generales stm32f429 2.1: Características generales stm32f429

Diagrama de bloques Stm

Figure 4. STM32F427xx and STM32F429xx block diagram

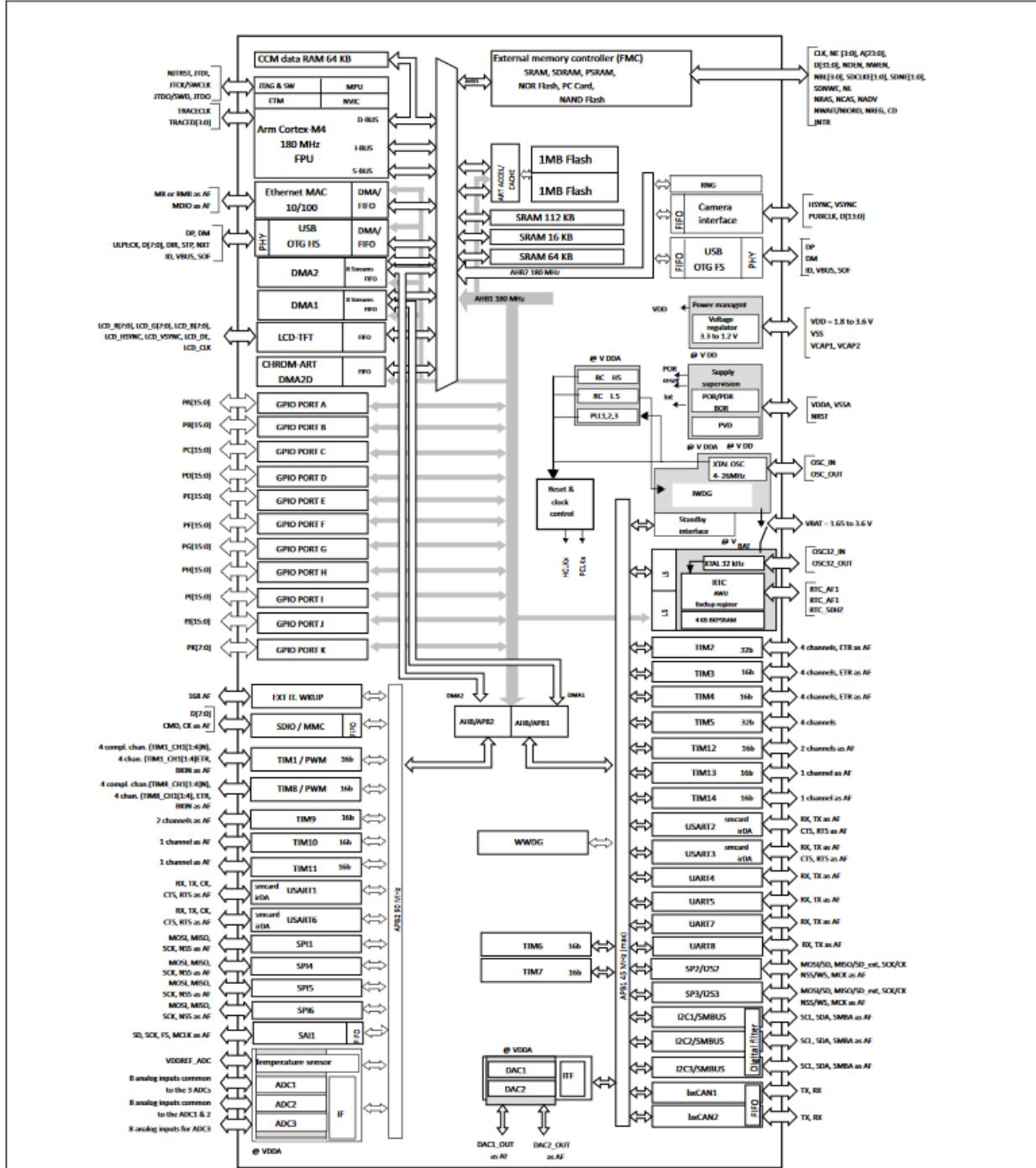
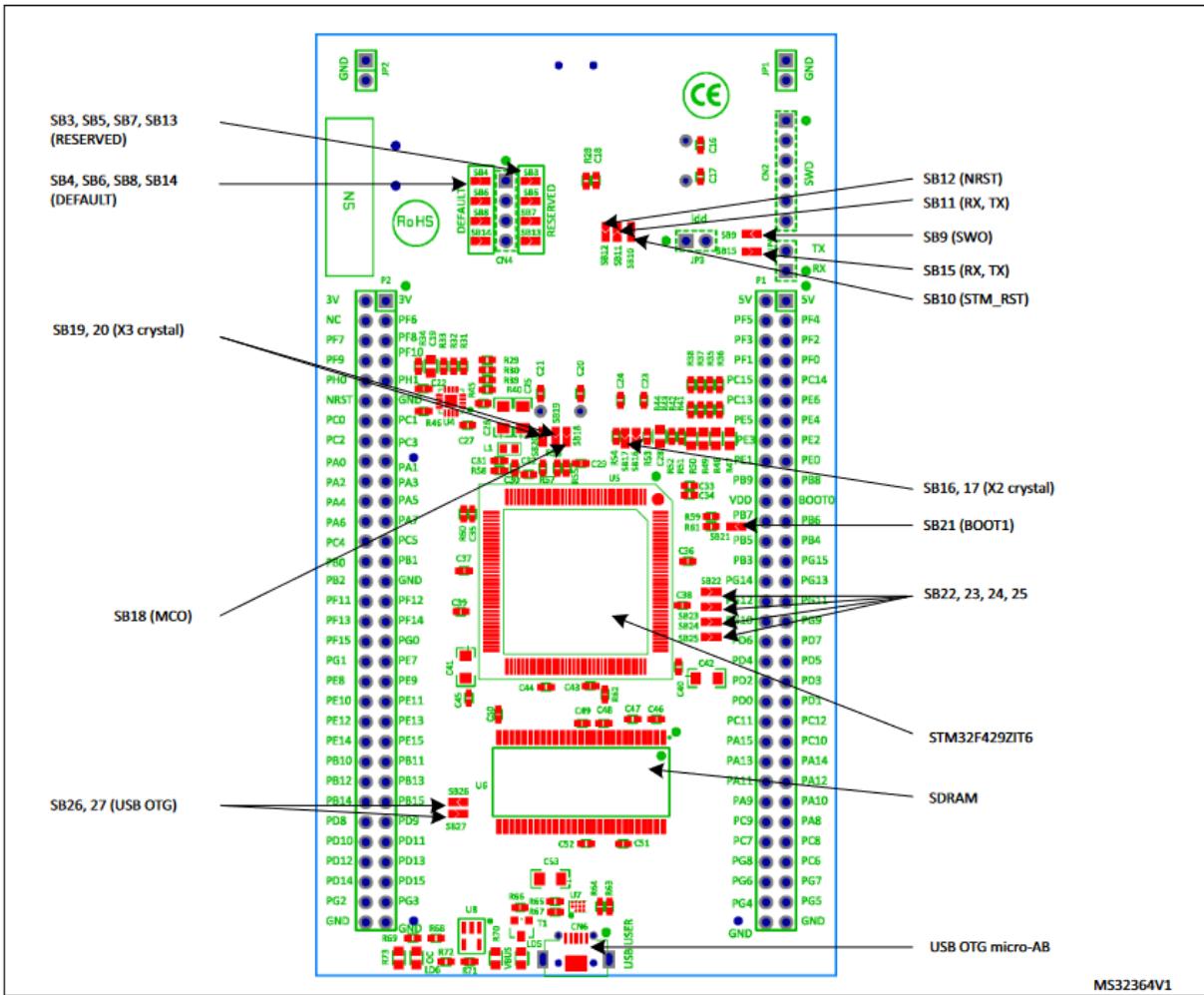


Diagrama stm32f429 2.2: Diagrama stm32f429

Diagrama de pines

Figure 3. Bottom layout



Pines stm32f429 2.3: Pines stm32f429

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	
$ AV_{DDx} $	Variations between different V_{DD} power pins	-	50	mV
$ V_{SSx} - V_{SSl} $	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)		

Características eléctricas stm32f429 2.4: Características eléctricas stm32f429

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	

Características eléctricas stm32f429 2.5: Características eléctricas stm32f429

Como se observa de las imágenes anteriores, la tensión máxima en los pines es de 3.3 V y la corriente máxima que pueden manejar es de 25 mA.

2.2. Componentes Electrónicos Complementarios

Pantalla ILI9341

3. Block Diagram

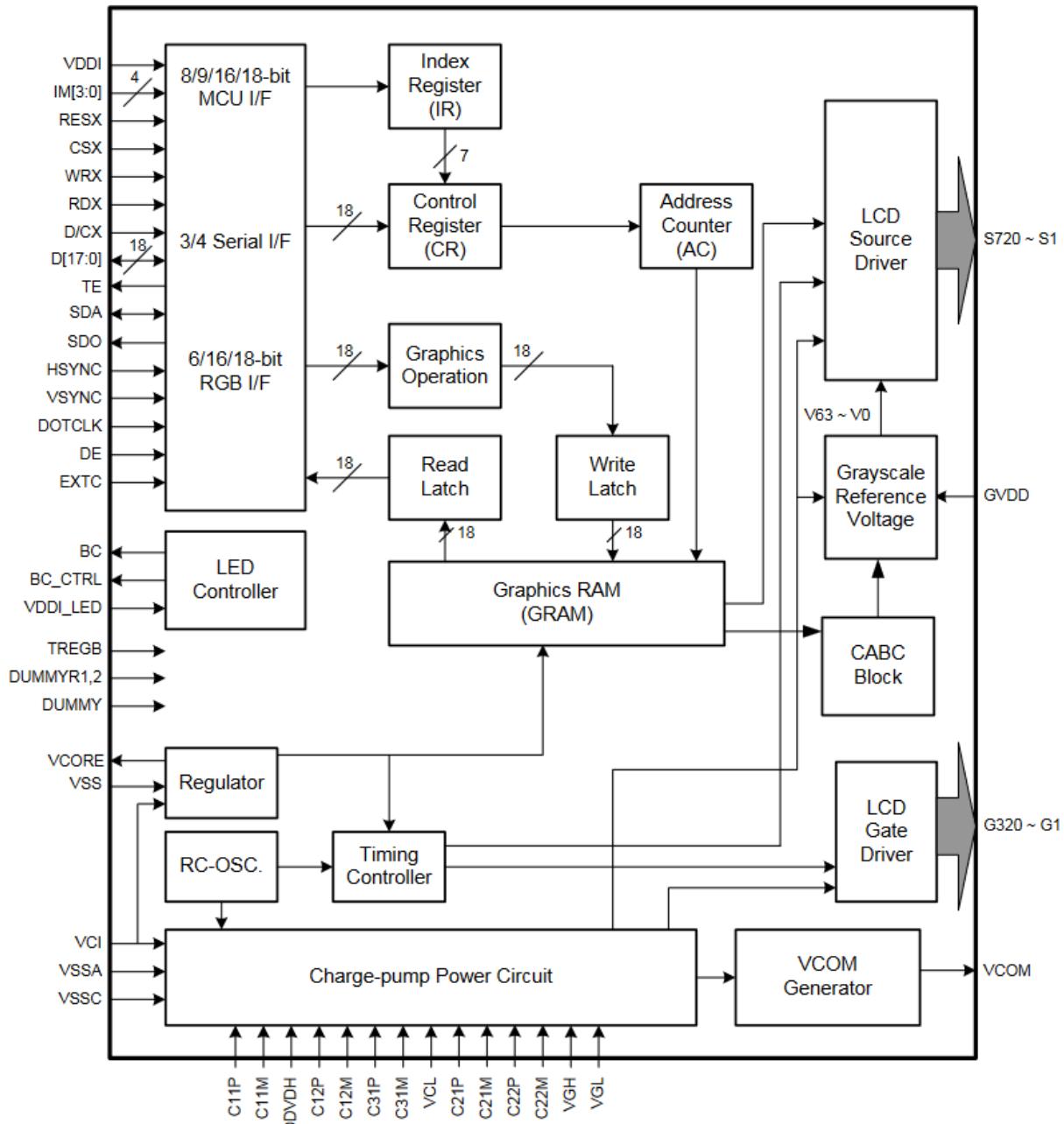


Diagrama ILI9341 2.6: Diagrama ILI9341

Item	Symbol	Unit	Value
Supply voltage	VCI	V	-0.3 ~ +4.6
Supply voltage (Logic)	VDDI	V	-0.3 ~ +4.6
Supply voltage (Digital)	VCORE	V	-0.3 ~ +2.0
Driver supply voltage	VGH-VGL	V	-0.3 ~ +28.0
Logic input voltage range	VIN	V	-0.3 ~ VDDI + 0.3
Logic output voltage range	VO	V	-0.3 ~ VDDI + 0.3
Operating temperature	Topr	°C	-40 ~ +85
Storage temperature	Tstg	°C	-55 ~ +110

Note: If the absolute maximum rating of even one of the above parameters is exceeded even momentarily, the quality of the product may be degraded. Absolute maximum ratings, therefore, specify the values exceeding which the product may be physically damaged. Be sure to use the product within the range of the absolute maximum ratings.

Características eléctricas ILI9341 2.7: Características eléctricas ILI9341

MEMS sensor de movimiento L3GD20

Features

- Three selectable full scales (250/500/2000 dps)
- I²C/SPI digital output interface
- 16 bit-rate value data output
- 8-bit temperature data output
- Two digital output lines (interrupt and data ready)
- Integrated low- and high-pass filters with user-selectable bandwidth
- Wide supply voltage: 2.4 V to 3.6 V
- Low voltage-compatible IOs (1.8 V)
- Embedded power-down and sleep mode
- Embedded temperature sensor
- Embedded FIFO
- High shock survivability
- Extended operating temperature range (-40 °C to +85 °C)
- ECOPACK® RoHS and "Green" compliant

Características generales L3GD20 2.8: Características generales L3GD20

Figure 1. Block diagram

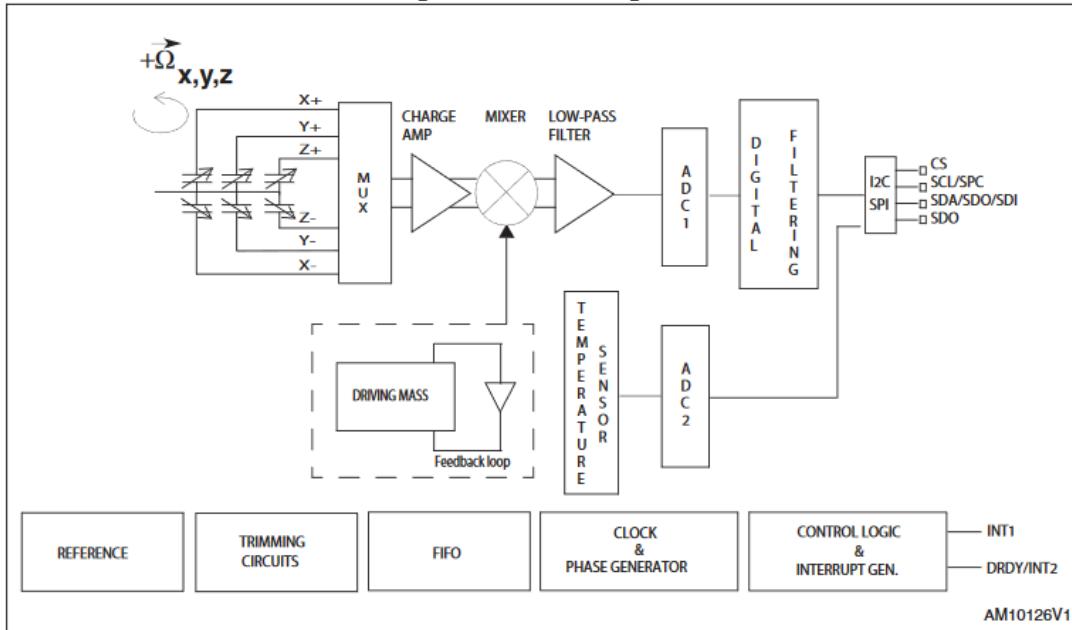
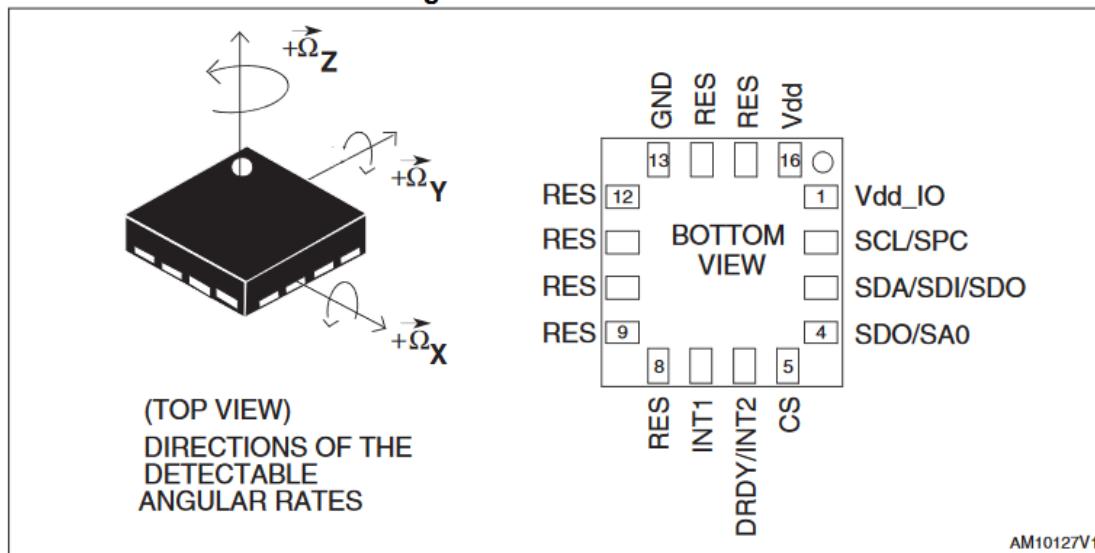


Diagrama L3GD20 2.9: Diagrama L3GD20

Pin description

Figure 2. Pin connection



Pines L3GD20 2.10: Pines L3GD20

Table 5. Electrical characteristics (1)

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
Vdd	Supply voltage		2.4	3.0	3.6	V
Vdd_IO	I/O pins supply voltage ⁽³⁾		1.71		Vdd+0.1	V
Idd	Supply current			6.1		mA
IddSL	Supply current in sleep mode ⁽⁴⁾	Selectable by digital interface		2		mA
IddPdn	Supply current in power-down mode	Selectable by digital interface		5		µA
VIH	Digital high level input voltage		0.8*Vdd_I_O			V
VIL	Digital low level input voltage				0.2*Vdd_I_O	V
Top	Operating temperature range		-40		+85	°C

Características eléctricas L3GD20 2.11: Características eléctricas L3GD20

Resistencias

- 3x 1 kΩ
- 4x 100Ω

Diodos LED

- 1x LED blanco
- 1x LED rojo

Botón

- 1x botón 'push'

Batería 9V

- 1x Batería 9V

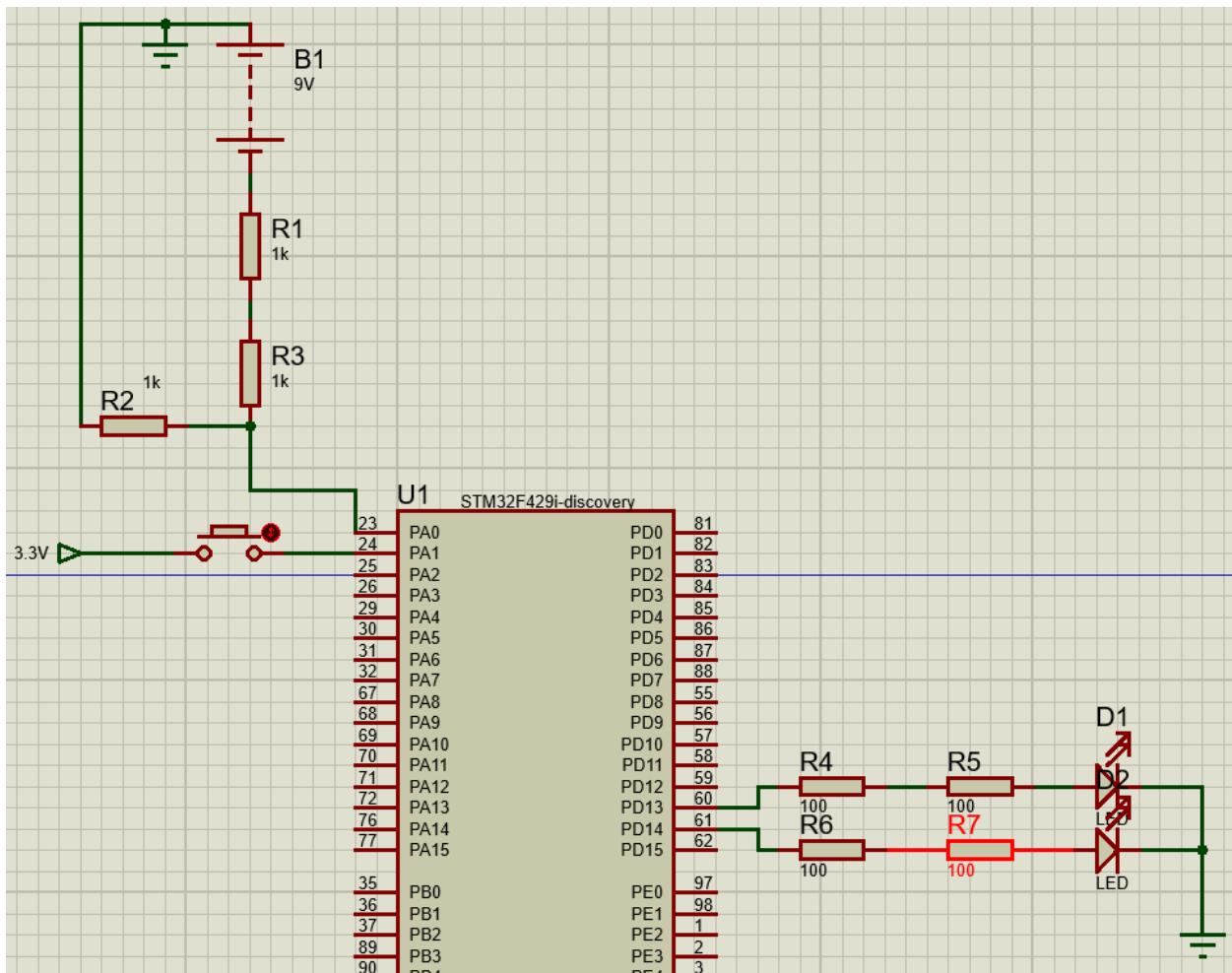
2.3. Precios

- 2x LEDS: \$0.66
- 7x Resistencias: \$1.28
- 1x Botón: \$0.47
- 1x Batería: \$1.5
- Total: **\$3.91**

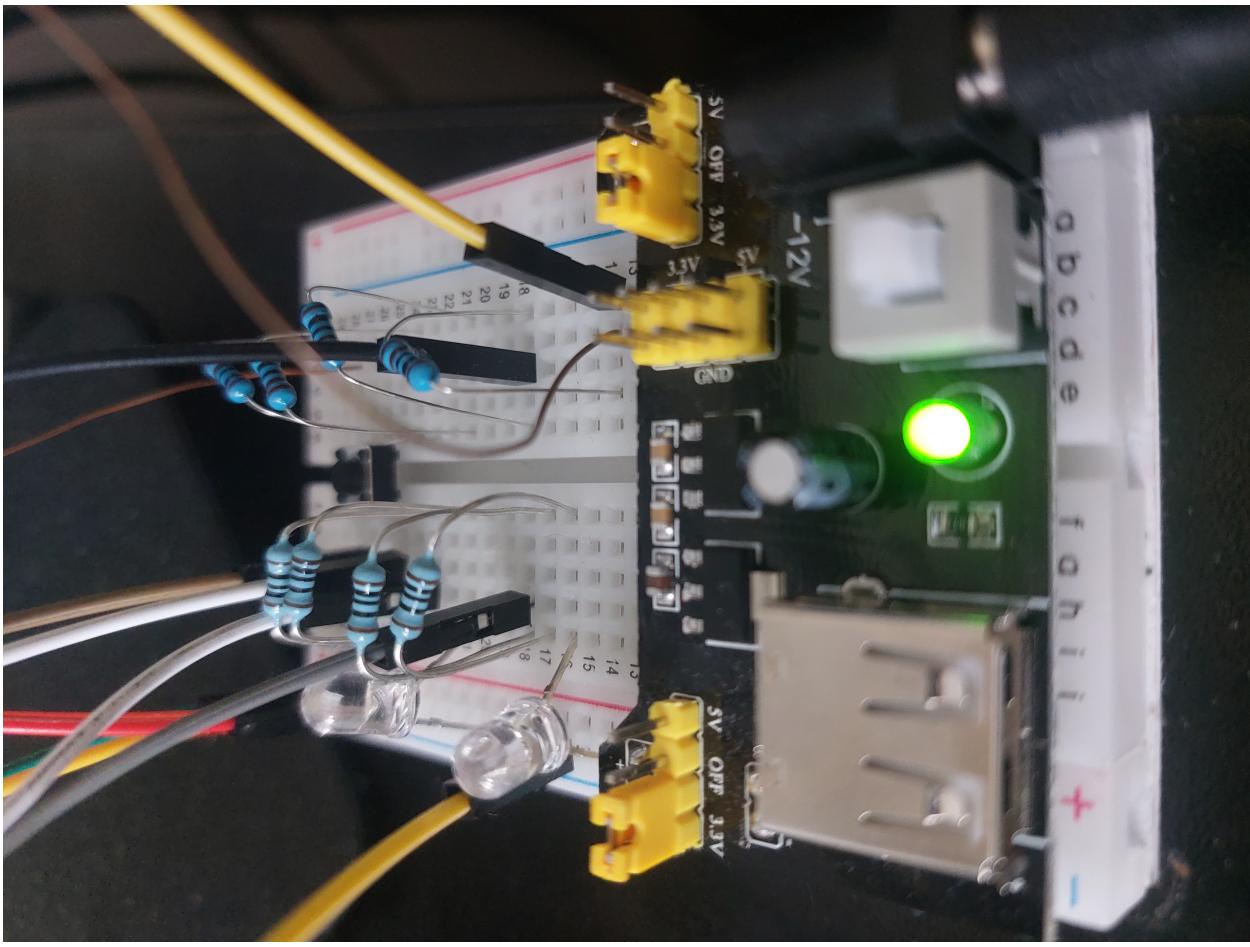
Obtenido de: [1]

2.4. Diseño del circuito

Esquemático



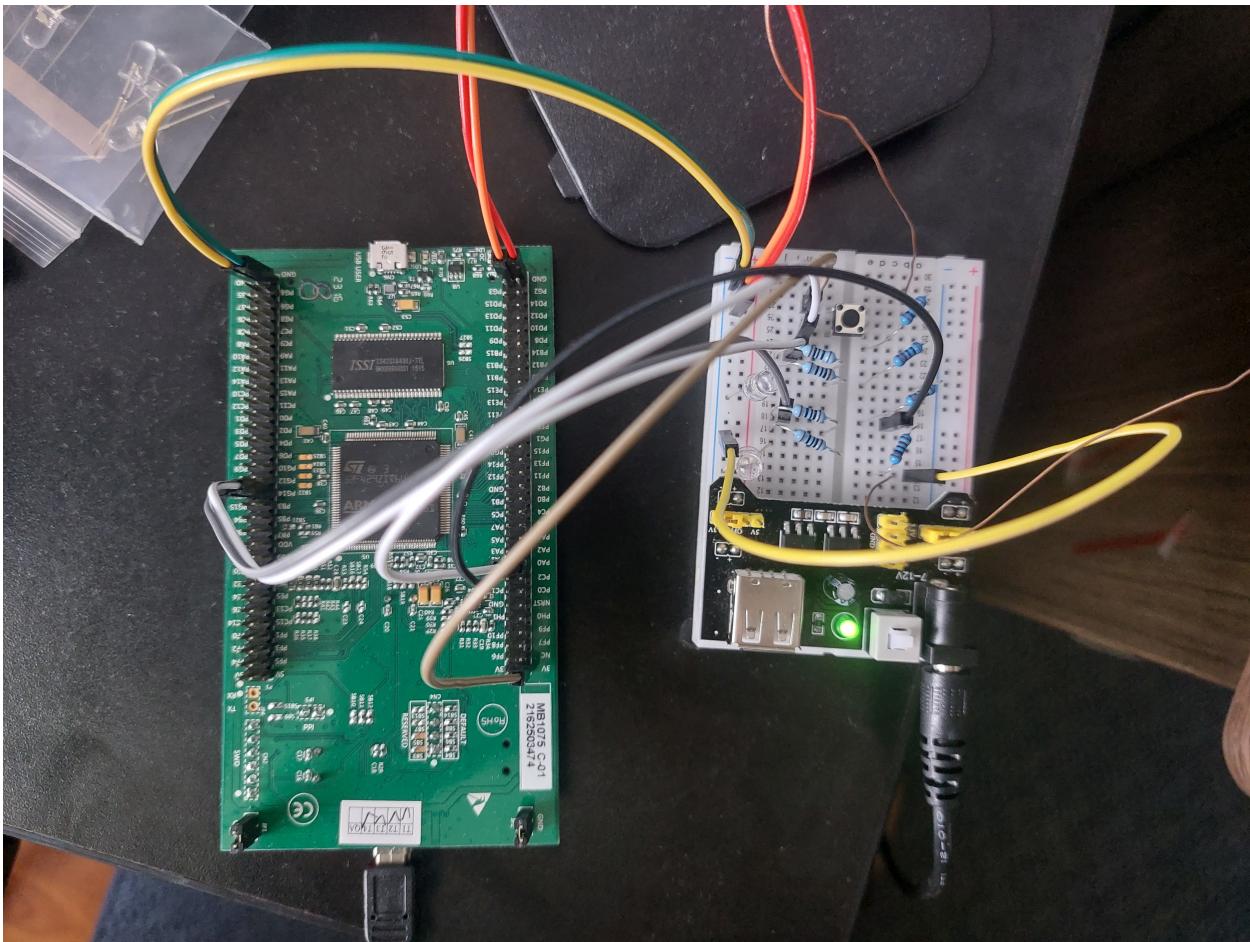
Esquemático stm-discovery 2.12: Esquemático stm-discovery



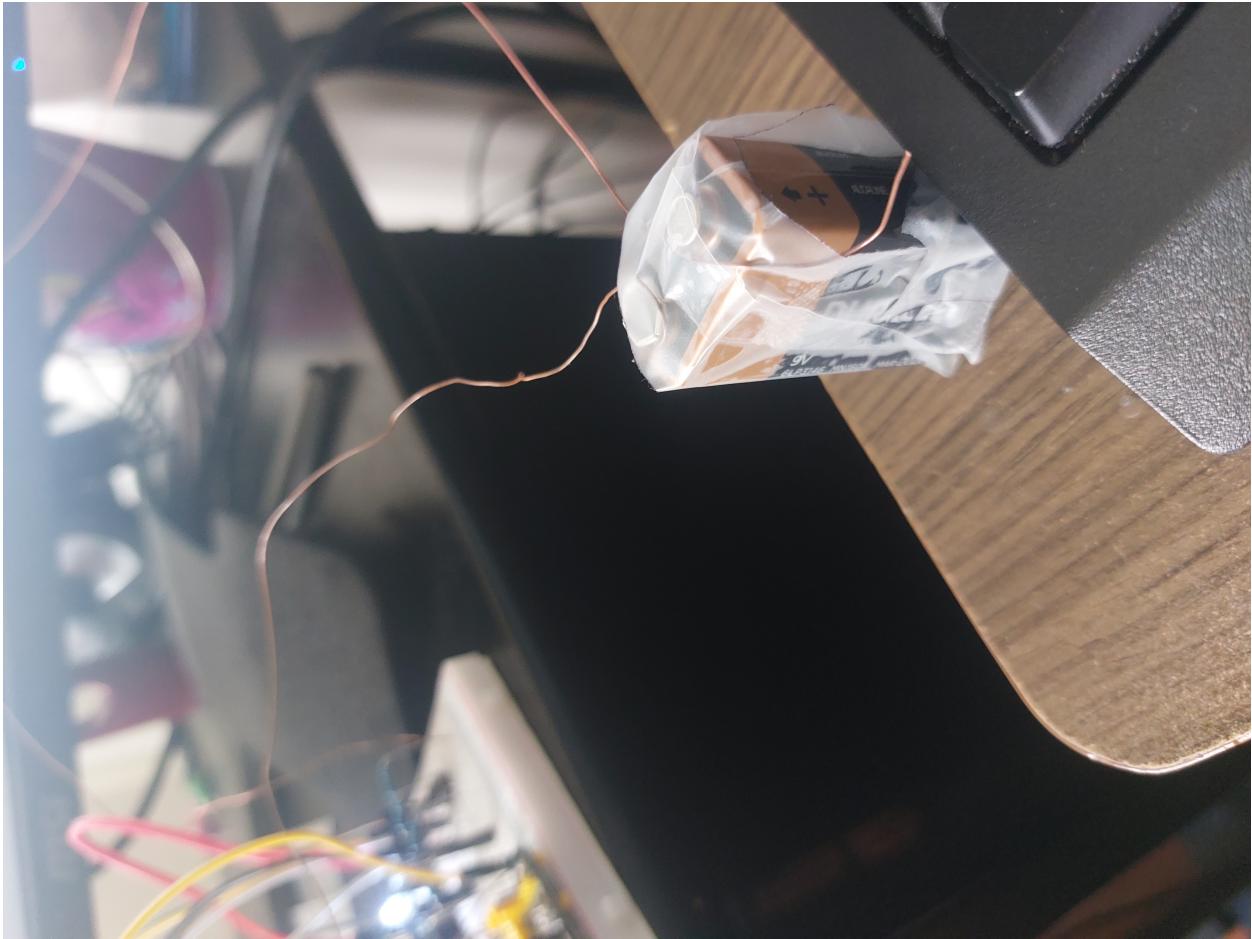
Esquemático breadboard 2.13: Esquemático breadboard



Conexión con stm32 2.14: Conexión con stm32



Circuito completo 2.15: Circuito completo



Batería 2.16: Batería

El circuito consta de 4 partes principales.

Botón USART

Al pin A1 se conecta un botón alimentado por una tensión de 3,3 V. Este botón, al ser presionado por el usuario, habilita o deshabilita la comunicación UART.

Batería con etapa de división de tensión

Al pin A0, se conecta una batería de 9V por medio de una etapa de división de tensión para ser leída por el convertidor analógico digital del microprocesador.

Al poner 3 resistencias de 1k en serie, la caída de tensión en cada resistencia va a ser de 3V. De esta forma, en la salida de la segunda resistencia, la tensión va a ser de 3V lo cual es necesario ya que la tensión máxima que toleran los pines IO del microcontrolador es de 3,3V.

LED USART

Al pin D14 se le conecta dos resistencias de $100\ \Omega$ y un diodo LED que parpadea cuando el usuario habilita la comunicación USART por medio del botón.

Como se tiene que cumplir que:

$$\frac{3,3V}{R_6 + R_7} < 20mA \quad (1)$$

Entonces, eligiendo $R_6/7=100\Omega$, se obtiene una resistencia total de 200Ω , con una corriente de 16,5mA ¡20 mA.

LED alarma batería

Al pin D13 se le conecta dos resistencias de $100\ \Omega$ y un diodo LED se enciende cuando la tensión en la batería es menor a 7V.

Como se tiene que cumplir que:

$$\frac{3,3V}{R_6 + R_7} < 20mA \quad (2)$$

Entonces, eligiendo $R_4/5=100\Omega$, se obtiene una resistencia total de 200Ω , con una corriente de 16,5mA ¡20 mA.

2.5. Temas de laboratorio

USART

La UART (Universal Asynchronous Receiver/Transmitter) es un componente fundamental en la comunicación de datos digitales entre dispositivos electrónicos. Es una interfaz serial asíncrona que permite la transmisión y recepción de datos binarios mediante un par de hilos, uno para enviar (TX) y otro para recibir (RX).

La transmisión UART se realiza mediante un proceso de envío de bits secuenciales, donde cada bit es precedido por un bit de inicio y seguido por uno o más bits de parada. La velocidad de transmisión se determina por una tasa de baudios, que representa la cantidad de veces por segundo que se transmiten los bits.

[2]

SPI

SPI (Serial Peripheral Interface) es un protocolo de comunicación sincrónico utilizado para la transferencia de datos entre dispositivos electrónicos. Se basa en una arquitectura de maestro y esclavo, donde un dispositivo maestro controla la comunicación con uno o varios dispositivos esclavos.

La comunicación SPI se realiza mediante la transmisión de bits secuenciales a través de cuatro líneas: la línea de reloj (SCK), la línea de selección de esclavo (SS), la línea de salida del maestro (MOSI) y la línea de entrada del maestro (MISO). El maestro genera una señal de reloj que se utiliza para sincronizar la transmisión de los bits de datos entre el maestro y el esclavo.

La velocidad de transmisión en SPI se determina por la frecuencia del reloj, que se puede configurar para operar a diferentes velocidades. Además, SPI permite la configuración de varios parámetros, como el modo de transmisión, la configuración de bit de datos y la activación de la interrupción. [2]

ADC

Las conversiones analógicas a digitales (ADC) en Arduino se realizan utilizando un conversor analógico a digital de 10 bits. El ADC mide el voltaje de entrada en la entrada analógica y lo convierte en un valor digital de 0 a 1023, donde 0 representa un voltaje de 0 voltios y 1023 representa un voltaje de referencia de 5 voltios. La precisión de la conversión ADC se puede mejorar utilizando un voltaje de referencia externo o ajustando la resolución del conversor a través de la biblioteca de Arduino.

La biblioteca de Arduino proporciona funciones para realizar conversiones ADC de manera sencilla. Los usuarios pueden usar la función `.analogRead()` para leer el valor de voltaje en una entrada analógica determinada y convertirlo en un valor digital. Además, los usuarios pueden configurar la resolución del conversor y el voltaje de referencia utilizando las funciones `.analogReference()` y `.analogReadResolution()`. [2]

IoT

El concepto de Internet de las cosas (IoT, por sus siglas en inglés) se refiere a la interconexión de objetos físicos, dispositivos y sistemas a través de internet, permitiéndoles recopilar e intercambiar datos.

Incluye una amplia gama de dispositivos, como electrodomésticos inteligentes, dispositivos portátiles, maquinaria industrial e incluso vehículos. Estos dispositivos interconectados están equipados con sensores, actuadores y capacidades de comunicación que les permiten recopilar y transmitir datos a la nube u otros dispositivos, facilitando el monitoreo en tiempo real, el control y la automatización.

El objetivo fundamental del IoT es crear una red inteligente e interconectada donde los objetos físicos puedan comunicarse y colaborar sin problemas, lo que permite nuevas aplicaciones y servicios que mejoran la eficiencia, productividad y comodidad en diversos ámbitos, como la salud, el transporte, la agricultura y las ciudades inteligentes.

MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería ligero y de bajo consumo de energía diseñado para la comunicación entre dispositivos en el contexto del Internet de las cosas (IoT).

MQTT se basa en un modelo de publicación/suscripción, en el que los dispositivos se comunican a través de un intermediario llamado broker. Los dispositivos pueden publicar mensajes en ciertos temas (topics) y otros dispositivos pueden suscribirse a esos temas para recibir los mensajes. Esto permite una comunicación eficiente y escalable entre una gran cantidad de dispositivos distribuidos en diferentes redes y ubicaciones geográficas.

MQTT es especialmente adecuado para entornos con ancho de banda limitado y conexiones poco confiables, ya que utiliza un formato de mensaje compacto y permite la entrega de mensajes confiable.

3. Desarrollo y análisis

3.1. Programa

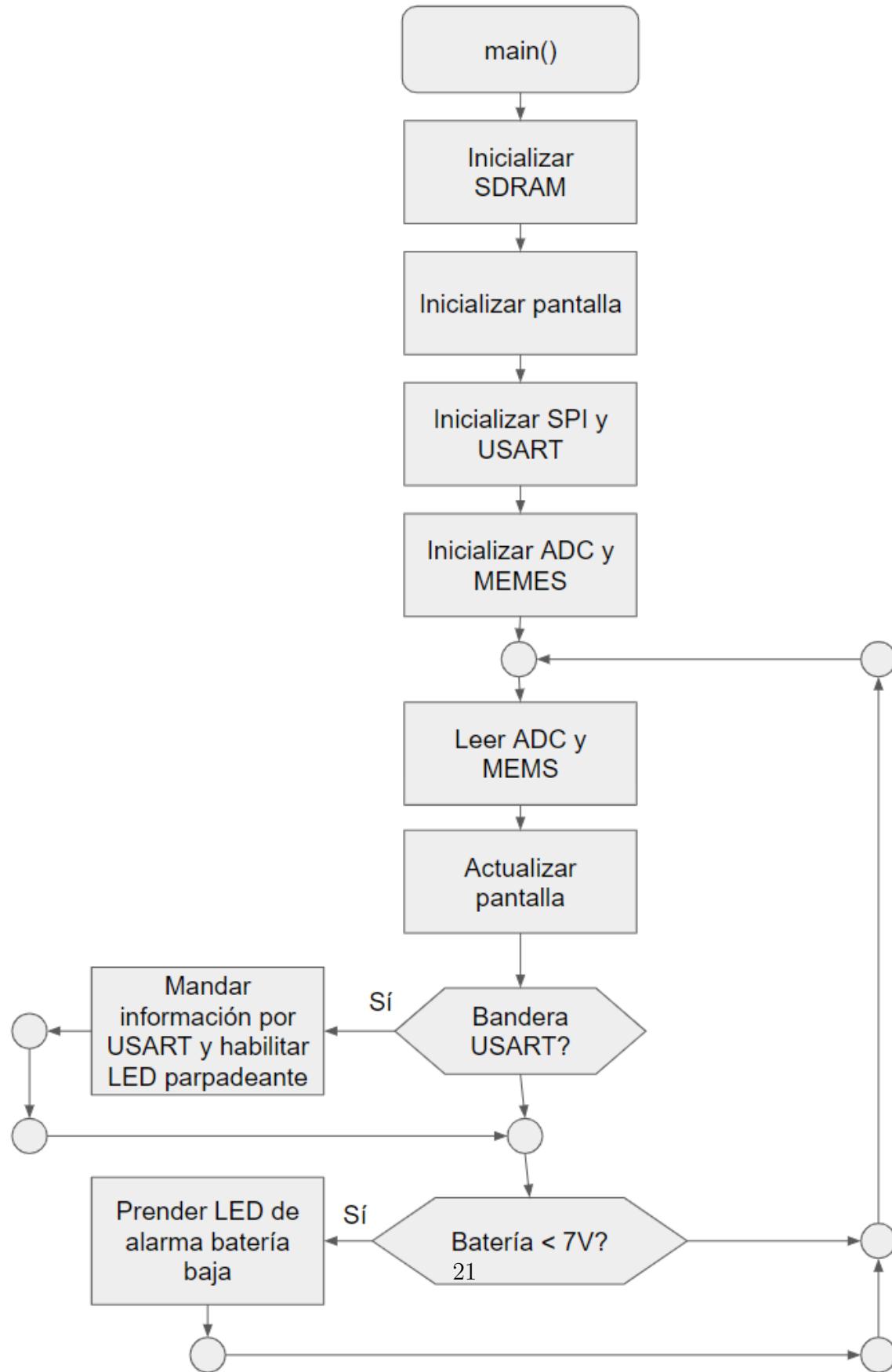
Descripción del programa

El programa fue escrito en lenguaje C utilizando la biblioteca libopencm3. El programa no hace uso de ninguna interrupción, todas las lecturas de entradas se hacen por medio de 'polling'.

El programa tiene la siguiente estructura: Primero se inicializa el SDRAM, el módulo spi y usart, el ADC y la pantalla.

Luego se lee los valores del ADC para la batería, el valor del botón y los valores del MEMS por medio de SPI. Seguidamente, se actualiza en la pantalla los valores leídos y se mandan por USART a la PC en caso de que la opción haya sido habilitada por el usuario por medio del botón. Por último, según los valores leídos del botón y la batería, se encienden o se apagan los diodos LED de alarma de batería y comunicación USART.

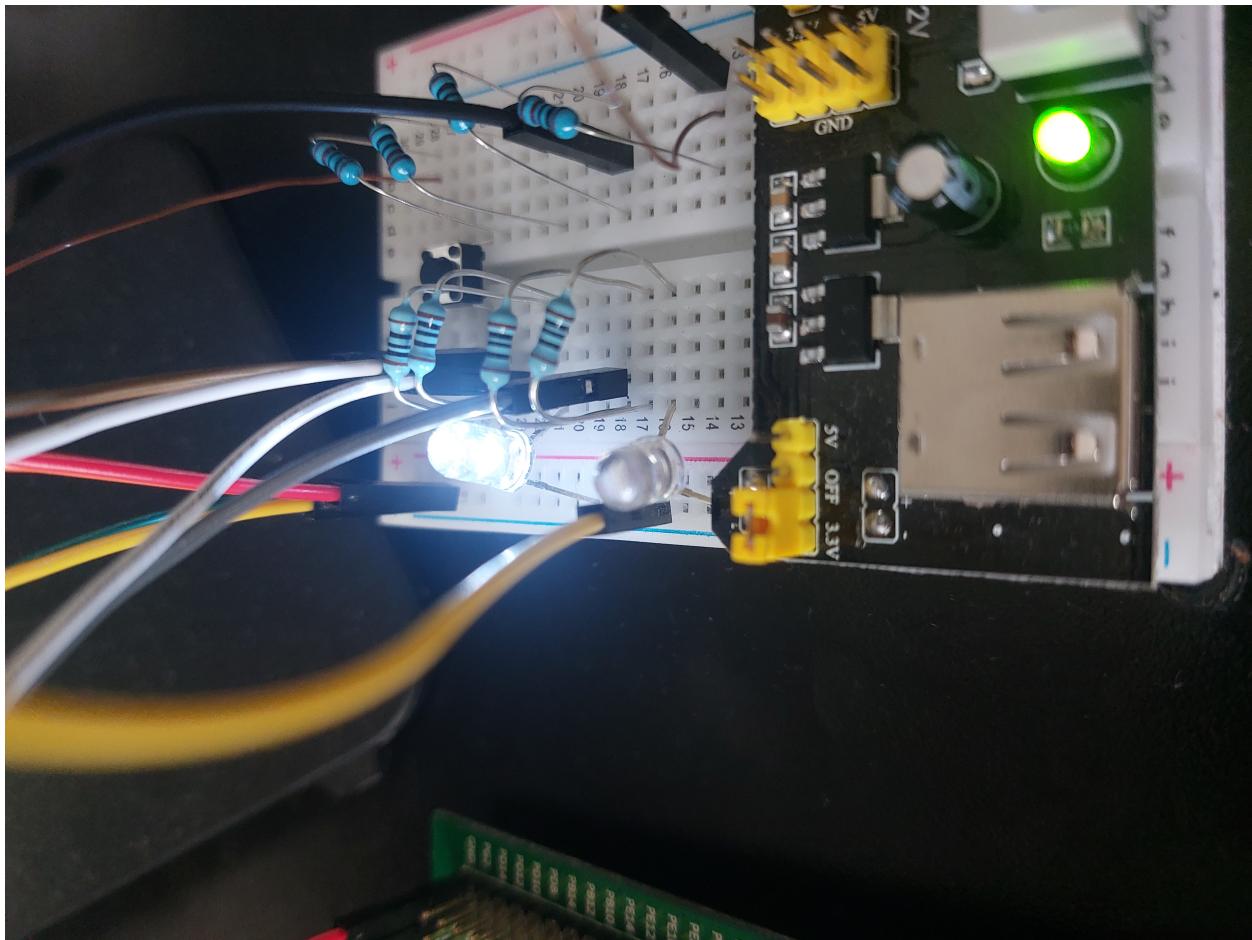
Diagrama de bloques



3.2. Componentes y funcionamiento

LED comunicación Usart

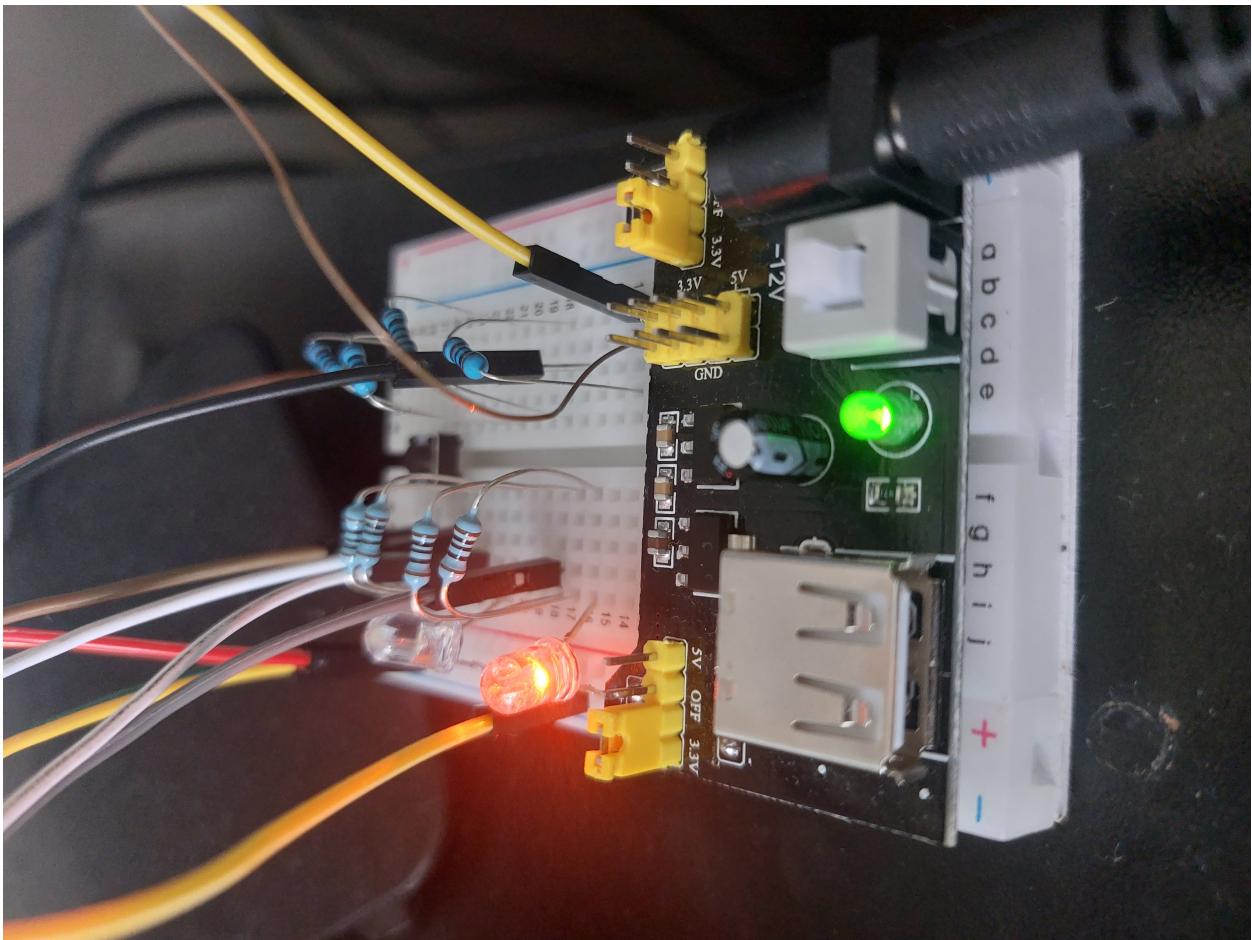
En la siguiente imagen se muestra el LED parpadeante de la comunicación USART habilitada por el usuario por medio del botón.



LED Usart 3.2: LED Usart

LED alarma baja batería

En la siguiente imagen se conecta el pin del ADC que lee la batería a tierra para probar si el LED de alarma se enciende.



LED alarma 3.3: LED alarma

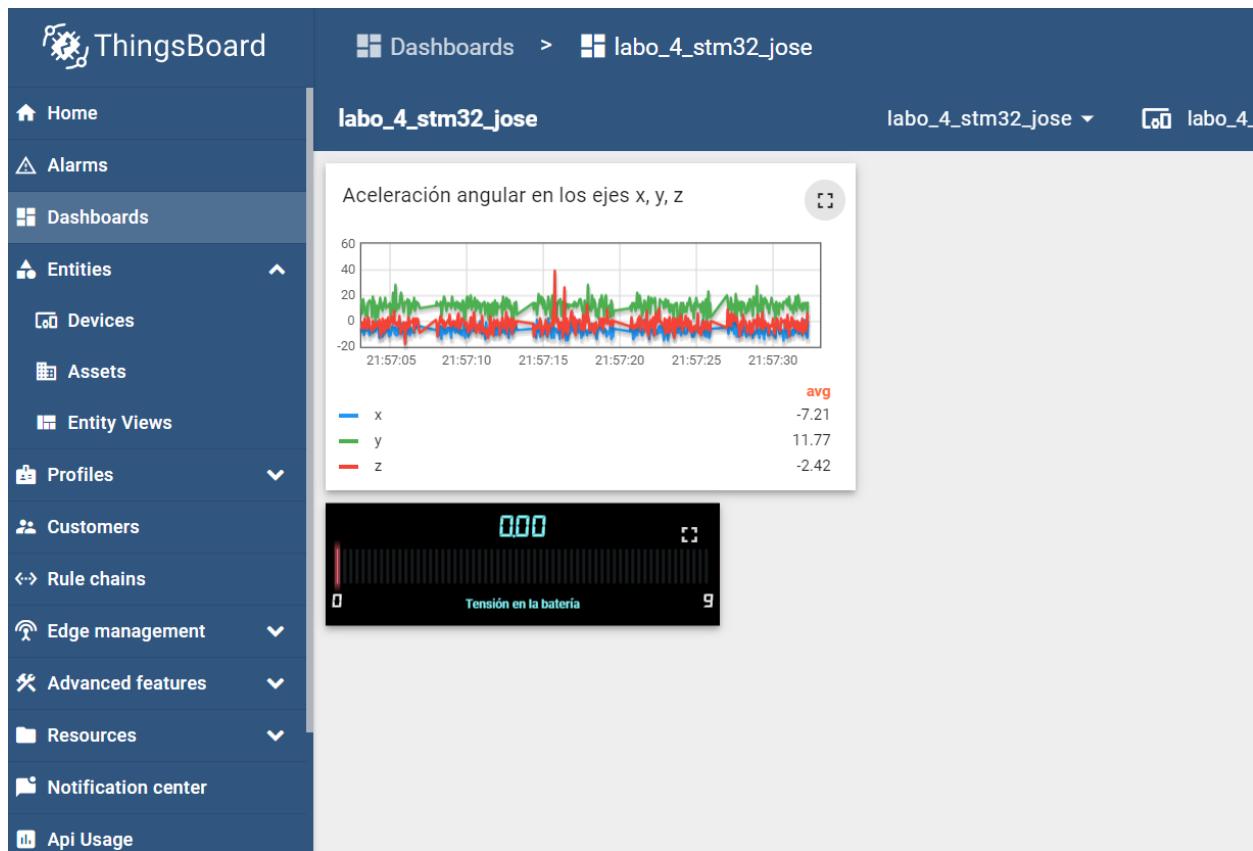
Lectura de nivel de batería y giroscopio

En la siguiente imagen se muestra el script de python mandando información al servidor mqtt.

```
{'x': '-6', 'y': '8', 'z': '0', 'bateria': 9.89}
{'x': '-2', 'y': '6', 'z': '-1', 'bateria': 9.89}
{'x': '-9', 'y': '14', 'z': '-6', 'bateria': 9.89}
{'x': '-4', 'y': '19', 'z': '5', 'bateria': 9.89}
{'x': '-7', 'y': '7', 'z': '-8', 'bateria': 9.89}
{'x': '-10', 'y': '6', 'z': '-4', 'bateria': 9.89}
{'x': '-7', 'y': '13', 'z': '-8', 'bateria': 9.89}
{'x': '-6', 'y': '12', 'z': '-7', 'bateria': 9.89}
{'x': '-5', 'y': '14', 'z': '0', 'bateria': 9.89}
{'x': '-10', 'y': '12', 'z': '0', 'bateria': 9.89}
{'x': '-9', 'y': '15', 'z': '-4', 'bateria': 9.89}
{'x': '-8', 'y': '5', 'z': '5', 'bateria': 9.89}
{'x': '-8', 'y': '14', 'z': '1', 'bateria': 9.89}
{'x': '-4', 'y': '13', 'z': '-5', 'bateria': 9.89}
{'x': '-6', 'y': '14', 'z': '-4', 'bateria': 9.89}
{'x': '-9', 'y': '5', 'z': '-2', 'bateria': 9.89}
{'x': '-3', 'y': '16', 'z': '3', 'bateria': 9.89}
{'x': '-7', 'y': '11', 'z': '1', 'bateria': 9.89}
{'x': '-11', 'y': '5', 'z': '-4', 'bateria': 9.89}
{'x': '-8', 'y': '18', 'z': '-12', 'bateria': 9.89}
{'x': '-8', 'y': '11', 'z': '-3', 'bateria': 9.89}
{'x': '-5', 'y': '25', 'z': '-1', 'bateria': 9.89}
```

mqtt python 3.4: mqtt python

En la siguiente imagen se muestra el dashboard de thingsboard.



Dashboard thingsboard 3.5: Dashboard thingsboard

En la siguiente imagen se muestra la lectura de la batería cuando el pin del ADC se conecta a una batería de 9V nueva.



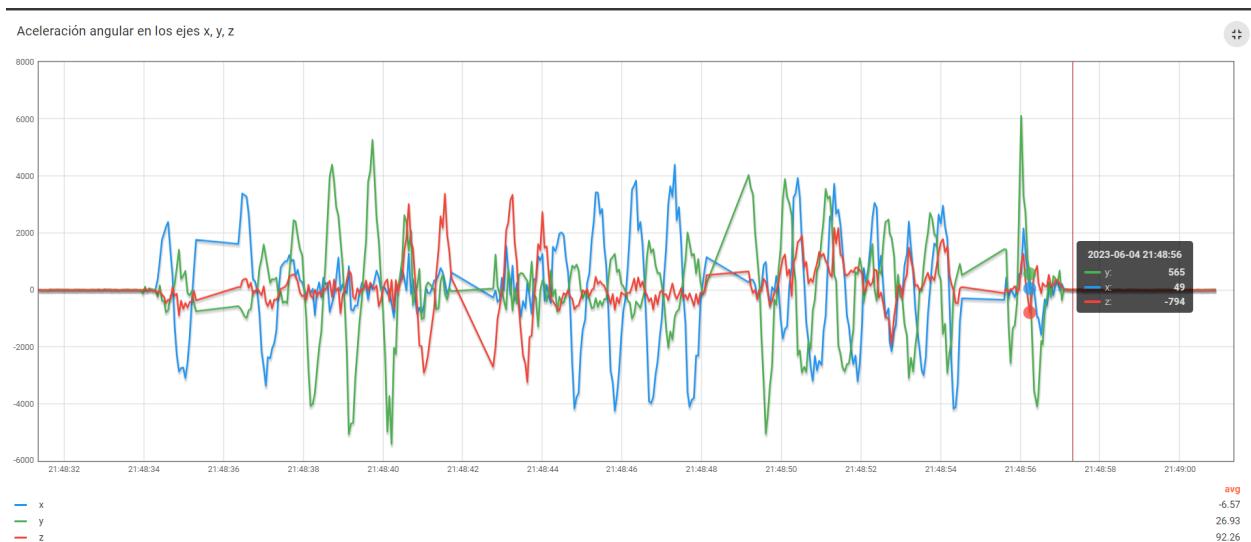
Batería nivel alto 3.6: Batería nivel alto

En la siguiente imagen se muestra la lectura de la batería cuando el pin del ADC se conecta a tierra.



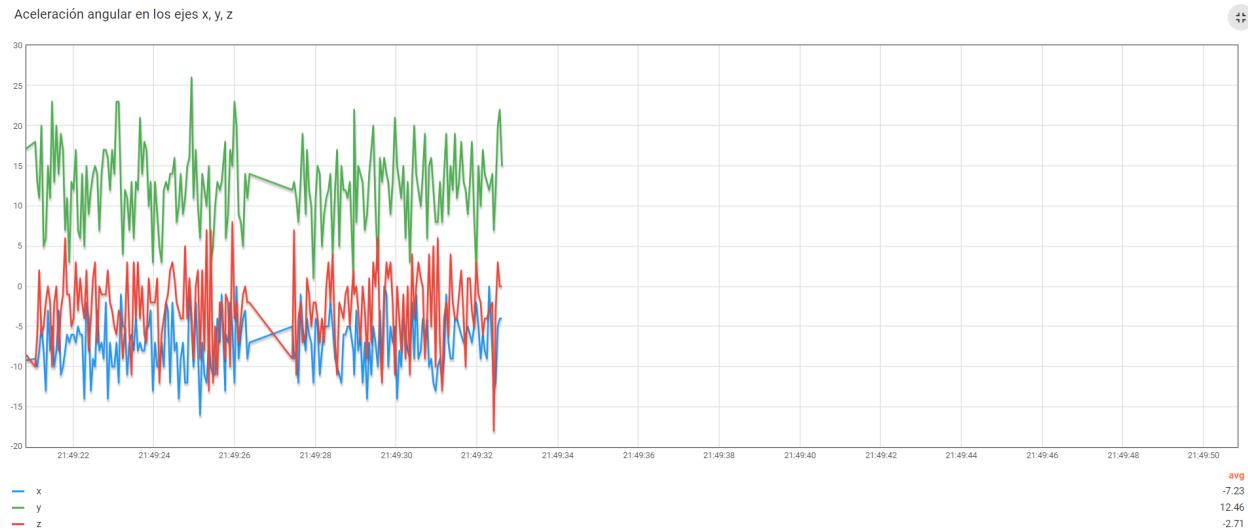
Batería nivel bajo 3.7: Batería nivel bajo

En la siguiente imagen se muestra la lectura de los 3 ejes del giroscopio en thingsboard al mover el microcontrolador de forma aleatoria.



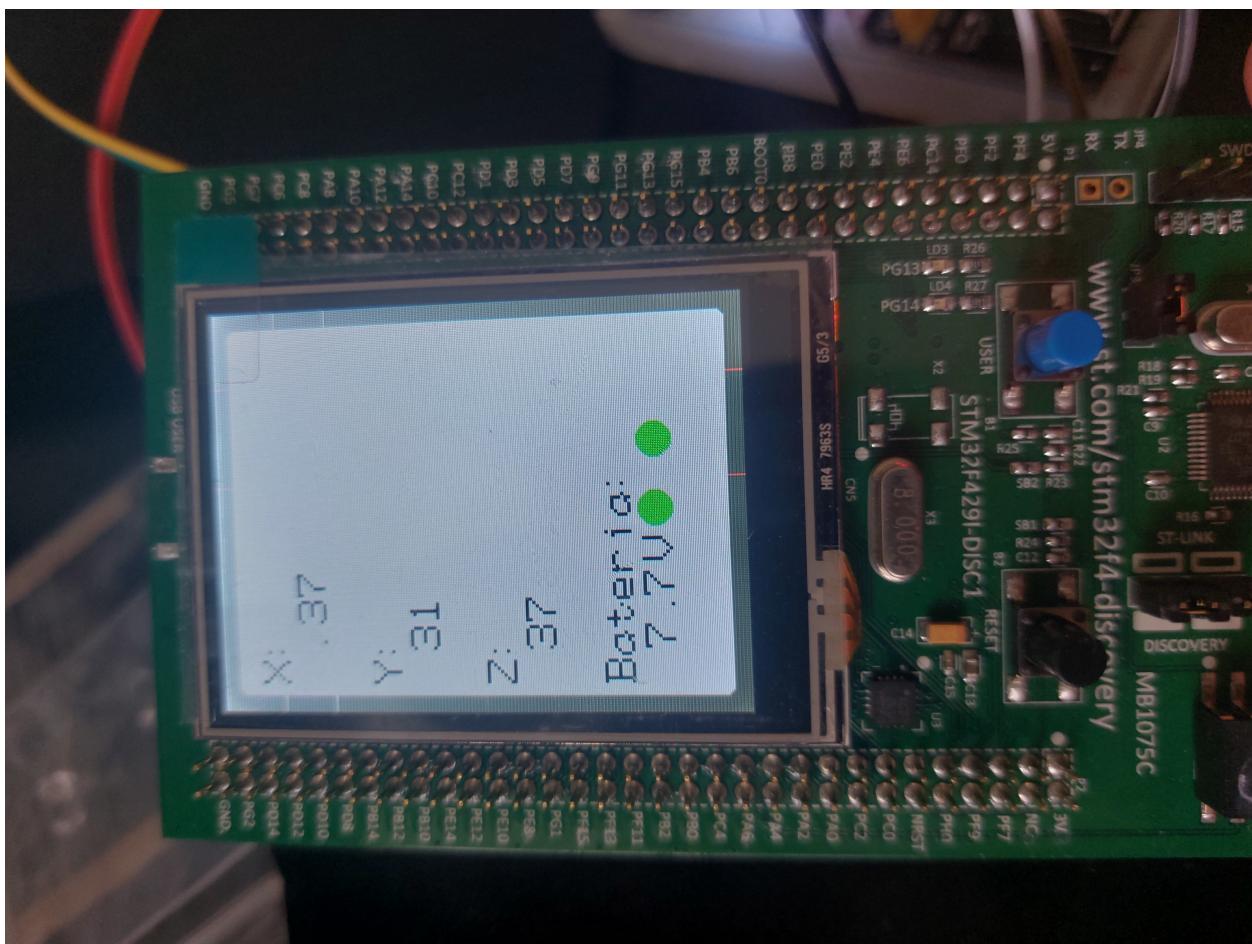
Ejes MEMS 3.8: Ejes MEMS

En la siguiente imagen se muestra la interrupción de datos cuando el usuario presiona el botón de habilitación de USART para detener la comunicación.

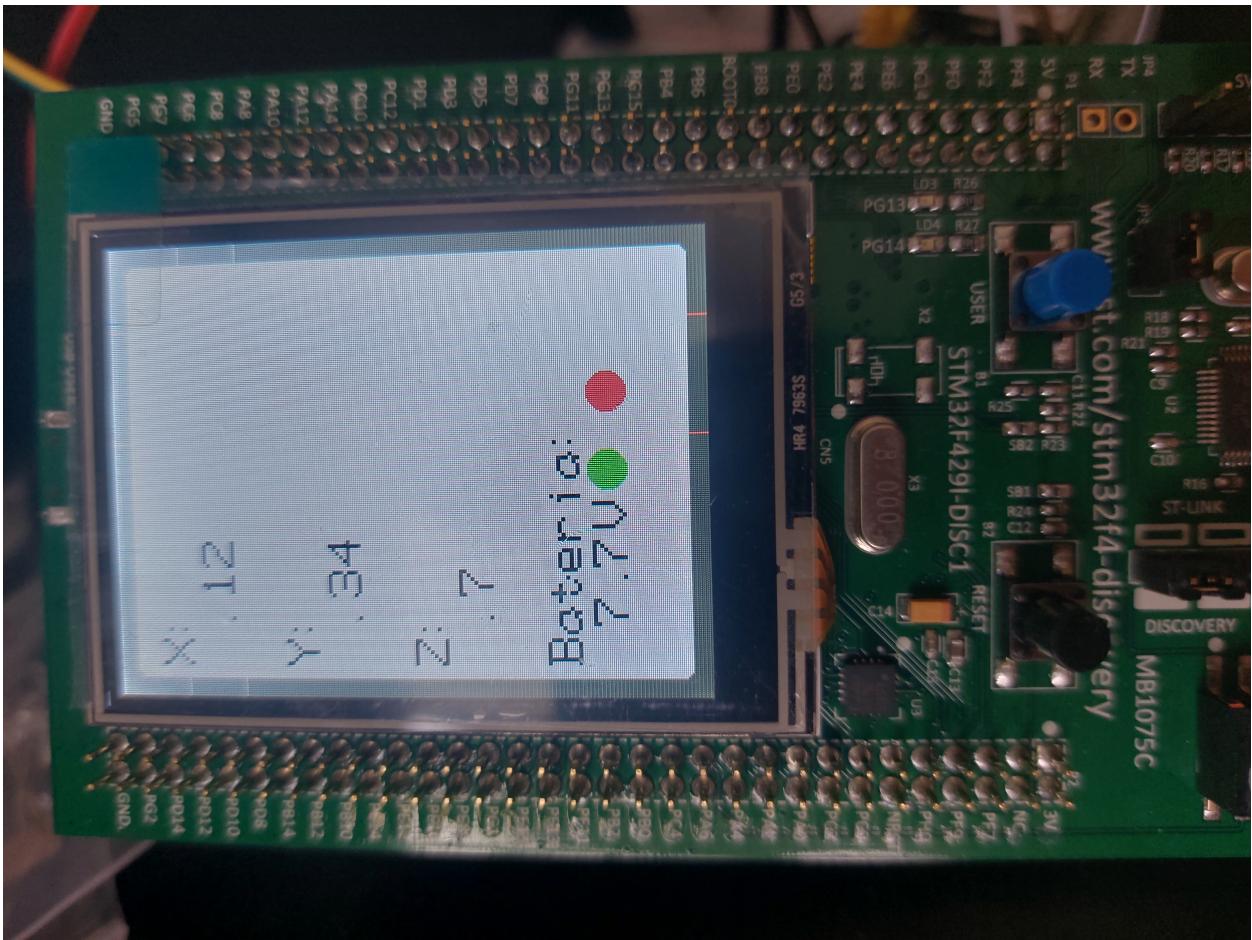


Parar comunicación USART 3.9: Parar comunicación USART

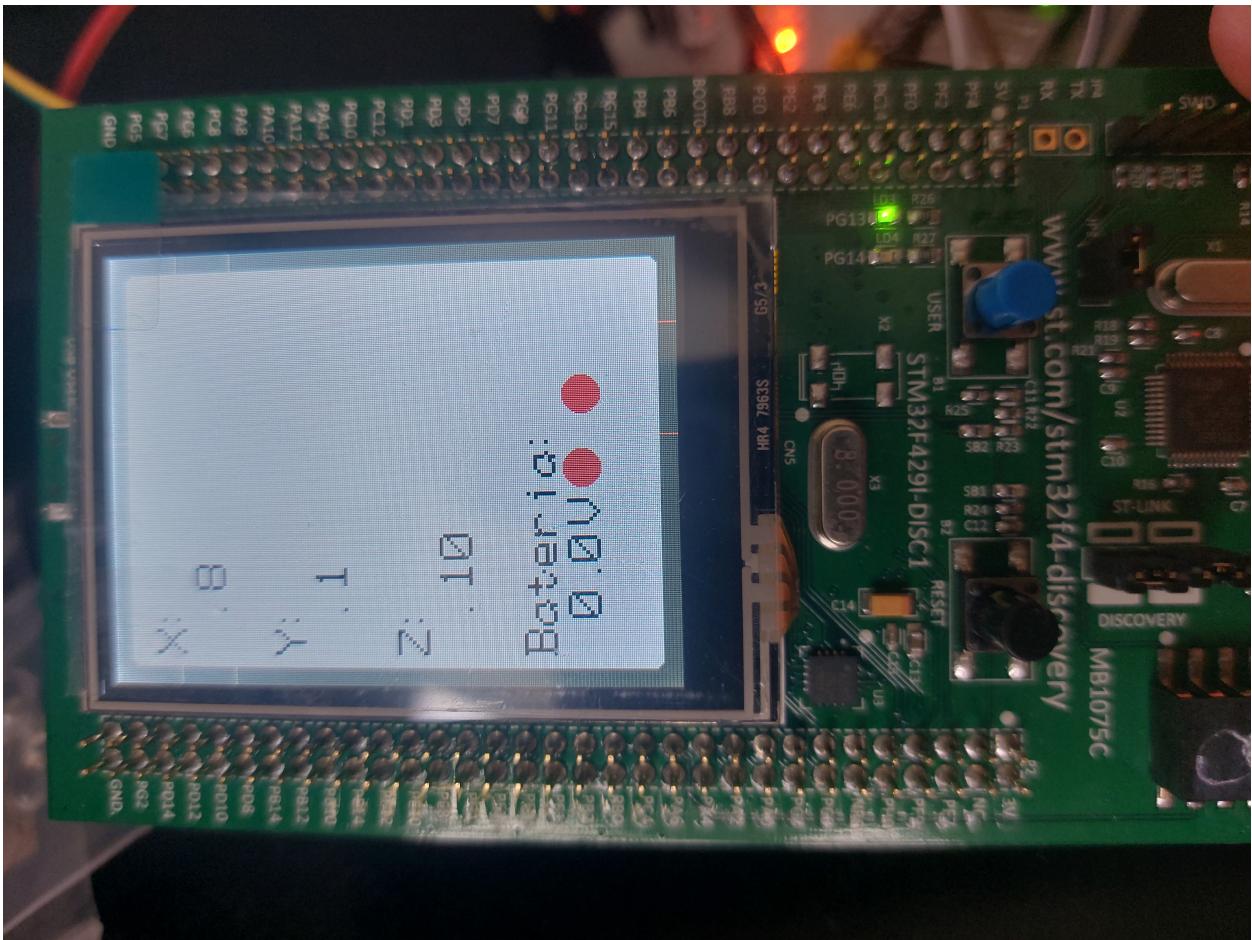
En las siguientes imágenes se muestra la pantalla en los siguientes casos respectivamente:
 Batería nivel alto y comunicación USART habilitado, batería nivel bajo y comunicación serial habilitado, batería nivel bajo y comunicación serial deshabilitado.



Caso 1 3.10: Caso 1



Caso 2 3.11: Caso 2



Caso 3 3.12: Caso 3

4. Conclusiones

Se concluyé que la solución propuesta para este cuarto laboratorio cumple con los requerimientos del enunciado y se comporta según el diseño teórico realizado.

4.1. Recomendaciones

Una recomendación para este laboratorio es encontrar la forma de poder seguir mandando datos del MEMS mientras se actualiza la pantalla. Utilizando el código proporcionado en los ejemplos del stm32, si se intenta de mandar datos USART por medio de interrupcio-

nes y correr comandos de actualización de pantalla al mismo tiempo, la temporización de actualización de pantalla no funciona correctamente y en la pantalla solo se muestra basura.

Bibliografía

- [1] “Componentes electrónicos.” <https://www.crcibernetica.com/>, 2003. (Accessed on 25/03/2023).
- [2] “Microcontroladores.” <https://en.wikipedia.org/wiki/Microcontroller>, 2006. (Accessed on 21/04/2023).

5. GIT

https://github.com/JAR1224/Laboratorio_4_Jose_Ramos