

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE-0624: LABORATORIO DE MICROCONTROLADORES

Laboratorio 4

STM32: GPIO, ADC, comunicaciones, Iot

Prof. Marco Villalta

Estudiantes:
Sofía Fonseca Muñoz, B42634

19 de febrero de 2023

Índice

1. Introducción/Resumen	2
2. Nota teórica	3
2.1. Microcontrolador	3
2.1.1. Características eléctricas	5
2.2. Bibliotecas	6
2.3. Diseño	6
2.4. Componentes electrónicos	6
3. Desarrollo/Análisis de resultados	8
3.1. Lectura de los valores del giroscopio	8
3.2. Transmisión de los datos por medio de USART	9
3.3. Botón y LED asociados al protocolo USART	9
3.4. Lectura del valor de la batería	10
3.5. Utilización de la pantalla para mostrar datos	11
3.6. Script de python para enviar datos a ThingsBoard	12
4. Conclusiones y recomendaciones	14
Referencias	15
5. Apéndice	16

1. Introducción/Resumen

Este proyecto consiste en la creación de un sismógrafo por medio del microcontrolador STM32f429 Discovery kit. Este kit está conformado por el microcontrolador ARM STM32F429, varios botones, una pantalla LCD táctil, varios leds y otros componentes que permiten la creación de proyectos gracias a su integralidad.

El sismógrafo utiliza el giroscopio interno que posee el kit para detectar movimiento. Las aceleraciones dadas se muestran en la pantalla junto con el nivel de batería con el que está funcionando y si se están transmitiendo datos. Este sismógrafo utiliza el protocolo USART para enviar datos a la computadora, la cual se encarga de enviar esta información a ThingsBoard.

ThingsBoard es una plataforma IoT que permite manipular y desplegar datos para diferentes funcionalidades. En este caso muestra los datos e indica si el sismógrafo detectó un temblor.

El repositorio correspondiente a este laboratorio se puede encontrar en el enlace https://github.com/sofifon/IE0624-2023/tree/main/Lab4_Seismograph.

2. Nota teórica

2.1. Microcontrolador

Para este laboratorio se utilizó la placa STM32F429i Discovery kit. Esta placa permite el desarrollo de aplicaciones gracias a el microprocesador de alto rendimiento STM32F4 que incluye la placa. Es una placa que ha sido desarrollada por medio de software libre por lo que contiene una extensa cantidad de librerías y ejemplos para el desarrollo de proyectos. [1]

La placa cuenta con un procesador STM32F429ZIT6 con una memoria flash de 2Mb y 256 Kb de RAM, una pantalla LCD táctil de 2.4 pulgadas, conexión por mini USB, un giroscopio, 6 LEDs, dos botones, funciones de datos USB, poder por medio de batería externa. [1] En la siguiente imagen se puede observar el diagrama de la placa y los componentes que posee en su lado superior:

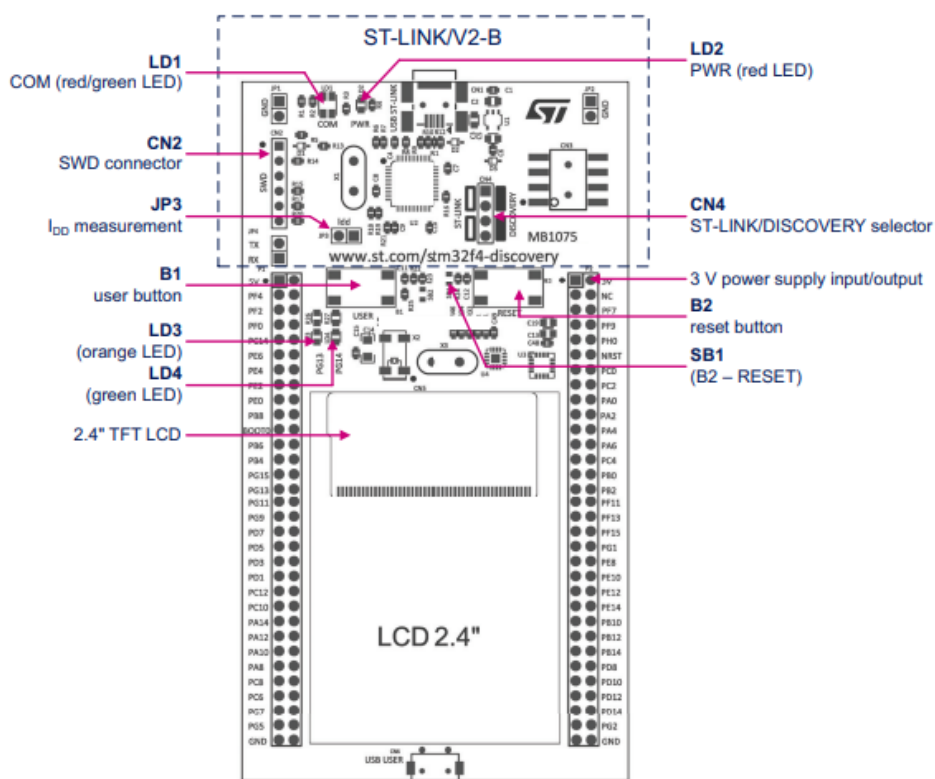


Figura 1: Diagrama frontal de la placa STM32F429 Discovery Kit. Tomada de <https://www.st.com/en/evaluation-tools/32f429idiscovery.html>

En la siguiente imagen se muestra el lado trasero de la placa donde incluso se puede observar el microcontrolador:

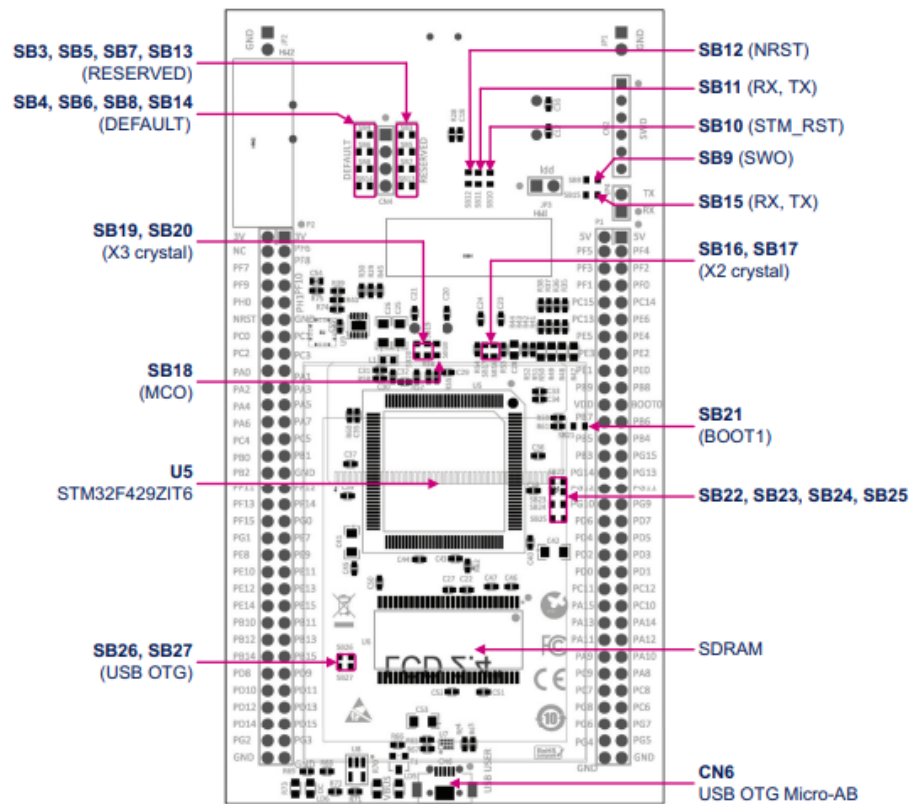


Figura 2: Diagrama trasero de la placa STM32F429 Discovery Kit. Tomada de <https://www.st.com/en/evaluation-tools/32f429idiscovery.html>

Es importante destacar que muchas de las funcionalidades que logra este kit se deben a la placa externa que contiene el microcontrolador. El microcontrolador en sí tiene buen rendimiento pero lo que se utiliza por ejemplo en este laboratorio se logra gracias a la placa. En el siguiente diagrama de bloques se muestra como se realizaron las conexiones de los componentes.

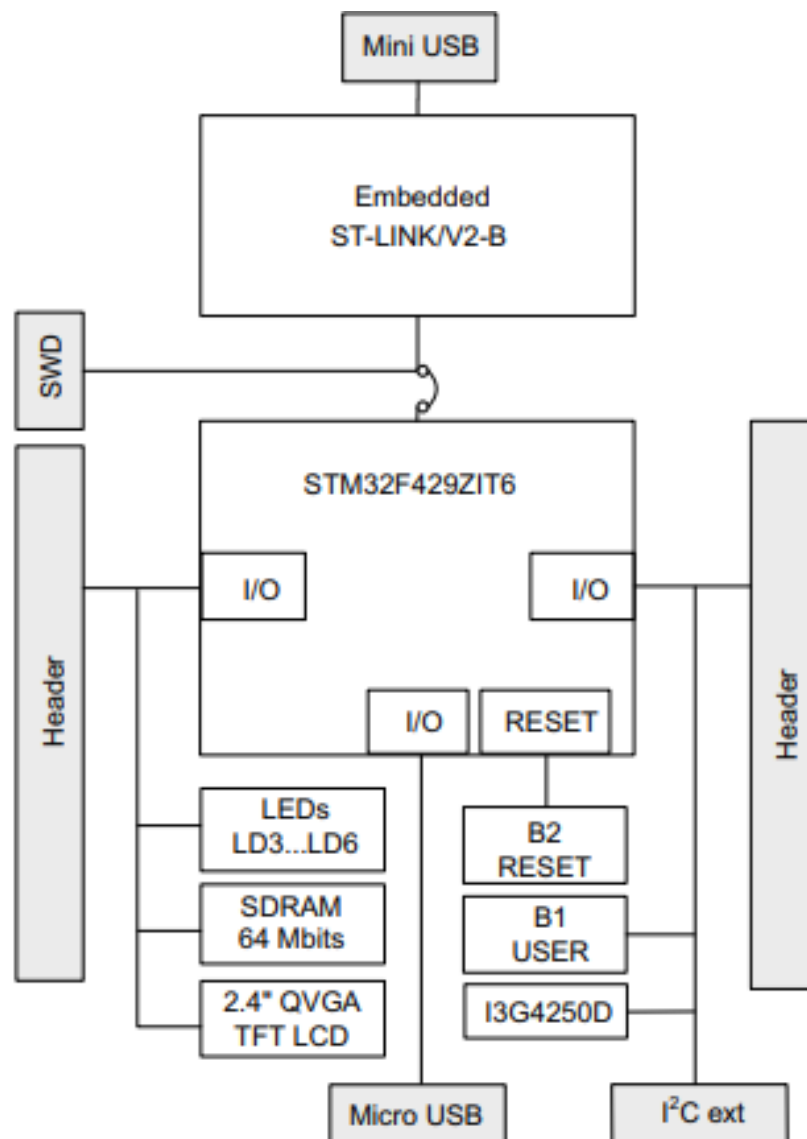


Figura 3: Diagrama de bloques de la placa. Tomada de <https://www.st.com/en/evaluation-tools/32f429idiscovery.html>

El giroscopio que contiene esta placa es un I3G4250D de bajo poder, el sensor detecta en tres ejes por medio de su elemento. Este dispositivo además contiene una interfaz IC que permite enviar los datos por medio de un protocolo I2C o SPI. Este kit lo controla por medio de la interfaz SPI. [1]

2.1.1. Características eléctricas

Debido a que esta es una placa que contiene todos los componentes, no se utilizaron las características eléctricas más allá del uso de una fuente de poder externo. Esta debe ser de 5V y existen dos diodos que protegen la placa de otros valores. Los pines de 5V y 3V entregan 5V y 3V respectivamente y el consumo de poder debe ser menor a 100 mA. [1]

2.2. Bibliotecas

A nivel de software, se utilizó la biblioteca `libopenm3`, la cual es una biblioteca de software libre hecha para la familia de microprocesadores STM. La biblioteca posee sub bibliotecas que facilitan el uso de cada uno de los elementos de la tarjeta. Además `libopenm3` posee una serie de ejemplos que tienen funcionalidades que se necesitaban para este proyecto, por lo que se tomaron algunas de estas como bibliotecas para el desarrollo del proyecto. A continuación un alista de las bibliotecas:

- **math.h**: para realizar operaciones matemáticas relacionadas a los ángulos del giroscopio.
- **rcc.h**: para configurar el reloj interno del microcontrolador.
- **gpio.h**: para controlar los elementos externos como los leds y botones.
- **math.h**: para realizar operaciones matemáticas relacionadas a los ángulos del giroscopio.
- **spi.h**: para la conexión del giroscopio.
- **adc.h**: para la conexión de la pantalla.
- **usart.h**: para el envío de la información a la computadora por medio del protocolo USART.

Por otro lado, para la creación del script de python que recibe los datos para enviarlos a Thingsboard se utilizaron otras bibliotecas dentro de python:

- **serial**: que permite leer los datos que se transmiten por medio de USART.
- **time**: para distinguir el tiempo con que se envían los datos.
- **json**: para crear la estructura de datos que se envía a ThingsBoard.
- **paho de mqtt**: permite la transmisión de datos de tamaño pequeño para enviar los datos a ThingsBoard.

2.3. Diseño

Debido a que para este proyecto se está utilizando solamente los componentes que incluye la placa, no se realizó un diseño físico. La única conexión que presenta la placa es hacia la fuente de poder.

La conexión de la batería se hace por medio del pin de 5V presente en la placa. Para eso se utiliza un adaptador de batería que cuenta con los cables positivo y negativo que se conectan respectivamente a la placa. La batería se conecta al adaptador.

2.4. Componentes electrónicos

Para la creación del sismógrafo se utilizó la siguiente lista de componentes. Esta lista contiene la placa, la cual es propiedad de la Escuela, sin embargo, si se quisiera comprar se comparte el precio en colones basado en el precio en dólares. Esto debido a que la placa es difícil de conseguir en el país y la manera de conseguirla sería obtenerla en Estados Unidos. Para los precios se toma en cuenta un tipo de cambio de 565 colones equivalente a 1 USD.

Componente	Precio (colones)
STM32F429 Discovery kit	20385
Adaptador de batería	848
Batería	4300
Precio total	25535

El precio total del sismógrafo sería 25535 colones.

3. Desarrollo/Análisis de resultados

Existe una serie de requerimientos que se pidieron para la creación del sismógrafo que se van a analizar a continuación:

1. Se leen los valores de cada uno de los ejes (X,Y,Z) del giroscopio.
2. Se utiliza un botón o switch para habilitar la comunicación por USART.
3. Se utiliza un LED para indicar que las comunicaciones por medio del protocolo USART estan activas.
4. Se debe leer el valor de la batería de 9V y debe enviar una notificación a ThingsBoard indicando si el valor de la batería es bajo.
5. Debe desplegar en la pantalla los valores del giroscopio, el nivel de batería y si la comunicación serial está habilitada.
6. Se utiliza un script de python para leer los valores enviados por USART y los enví a ThingsBoard.

A continuación se va a mostrar tanto la forma en que se implementó así como el resultado obtenido de cada uno de los requerimientos.

3.1. Lectura de los valores del giroscopio

Para obtener los valores del giroscopio, se utilizó como base el ejemplo spi presente en los ejemplos de la biblioteca libopenm3. Este ejemplo demuestra como utilizar la comunicación SPI. Una vez leídos los datos del giroscopio se despliegan en la consola de la placa.

Adicionalmente a esto se procesaron los datos para que se puedan desplegar en radianes y no en grados. En la siguiente imagen de muestra como estos datos, en conjunto con el nivel de batería son transmitidos a la computadora.

[illegible]

Figura 4: Transmisión de cada uno de los ejes del giroscopio. Imagen propia.

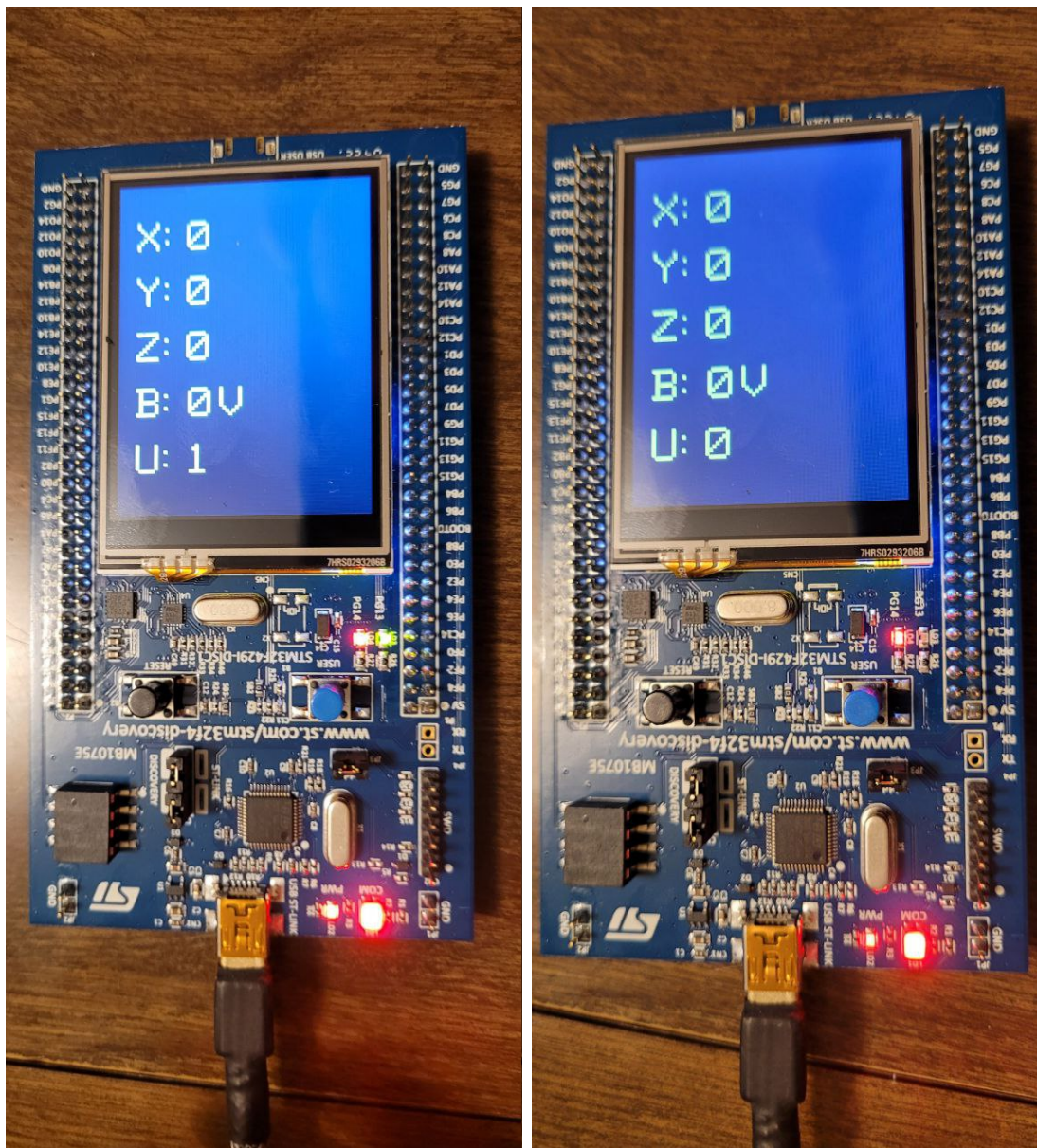
3.2. Transmisión de los datos por medio de USART

Para la implementación de esta parte se tomó como base el ejemplo usart de la biblioteca. Este ejemplo envía los datos a la computadora usando el baudrate 11520 y 8 bits para la transmisión de datos. Tomando como base lo que se tenía de la lectura del giroscopio, se modificaron los datos que se transmiten a ser las lecturas del giroscopio. Adicionalmente se agregó el nivel de la batería como la señal de batería baja. En la figura 4 se muestra como esos datos son recibidos por la computadora y se pudieron desplegar cada uno de los ejes.

3.3. Botón y LED asociados al protocolo USART

Para esta sección se utilizaron los ejemplos miniblink y button de la biblioteca. El botón escogido fue el de USER ya que el de RESET tiene la función de reiniciar el programa y se desea que esa función se siga cumpliendo. Se utilizó el LED verde correspondiente al pin PG13. Este link parpadea cuando está habilitada la comunicación. La variable **com_en** es modificada por el botón, si esta tiene un valor de 1 se da la transferencia y si tiene un valor de 0 no se da.

En la siguiente imagen se muestra como después de presionar el botón, se enciende el LED verde y se muestra en la pantalla que USART está habilitado y cuando se vuelve a presionar se quitan los dos.



(a) Transmisión USART habilitada.

(b) Transmisión USART deshabilitada.

Figura 5: Funcionamiento del botón. Figura propia.

3.4. Lectura del valor de la batería

Para la lectura del valor de la batería se basó en el ejemplo de la biblioteca. Para las siguientes demostraciones, se muestra el valor de la batería como 0V ya que no se pudo conseguir el adaptador de la batería a tiempo. Sin embargo, se muestran todas las funcionalidades.

En la figura 5 mostrada anteriormente se puede observar como se despliega el valor de la batería en la pantalla, y en la figura 4 cómo se pasó su valor, además de la señal para indicar que la batería está baja.

Esta información se transmitió por medio de USART hasta ThingsBoard donde se creó un Widget que permite mostrar la información de una manera gráfica. La siguiente imagen muestra el nivel de la batería así como la señal de batería baja.

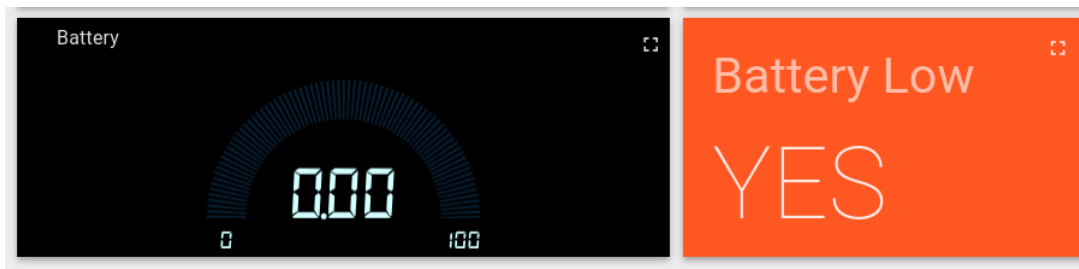


Figura 6: Widgets de ThingsBoard para batería. Imagen propia.

3.5. Utilización de la pantalla para mostrar datos

Para esta funcionalidad se basó en los ejemplos `ldc-serial` y `lcd-dma` de la biblioteca para hacerlo. Se tomó como base la funcionalidad de que cada vez que se inicializa el programa se muestra una pantalla de inicio por 2 s, una vez que han pasado se despliegan el resto de los datos. En la siguiente imagen se muestra este proceso.



(a) Pantalla de inicio.

(b) Muestra de datos.

Figura 7: Funcionamiento de la pantalla. Figura propia.

3.6. Script de python para enviar datos a ThingsBoard

Para ello se utilizó el tipo de dato mqtt que permite transmitir datos pequeños. La información se agrupó en objetos de Json para ser enviada. El script debe tener en cuenta el token de acceso a Thingsboard así como la dirección de conexión que en este caso corresponde a la de la Escuela. Los datos son recibidos como se muestran en la siguiente imagen:

Sismografo_Sofi

Device details

Details

Attributes

Latest telemetry

Alarms

Events

Relations

Audit Logs

Latest telemetry

<input type="checkbox"/>	Last update time	Key ↑	Value
<input type="checkbox"/>	2023-02-18 21:29:57	Battery Low	YES
<input type="checkbox"/>	2023-02-18 21:29:57	Battery Lvl	0
<input type="checkbox"/>	2023-02-18 21:29:57	x	-11
<input type="checkbox"/>	2023-02-18 21:29:57	y	52
<input type="checkbox"/>	2023-02-18 21:29:57	z	0

Figura 8: Recepción de datos en ThingsBoard. Imagen propia.

Estos datos una vez recibidos en ThingsBoard, fueron procesados en Widgets para que fueran visualmente más atractivos. Se utilizó el widget de gráfico ya que simula bastante bien un sismógrafo. Además se creó un widget que detecta si hay un temblor. La siguiente imagen muestra todos los widgets:

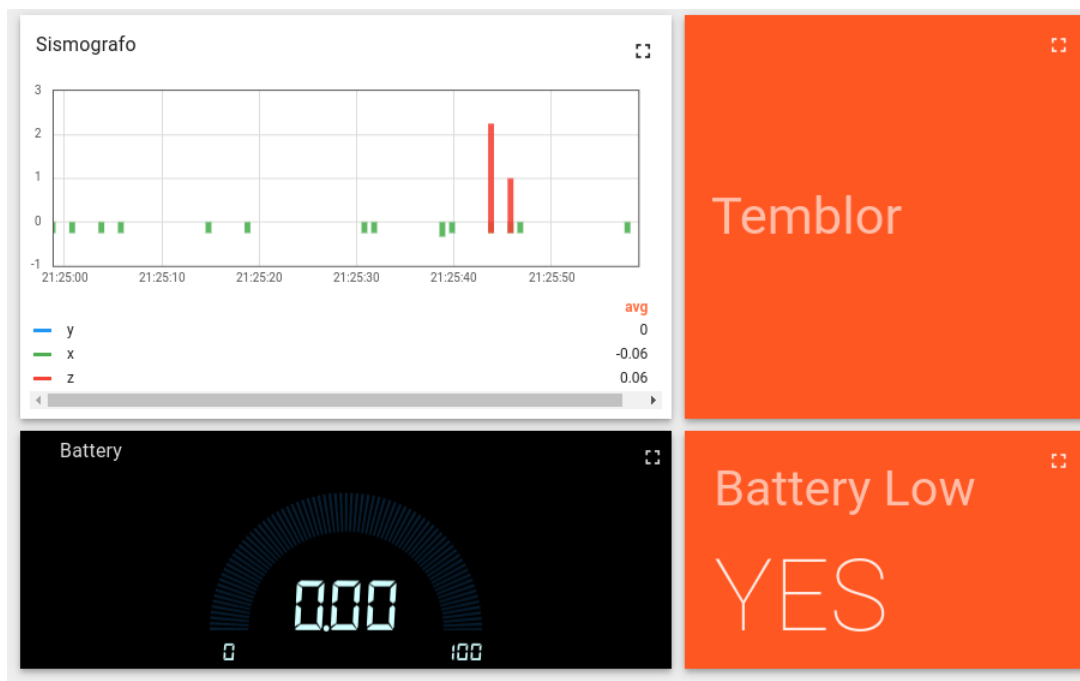


Figura 9: Dashboard en ThingsBoard. Imagen propia.

En la siguiente imagen se muestra además como al recibir una señal de que hubo movimiento se indica que hubo un temblor:

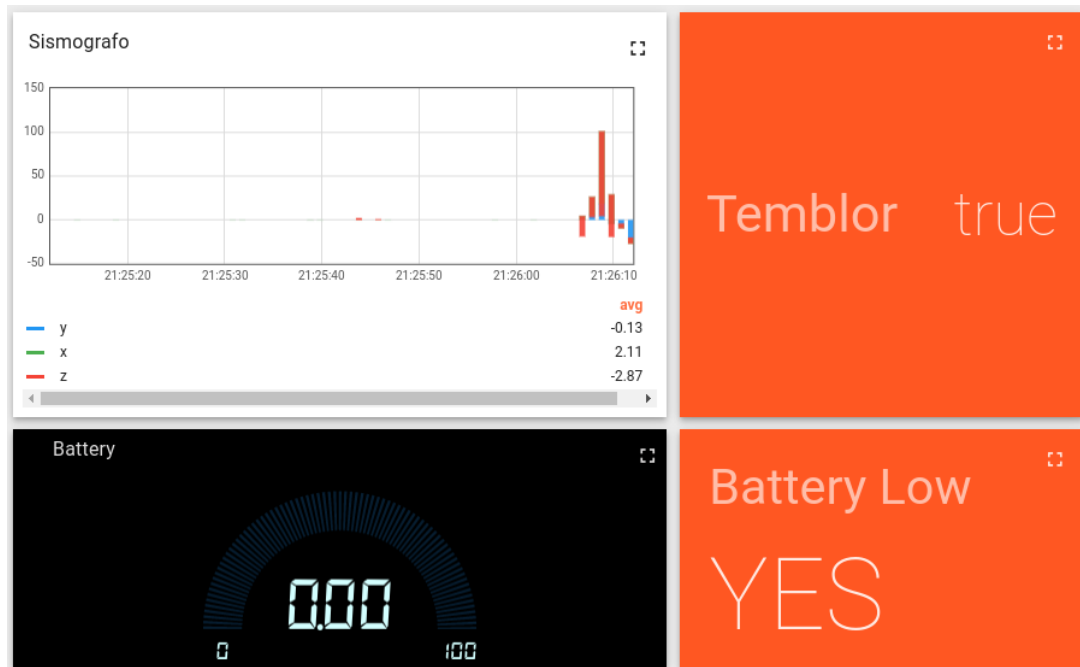


Figura 10: Dashboard en ThingsBoard con temblor. Imagen propia.

4. Conclusiones y recomendaciones

- Se recomienda empezar probando cada uno de los ejemplos de la biblioteca libopencm3 ya que son muy completos y muestran la mayoría de las funcionalidades del STM32f429.
- El uso de una placa física permitió aprender otras partes del uso de un microprocesador que no se habían podido ver previamente como el "flasheo" de la información.
- Entender los alcances del microcontrolador permite realizar una gran cantidad de proyectos que pueden disminuir los costos de materiales.
- Utilizar el internet en microcontroladores permite que los alcances de cada proyecto lleguen a más personas, así como pueden tener otras funcionalidades que no se lograrían de otra manera.
- En este proyecto se puede notar como el internet de las cosas ha evolucionado gracias a que las herramientas son sencillas de utilizar y ayudan a facilitar la vida cotidiana.

Referencias

1. ST Life Augmented (2019) Discovery kit with STM32F429ZI MCU. Recuperado de: <https://www.st.com/en/evaluation-tools/32f429idiscovery.html#documentation>
2. Libopencm3 (2023) Libopencm3 Examples. Recuperado de: <https://github.com/libopencm3/libopencm3-examples>
3. Riyo santo yosep. (2021) MQTT using Thingsboard. Recuperado de: <https://youtu.be/iEvu1kFHKUE>

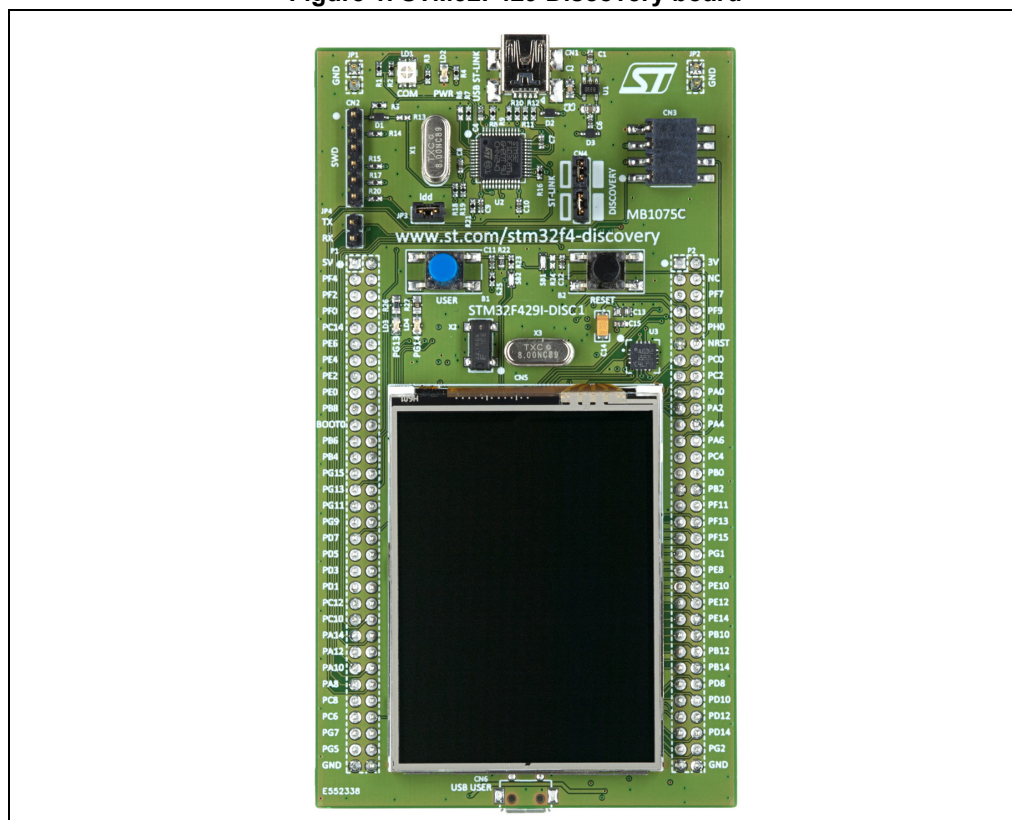
5. Apéndice

Introduction

The 32F429IDISCOVERY Discovery kit allows users to easily develop applications with the STMicroelectronics Arm® Cortex®-M4 core-based STM32F429 high-performance microcontroller. It includes an ST-LINK/V2-B embedded debug tool, a 2.4" QVGA TFT LCD, an external 64-Mbit SDRAM, an ST MEMS gyroscope, a USB OTG Micro-AB connector, LEDs and push-buttons.

The board comes with the STM32 comprehensive free software libraries and examples available with the STM32CubeF4 MCU Package, as well as direct access to the Arm® Mbed Enabled™ resources at the <http://mbed.org> website.

Figure 1. STM32F429 Discovery board



Picture is not contractual.



1 Features

- STM32F429ZIT6 microcontroller featuring 2 Mbytes of Flash memory, 256 Kbytes of RAM in an LQFP144 package
- 2.4" QVGA TFT LCD
- USB OTG with Micro-AB connector
- I3G4250D, ST MEMS motion sensor 3-axis digital output gyroscope
- Six LEDs:
 - LD1 (red/green) for USB communication
 - LD2 (red) for 3.3 V power-on
 - Two user LEDs: LD3 (green), LD4 (red)
 - Two USB OTG LEDs: LD5 (green) V_{BUS} and LD6 (red) OC (over-current)
- Two push-buttons (user and reset)
- 64-Mbit SDRAM
- Extension header for LQFP144 I/Os for a quick connection to the prototyping board and an easy probing
- On-board ST-LINK/V2-B
- USB functions:
 - Debug port
 - Virtual COM port
 - Mass storage
- Mbed Enabled™ (see <http://mbed.org>)
- Board power supply: through the USB bus or from an external 5 V supply voltage
- Comprehensive free software including a variety of examples, part of STM32CubeF4 MCU Package or STSW-STM32138, for using legacy standard libraries

6 Hardware layout

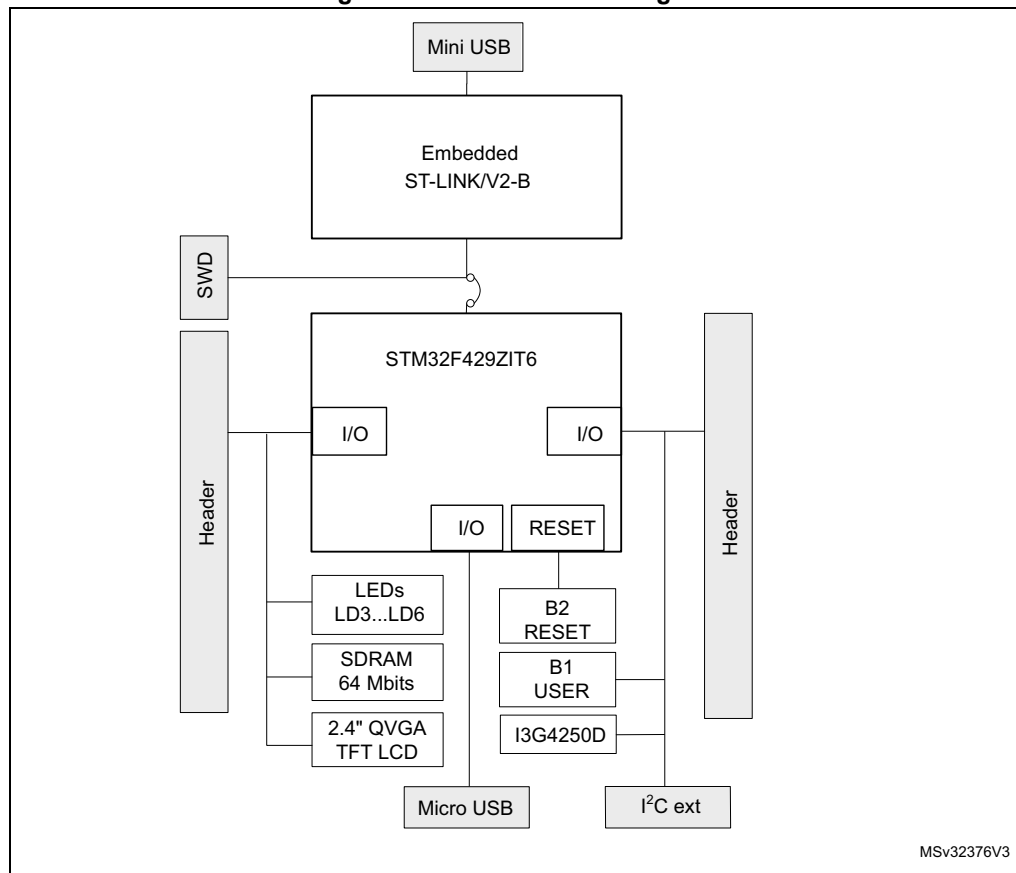
The STM32F429 Discovery board has been designed around the STM32F429ZIT6 microcontroller in a 144-pin LQFP package.

Figure 2 illustrates the connections between the STM32F429ZIT6 and its peripherals (ST-LINK/V2-B, push-buttons, LEDs, USB OTG, ST-MEMS gyroscope, accelerometer, magnetometer, and connectors).

Figure 3 and *Figure 4* show the location of these features on the STM32F429 Discovery board.

Figure 5 shows the mechanical dimensions of the STM32F429 Discovery board.

Figure 2. Hardware block diagram



6.1 STM32F429 Discovery board layout

Figure 3. Top layout

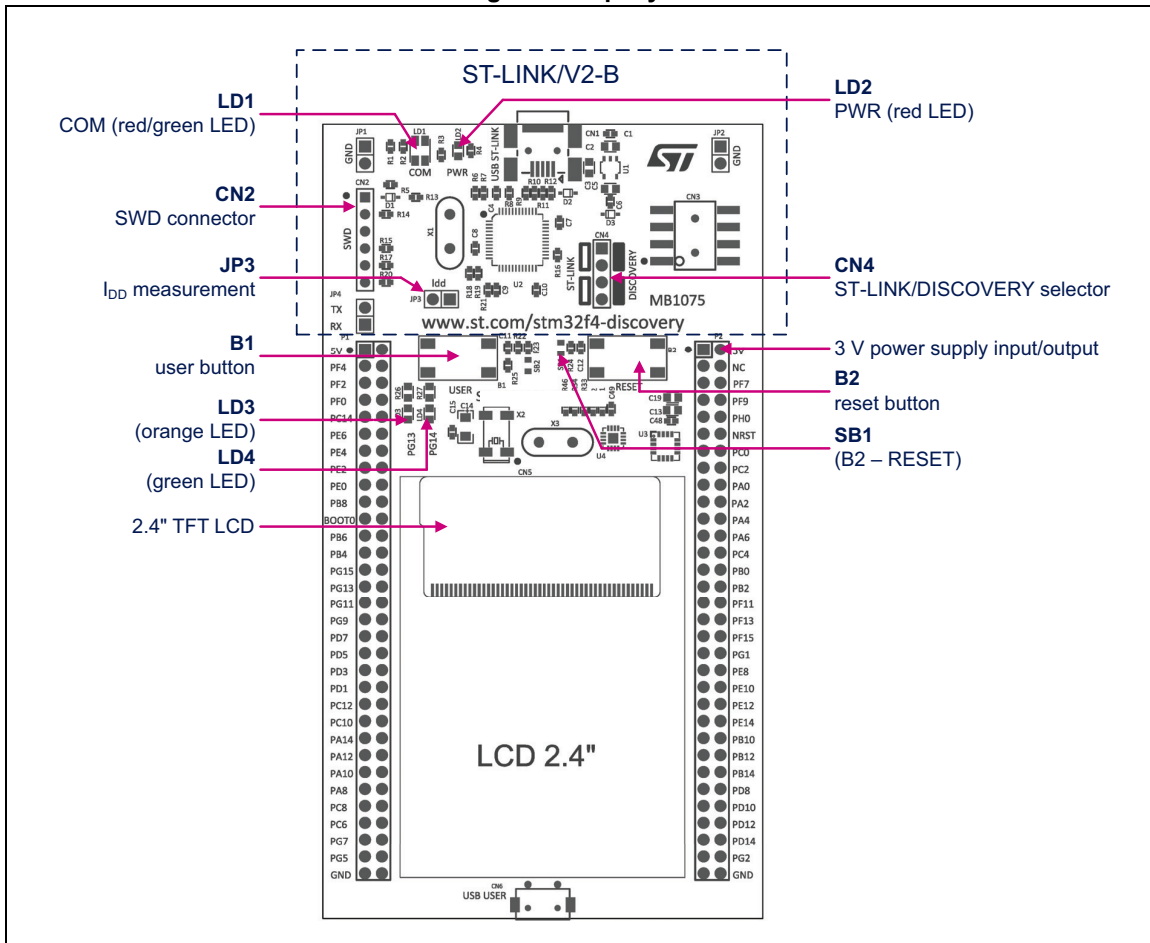
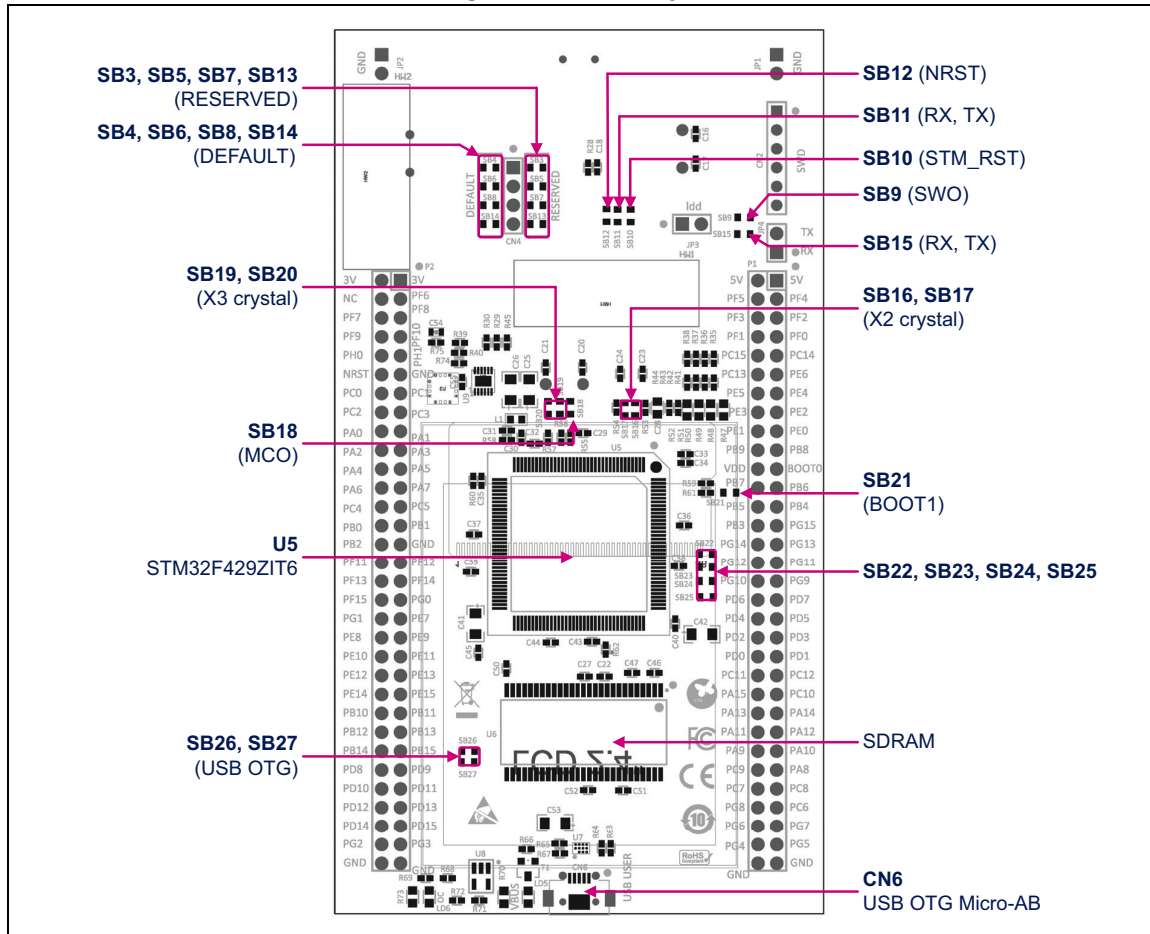
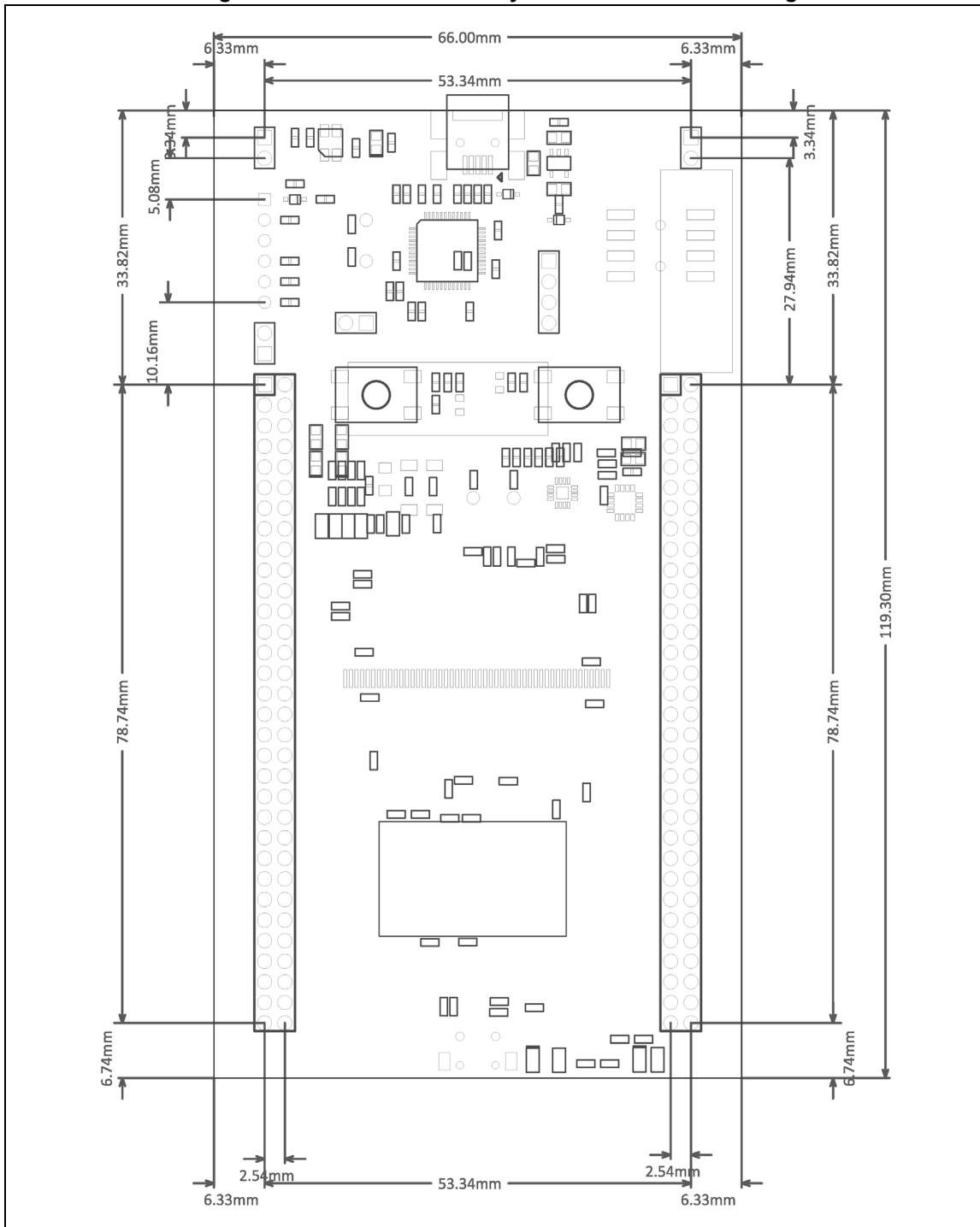


Figure 4. Bottom layout



6.2 Mechanical drawing

Figure 5. STM32F429 Discovery board mechanical drawing



6.4 Power supply and power selection

The power supply is provided either by the host PC through the USB cable or by an external 5 V power supply.

The D2 and D3 diodes protect the 5 V and 3 V pins from external power supplies:

- 5 V and 3 V can be used as output power supplies when another application board is connected to pins P1 and P2.
In this case, the 5 V and 3 V pins deliver a 5 V or 3 V power supply and the power consumption must be lower than 100 mA.
- 5 V and 3 V can also be used as input power supply, e.g. when the USB connectors are not connected to the PC.
In this case, the STM32F429 Discovery board must be powered by a power supply unit or by auxiliary equipment complying with the standard EN-60950-1: 2006+A11/2009, and must be Safety Extra Low Voltage (SELV) with limited power capability.

6.5 LEDs

- LD1 COM:
The LD1 default status is red. LD1 turns to green to indicate that communications are in progress between the PC and the ST-LINK/V2-B.
- LD2 PWR:
The red LED indicates that the board is powered.
- User LD3:
The green LED is a user LED connected to the I/O PG13 of the STM32F429ZIT6.
- User LD4:
The red LED is a user LED connected to the I/O PG14 of the STM32F429ZIT6.
- User LD5:
The green LED indicates when VBUS is present on CN6 and is connected to PB13 of the STM32F429ZIT6.
- User LD6:
The red LED indicates an overcurrent from VBUS of CN6 and is connected to the I/O PC5 of the STM32F429ZIT6.

6.6 Push-buttons

- B1 USER:
User and Wake-Up button connected to the I/O PA0 of the STM32F429ZIT6.
- B2 RESET:
The push-button connected to NRST is used to RESET the STM32F429ZIT6.

6.7 USB OTG supported

The STM32F429ZIT6 drives USB OTG High-Speed through its internal PHY, which limits it to USB OTG Full Speed on this board. The USB Micro-AB connector (CN6) allows the user to connect a host or device component, such as a USB key, mouse, or other.