

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

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

Prof. MSc. Marco Villalta Fallas

Andrés Artavia Solano - B90789
Mario Navarro Bejarano - B75398

Índice

1. Introducción	1
2. Nota Teórica	2
2.1. Componentes empleados	2
2.2. STM32F429ZI	2
2.3. Giroscopio y pantalla LCD	3
2.4. ThingsBoard	3
3. Desarrollo	5
3.1. Lectura de Giroscopio	5
3.2. LED's, botón y ADC	6
3.3. Nivel de batería	6
3.4. Dashboard en ThingsBoard	7
4. Análisis de los resultados	9
5. Conclusiones	11
6. Apéndice	13

1. Introducción

En este laboratorio se busca desarrollar un sismógrafo básico utilizando la placa STM32F429 Discovery Kit y la biblioteca libopencm3, aprovechando la capacidad del giroscopio para detectar movimientos en los ejes X, Y, Z. Este proyecto representa una aplicación práctica del uso de microcontroladores en sistemas de monitoreo sísmico, permitiendo adquirir habilidades en la integración de sensores, manejo de periféricos y transmisión de datos en un entorno de Internet de las Cosas (IoT). El diseño del sismógrafo se enfoca en capturar las variaciones de los ejes del giroscopio, indicando potenciales eventos sísmicos. A través de un sistema de comunicación USART/USB, se habilitará la transmisión de los datos a una computadora para posteriormente transmitir los datos a una plataforma IoT. Además, se integrará un sistema de monitoreo del nivel de batería, crucial para asegurar la operación continua del dispositivo. El principal objetivo de este laboratorio es diseñar e implementar un sistema de monitoreo sísmico básico utilizando la placa STM32F429 Discovery Kit, con la finalidad de capturar movimientos en los ejes X, Y, Z mediante el giroscopio incorporado, simulando las funciones de un sismógrafo. Además, se busca proporcionar al usuario la capacidad de habilitar o deshabilitar la transmisión de datos mediante un switch físico, permitiendo controlar la comunicación serial a conveniencia. El sistema incluirá un LED indicador, que parpadeará cuando la transmisión USART esté habilitada. También se implementará un mecanismo de monitoreo del nivel de batería, en caso de que la tensión descienda por debajo del límite de operación seguro (7V), se encenderá un LED de alarma y se enviará una notificación de batería baja al dashboard IoT en ThingsBoard. También se planea desplegar en la pantalla LCD los valores del giroscopio en los tres ejes, el nivel de batería, y el estado de la comunicación USART/USB. Finalmente, se desarrollará un script en Python que permitirá transmitir la información del giroscopio y el estado de la batería al dashboard de ThingsBoard, donde los datos podrán ser visualizados de forma intuitiva mediante widgets.

Nota: El código fuente se puede encontrar en la rama “main” del repositorio, en la carpeta “lab4”. <https://github.com/artavias/IE0624/tree/main>

2. Nota Teórica

A continuación se muestra un diagrama del diseño general de la placa:

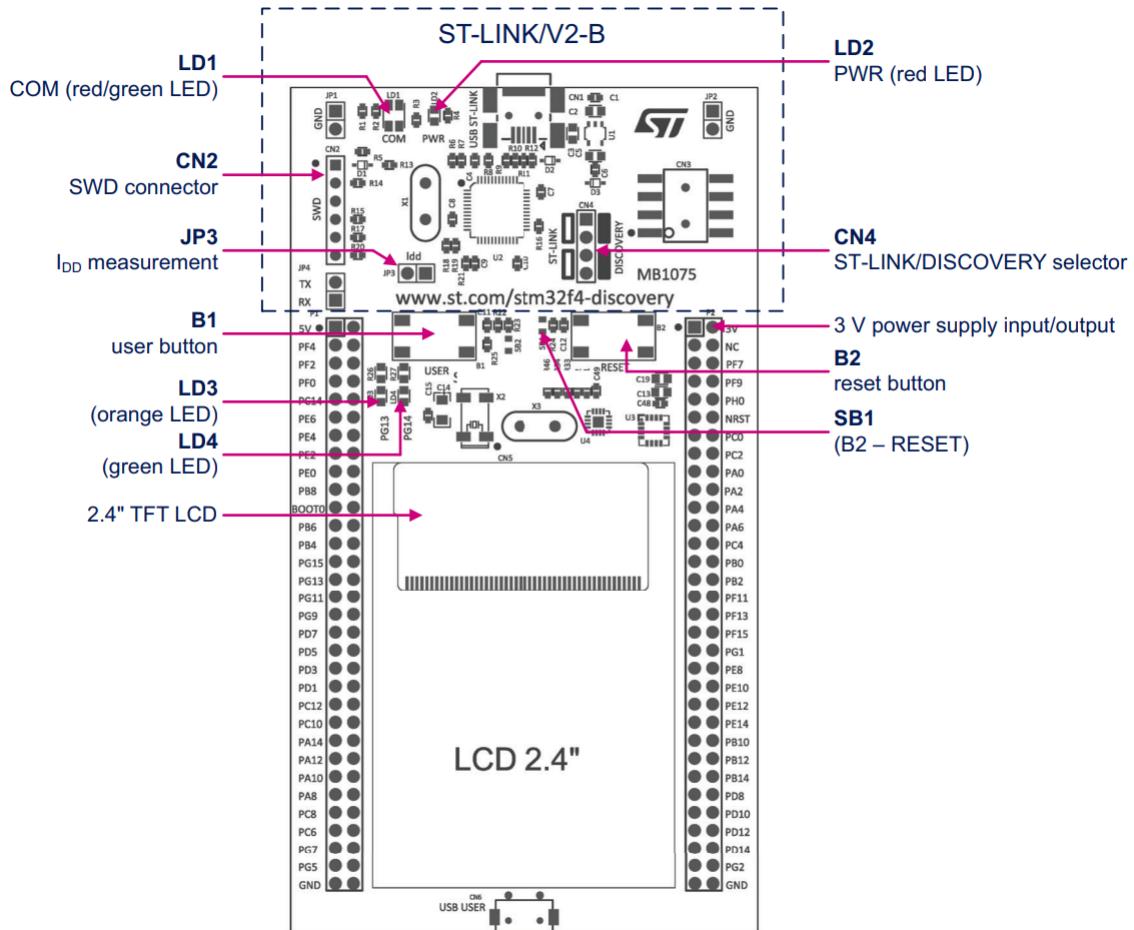


Figura 1: Diseño general del STM32F429i Discovery Kit [1]

2.1. Componentes empleados.

La siguiente tabla detalla los componentes y la cantidad utilizada.

Componente	Cantidad
STM32F429i Discovery Kit	1
Bateria 9V	1
Resistencia 10kΩ	2
Resistencia 2.2kΩ	1

2.2. STM32F429ZI

Los siguientes datos de la placa STM32F429 Discovery kit fueron tomados de la hoja de datos oficial del fabricante [1]. La placa STM32F429 Discovery kit se basa en el microcontrolador STM32F429ZI, el cual es un microcontrolador de alto rendimiento de la serie STM32F4 desarrollado por STMicroelectronics, basado en el núcleo ARM Cortex-M4. Este procesador está diseñado para aplicaciones de tiempo real que requieren cálculos intensivos y alto control sobre periféricos. A continuación, se detalla su arquitectura, características clave del procesador, y el uso de registros. El procesador ARM Cortex-M4 integrado en el STM32F429ZI opera a

una frecuencia de 180 MHz. Este procesador incorpora una Unidad de Punto Flotante (FPU) compatible con el estándar IEEE 754, lo que permite realizar operaciones de punto flotante de forma rápida y eficiente. Además, incluye una caché de instrucciones y datos (I/D), que optimiza la velocidad de acceso a memoria y minimiza las latencias durante la ejecución del código. El Cortex-M4 soporta procesamiento digital de señales (DSP), incorporando instrucciones SIMD (Single Instruction Multiple Data), que mejoran el rendimiento en algoritmos matemáticos, como filtros y procesamiento de señales, aumentando la eficiencia del sistema. Ofrece un sistema de gestión de interrupciones capaz de manejar hasta 240 interrupciones con 16 niveles de prioridad. Estas interrupciones son gestionadas por el NVIC (Nested Vectored Interrupt Controller), asegurando respuestas rápidas y eficientes a eventos de hardware software críticos. El microcontrolador dispone de 2 MB de memoria Flash interna, utilizada para almacenar el firmware y las aplicaciones críticas. Esta memoria permite múltiples ciclos de lectura y escritura, garantizando la estabilidad del sistema durante las actualizaciones. Además, cuenta con 256 KB de SRAM interna para la ejecución de tareas en tiempo real. Para aplicaciones que requieren un manejo intensivo de datos, como el procesamiento de señales, el sistema puede hacer uso de una memoria externa SDRAM de 64 Mbits, útil para almacenamiento temporal y creación de buffers. El STM32F429ZI ofrece varios puertos GPIO (General Purpose I/O), los cuales se configuran a través de registros específicos para definir su modo de operación, ya sea como entrada, salida, función alternativa o analógica. El microcontrolador dispone de hasta 14 temporizadores, algunos de ellos con funciones avanzadas como la generación de PWM (Pulse Width Modulation). Cada temporizador se gestiona mediante los registros de control de temporizadores.

2.3. Giroscopio y pantalla LCD

La interfaz SPI permite la comunicación entre el microcontrolador y el giroscopio MEMS I3G4250D. El giroscopio MEMS I3G4250D integrado en la placa es un sensor de movimiento de tres ejes que permite medir la velocidad angular en rangos ajustables de ± 245 , ± 500 o ± 2000 grados por segundo (dps), lo que lo hace adecuado para aplicaciones de monitoreo de vibraciones y detección de movimientos sísmicos. Este giroscopio cuenta con una interfaz de salida digital compatible con SPI e I2C, lo que facilita su integración con el microcontrolador y permite la captura eficiente de datos en tiempo real. Además, el I3G4250D incluye un filtro pasa bajos configurable que mejora la precisión en la medición de cambios de orientación y detecta variaciones sutiles en el movimiento en los ejes X, Y y Z. La pantalla LCD TFT integrada es de 2.4 pulgadas con una resolución QVGA de 240 x 320 píxeles y es capaz de mostrar hasta 262,000 colores. Es controlada mediante un controlador ILI9341 a través de un protocolo RGB, esta pantalla facilita la visualización de información como el nivel de batería, los datos de los ejes del giroscopio y el estado de la comunicación USART/USB. Además, su bajo consumo de energía y compatibilidad con voltajes típicos de 2.8 V la hacen ideal para aplicaciones en sistemas portátiles.

2.4. ThingsBoard

ThingsBoard es una plataforma de Internet de las Cosas (IoT) de código abierto que permite recopilar, procesar y visualizar datos en tiempo real desde dispositivos y sensores conectados. Facilita la administración remota y el monitoreo de dispositivos IoT, ofreciendo herramientas avanzadas para crear visualizaciones interactivas, analizar datos y generar alertas según condiciones específicas. Entre sus características principales se encuentra la gestión de dispositivos, que permite registrar y controlar dispositivos IoT de manera eficiente. Cada dispositivo puede configurarse para enviar datos en intervalos regulares o cuando ocurren eventos específicos.

cos, utilizando protocolos de comunicación como MQTT, HTTP y CoAP, lo cual permite una fácil integración con diversos dispositivos. ThingsBoard también permite el procesamiento y almacenamiento de datos a gran escala en tiempo real, aplicando reglas de procesamiento para ejecutar acciones basadas en eventos, como activar alertas o enviar datos a servicios en la nube cuando se cumplen condiciones específicas. Permite crear dashboards personalizables con widgets interactivos como gráficos de líneas, medidores, tablas y mapas, facilitando el análisis y la monitorización de datos en tiempo real. La plataforma permite configurar alertas y notificaciones basadas en reglas definidas por el usuario, ideal para aplicaciones que requieren alertas inmediatas en situaciones críticas, como baja batería o detección de actividad sísmica. Estas notificaciones pueden enviarse por correo electrónico o integrarse con otros sistemas de mensajería en tiempo real. Diseñada para aplicaciones de cualquier escala, ThingsBoard es altamente escalable y cuenta con características de seguridad avanzadas que incluyen autenticación de dispositivos, control de acceso basado en roles y cifrado de datos. [2] En este laboratorio, ThingsBoard se utilizará para recibir y visualizar datos del giroscopio y el nivel de batería de la placa STM32F429 Discovery Kit. Un script en Python enviará estos datos a la plataforma, representándolos en un dashboard interactivo para observar en tiempo real las variaciones de movimiento y el estado de la energía. Además, las alertas configuradas en ThingsBoard enviarán notificaciones en caso de batería baja, garantizando una respuesta rápida ante condiciones críticas.

3. Desarrollo

3.1. Lectura de Giroscopio

El primer punto que se debe resolver es realizar la lectura de los valores en los ejes X, Y y Z del giroscopio. Para realizar esta sección, se utilizó parte del código que se proporciona por los ejemplos de libopemcm3, esto para funciones no relevantes para los objetivos de este laboratorio, como configurar el reloj del sistema por ejemplo.

En la figura 2 se muestra el código utilizado para configurar los pines y el SPI5, estos son los elementos necesarios para comunicarse con el giroscopio. Primero se configuran los puertos GPIO, como salidas, así como las frecuencias, luego se configuran los registros SPI5 para habilitarlo a funcionar en modo maestro, dadas los parámetros de funcionamiento de la interfaz SPI5 del giroscopio. Es importante mencionar que el orden en el que se llaman estas funciones es el orden que se indica en las diapositivas del curso.

```
83 // Configurar pines para Giroscopio
84 rcc_periph_clock_enable(RCC_GPIOC | RCC_GPIOF);
85 rcc_periph_clock_enable(RCC_SPI5);
86 gpio_mode_setup(GPIOF, GPIO_MODE_AF, GPIO_PUPD_NONE, GPIO7 | GPIO8 | GPIO9);
87 gpio_set_af(GPIOF, GPIO_AF5, GPIO7 | GPIO8 | GPIO9);
88 gpio_set_output_options(GPIOF, GPIO_OTYPE_PP, GPIO_OSPEED_25MHZ,
89                         GPIO7 | GPIO9);
90 gpio_mode_setup(GPIOC, GPIO_MODE_OUTPUT, GPIO_PUPD_NONE, GPIO1);
91 gpio_set(GPIOC, GPIO1);
92
93 // Configurar SPI5 para giroscopio
94 spi_set_master_mode(SPI5);
95 spi_set_baudrate_prescaler(SPI5, SPI_CR1_BAUDRATE_FPCLK_DIV_8);
96 spi_set_clock_polarity_0(SPI5);
97 spi_set_clock_phase_0(SPI5);
98 spi_set_full_duplex_mode(SPI5);
99 spi_set_unidirectional_mode(SPI5);
100 spi_enable_software_slave_management(SPI5);
101 spi_send_msb_first(SPI5);
102 spi_set_nss_high(SPI5);
103 spi_enable(SPI5);
104 spi_set_dff_8bit(SPI5);
105 spi_enable_ss_output(SPI5);
```

Figura 2: Código para configurar comunicación con SPI5

Luego de realizar estas configuraciones, se pueden configurar los registros del giroscopio y proceder a leer los registros correspondientes a los valores X, Y y Z, tal como se muestra en la figura 3

```
155 // Configurar registros de giroscopio
156 write_reg(0x20, 0xcf);
157 write_reg(0x21, 0x07);
158 write_reg(0x23, 0xb0);
159
160 valor_X = read_reg(0x28);
161 valor_Y = read_reg(0x2A);
162 valor_Z = read_reg(0x2C);
```

Figura 3: Escritura y lectura de registros del giroscopio.

Nota importante: Se realizaron múltiples intentos de establecer la comunicación con el giroscopio, desde modificaciones en la configuración de SPI5, simular las comunicaciones SPI5 manualmente, probar ejemplos proporcionados hasta probar código proporcionado por compañeros del curso a los cuales sí les funcionaba, sin embargo, no fue posible obtener datos fuera

de un constante valor de 0, por ello se llegó a la conclusión de que puede existir algún daño físico en la placa, lo que impide la comunicación con el giroscopio, o que el daño existe en sí en el giroscopio.

Para compensar esta carencia, se comentó esta sección del código (en teoría el código sigue siendo funcional y se podría utilizar en una placa en buen estado) y se agregaron unos contadores que simulan el cambio en los valores que proporcionaría el giroscopio. Más adelante en la sección de resultados se mostrará este funcionamiento.

3.2. LED's, botón y ADC

Para realizar la configuración de los leds que se usarán para indicar que la tensión de la batería está baja, que la comunicación USART está activada, del botón para habilitar esta comunicación y del convertidor analógico digital, se realizó un proceso de configuración muy similar al que se utilizó para configurar la comunicación con el giroscopio, con la diferencia del cambio de los pines. Así que para evitar extender este reporte, se omite agregar esta sección del código.

Fuera de la configuración, en la figura 4 se muestra el código con la lógica para controlar estas funciones. En las líneas 192-164 se muestra la lógica para encender el led cuando se presiona el botón, esta información se utiliza también para habilitar la comunicación USART. En las líneas 165-167 se muestra el código que se encarga de leer la información sobre la tensión de la batería. En este caso esta información muestra la tensión en mili-volts, para aprovechar la precisión del ADC. Finalmente en las líneas 169-177 se muestra la lógica que se utiliza para encender y apagar el led de color rojo, esta función se activa cuando la tensión es inferior a 7.5V, por lo que se acerca a la tensión de peligro de 7V.

```

161 // Leer botón para ver si hay que activar la comunicación USART
162 if(gpio_get(GPIOA, GPIO0)){
163     gpio_toggle(GPIOG, GPIO13);
164 }
165 //Leer datos de adc
166 bateria = read_adc_naiive(1)*9000/4095;
167 sprintf(str_adc, "%u", bateria);
168 //Encender led rojo si bateria < 7500
169 if (bateria < 7500){
170     if (counter2==4){
171         gpio_toggle(GPIOG, GPIO14);
172         counter2=0;
173     }
174     if (bateria>7500){
175         gpio_clear(GPIOG, GPIO14);
176     }
177     counter2++;

```

Figura 4: Código para controlar leds y comunicación.

3.3. Nivel de batería

Para poder medir el nivel de la batería de 9V se puede utilizar un ADC de 12 bits incluido en la placa, pero primero es necesario condicionar la tensión de la batería a un rango seguro para la placa de 5V. Para esto se construyó un divisor de tensión utilizando las resistencias disponibles en la bodega, se optó por utilizar dos resistencias de $10\text{k}\Omega$ y una de $2.2\text{k}\Omega$. El siguiente diagrama muestra el circuito construido:

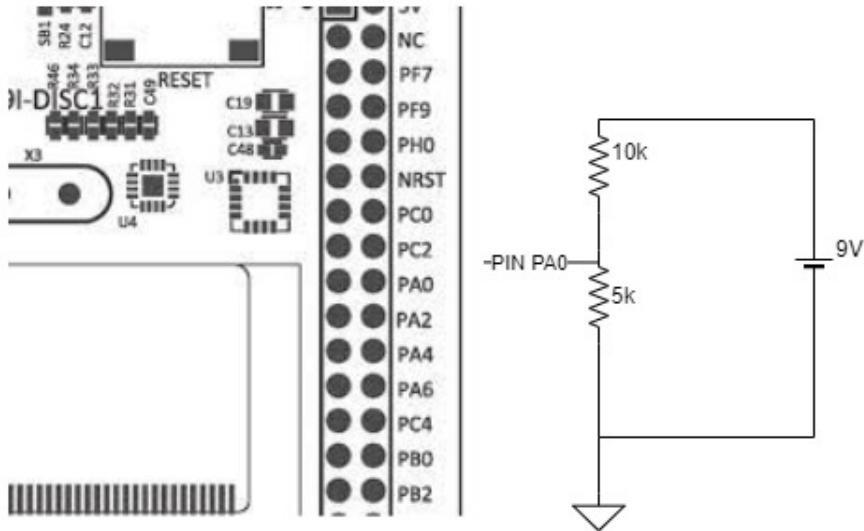


Figura 5: Esquemático del divisor de tensión. Diagrama de la placa tomado de [1]

Como se muestra en la figura 5 la tensión de la batería pasaría a condicionarse a un nivel en donde la tensión máxima, es decir cuando la batería se encuentre a 9V, viene dado por:

$$V_{out} = 9 \cdot \frac{5k}{10k + 5k} = 3V \quad (1)$$

De esta manera podemos conectar el pin PA0 al divisor de tensión de manera segura sin sobrepasar los límites de la placa.

Ahora utilizando el ADC de 12 bits de la placa se configura para leer el nivel de tensión del pin PA1 y se obtiene un valor periódicamente que va de 0 a 4095. Este rango del ADC corresponde a un rango de tensión en el pin de 0 a 3.3V pero como calculamos anteriormente, el máximo valor de tensión que puede alcanzar a medirse en el pin es de 3V, gracias al circuito de acondicionamiento. Si realizamos un cálculo rápido podemos ver que este valor corresponde a 3723 en el rango del ADC:

$$3V \cdot \frac{4095}{5} = 3722,72 \quad (2)$$

El ADC en este caso nos daría mediciones en un rango de 0 a 3723, tomando esto en cuenta se puede mapear el valor a un rango de 0 a 9V nuevamente para obtener finalmente el nivel de tensión de la batería. Para mapear el valor a este nuevo rango simplemente se multiplica por 9 y se divide entre 3723 el valor obtenido del ADC.

En nuestro caso no fue posible encontrar un adaptador adecuado para conectar la batería y realizar el circuito, pero el código necesario para medir la tensión de la batería utilizando el divisor de tensión diseñado se puede encontrar en el repositorio del laboratorio.

3.4. Dashboard en ThingsBoard

El primer paso para enviar los datos al dashboard de ThingsBoard es leer los datos enviados por la placa, para esto se establece conexión creando un objeto de tipo `Serial()` con el puerto serial que en este caso corresponde a ‘COM14’ con una tasa de 115200 baudios. Para enviar los datos al dashboard de ThingsBoard primero fue necesario agregar un dispositivo y generar un token necesario para realizar la conexión al host. Luego es necesario crear un objeto de tipo ‘`mqtt.Client()`’ y especificarle el host y el puerto para iniciar un loop infinito en donde se leen los datos del puerto serial y se ‘parsean’, es decir que se ordenan y se extraen los datos de los ejes, así como el nivel de la batería. Estos datos se utilizan para formar un diccionario el cual se envía como un objeto de tipo ‘`json`’ mediante la función ‘`publish`’.

A continuación se muestra el código utilizado:

```
5 # Se debe configurar dependiendo de cada puerto
6 puerto_serial = serial.Serial('COM14', 115200, timeout=1)
7
8 datos = 0
9
10 """ Este codigo se puede usar para guardar
11 datos si se requiere un registro
12 with open('reg_datos.csv', mode='a', newline='') as reg_datos:
13
14 writer = csv.writer(reg_datos)
15 writer.writerow(datos)
16 """
17
18
19 def enviar_TB():
20     valor_X = 0
21     valor_Y = 0
22     valor_Z = 0
23     bateria = 0
24     alarma = 0
25     token = "eiCEzMR90IHO0grSHBia"
26     host = "iot.eie.ucr.ac.cr"
27     puerto = 1883
28     TOPIC = "v1/devices/me/telemetry"
29     cliente = mqtt.Client()
30     cliente.username_pw_set(token)
31     cliente.connect(host, puerto, 60)
32     cliente.loop_start()
33     while True:
34         datos = puerto_serial.readline().decode('utf-8').rstrip()
35         print(datos)
36
37         datos = datos.split(": ")
38
39         if (datos[0] == "Valor X"):
40             valor_X = int(datos[1])
41         if (datos[0] == "\rValor Y"):
42             valor_Y = int(datos[1])
43         if (datos[0] == "\rValor Z"):
44             valor_Z = int(datos[1])
45         if (datos[0] == "\rBateria"):
46             bateria = float(datos[1])/1000
47         if (bateria < 7500):
48             alarma = 1
49         if (bateria > 7500):
50             alarma = 0
51         data = {
52             "nivelBateria": bateria,
53             "valorX": valor_X,
54             "valorY": valor_Y,
55             "valorZ": valor_Z,
56             "alarm": alarma
57         }
58         cliente.publish(TOPIC, json.dumps(data))
59
```

Figura 6: Script para leer y enviar datos a ThingsBoard

4. Análisis de los resultados

A continuación se muestran capturas del funcionamiento del código implementado.

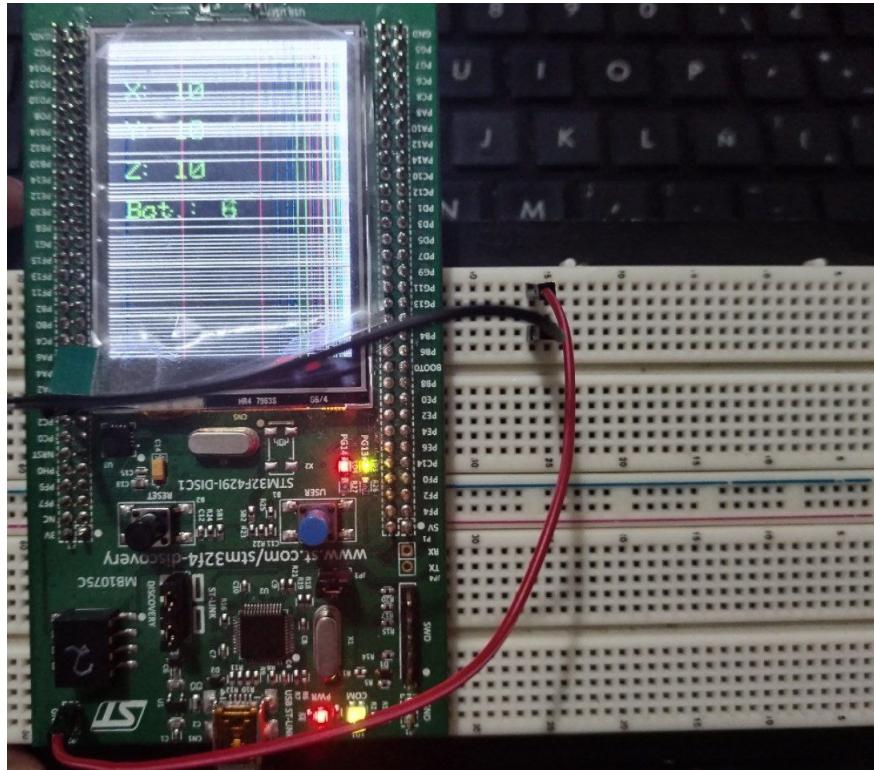


Figura 7: Muestra de información en pantalla.

En la figura 7 se muestra el funcionamiento de la placa, se pueden observar las siguientes características:

- En pantalla se muestra el valor del giroscopio (simulado) en los tres ejes. No es claramente visible debido al daño que presenta la pantalla.
- Se muestra encendido el led verde, este indica que la comunicación USART se encuentra activa. Cuando se presiona el botón la comunicación se desactiva y el led se apaga.
- Se puede observar el led rojo encendido. Este led parpadea indicando que la batería tiene un nivel de tensión inferior a 7.5V, esto también se puede observar en la pantalla, que indica una medida de 5mV, casi cero. Esto porque el pin ADC se conectó con jumpers al pin GND. Si este pin se conecta a tensión, el valor en pantalla indica 9000mV, es decir, indica que la batería está cargada y por lo tanto el led rojo deja de parpadear.

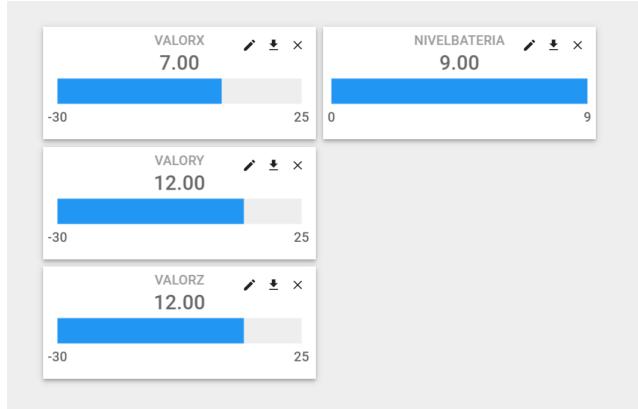


Figura 8: Muestra del dashboard en thingsboard al que se conecta la placa

En la figura 8 se muestra el dashboard de thingsboard. Se puede observar en la columna de la izquierda que se muestran los valores de los 3 ejes. En este caso se están simulando valores de 7, 12 y 12 para X, Y y Z respectivamente. A la derecha se observa el indicador de la batería, en este caso el ADC se conectó a tensión máxima, por lo que se indica una tensión de 9V.

5. Conclusiones

En este laboratorio, el objetivo era implementar un sistema de monitoreo basado en la lectura de datos del giroscopio en la placa STM32F429 Discovery Kit y su transmisión a la plataforma IoT ThingsBoard. Sin embargo, debido a un fallo en el giroscopio, no fue posible obtener datos reales de los ejes X, Y y Z. Para completar el proyecto, se optó por simular los datos del giroscopio, lo que permitió probar la funcionalidad de adquisición, procesamiento y envío de información a ThingsBoard, cumpliendo así los objetivos de transmisión y visualización en tiempo real.

A pesar de la limitación técnica con el sensor, la simulación de los datos fue útil para verificar el funcionamiento del sistema de comunicación y el correcto envío de la información al dashboard de ThingsBoard. Esta simulación permitió observar en tiempo real cómo los valores del giroscopio (aunque simulados) se desplegaban en la plataforma, confirmando que el sistema era capaz de transmitir datos de manera eficiente y que el dashboard estaba correctamente configurado para recibir y visualizar la información.

La transmisión de datos vía USART se configuró para enviar la información simulada del giroscopio y el estado de la batería. La comunicación mediante USART fue exitosa, ya que el sistema transmitió datos a ThingsBoard en tiempo real, donde se visualizaron correctamente en el dashboard. Adicionalmente, se implementó un control de habilitación mediante un switch físico, lo cual facilitó la activación y desactivación de la transmisión.

Para mejorar futuros laboratorios, se recomienda comenzar con una verificación previa de los componentes, asegurando que el giroscopio y otros sensores funcionen adecuadamente antes de iniciar la implementación. Ante posibles fallas, sería útil contar con un sistema de simulación de datos automatizado que permita evaluar el rendimiento del sistema bajo condiciones realistas.

Referencias

- [1] STMicroelectronics. Discovery kit with stm32f429zi mcu. https://www.st.com/resource/en/user_manual/um1670-discovery-kit-with-stm32f429zi-mcu-stmicroelectronics.pdf.
- [2] ThingsBoard. Thingsboard architecture. <https://thingsboard.io/docs/reference/>, 2024.

6. Apéndice

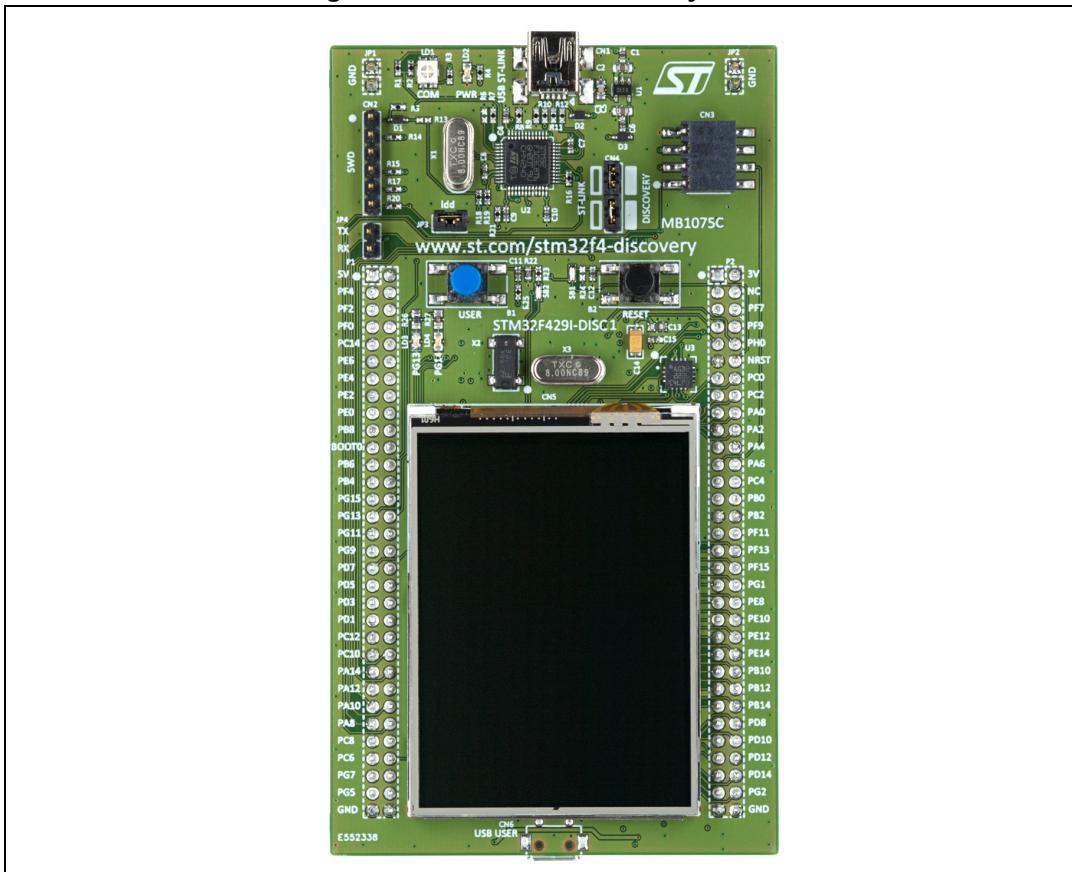
Discovery kit with STM32F429ZI MCU

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.



Contents

1	Features	6
2	Ordering information	7
2.1	Codification	7
3	Development environment	8
3.1	System requirements	8
3.2	Development toolchains	8
3.3	Demonstration software	8
4	Conventions	8
5	Quick start	9
5.1	Getting started	9
6	Hardware layout	10
6.1	STM32F429 Discovery board layout	11
6.2	Mechanical drawing	13
6.3	Embedded ST-LINK/V2-B	14
6.3.1	Drivers	15
6.3.2	ST-LINK/V2-B firmware upgrade	15
6.3.3	VCP configuration	15
6.3.4	Using ST-LINK/V2-B to program/debug the STM32F429ZIT6 on board	16
6.3.5	Using ST-LINK/V2-B to program/debug an external STM32 application	17
6.4	Power supply and power selection	18
6.5	LEDs	18
6.6	Push-buttons	18
6.7	USB OTG supported	18
6.8	Gyroscope MEMS (ST-MEMS I3G4250D)	19
6.9	TFT LCD (Thin-film-transistor liquid-crystal display)	19
6.10	64-Mbit SDRAM (1Mbit x 16-bit x 4-bank)	19
6.11	JP3 (Idd)	19

6.12	OSC clock	20
6.12.1	OSC clock supply	20
6.12.2	OSC 32 kHz clock supply	20
6.13	Solder bridges	21
6.14	Extension connectors	22
7	32F429IDISCOVERY information	29
7.1	Product marking	29
7.2	Board revision history	29
7.2.1	MB1075	29
7.3	Known limitations	30
7.3.1	MB1075-F429	30
8	Revision history	31

List of tables

Table 1.	Ordering information	7
Table 2.	Codification explanation	7
Table 3.	ON/OFF conventions	8
Table 4.	Jumper states	14
Table 5.	Debug connector CN2 (SWD)	17
Table 6.	Solder bridges	21
Table 7.	STM32 pin description versus board functions	22
Table 8.	Document revision history	31

List of figures

Figure 1.	STM32F429 Discovery board	1
Figure 2.	Hardware block diagram.....	10
Figure 3.	Top layout.....	11
Figure 4.	Bottom layout	12
Figure 5.	STM32F429 Discovery board mechanical drawing	13
Figure 6.	Updating the list of drivers in Device Manager	15
Figure 7.	STM32F429 Discovery board connections.....	16
Figure 8.	ST-LINK/V2-B connections.....	17

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

2 Ordering information

To order the Discovery kit with the STM32F429ZI microcontroller, refer to [Table 1](#). Additional information is available from the datasheet and reference manual of the target microcontroller.

Table 1. Ordering information

Order code	Board reference	Target STM32
STM32F429I-DISC1 ⁽¹⁾	MB1075	STM32F429ZIT6

1. Mbed Enabled™ STM32F429I-DISC1 with ST-LINK/V2-B replaces obsolete STM32F429I-DISCO with ST-LINK/V2.

2.1 Codification

The meaning of the codification is explained in [Table 2](#).

Table 2. Codification explanation

32XXYYZDISCOVERY	Description	Example: 32F429IDISCOVERY
32XX	MCU series in STM32 32-bit Arm Cortex MCUs	STM32F4 Series
YY	MCU product line in the series	STM32F429
Z	STM32 Flash memory size: – I for 2 Mbytes	2 Mbytes
DISCOVERY	Discovery kit	Discovery kit

The order code is mentioned on a sticker placed on the top or bottom side of the board.

3 Development environment

3.1 System requirements

- Windows® OS (7, 8 and 10), Linux® 64-bit or macOS®^(a)
- USB Type-A to Mini-B cable

3.2 Development toolchains

- IAR Systems - IAR Embedded Workbench®^(b)
- Keil® - MDK-ARM^(b)
- STMicroelectronics - STM32CubeIDE
- Arm® Mbed™^(c) online (see <http://mbed.org>)

3.3 Demonstration software

The demonstration software, included in the STM32Cube MCU Package corresponding to the on-board microcontroller, is preloaded in the STM32 Flash memory for easy demonstration of the device peripherals in standalone mode. The latest versions of the demonstration source code and associated documentation can be downloaded from www.st.com.

4 Conventions

Table 3 provides the definition of some conventions used in the present document.

Table 3. ON/OFF conventions

Convention	Definition
Jumper JPx ON	Jumper fitted
Jumper JPx OFF	Jumper not fitted
Solder bridge SBx ON	SBx connections closed by solder
Solder bridge SBx OFF	SBx connections left open

-
- a. macOS® is a trademark of Apple Inc. registered in the U.S. and other countries.
 - b. On Windows® only.
 - c. Arm and Mbed are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and or elsewhere.

5 Quick start

The STM32F429 Discovery is a low-cost and easy-to-use development kit to quickly evaluate and start development with an STM32F4 microcontroller.

Before installing and using the product, accept the Evaluation Product License Agreement from www.st.com/stm32f4-discovery.

For more information on the STM32F429 Discovery board and for demonstration software, visit www.st.com/stm32f4-discovery.

5.1 Getting started

Follow the sequence below to configure the STM32F429 Discovery board and launch the DISCOVER application:

1. Ensure that the jumpers JP3 and CN4 are set ON (Discovery mode).
2. Connect the STM32F429 Discovery board to a PC using a USB cable Type-A/Mini-B through the USB ST-LINK connector CN1, to power the board, the LEDs LD2 (PWR) and LD1 (COM).
3. The following applications are available on the screen:
 - Clock/Calendar and Game
 - Video Player and Image Browser (play videos and view images from the USB mass storage connected to CN6)
 - Performance monitor (watch the CPU load and run a graphical benchmark)
 - System Info
4. The demonstration software, as well as other software examples, are available at the www.st.com/stm32f4-discovery web page.
5. Develop applications starting from the examples.

6 Hardware layout

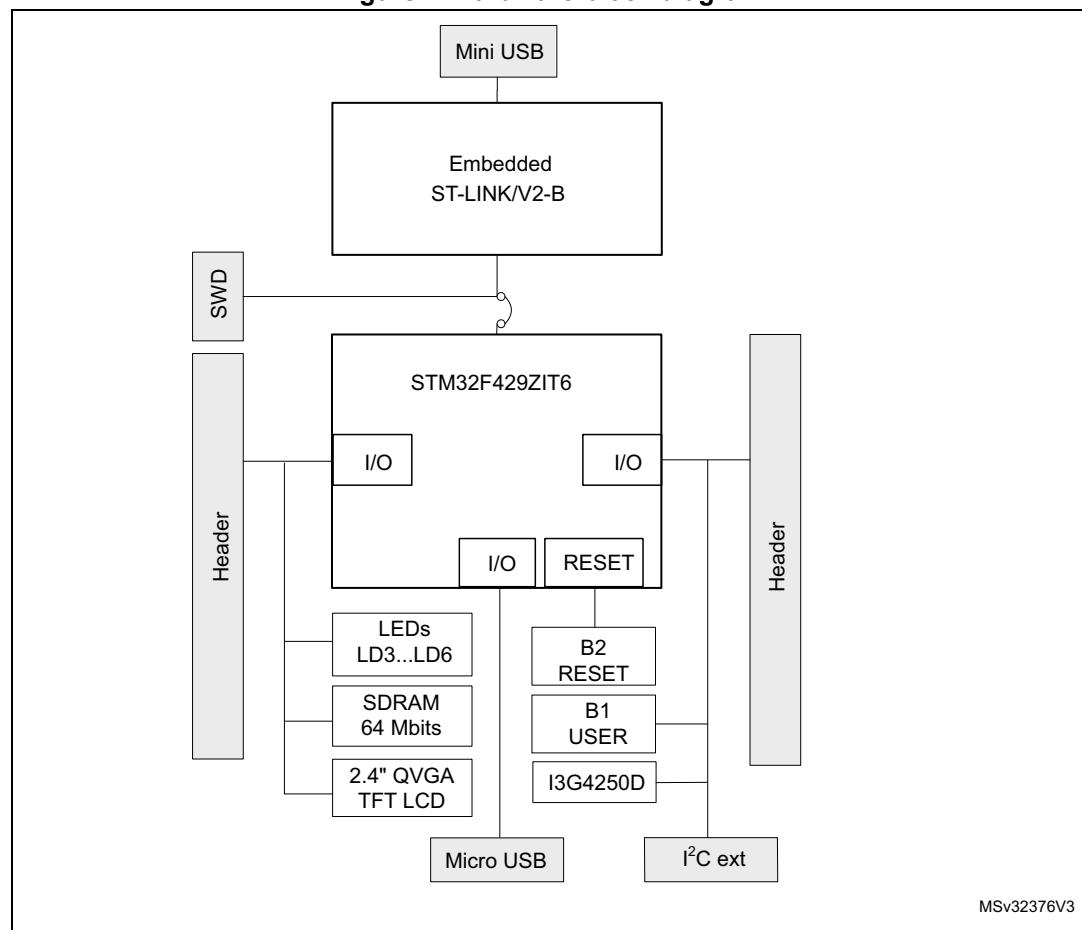
The STM32F429 Discovery board has been designed around the STM32F429ZIT6 microcontroller in a 144-pin LQFP package.

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

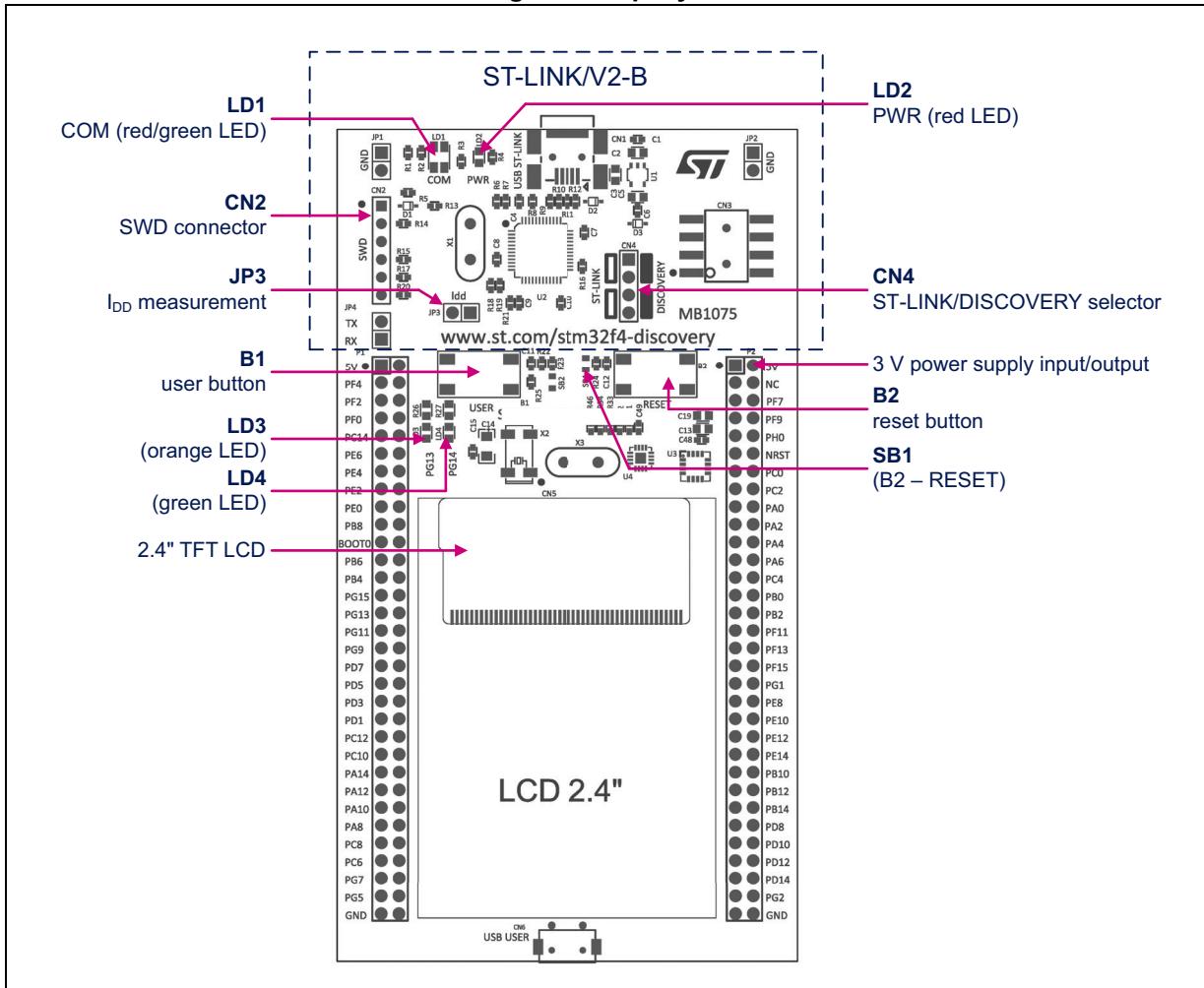
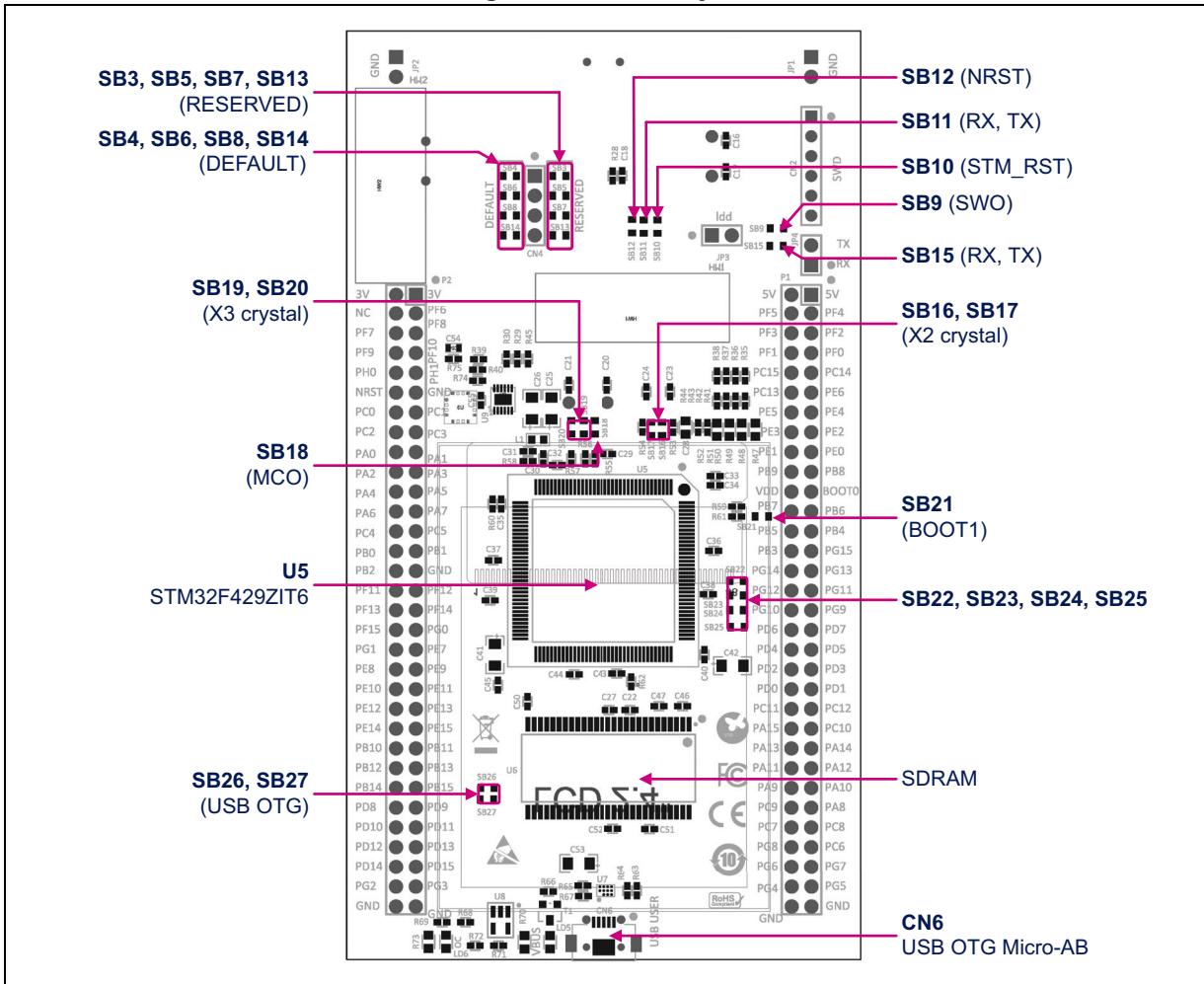
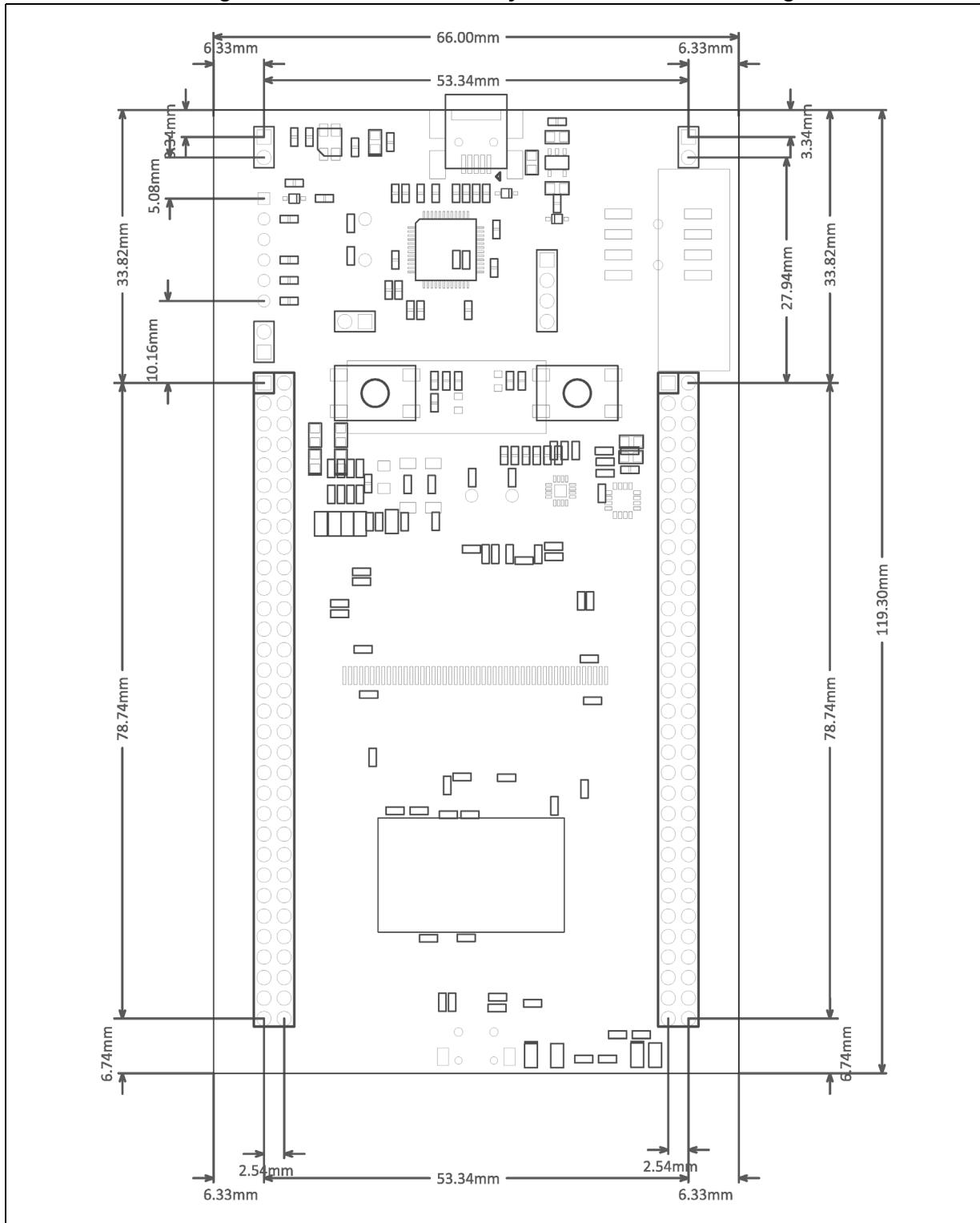


Figure 4. Bottom layout



6.2 Mechanical drawing

Figure 5. STM32F429 Discovery board mechanical drawing



6.3 Embedded ST-LINK/V2-B

The ST-LINK/V2-B on STM32F429I-DISC1 is embedded as a programming and debugging tool. Virtual COM port and USB mass storage features are supported by the ST-LINK/V2-B only for Mbed compatibility.

The ST-LINK/V2-B makes the STM32F429I-DISC1 boards Mbed Enabled. The embedded ST-LINK/V2-B supports only SWD for STM32 devices. For information about debugging and programming features, refer to *ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32, User manual (UM1075)*, which describes in detail all the ST-LINK/V2 features.

The changes in ST-LINK/V2-B versus ST-LINK/V2 version are listed below.

New features supported on ST-LINK/V2-B are:

- Virtual COM port interface on USB
- Mass storage interface on USB

Features not supported on ST-LINK/V2-B are:

- SWIM interface
- Minimum supported application voltage limited to 3 V
- USB power management request for more than 100 mA power on USB

Known limitation:

- Activating the readout protection on the ST-LINK/V2-B target, prevents the target application from running afterward. The target readout protection must be kept disabled on the ST-LINK/V2-B boards.

There are two different ways to use the embedded ST-LINK/V2-B, depending on the jumper states, as shown in [Table 4](#):

- Programming or debugging the STM32 on board. Refer to [Section 6.3.4: Using ST-LINK/V2-B to program/debug the STM32F429ZIT6 on board](#) to program or debug the STM32F429ZIT6 on-board.
- Programming or debugging the STM32 in an external application board, using a cable connected to the SWD connector CN2. Refer to chapter [Section 6.3.5: Using ST-LINK/V2-B to program/debug an external STM32 application](#).

Table 4. Jumper states

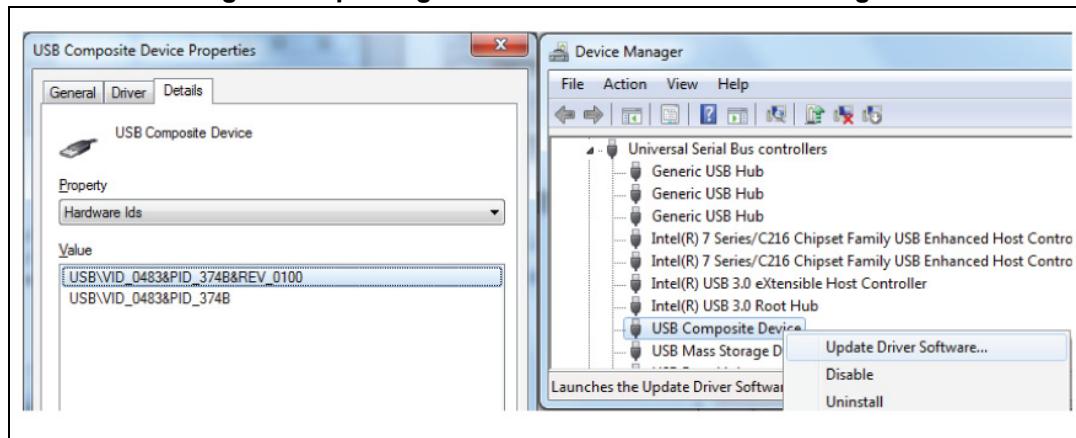
Jumper state	Description
Both CN4 jumpers ON	ST-LINK/V2-B functions enabled for on-board programming (default)
Both CN4 jumpers OFF	ST-LINK/V2-B functions enabled for application through external CN2 connector (SWD supported)

6.3.1 Drivers

The ST-LINK/V2-B requires a dedicated USB driver, which, for Windows® 7, 8, and 10 can be found at the www.st.com website. In case the STM32 Discovery is connected to the PC before the driver is installed, some Discovery interfaces may be declared as “Unknown” in the PC device manager. In this case, the user must install the driver files (see [Figure 6: Updating the list of drivers in Device Manager](#)) and update the driver of the connected device from the device manager.

Note: Prefer using the “USB Composite Device” handle for a full recovery.

Figure 6. Updating the list of drivers in Device Manager



6.3.2 ST-LINK/V2-B firmware upgrade

The ST-LINK/V2-B embeds a firmware upgrade mechanism for the in-situ upgrade through the USB port. As the firmware may evolve during the lifetime of the ST-LINK/V2-B product (for example new functionalities, bug fixes, support for new microcontroller families), it is recommended to visit the www.st.com website, before starting to use the Discovery board and periodically, to stay up-to-date with the latest firmware version.

6.3.3 VCP configuration

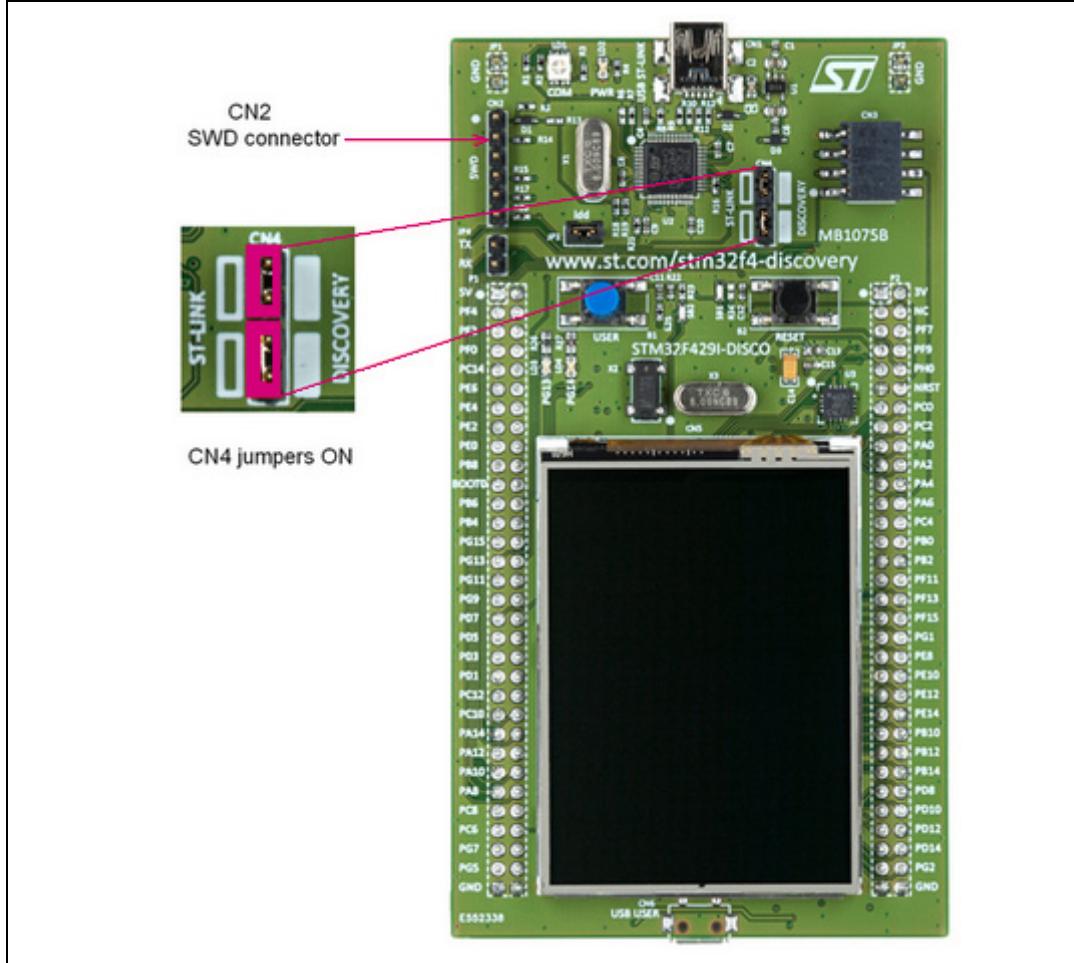
The ST-LINK/V2-B on STM32F429I-DISC1 supports Virtual COM port (VCP) on U2 pin 12 (ST-LINK_TX) and U2 pin 13 (ST-LINK_RX), which are connected to the STM32F429 target STM32 USART1 (PA9, PA10) for Mbed support, thanks to the SB11 and SB15 solder bridges.

The SB11 (PA9) and SB15 (PA10) default configurations for 32F429IDISCOVERY and STM32F429I-DISC1 are given in [Table 6: Solder bridges](#).

6.3.4 Using ST-LINK/V2-B to program/debug the STM32F429ZIT6 on board

To program the STM32F429ZIT6 on board, simply plug in the two jumpers on CN4, as shown marked in red in [Figure 7](#), but do not use the CN2 connector as it could disturb the communication with the STM32F429ZIT6 of the STM32F429 Discovery board.

Figure 7. STM32F429 Discovery board connections



6.3.5 Using ST-LINK/V2-B to program/debug an external STM32 application

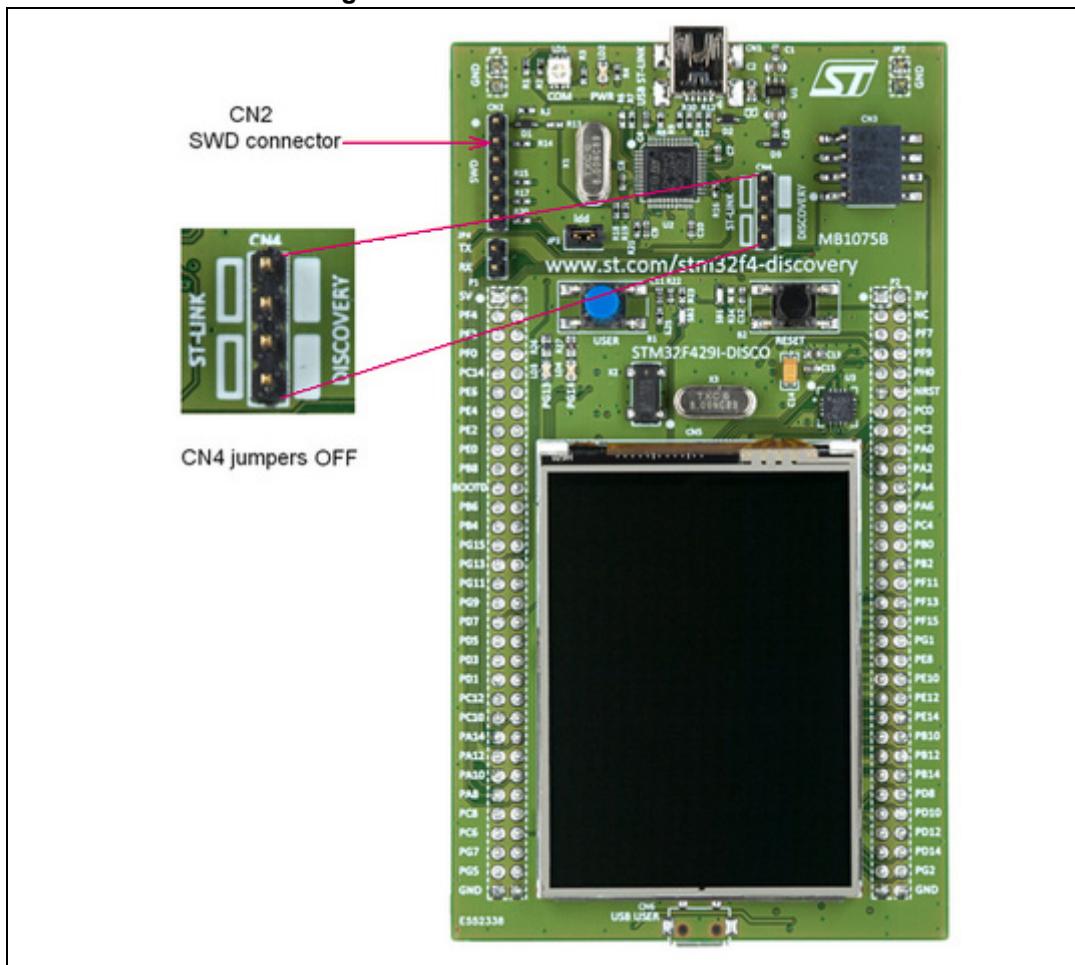
It is very easy to use the ST-LINK/V2-B to program the STM32 on an external application. Simply remove the two jumpers from CN4 as shown in [Figure 8](#) and connect the application to the CN2 debug connector according to [Table 5](#).

Note: SB7 must be OFF if CN2 pin 5 is used in an external application.

Table 5. Debug connector CN2 (SWD)

Pin	CN2	Designation
1	VDD_TARGET	VDD from application
2	SWCLK	SWD clock
3	GND	Ground
4	SWDIO	SWD data input/output
5	NRST	RESET of target STM32
6	SWO	Reserved

Figure 8. ST-LINK/V2-B connections



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.

Two LEDs are dedicated to this module:

- LD5 (green LED) indicates when VBUS is active
- LD6 (red LED) indicates an overcurrent from a connected device

Note: *The USB OTG host operation is realized once the Discovery is connected to the PC via the ST-LINK cable (mini USB). Once the USB OTG host device is detected and identified, then the USB ST-LINK cable can be removed.*

6.8 Gyroscope MEMS (ST-MEMS I3G4250D)

The I3G4250D is a low-power, three-axis angular rate sensor. It includes a sensing element and an IC interface able to provide the measured angular rate to the external world through the I²C/SPI serial interface.

The I3G4250D has a full-scale of $\pm 245/\pm 500/\pm 2000$ dps and is capable of measuring rates with a user-selectable bandwidth.

The STM32F429ZIT6 controls this motion sensor through the SPI interface.

6.9 TFT LCD (Thin-film-transistor liquid-crystal display)

The TFT LCD is a 2.4" display of 262 K colors. Its definition is QVGA (240 x 320 dots) and is directly driven by the STM32F429ZIT6 using the RGB protocol. It includes the ILI9341 LCD controller and can operate with a typical 2.8 V voltage.

6.10 64-Mbit SDRAM (1Mbit x 16-bit x 4-bank)

The 64-Mbit SDRAM is a high-speed CMOS, dynamic random-access memory designed to operate in 3.3 V memory systems containing 67,108,864 bits. It is internally configured as a quad-bank DRAM with a synchronous interface. Each 16,777,216-bit bank is organized as 4,096 rows by 256 columns by 16 bits. The 64-Mbit SDRAM includes auto-refresh, power-saving, and power-down modes. All signals are registered on the positive edge of the clock signal, CLK.

The STM32F429ZIT6 reads and writes data at 80 MHz.

6.11 JP3 (Idd)

Jumper JP3, labeled Idd, allows the consumption of STM32F429ZIT6 to be measured by removing the jumper and connecting an ammeter.

- Jumper ON: STM32F429ZIT6 is powered (default).
- Jumper OFF: an ammeter must be connected to measure the STM32F429ZIT6 current, (if there is no ammeter, the STM32F429ZIT6 is not powered).

6.12 OSC clock

6.12.1 OSC clock supply

The following information indicates all configurations for clock supply selection.

- **MCO from ST-LINK** (from MCO of the STM32F429ZIT6)
This frequency cannot be changed, it is fixed at 8 MHz and connected to PH0-OSC_IN of the STM32F429ZIT6. The configuration needed is:
 - SB18 closed, SB19 open, R56 removed
 - SB20, R57, C20, C21, X3 = do not care
- **Oscillator on board** (from X3 crystal)
For typical frequencies and its capacitors and resistors, refer to the STM32F429ZIT6 datasheet. The configuration needed is:
 - SB18, SB19, SB20 open
 - R56, R57, C20, C21, X3 soldered
- **Oscillator from external PH0** (from external oscillator through pin 10 of the P2 connector)
The configuration needed is:
 - SB19 closed, SB18 open, R56 removed
 - SB20, R57, C20, C21, X3 = do not care
- **No external oscillator** (from internal oscillator HSI only).
PH0 and PH1 can be used as GPIO. The configuration needed is:
 - SB18 open, SB19 closed, SB20 closed, R56 removed, R57 removed
 - C20, C21, X3 = do not care

6.12.2 OSC 32 kHz clock supply

The following information indicates all configurations for the 32 kHz clock supply selection.

- **Oscillator on board** (from X2 Crystal, not provided).
The configuration needed is:
 - SB16 open, SB17 open
 - R53, R54, C23, C24, X2 soldered
- **Oscillator from external PC14** (from external oscillator through pin 9 of P1 connector)
The configuration needed is:
 - SB16 closed, R53 removed
 - SB17, R54, C23, C24, X2 = do not care
- **No external oscillator** (PC14 and PC15 can be used as GPIO).
The configuration needed is:
 - SB16 closed, SB17 closed, R53 removed, R54 removed
 - C23, C24, X2 = do not care

6.13 Solder bridges

Table 6. Solder bridges

Bridge	State ⁽¹⁾	Description
SB19, 20 (X3 crystal)	OFF	X3, C20, C21, R56, and R57 provide a clock. PH0 and PH1 are disconnected from P2.
	ON	PH0 and PH1 are connected to P2. Remove only R56 and R57.
SB4, 6, 8, 14 (default)	ON	Reserved, do not modify.
SB3, 5, 7, 13 (reserved)	OFF	Reserved, do not modify.
SB22, 23, 24, 25	OFF	Reserved, do not modify.
SB16, 17 (X2 crystal)	OFF	X2, C23, C24, R53, and R54 deliver a 32 kHz clock. PC14 and PC15 are not connected to P2
	ON	PC14 and PC15 are only connected to P2. Remove only R53 and R54.
SB1 (B2-RESET)	ON	B2 push-button is connected to the NRST of STM32F429ZIT6.
	OFF	B2 push-button is not connected to the NRST of STM32F429ZIT6.
SB2 (B1-USER)	ON	B1 push-button is connected to PA0.
	OFF	B1 push-button is not connected to PA0.
SB11, 15 (RX,TX)	OFF	STM32F429 USART1 is not connected to ST-LINK, so VCP is disabled (Default configuration on 32F429IDISCOVERY).
	ON	STM32F429 USART1 is connected to ST-LINK, so VCP is enabled (default configuration on STM32F429I-DISC1).
SB12 (NRST)	ON	NRST signal of connector CN2 is connected to the NRST of STM32F429ZIT6.
	OFF	NRST signal is not connected.
SB9 (SWO)	OFF	SWO signal is not connected.
	ON	SWO signal of connector CN2 is connected to PB3.
SB10 (STM_RST)	OFF	No incidence on the NRST signal of STM32F429ZIT6.
	ON	NRST signal of STM32F429ZIT6 is connected to GND.
SB21 (BOOT0)	ON	BOOT0 signal of STM32F429ZIT6 is at level LOW through 510 Ω pull-down.
	OFF	BOOT0 signal of STM32F429ZIT6 is at level HIGH through 10 KΩ pull-up (not provided).
SB26, 27 (USB OTG)	OFF	PB14 and PB15 are only used for USB OTG and not connected to P2 to avoid noise.
	ON	PB14 and PB15 are connected to P2.
SB18 (MCO)	OFF	MCO signal of STM32F429ZIT6 is not used.
	ON	MCO clock signal from STM32F429ZIT6 is connected to OSC_IN of STM32F429ZIT6.

1. The default SBx state is shown in bold.

6.14 Extension connectors

The male headers P1 and P2 can connect the STM32F429 Discovery board to a standard prototyping/wrapping board. STM32F429ZIT6 GPIOs are available on these connectors. P1 and P2 can also be probed by an oscilloscope, a logic analyzer, or a voltmeter.

Table 7. STM32 pin description versus board functions

STM32 pin	Board functions																				
	Main function	LQFP144	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	LED	Push-button	I ² C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2
BOOT0	138	NRST	BOOT0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	21	-	
NRST	25	-	-	-	-	RESET	-	-	-	-	-	-	-	-	-	-	-	-	-	12	
PA0	34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	18	
PA1	35	-	-	-	-	-	-	-	INT1	-	-	-	-	-	-	-	-	-	-	17	
PA2	36	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20	
PA3	37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	19	
PA4	40	-	-	-	-	VSYNC	DB3	-	-	-	-	-	-	-	-	-	-	-	-	22	
PA5	41	-	-	-	-	G2	VSYNC	B5	-	-	-	-	-	-	-	-	-	-	-	21	
PA6	42	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24	
PA7	43	-	-	-	-	DB6	-	-	-	-	-	-	-	-	-	-	-	-	-	23	
PA8	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	53	-
PA9	101	-	USART1_RX	USART1_TX	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	52	-
PA10	102	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	51	-

Table 7. STM32 pin description versus board functions (continued)

STM32 pin		Board functions																		
Main function	LQFP144	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	LED	Push-button	I ² C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2
PA11	103	-	-	-	DB15	DB14	-	-	-	-	-	-	-	-	-	-	-	50	-	
PA12	104	-	-	-	R5	R4	-	-	-	-	-	-	-	-	-	-	-	49	-	
PA13	105	SWCLK	SWDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48	-	
PA14	109	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	47	-	
PA15	110	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46	-	
PB0	46	-	-	-	R6	R3	-	-	-	-	-	-	-	-	-	-	-	-	28	
PB1	47	-	-	-	DB16	DB13	-	-	-	-	-	-	-	-	-	-	-	-	27	
PB2	48	SWO	BOOT1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	30	
PB3	133	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	28	-	
PB4	134	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25	
PB5	135	-	-	SDCKE1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26	
PB6	136	-	-	SDNE1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23	
PB7	137	-	-	-	DB9	DB8	DB5	DB4	-	-	-	-	-	-	-	-	-	-	24	
PB8	139	-	-	-	G5	G4	B7	B6	-	-	-	-	-	-	-	-	-	-	19	
PB9	140	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	20	
PB10	69	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48	
PB11	70	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	47	
PB12	73	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	50	

Table 7. STM32 pin description versus board functions (continued)

STM32 pin		Board functions																						
Main function	LQFP144	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	DM	VBUS	LED	Push-button	I ² C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2		
PB13	74	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	49				
PB14	75	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	52 (1)				
PB15	76	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	51 (2)				
PC0	26	-	-	-	SDNWE	-	-	-	DP	DM	VBUS	-	-	-	-	-	-	-	-	-	14			
PC1	27	-	-	-	-	CSX	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13		
PC2	28	-	-	-	-	CSX	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	16		
PC3	29	-	-	-	-	CSX	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15		
PC4	44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	26		
PC5	45	-	-	-	-	-	-	-	-	-	-	Red	-	-	-	-	-	-	-	-	-	-	25	
PC6	96	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	57	-	
PC7	97	-	-	-	-	DB10	H SYNC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	56	-	
PC8	98	-	-	-	-	R2	-	G6	H SYNC	-	-	-	-	-	-	-	-	-	-	-	-	55	-	
PC9	99	-	-	-	-	-	-	-	-	-	-	-	-	-	SDA	-	-	-	1	-	54	-		
PC10	111	-	-	-	-	DB12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45	-	
PC11	112	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44	-	
PC12	113	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	43	-	
PC13	7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	
PC14	8	OSC32_IN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	-	

Table 7. STM32 pin description versus board functions (continued)

STM32 pin		Board functions																			
Main function	LQFP144	OSC32_OUT	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	LED	Push-button	I²C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2
PC15	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10	-	
PD0	114	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	42	-	
PD1	115	-	-	-	-	D3	D2	-	-	-	-	-	-	-	-	-	-	-	41	-	
PD2	116	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40	-	
PD3	117	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39	-	
PD4	118	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38	-	
PD5	119	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37	-	
PD6	122	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	36	-	
PD7	123	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	35	-	
PD8	77	-	-	-	-	D15	D14	D13	-	-	-	-	-	-	-	-	-	-	-	54	-
PD9	78	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	53	-
PD10	79	-	-	-	-	NBL0	D1	D0	-	-	-	-	-	-	-	-	-	-	-	-	56
PD11	80	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	55
PD12	81	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	58
PD13	82	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	57
PD14	85	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60
PD15	86	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	59
PE0	141	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	17	-
PE1	142	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	18	-
PE2	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15	-
PE3	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	16	-
PE4	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13	-
PE5	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	14	-

Table 7. STM32 pin description versus board functions (continued)

STM32 pin		Board functions																		
Main function	LQFP144	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	LED	Push-button	I ² C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2
PE6	5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11	-	
PE7	58	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37	
PE8	59	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40	
PE9	60	-	-	-	D4	D5	D6	-	-	-	-	-	-	-	-	-	-	-	39	
PE10	63	-	-	-	D7	D8	D9	-	-	-	-	-	-	-	-	-	-	-	42	
PE11	64	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	41	
PE12	65	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	44	
PE13	66	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	43	
PE14	67	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46	
PE15	68	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45	
PF0	10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	-	
PF1	11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8	-	
PF2	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	
PF3	13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	-	
PF4	14	-	-	-	A5	A4	A3	A2	A1	A0	-	-	-	-	-	-	-	3	-	
PF5	15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	-	
PF6	18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	-	
PF7	19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6	-	
PF8	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	
PF9	21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8	-	
PF10	22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	-	
PF11	49	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32	-	
PF12	50	-	-	-	A7	A6	SDNRAS	-	-	-	-	-	-	-	-	-	-	-	31	
PF13	53	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	34	

Table 7. STM32 pin description versus board functions (continued)

STM32 pin		Board functions																		
Main function	LQFP144	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	LED	Push-button	I ² C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2
PF14	54	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	33	
PF15	55	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	36	
PG0	56	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	35	
PG1	57	-	-	-	A11	A10	A9	A8	-	-	-	-	-	-	-	-	-	-	38	
PG2	87	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	62	
PG3	88	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61	
PG4	89	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	62	-	
PG5	90	-	-	-	BA1	BA0	-	-	-	-	-	-	-	-	-	-	-	61	-	
PG6	91	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60	-	
PG7	92	-	-	-	-	DOTCLK	DB17	-	-	-	-	-	-	-	-	-	-	59	-	
PG8	93	-	-	-	SDCLK	-	-	-	-	-	-	-	-	-	-	-	-	58	-	
PG9	124	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	33	-	
PG10	125	-	-	-	-	B4	B3	G3	-	-	-	-	-	-	-	-	-	34	-	
PG11	126	-	-	-	-	B4	B3	G3	-	-	-	-	-	-	-	-	-	31	-	
PG12	127	-	-	-	-	DB2	DB1	DB7	-	-	-	-	-	-	-	-	-	32	-	
PG13	128	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	29	-	
PG14	129	-	-	-	-	-	-	-	-	-	Red	Green	-	-	-	-	-	30	-	
PG15	132	-	-	SDNCAS	-	-	-	-	-	-	-	-	-	-	-	-	-	27	-	
PH0	23	OSC_IN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10	

Table 7. STM32 pin description versus board functions (continued)

STM32 pin		Board functions																			
Main function	LQFP144	OSC_OUT	System	VCP	SDRAM	LCD-TFT	LCD-RGB	LCD-SPI	I3G4250D	USB	LED	Push-button	I ² C Ext	Touch panel	Free I/O	Power supply	CN2	CN3	CN6	P1	P2
PH1	24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	22	-	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	-	-	1	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8	-	1	-	-	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	7	5	63	11	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	64	29	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

1. If SB27 is ON.
2. If SB26 is ON.

7 32F429IDISCOVERY information

7.1 Product marking

The sticker located on the top or bottom side of the PCB board shows the information about product identification such as board reference, revision, and serial number.

The first identification line has the following format: "MBxxxx-Variant-yzz", where "MBxxxx" is the board reference, "Variant" (optional) identifies the mounting variant when several exist, "y" is the PCB revision and "zz" is the assembly revision: for example B01.

The second identification line is the board serial number used for traceability.

Evaluation tools marked as "ES" or "E" are not yet qualified and therefore not ready to be used as reference design or in production. Any consequences deriving from such usage will not be at ST charge. In no event, ST will be liable for any customer usage of these engineering sample tools as reference designs or in production.

"E" or "ES" marking examples of location:

- On the target STM32 that is soldered on the board (for illustration of STM32 marking, refer to the STM32 datasheet "Package information" paragraph at the www.st.com website).
- Next to the evaluation tool ordering part number that is stuck or silk-screen printed on the board.

The board reference for the 32F429IDISCOVERY base board is MB1075.

7.2 Board revision history

7.2.1 MB1075

Revision B01

The revision B-01 of the MB1075 board is the initial official release.

Revision C01

The revision C-01 of the MB1075 board removes the limitations of the revision B-01.

1. STM32F103C8T6 replaced by STM32F103CBT6 for ST-LINK/V2-B
2. Mbed Enabled™, SB11 and SB15 closed

Revision E01

The revision E-01 of the MB1075 board removes the limitations of the revision C-01.

1. U3: L3GD20 replaced by I3G4250D and R75, C54 added
2. Old STMPE811QTR kept and SX8651 added for two footprints, and U9, C55, and R74 added for SX8651, and original STMPE811QTR circuit kept
3. CN5 SF-TC240T-9370-T replaced by FRD240C48003-B: pin1 and pin3 for YU and YD swapped, pin8 connected with 3V
4. Updated solder bridges:
 - ON: SB3, SB5, SB7, SB13, SB19
 - OFF: SB4, SB6, SB8, SB14, SB18
5. For BOM for SX8651: as U9 SX8651IWLRT is soldered, added R29, R30, and R45 4.7 KΩ resistors
6. Updated CN6 footprint
7. Line added on top silkscreen for CN5 when FPC is mounted

7.3 Known limitations

7.3.1 MB1075-F429

Revision B01

ST-LINK update from ST-LINK/V2 to ST-LINK/V2-B

Revision C01

EOL request for L3GD20 and STMPE811QTR

8 Revision history

Table 8. Document revision history

Date	Revision	Changes
10-Sep-2013	1	Initial release.
04-Mar-2016	2	New revision to introduce STM32F429I-DISC1 additional CPN that corresponds to Mbed-Enabled Discovery Kit. Updated: <i>Section : Introduction, Figure 1: STM32F429 Discovery board, Section 5: Quick start, Section 6: Hardware layout, Figure 2: Hardware block diagram, Figure 3: Top layout, Section 6.3: Embedded ST-LINK/V2 (or V2-B), Table 5: Solder bridges, Table 6: STM32 pin description versus board functions, Section 7: Electrical schematics.</i>
20-Sep-2017	3	Updated <i>Section 6.9: TFT LCD (Thin-film-transistor liquid-crystal display).</i>
02-Jul-2020	4	Reshuffle of document to align with latest standards: – <i>Introduction</i> to <i>Quick start</i> reordering – <i>Electrical schematics</i> removed – New <i>Table 2: Codification explanation</i> and <i>Section 7: 32F429IDISCOVERY information</i> Updated: – <i>Figure 2, Figure 3, Figure 4, Figure 5, Figure 7, and Figure 8.</i> – <i>Table 4: Jumper states</i> – <i>Section 6.3.4: Using ST-LINK/V2-B to program/debug the STM32F429ZIT6 on board</i> and <i>Section 6.3.5: Using ST-LINK/V2-B to program/debug an external STM32 application</i> for connector CN2, and <i>Table 6: Solder bridges</i> for connector CN3. – <i>Section 6.7: USB OTG supported</i> – Gyroscope ST-MEMS I3G4250D.
26-Aug-2020	5	Updated: – Clarification on protection diodes in <i>Section 6.4</i> Added: – Note on USB OTG host device management in <i>Section 6.7</i>

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2020 STMicroelectronics – All rights reserved