Laboratorio 4:
Arduino: GPIO, ADC, timer, comunicaciones, Iot

Jafet Soto Arrieta, B77543
Universidad de Costa Rica, Escuela de Ingeniería Eléctrica
IE-0623 Laboratorio de Microcontroladores
29 de mayo del 2024
jafet.soto@ucr.ac.cr

***Resumen*—**
Se desarrolla un detector de cambios de pendientes utilizando el MCU STM32F429 Discovery Kit, que detecta variaciones de pendiente mayores a 5 grados para indicar un posible deslizamiento.

I. Introducción

En entornos geográficos montañosos, especialmente aquellos propensos a fuertes lluvias, las pendientes presentan riesgos significativos de deslizamientos, Para mitigar estos riesgos es esencial contar con un sistema confiable de monitoreo de pendientes. Este sistema permite la detección temprana de posibles peligros, proporcionando alertas para tomar medidas preventivas. El presente proyecto tiene como finalidad desarrollar dicho sistema utilizando microcontroladores y sensores, integrando alertas visuales y comunicación con una plataforma IoT para un monitoreo efectivo y en tiempo real.

II. Nota teórica

II-A. Microcontrolador STM32F429:

El STM32F429ZIT6 es un microcontrolador avanzado basado en el núcleo ARM Cortex-M4 con FPU, adecuado para aplicaciones de alto rendimiento y bajo consumo. Sus principales características son:

- *Núcleo:* ARM Cortex-M4 de 32 bits con FPU (RISC).
- *Frecuencia:* Hasta 180 MHz.
- *Memoria:* 2MB de Flash, 256KB de SRAM.
- *Controlador LCD-TFT:* Integrado.
- *Operación de Bajo Consumo:* Diseñado para aplicaciones de bajo consumo.
- *Convertidores A/D y D/A:* 3 ADCs de 12 bits y 2 DACs de 12 bits.
- *Timers:* 17 timers (12 de 16 bits y 2 de 32 bits hasta 180 MHz, cada uno con 4 IC/OC/PWM).
- *Depuración:* Soporta SWD y JTAG.
- *I/O:* 168 pines de I/O con capacidad de interrupción.
- *Interfaces de Comunicación:* 21 interfaces (I2C, USART, SPI, SAI, CAN).
- *Conectividad USB:* USB 2.0 avanzada.
- *Interfaz de Cámara:* Integrada.
- *Generador de Números Aleatorios (RNG):* Integrado.
- *CRC y Controladores DMA:* Incluidos.

El STM32F429 opera con una tensión de alimentación que varía entre 1.8V y 3.6V, dependiendo del modo de operación. Las corrientes de los puertos de I/O varían según la configuración, pero típicamente, cada pin de I/O puede suministrar o consumir hasta 25mA; pero la corriente total del microcontrolador se puede reducir para minimizar el consumo.

Para el desarrollo con el STM32F429 Discovery Kit, se utiliza la biblioteca libopencm3, una biblioteca de código abierto para microcontroladores ARM Cortex-M que proporciona funciones básicas de hardware y control de periféricos.

II-B. Discovery Kit:

El STM32F429 Discovery Kit es una plataforma de desarrollo integral diseñada para evaluar y desarrollar aplicaciones utilizando el microcontrolador STM32F429ZIT6. Ofrece una variedad de periféricos y recursos para facilitar la creación de prototipos y pruebas. Sus principales características incluyen:

- *Microcontrolador:* STM32F429ZIT6.
- *Pantalla:* 2.4”QVGA TFT LCD.
- *Conectividad USB:* USB OTG con conector Micro-AB.
- *Sensor de Movimiento:* Giroscopio I3G4250D de 3 ejes ST MEMS.
- *Indicadores LED:* 6 LEDs (LD1 para comunicaciones USB, LD2 para indicar 3.3V Power On, LD3 y LD4 para uso del usuario, LD5 y LD6 para USB OTG).
- *Botones:* 2 push-buttons (uno para usuario y otro para reset).
- *Memoria:* 64-Mbit SDRAM.
- *Headers:* Header para LQFP144 I/Os.
- *Depuración:* ST-LINK/V2-B on-board.
- *Funciones USB:* Debug, virtual COM, almacenamiento.

El STM32F429 Discovery Kit, con su robusto microcontrolador STM32F429ZIT6, es una plataforma potente y flexible para el desarrollo de aplicaciones embebidas de alto rendimiento, facilitando la integración de diversos periféricos y comunicaciones avanzadas en un solo paquete.

El sistema incorpora un **giroscopio** de 3 ejes ST MEMS, específicamente el I3G4250D. Este giroscopio es un sensor de movimiento altamente sensible y preciso que mide la velocidad angular en tres ejes perpendiculares (X, Y, Z). Su función es detectar cambios en la orientación del sistema, lo que permite

monitorear cualquier deformación angular significativa en las pendientes que se están evaluando.

A continuación se presenta el diagrama de bloques del microcontrolador STM32F429:

Figura 1: Diagrama de bloques del STM32F429 [1].

II-C. Libopencm3:

Libopencm3 es un proyecto de código abierto que tiene como objetivo crear una biblioteca de firmware gratuita y libre para varios microcontroladores ARM Cortex-M. Anteriormente conocido como libopenstm32, este proyecto proporciona soporte para una amplia gama de microcontroladores, incluyendo los de fabricantes como STMicroelectronics (STM32), Atmel (SAM), NXP (LPC), Texas Instruments (Stellaris/Tiva/MSP432), y Silicon Labs (Energy Micro), entre otros.

Para compilar el código de los ejemplos de la biblioteca y generar los archivos .elf se debe ejecutar el siguiente comando **make** en el directorio /libopencm3-examples.

Seguido a esto de ejecuta el comando **arm-none-eabi-objcopy -O binary lcd-serial.elf lcd-serial.bin** para generar el archivo .bin necesario para el MCU.

Finalmente para cargar nuestro codigo en el STM32F429i ejecutamos el comando **st-flash –reset write lcd-serial.bin 0x8000000**

Esto nos permite tener una base para aprovechar los archios *MakeFile* diseñados por libopen que facilitan el trabajo. Para el desarrollo de esta practica se parte del diseño **lcd-serial**.

II-D. Thingsboard:

Para el laboratorio se requiere la conexión a un dashboard que muestre los datos del giroscopio, la temperatura y el nivel de la batería, el dashboard diseñado se puede ver en hipervínculo; a continuación se presenta una vista general del dashboard:

Figura 2: Dashboar de Thinksboard.

Para desarrollar la conección con el dashboard y el STM32F429i se debe generar un archivo de conexión MQTT, para este laboratorio desarrollamos este archivo de configuración utilizando los ejemplos del repositorio de ejemplos de ThingsBoard.

III. Desarrollo y análisis.

III-A. MakeFlile:

En primera instancia se modifica el archivo MakeFile para contener 2 nuevas reglas, la primera denominada **run** , que ejecuta la compilación del código c y la carga del archivo *.bin* al MCU, resumiendo los pasos descritos para en la sección de *libopencm3* .

```
run: all
    @arm-none-eabi-objcopy -O binary lcd-serial.elf lcd-serial.bin
    @st-flash --reset write $(BINARY).bin 0x8000000
```

Figura 3: Regla para compilación y carga.

Se puede modificar la regla anterior variando los nombres de los archivos lcd-serial.elf y lcd-serial.bin, con el archivo binario y el de salida que deseamos procesar y obtener respectivamente.

La segunda regla que agregamos nos permite ejecutar la conexión con el Dashboard por medio de MQTT.

```
mqtt:
    @python3 stm32f429i_to_mqtt.py
```

Figura 4: Regla para generar la conexión MQTT.

III-B. Conexión MQTT:

Bibliotecas Importadas:

- **serial:** Para la comunicación serial.
- **paho.mqtt.client:** Para manejar la conexión MQTT.
- **json:** Para manipular datos en formato JSON.
- **time:** Para manejar retardos en la ejecución.

Configuración Serial:

Se establece la comunicación serial a través del puerto /dev/ttyACM1 a una velocidad de 115200 baudios. Funciones de Callback para MQTT:

- **on_connect:** Notifica cuando la conexión al broker MQTT es exitosa o si hay un error.
- **on_disconnect:** Notifica cuando la conexión al broker MQTT se termina.
- **on_publish:** Confirma la publicación de un mensaje en el dashboard.

Configuración del Cliente MQTT Se crea una instancia del cliente MQTT. Se configuran los callbacks (on_connect, on_disconnect, on_publish). Se establecen los detalles del broker MQTT:

- **Broker:** iot.eie.ucr.ac.cr
- **Puerto:** 1883
- **Usuario:** Monitor_Pendientes_B77543
- **Contraseña:** ae79c8xvqqc1z3x1yv2c

El proceso de conexión y publicación comienza estableciendo las credenciales y conectándose al broker mediante un bucle while que asegura intentos repetidos hasta lograr la conexión. Los datos seriales se leen, se procesan y se dividen en componentes (Eje X, Eje Y, Eje Z, Batería). La información de la batería se valida y transforma, y los datos se convierten a formato JSON para publicarse en el tópico v1/devices/me/telemetry. Posteriormente, los datos se publican en el dashboard de ThingsBoard, imprimiendo el payload publicado. El flujo incluye la conexión serial, configuración del cliente MQTT, intentos de conexión al broker, lectura y procesamiento de datos, publicación, y repetición desde la lectura. Se maneja errores básicos en la conexión y publicación, validando el estado de la batería y asegurando la corrección del puerto serial y las credenciales MQTT.

Para conectar con el serbidod de Thingsboard se debe realizar una conexión parecida a la siguiente:

curl -v -X POST http://iot.eie.ucr.ac.cr:8080/api/v1/ae79c8xvqqc1z3x1yv2c/telemetry –header Content-Type:application/json –data "temperature:25"

Se leccionando el dispositivo en la pagina que deseamos asociar y corriendo el comando mencionado ateriormente activamos el dispositivo en la pagina de Thingboard, lo que nos permite correr el comando *make mqtt* para generar la conexión por medio del archivo **stm32f4i_to_mqtt.py.**

Figura 5: Sistema en estado Ok.

Figura 6: Sistema en estado activo.

III-C. Código principal:

Se desarrolla un firmware para el MCU escrito en C, que integra una variedad de funciones para el monitoreo y control de dispositivos. A lo largo del código, se encuentran diferentes funciones diseñadas para configurar periféricos, leer sensores, comunicarse con otros dispositivos y actualizar la interfaz de usuario. Cada función desempeña un papel crucial en el funcionamiento del sistema, desde la inicialización de hardware hasta la presentación de datos al usuario.

Para la interfase de usuario, el diseño final que se muestra en la pantalla es el siguiente:

En las figuras anteriores se puede apreciar las diferentes componentes de la UI, existe una señal de alerta especial para la condición de la batería, además, al inicial la comunicación serial o la detección de cambios en la pendiente el controlador realiza el parpadeo del led de placa.

Ahora podemos repasar las funciones que componen el código principal y completan el funcionamiento del dispositivo detector de cambios de pendiente.

- **ALERT:** Esta función se encarga de gestionar el estado del sistema, mostrando en la pantalla si está en estado IDLE (funcionando correctamente) o ACTIVE (en estado de alerta debido a desviaciones significativas en los datos del giroscopio). Además de actualizar la pantalla, controla el LED correspondiente para reflejar el estado actual del sistema.
- **spi_setup:** La función spi_setup configura el periférico SPI5 para comunicarse con el giroscopio. Establece la configuración de los pines GPIO necesarios, define la velocidad de transmisión, el modo de operación y otros parámetros importantes para garantizar una comunicación adecuada con el dispositivo.
- **GPIO_CONFIG:** Esta función se encarga de configurar los pines GPIO necesarios para el funcionamiento del sistema. Esto incluye configurar los pines como entrada o salida, así como activar las resistencias de pull-up o pull-down según sea necesario para garantizar un funcionamiento correcto de los dispositivos conectados.
- **adc_setup:** La función adc_setup configura el ADC1 para realizar conversiones analógico-digitales. Define los parámetros de configuración del ADC, como el tiempo de muestreo y la secuencia de canales a convertir, para garantizar mediciones precisas de las señales analógicas.
- **read_adc_naiive:** Esta función realiza una conversión ADC en el canal especificado y devuelve el valor digital correspondiente. Utiliza un enfoque simple y directo para realizar la conversión, sin utilizar características avanzadas del ADC.
- **GYRO_DATA:** GYRO_DATA se encarga de leer los datos del giroscopio a través de la interfaz SPI. Lee los valores de los ejes X, Y, Z y la temperatura del giroscopio, los procesa y los devuelve en una estructura de datos GYRO para su uso posterior en el programa principal.
- **usart_setup:** La función usart_setup configura el USART1 para la comunicación serie. Define la velocidad de baudios, el número de bits de datos, los bits de parada y otros parámetros de comunicación para establecer una conexión serial fiable con otros dispositivos.
- **PW_State:** PW_State se encarga de actualizar la interfaz de usuario para mostrar el estado de la batería. Dependiendo de si la batería está en buen estado o baja, esta función dibuja indicadores gráficos adecuados en la pantalla para informar al usuario sobre el estado de la batería.
- **delay:** La función delay implementa una breve pausa en la ejecución del programa mediante un bucle vacío. Esta pausa se utiliza para introducir retrasos en la ejecución del programa cuando sea necesario, por ejemplo, para esperar un tiempo determinado entre operaciones.

IV. Conclusiones y recomendaciones.

Se desarrolló un firmware para el MCU escrito en C, que cumplió con los objetivos establecidos para el proyecto. A pesar de las dificultades encontradas en el aprendizaje del funcionamiento del giroscopio y la implementación de la comunicación MQTT, se logró completar ambas partes satisfactoriamente. Sin embargo, se reconoce que el trabajo podría haber sido mejor depurado para mejorar la eficiencia y la robustez del sistema.

Se identificaron algunas áreas de mejora durante el desarrollo del proyecto. En primer lugar, se observó que el valor de la batería no se convirtió a un rango de 0 a 100 como se esperaba. Esto podría corregirse en futuras iteraciones del firmware para proporcionar una indicación más precisa del nivel de la batería.

Además, se omitió la inclusión de la lectura de temperatura en el diccionario de código para la comunicación MQTT debido a limitaciones de tiempo. En futuros trabajos, se recomienda agregar esta funcionalidad para proporcionar una visión más completa de las condiciones ambientales en el sistema de monitoreo de pendientes.

También se observó que no se cambiaron los nombres de los archivos del ejemplo de lcd-serial, aunque se generó un Makefile que simplifica la compilación y carga del firmware en el STM32F429. Se sugiere revisar y corregir esta omisión en futuras versiones del código para mantener una estructura de archivos más coherente y organizada.A pesar de los desafíos encontrados, el proyecto fue exitoso y se logró un profundo aprendizaje en el desarrollo de firmware para microcontroladores, la integración de sensores y la comunicación con plataformas IoT.

Referencias

[1] *STM32F427xx STM32F429xx 32b Arm® Cortex®-M4 MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 com. interfaces, camera & LCD-TFT Datasheet - production data.* Retrieved from https://www.st.com/resource/en/datasheet/stm32f427vg.pdf (STMicroelectronics).

STM32F427xx STM32F429xx

32b Arm® Cortex®-M4 MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 com. interfaces, camera & LCD-TFT

Datasheet - production data

Features

- Core: Arm® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 180 MHz, MPU, 225 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- Memories
  - Up to 2 MB of Flash memory organized into two banks allowing read-while-write
  - Up to 256+4 KB of SRAM including 64-KB of CCM (core coupled memory) data RAM
  - Flexible external memory controller with up to 32-bit data bus: SRAM, PSRAM, SDRAM/LPSDR SDRAM, Compact Flash/NOR/NAND memories
- LCD parallel interface, 8080/6800 modes
- LCD-TFT controller with fully programmable resolution (total width up to 4096 pixels, total height up to 2048 lines and pixel clock up to 83 MHz)
- Chrom-ART Accelerator™ for enhanced graphic content creation (DMA2D)
- Clock, reset and supply management
  - 1.7 V to 3.6 V application supply and I/Os
  - POR, PDR, PVD and BOR
  - 4-to-26 MHz crystal oscillator
  - Internal 16 MHz factory-trimmed RC (1% accuracy)
  - 32 kHz oscillator for RTC with calibration
  - Internal 32 kHz RC with calibration
- Low power
  - Sleep, Stop and Standby modes
  - V_{BAT} supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
- 3×12-bit, 2.4 MSPS ADC: up to 24 channels and 7.2 MSPS in triple interleaved mode
- 2×12-bit D/A converters
- General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
- Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 180 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input

LQFP100 (14 × 14 mm)
LQFP144 (20 × 20 mm)
LQFP176 (24 × 24 mm)
LQFP208 (28 x 28 mm)

UFBGA176 (10 x 10 mm)
UFBGA169 (7 × 7 mm)
TFBGA216 (13 x 13 mm)

WLCSP143

- Debug mode
  - SWD & JTAG interfaces
  - Cortex-M4 Trace Macrocell™
- Up to 168 I/O ports with interrupt capability
  - Up to 164 fast I/Os up to 90 MHz
  - Up to 166 5 V-tolerant I/Os
- Up to 21 communication interfaces
  - Up to 3 × I^2C interfaces (SMBus/PMBus)
  - Up to 4 USARTs/4 UARTs (11.25 Mbit/s, ISO7816 interface, LIN, IrDA, modem control)
  - Up to 6 SPIs (45 Mbits/s), 2 with muxed full-duplex I^2S for audio class accuracy via internal audio PLL or external clock
  - 1 x SAI (serial audio interface)
  - 2 × CAN (2.0B Active) and SDIO interface
- Advanced connectivity
  - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
  - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
  - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII
- 8- to 14-bit parallel camera interface up to 54 Mbytes/s
- True random number generator
- CRC calculation unit
- RTC: subsecond accuracy, hardware calendar
- 96-bit unique ID

This is information on a product in full production. www.st.com

communicate at speeds of up to 11.25 Mbit/s. The other available interfaces communicate at up to 5.62 bit/s.

USART1, USART2, USART3 and USART6 also provide hardware management of the CTS and RTS signals, Smart Card mode (ISO 7816 compliant) and SPI-like communication capability. All interfaces can be served by the DMA controller.

Table 8. USART feature comparison(1)

USART name	Standard features	Modem (RTS/CTS)	LIN	SPI master	irDA	Smartcard (ISO 7816)	Max. baud rate in Mbit/s (oversampling by 16)	Max. baud rate in Mbit/s (oversampling by 8)	APB mapping
USART1	X	X	X	X	X	X	5.62	11.25	APB2 (max. 90 MHz)
USART2	X	X	X	X	X	X	2.81	5.62	APB1 (max. 45 MHz)
USART3	X	X	X	X	X	X	2.81	5.62	APB1 (max. 45 MHz)
UART4	X	-	X	-	X	-	2.81	5.62	APB1 (max. 45 MHz)
UART5	X	-	X	-	X	-	2.81	5.62	APB1 (max. 45 MHz)
USART6	X	X	X	X	X	X	5.62	11.25	APB2 (max. 90 MHz)
UART7	X	-	X	-	X	-	2.81	5.62	APB1 (max. 45 MHz)
UART8	X	-	X	-	X	-	2.81	5.62	APB1 (max. 45 MHz)

1. X = feature supported.

3.25 Serial peripheral interface (SPI)

The devices feature up to six SPIs in slave and master modes in full-duplex and simplex communication modes. SPI1, SPI4, SPI5, and SPI6 can communicate at up to 45 Mbits/s, SPI2 and SPI3 can communicate at up to 22.5 Mbit/s. The 3-bit prescaler gives 8 master mode frequencies and the frame is configurable to 8 bits or 16 bits. The hardware CRC generation/verification supports basic SD Card/MMC modes. All SPIs can be served by the DMA controller.

The SPI interface can be configured to operate in TI mode for communications in master mode and slave mode.

3.35 Digital camera interface (DCMI)

The devices embed a camera interface that can connect with camera modules and CMOS sensors through an 8-bit to 14-bit parallel interface, to receive video data. The camera interface can sustain a data transfer rate up to 54 Mbyte/s at 54 MHz. It features:

- Programmable polarity for the input pixel clock and synchronization signals
- Parallel data communication can be 8-, 10-, 12- or 14-bit
- Supports 8-bit progressive video monochrome or raw bayer format, YCbCr 4:2:2 progressive video, RGB 565 progressive video or compressed data (like JPEG)
- Supports continuous mode or snapshot (a single frame) mode
- Capability to automatically crop the image

3.36 Random number generator (RNG)

All devices embed an RNG that delivers 32-bit random numbers generated by an integrated analog circuit.

3.37 General-purpose input/outputs (GPIOs)

Each of the GPIO pins can be configured by software as output (push-pull or open-drain, with or without pull-up or pull-down), as input (floating, with or without pull-up or pull-down) or as peripheral alternate function. Most of the GPIO pins are shared with digital or analog alternate functions. All GPIOs are high-current-capable and have speed selection to better manage internal noise, power consumption and electromagnetic emission.

The I/O configuration can be locked if needed by following a specific sequence in order to avoid spurious writing to the I/Os registers.

Fast I/O handling allowing maximum I/O toggling up to 90 MHz.

3.38 Analog-to-digital converters (ADCs)

Three 12-bit analog-to-digital converters are embedded and each ADC shares up to 16 external channels, performing conversions in the single-shot or scan mode. In scan mode, automatic conversion is performed on a selected group of analog inputs.

Additional logic functions embedded in the ADC interface allow:

- Simultaneous sample and hold
- Interleaved sample and hold

The ADC can be served by the DMA controller. An analog watchdog feature allows very precise monitoring of the converted voltage of one, some or all selected channels. An interrupt is generated when the converted voltage is outside the programmed thresholds.

To synchronize A/D conversion and timers, the ADCs could be triggered by any of TIM1, TIM2, TIM3, TIM4, TIM5, or TIM8 timer.

3.39 Temperature sensor

The temperature sensor has to generate a voltage that varies linearly with temperature. The conversion range is between 1.7 V and 3.6 V. The temperature sensor is internally connected to the same input channel as V_{BAT}, ADC1_IN18, which is used to convert the sensor output voltage into a digital value. When the temperature sensor and V_{BAT} conversion are enabled at the same time, only V_{BAT} conversion is performed.

As the offset of the temperature sensor varies from chip to chip due to process variation, the internal temperature sensor is mainly suitable for applications that detect temperature changes instead of absolute temperatures. If an accurate temperature reading is needed, then an external temperature sensor part should be used.

3.40 Digital-to-analog converter (DAC)

The two 12-bit buffered DAC channels can be used to convert two digital signals into two analog voltage signal outputs.

This dual digital Interface supports the following features:

- two DAC converters: one for each output channel
- 8-bit or 10-bit monotonic output
- left or right data alignment in 12-bit mode
- synchronized update capability
- noise-wave generation
- triangular-wave generation
- dual DAC channel independent or simultaneous conversions
- DMA capability for each channel
- external triggers for conversion
- input voltage reference V_{REF+}

Eight DAC trigger inputs are used in the device. The DAC channels are triggered through the timer update outputs that are also connected to different DMA streams.

3.41 Serial wire JTAG debug port (SWJ-DP)

The Arm SWJ-DP interface is embedded, and is a combined JTAG and serial wire debug port that enables either a serial wire debug or a JTAG probe to be connected to the target.

Debug is performed using 2 pins only instead of 5 required by the JTAG (JTAG pins could be re-use as GPIO with alternate function): the JTAG TMS and TCK pins are shared with SWDIO and SWCLK, respectively, and a specific sequence on the TMS pin is used to switch between JTAG-DP and SW-DP.

L3GD20

MEMS motion sensor: three-axis digital output gyroscope

Datasheet - production data

LGA-16 (4x4x1 mm)

Features

- Three selectable full scales (250/500/2000 dps)
- I^2C/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

Applications

- Gaming and virtual reality input devices
- Motion control with MMI (man-machine interface)
- GPS navigation systems
- Appliances and robotics

Description

The L3GD20 is a low-power three-axis angular rate sensor.

It includes a sensing element and an IC interface capable of providing the measured angular rate to the external world through a digital interface (I^2C/SPI).

The sensing element is manufactured using a dedicated micro-machining process developed by STMicroelectronics to produce inertial sensors and actuators on silicon wafers.

The IC interface is manufactured using a CMOS process that allows a high level of integration to design a dedicated circuit which is trimmed to better match the sensing element characteristics. The L3GD20 has a full scale of ±250/±500/ ±2000 dps and is capable of measuring rates with a user-selectable bandwidth.

The L3GD20 is available in a plastic land grid array (LGA) package and can operate within a temperature range of -40 °C to +85 °C.

Table 1. Device summary

Order code	Temperature range (°C)	Package	Packing
L3GD20	-40 to +85	LGA-16 (4x4x1 mm)	Tray
L3GD20TR	-40 to +85	LGA-16 (4x4x1 mm)	Tape and reel

This is information on a product in full production. www.st.com

2 Mechanical and electrical specifications

2.1 Mechanical characteristics

@ Vdd = 3.0 V, T = 25 °C unless otherwise noted.

Table 4. Mechanical characteristics[1]

Symbol	Parameter	Test condition	Min.	Typ.[2]	Max.	Unit
FS	Measurement range	User-selectable		±250		dps
				±500		
				±2000		
So	Sensitivity	FS = 250 dps		8.75		mdps/digit
		FS = 500 dps		17.50		
		FS = 2000 dps		70		
SoDr	Sensitivity change vs. temperature	From -40 °C to +85 °C		±2		%
DVoff	Digital zero-rate level	FS = 250 dps		±10		dps
		FS = 500 dps		±15		
		FS = 2000 dps		±75		
OffDr	Zero-rate level change vs. temperature	FS = 250 dps		±0.03		dps/°C
		FS = 2000 dps		±0.04		dps/°C
NL	Non linearity	Best fit straight line		0.2		% FS
Rn	Rate noise density			0.03		$dps/(\sqrt{Hz})$
ODR	Digital output data rate			95/190/ 380/760		Hz
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3.0 V. The operational power supply range is specified in *Table 5*.
2. Typical specifications are not guaranteed.

2.2 Electrical characteristics

@ Vdd =3.0 V, T=25 °C unless otherwise noted.

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_IO			V
VIL	Digital low level input voltage				0.2*Vdd_IO	V
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3.0 V.
2. Typical specifications are not guaranteed.
3. It is possible to remove Vdd maintaining Vdd_IO without blocking the communication busses; in this condition the measurement chain is powered off.
4. Sleep mode introduces a faster turn-on time relative to power-down mode.

2.3 Temperature sensor characteristics

@ Vdd =3.0 V, T=25 °C unless otherwise noted.

Table 6. Electrical characteristics (1)

Symbol	Parameter	Test condition	Min.	Typ.(2)	Max.	Unit
TSDr	Temperature sensor output change vs. temperature	-		-1		°C/digit
TODR	Temperature refresh rate			1		Hz
Top	Operating temperature range		-40		+85	°C

1. The product is factory calibrated at 3.0 V.
2. Typical specifications are not guaranteed.

2.4 Communication interface characteristics

2.4.1 SPI - serial peripheral interface

Subject to general operating conditions for Vdd and T_{op}.

Table 7. SPI slave timing values

Symbol	Parameter	Value[1]		Unit
		Min	Max	
tc(SPC)	SPI clock cycle	100		ns
fc(SPC)	SPI clock frequency		10	MHz
tsu(CS)	CS setup time	5		ns
th(CS)	CS hold time	8		
tsu(SI)	SDI input setup time	5		
th(SI)	SDI input hold time	15		
tv(SO)	SDO valid output time		50	
th(SO)	SDO output hold time	6		
tdis(SO)	SDO output disable time		50	

1. Values are guaranteed at a 10 MHz clock frequency for SPI with both 4 and 3 wires, based on characterization results; not tested in production.

Figure 3. SPI slave timing diagram [a]

a. Measurement points are at 0.2·Vdd_IO and 0.8·Vdd_IO, for both input and output port.