Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0624 Laboratorio de Microcontroladores
II ciclo 2022

Laboratorio 5

STM32: GPIO, Giroscopio, comunicaciones, TinyML

Luis Javier Herrera B93840
Juan Ignacio Montealegre Salazar B95001

Profesor: Marco Villalta Fallas

Miércoles 30 de noviembre del 2022

Índice

Índice de figuras

Índice de tablas

1. Resumen

En esta práctica de laboratorio se busca desarrollar un reconocer de actvidad humana HAR (Human Activity Recognition), mediante el uso de la placa Discovery STM32F49 y la implementación de una red neuronal previamente entrenada utilizando la librería TensorFlow. Se busca entrenar una red neuronal que pueda detectar 3 tipos de movimientos distintos ante una entrada de datos correspondientes a cambios en movimiento captados por el giroscopio incorporado a la placa. Se propone que estos movimientos corresponden al movimiento que se realiza con un golpe (atrás hacia adelante de forma repetitiva), un movimiento circular y también se entrena para detectar un estado estacionario. Se realiza una captura de 2000 muestras con cada uno de estos movimientos utilizando un script de Python que lee los datos enviados por la placa y los guarda en un archivo csv. A partir de estos datos se modela una red neuronal con diversas capas y neuronas que por medio de aprendizaje automático podrá dar una predicción a un estímulo recibido luego de ser entrenada con los datos proporcionados. Finalmente, para que el modelo pueda ser utilizado para TinyML se requiere pasar el modelo de TensorFlow a TensorFLow Lite y exportarlo como un archivo de encabezado .h, el cual se incluye en otro programa que se cargará en la placa y podrá realizar una predicción según las lecturas del giroscopio. En la elaboración del proyecto se logró cumplir con todos los pasos, pero el modelo al ser cargado no es muy robusto.

Palabras clave: *giroscopio, TinyML, TensorFlow, microcontrolador, Python, STM32F427*

2. Nota teórica

En esta sección se describe el microcontrolador utilizado en esta ocasión, así como cada periférico utilizado (registros e instrucciones/funciones), componente electrónico complementario y el diseño del circuito final. El proceso de pruebas realizadas para llegar al circuito final se abarca en la sección "Desarrollo y Análisis de Resultados".

2.1. Microcontrolador STM32F427

El MCU STM32F427 cuenta con un núcleo Arm Cortex M4 CPU de 32 bits con un acelerador en tiempo real que le permite tener un tiempo de espera para ejecución de 0 de la memoria Flash. Este opera a una frecuencia de 180 MHz y tiene una memoria Flash de 2 MB organizada en dos bancos que le permite leer mientras escribe los datos [1]. Tiene una memoria SRAM de hasta 254+4 KB y un controlador de memoria externa flexible con memoria de datos de hasta 32 bits. Necesita una alimentación dentro del rango de 1.7 V - 3.6 V y además posee un modo de funcionamiento de bajo consumo de energía. En la Figura 1 se puede apreciar la distribución de pines para este MCU [1].

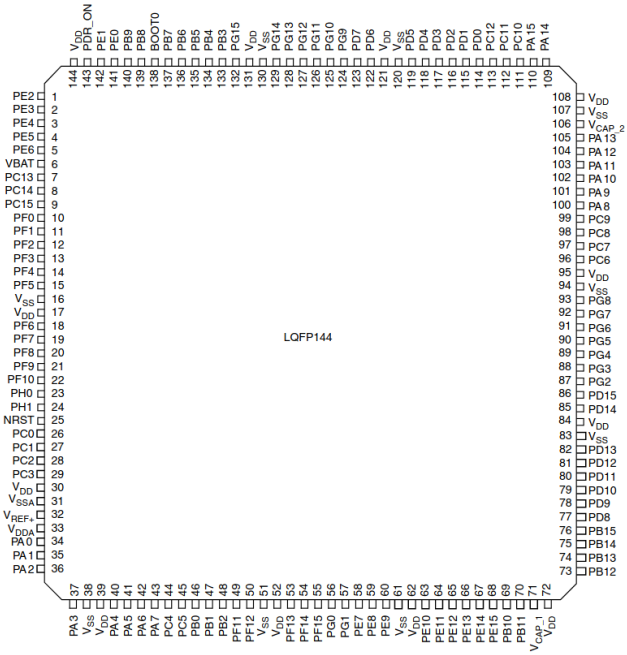


Figura 1: Distribución de pines del STM32 [1].

2.2. Placa STM32F429 Discovery kit

La placa de desarrollo STM32F429 Discovery Kit utiliza el MCU STM32 para así brindar al usuario un dispositivo orientado al desarrollo de aplicaciones y con variedad de periféricos. La lista de periféricos que vienen incorporados a esta tarjeta de desarrollo es la siguiente:

- MCU STM32
- ST-LINK/V2 integrado con interruptor de modo de selección para usar el kit como un ST- LINK/V2 (con conector SWD para programación y depuración).
- Alimentación de la placa que puede ser a través de de un bus USB o alimentación externa de 3 V o 5 V.
- L3GD20 sensor de movimiento.
- Pantalla LCD TFT 2.4", 262K colores RGB, 240 x 320 puntos.
- SDRAM de 64 Mbits.
- Seis LEDs.
- Dos botones.
- USB OTG con conector micro-AB.
- Encabezado de extensión para E/S LQFP144 para una conexión rápida a la placa de creación de prototipos.

Figura 2: Tarjeta de desarrollo STM32F429 Discovery Kit [2].

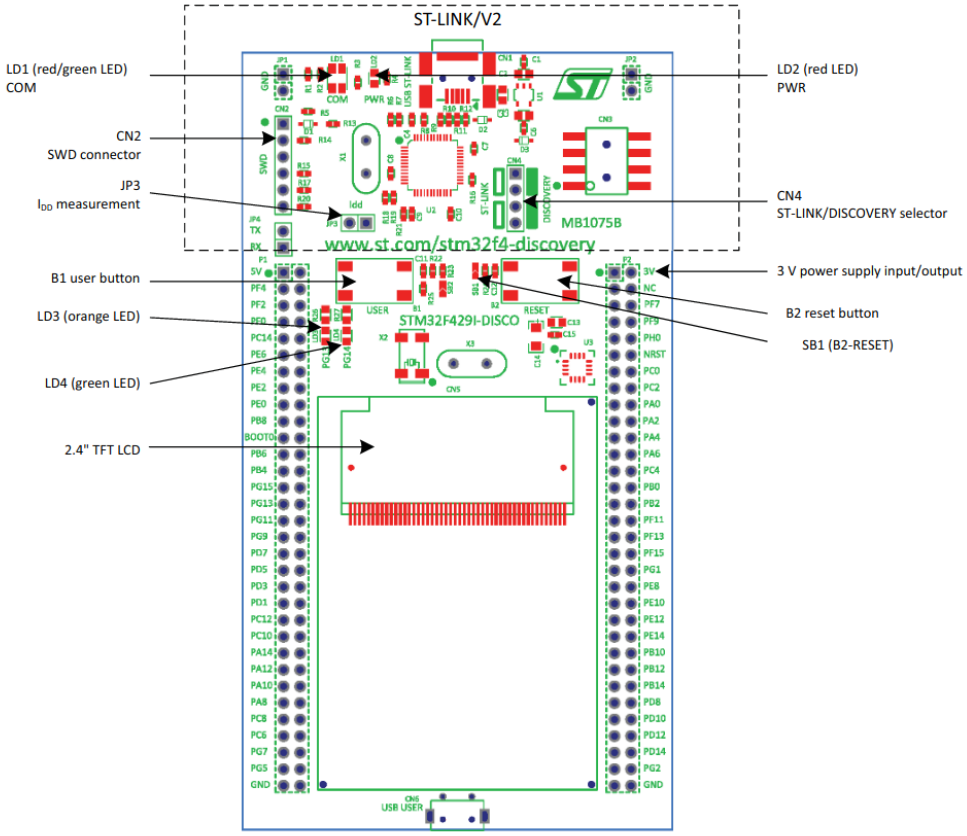


Figura 3: Layout de tarjeta de desarrollo STM32F429 Discovery Kit [2].

2.2.1. Giroscopio

El L3GD20 es un sensor de velocidad angular de tres ejes que cuenta con un bajo consumo de energía. Incluye un elemento de detección y una interfaz IC capaz de proporcionar la tasa angular medida al mundo externo a través de la interfaz serial I2C/SPI [2]. El L3GD20 tiene escalas completas que se selecciona dinámicamente por el usuario con ± 250 dps/500 y dps/±2000 dps [2]. El MCU STM32 se encarga de controlar el sensor mencionado anteriormente a través de la interfaz SPI.

2.3. TensorFlow Lite

TensorFlow es una biblioteca de software de código abierto para inteligencia artificial y aprendizaje automático con redes neuronales profundas [4]. TensorFlow fue desarrollado por Google Brain para uso interno en la empresa y de código abierto desde el 2015.

TensorFlow Lite es un marco de aprendizaje profundo de código abierto diseñado para la inferencia en el dispositivo. Esta librería proporciona un conjunto de herramientas que permite el aprendizaje automático en el dispositivo al dar la posibilidad de que los desarrolladores ejecuten sus modelos entrenados en dispositivos y computadoras móviles, integrados y de IoT [4]. Es compatible con plataformas como Linux, sistemas operativos de dispositivos móviles y MCU. TensorFlow Lite está especialmente optimizado para el aprendizaje automático en el dispositivo (Edge ML). Como modelo Edge ML, es adecuado para la implementación en dispositivos de borde con recursos limitados.

2.4. TinyML

El Tiny Machine Learning (TinyML) se define ampliamente como un campo de crecimiento exponencial en tecnologías y aplicaciones de aprendizaje automático que incluye hardware (circuitos integrados y dedicados), algoritmos y software capaces de realizar análisis de datos de sensores en dispositivos en las áreas de visión, biomédicos, audio, IMU, entre otros [5]. Estos se caracterizan por consumir potencia extremadamente baja, típicamente en el rango de mW y hasta menor. Gracias a los anteriormente mencionado permite una variedad de casos de uso continuamente encendidos y enfocado a dispositivos que funcionan con baterías.

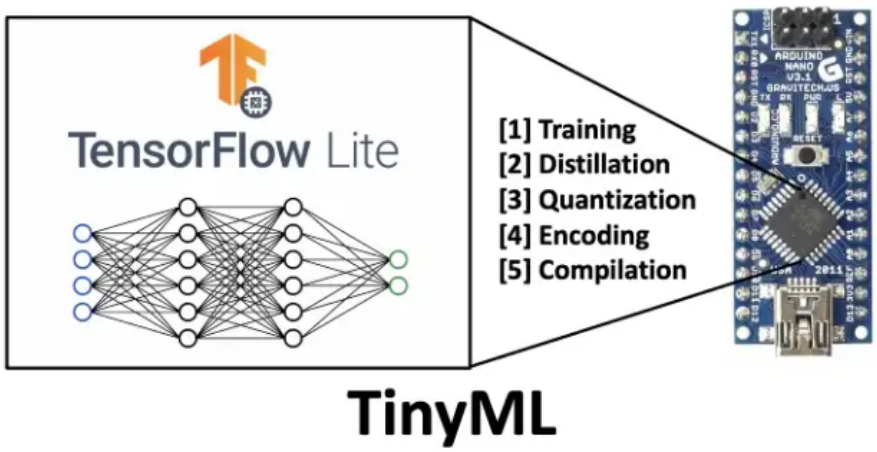


Figura 4: Funcionamiento del TinyML con TensorFlowLite [3].

2.5. Diseño del circuito

En esta ocasión, el circuito corresponde exclusivamente a la placa de desarrollo STM32, así como un cable USB mini.

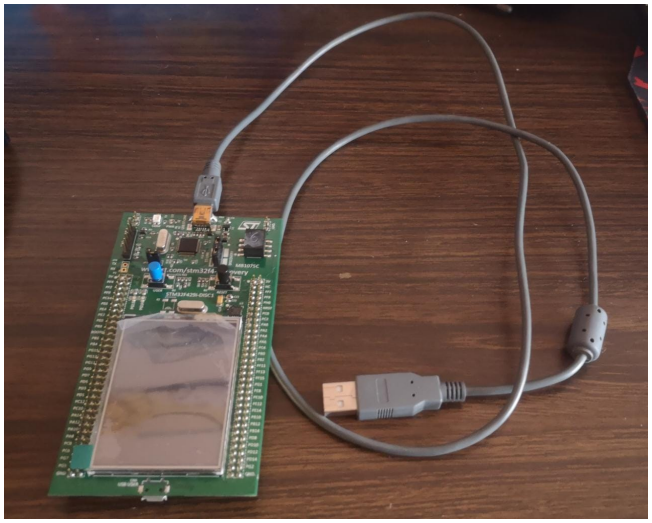


Figura 5: Circuito final del laboratorio. Elaboración propia.

2.6. Lista de componentes

La lista de componentes utilizada se puede apreciar en la Tabla 1.

Componete	Tipo	Precio	Cantidad
Microcontrolador	STM32F429 Discovery Kit	$94.99	1
Cable	USB mini	$5	1
Total	-	$99.99	1

Tabla 1: Lista de componentes utilizados

3. Desarrollo y Análisis de Resultados

En esta sección se comenta de forma detallada el desarrollo del proyecto. Primero se explica el desarrollo del circuito como tal, y luego el diseño del programa a partir del programa deseado.

3.1. Desarrollo del circuito

En esta ocasión se sabía de antemano que el circuito como tal sería únicamente la placa de desarrollo STM-32 así como un cable USB mini para conectarla a la computadora de uso personal. Esto se debe a que la placa ya cuenta con el giroscopio requerido para la elaboración de este laboratorio.

3.2. Desarrollo del programa

Para lograr el desarrollo correcto del laboratorio se tuvo que utilizar diversos scripts de código tanto en C++ como Python. Primeramente, se utilizó el script de Python con nombre *scrip_ datos.py* para realizar la recolección datos por medio de la comunicación con el puerto serial. La funcionalidad de este es crear un archivo en formato `.csv` con las lecturas de los valores de aceleración en los tres ejes de giroscopio. Para este caso se utilizaron tres gestos: uno completamente estacionario, uno realizando un movimiento giratorio en forma de círculos y otro realizando un golpe frontal. Para cada uno de estos gestos se tomaron 2000 muestras. Una vez teniendo los datos para entrenar el modelo listos, se utilizó el script de Python con nombre *modelo.py* el cual se encargaba de recibir los datos de entrada y realizar un entrenamiento y creación de modelo de predicción utilizando machine learning y la librería Tensor Flow. Al final del script se realiza un a conversión de Tensor Flow a Tensor Flow Lite con el fin de poder cargar el modelo de predicción de gestos creado al microcontrolador. El archivo de encabezado resultante es un arreglo de datos numéricos con el nombre de *model_ machine.h*.

Con el modelo de entrenamiento listo, este se añade a una de la funciones ubicadas en el archivo *main_ funcions.cc* que se ejecutan en el microntrolador y se encarga de cargar el modelo y correrlo continuamente realizando la predicción con base a las lecturas de aceleración que se toman del giroscopio del microcontrolador. Por último, solo basta con cargar el binario generado del diseño de software implementado al microcontrolador y observar el funcionamiento por medio de una terminal conectada al puerto serial. Toda la descripción del flujo del programa se puede apreciar en la Figura 16.

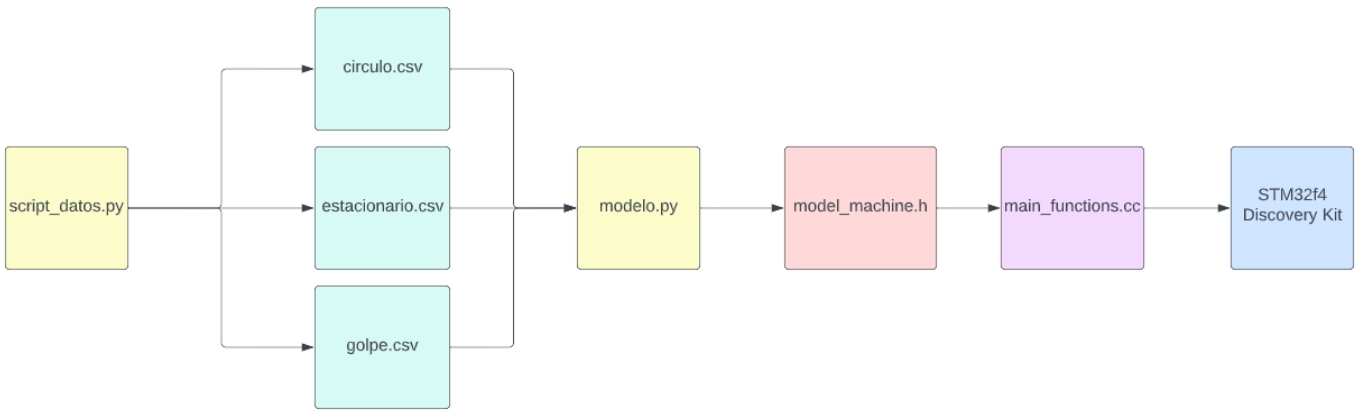


Figura 6: Diagrama de flujo para el funcionamiento del programa creado. Elaboración propia.

Primero se realiza la recolección de los resultados en 3 archivos csv separados. Esto se hace cargando en la placa un programa que envía datos captados por el giroscopio a la computadora y luego se corre un script de python que lee los datos enviados y los guarda en un archivo csv. Se realiza un archivo csv para cada movimiento (estacionario, golpe, y círculo) con un total de 2000 muestras por movimiento. Al graficar estos datos, se obtienen las siguientes gráficas:

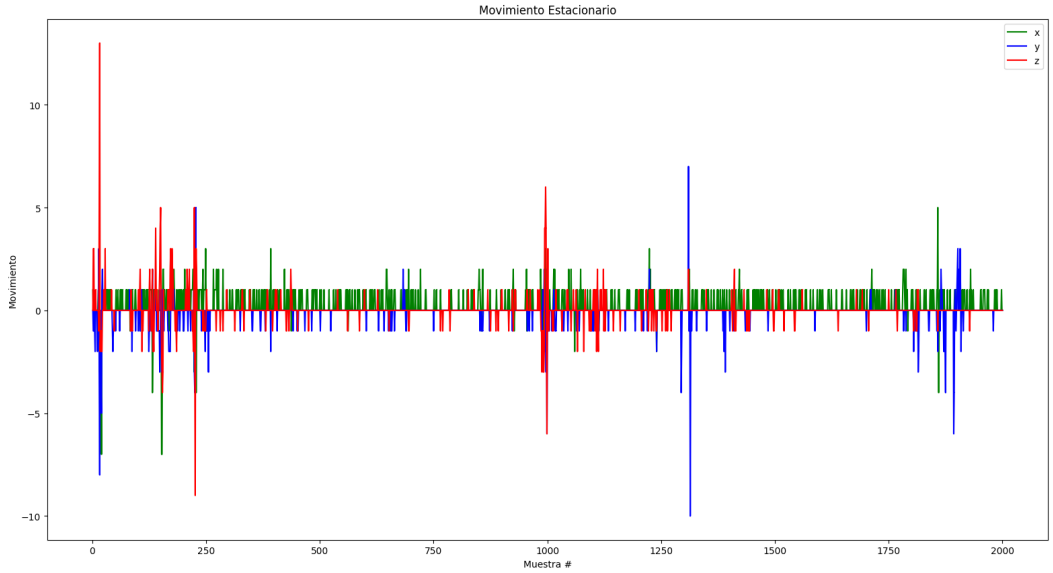


Figura 7: Gráfica de datos correspondientes a movimiento estacionario. Elaboración propia.

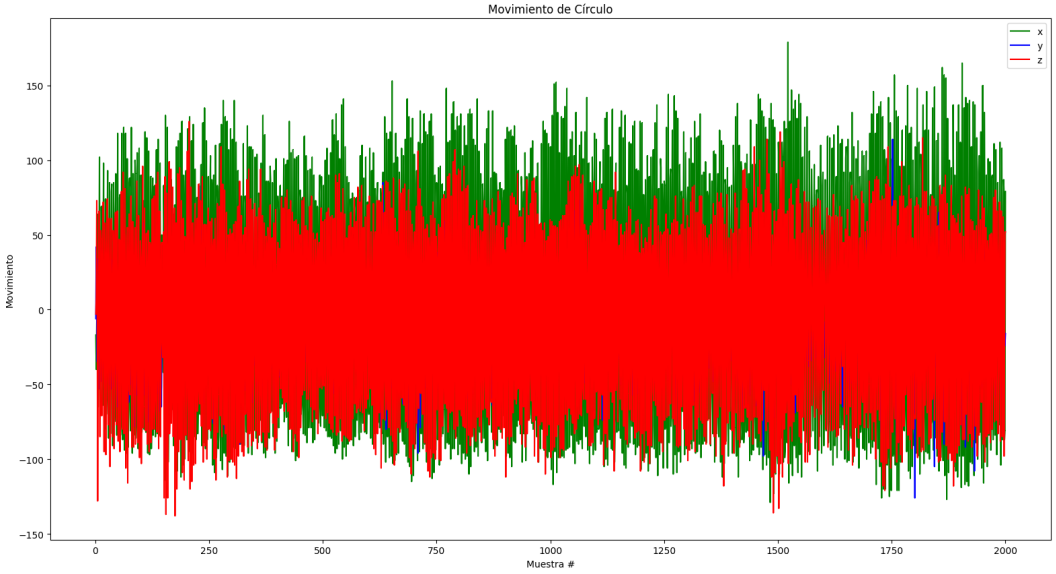


Figura 8: Gráfica de datos correspondientes a movimiento de círculo. Elaboración propia.

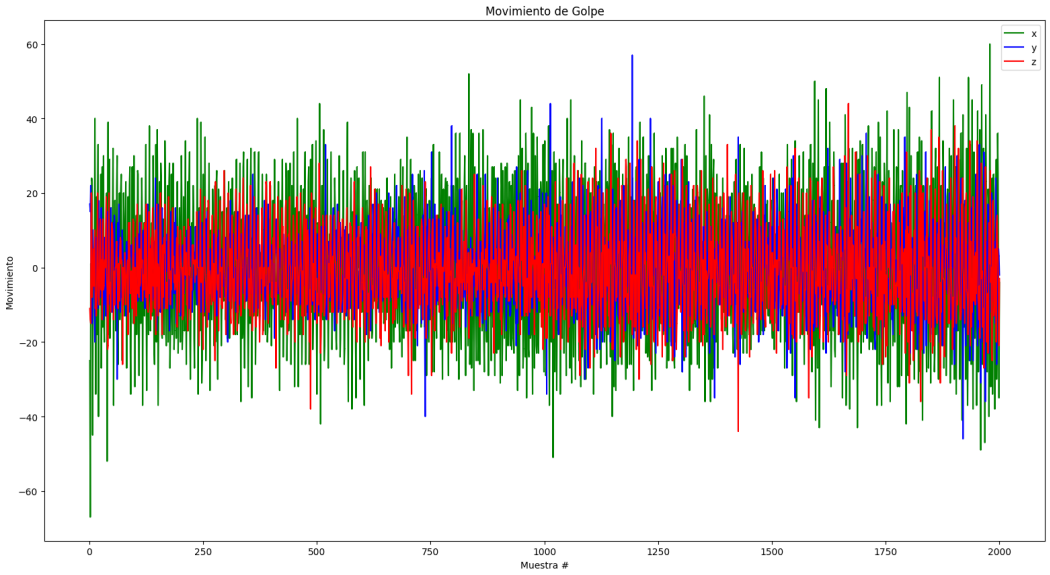


Figura 9: Gráfica de datos correspondientes a movimiento de golpe. Elaboración propia.

Una vez que se tienen los datos requeridos, se desarrolla la red neuronal en otro script de python y utilizando la librería de TensorFlow. Se configura una red neuronal con 3 capas de 125, 100 y 100 neuronas respectivamente. Al crear y entrenar dicho modelo con el 60 % de los datos, utilizar el 20 % de los datos para validar y el resto 20 % para pruebas, se obtienen las siguientes gráficas.

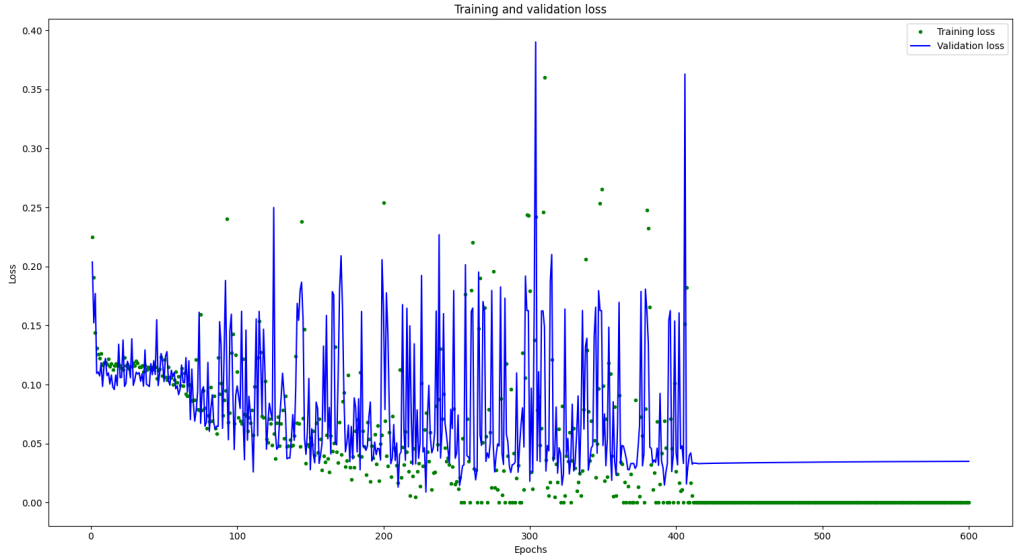


Figura 10: Gráfica de pérdidas de entrenamiento y validación. Elaboración propia.

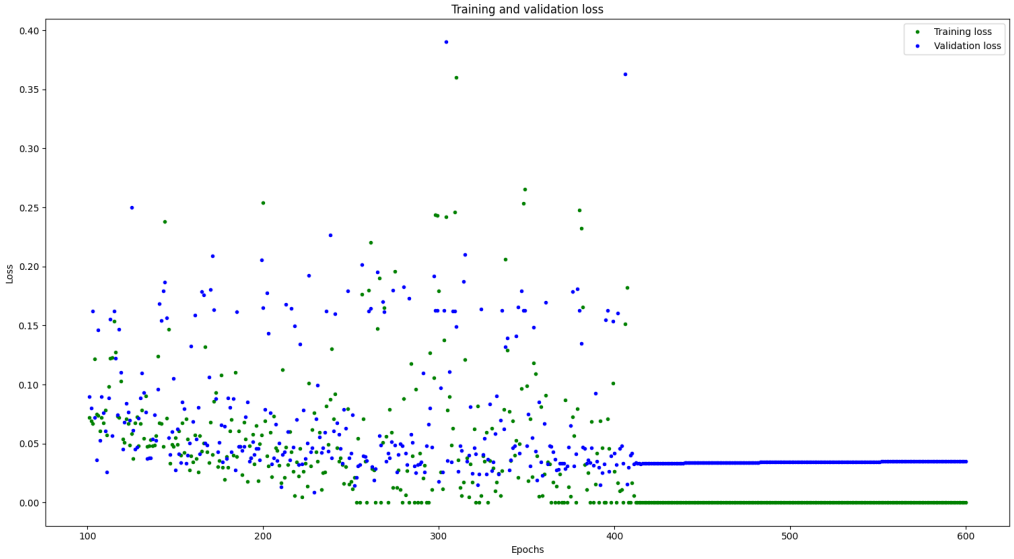


Figura 11: Gráfica de pérdidas de entrenamiento y validación. Elaboración propia.

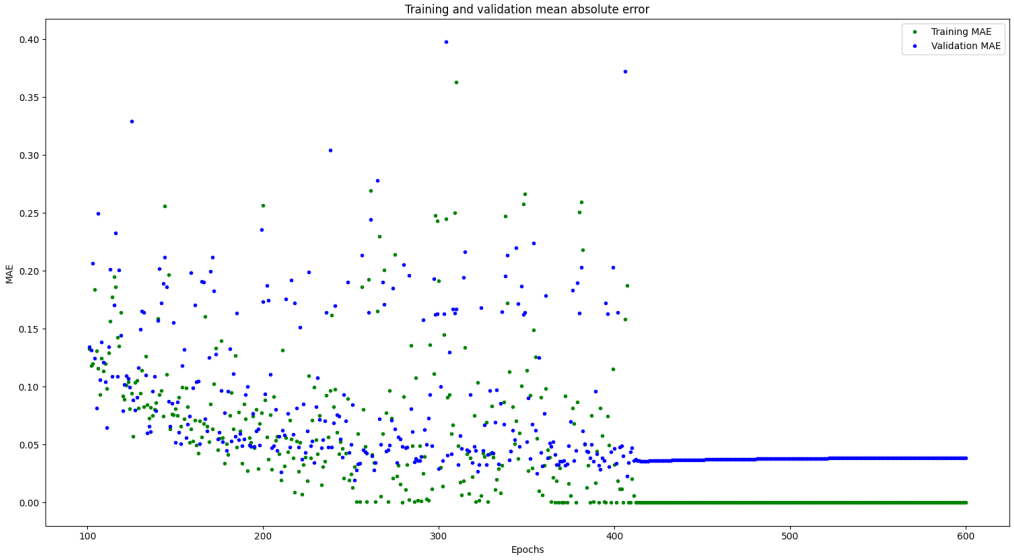


Figura 12: Mean Absolut Error para datos de validación y entrenamiento. Elaboración propia.

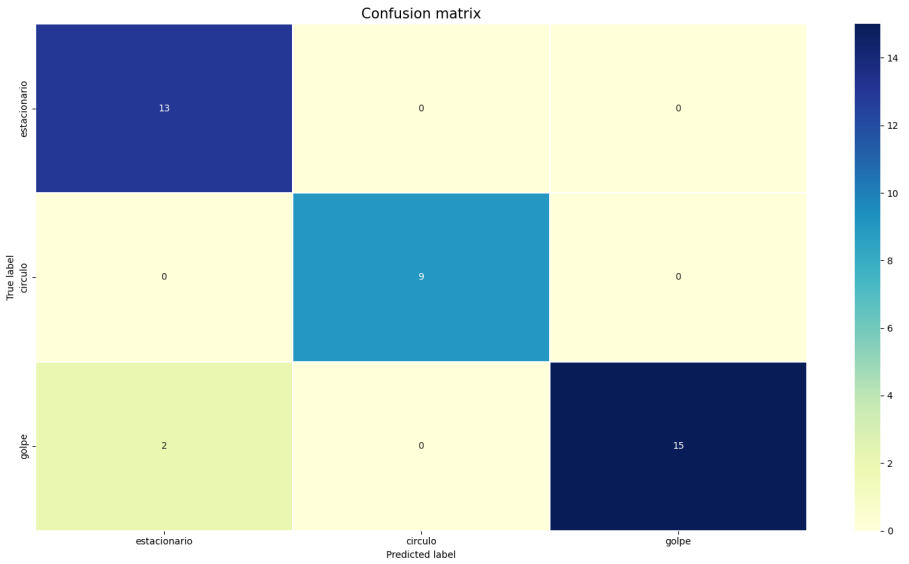


Figura 13: Matriz de confusión para evaluación de modelo obtenido. Elaboración propia.

En las gráficas se puede observar que la red generada no genera muy buenas predicciones, por lo que no se espera que las predicciones que realice la red cuando se implemente no sean buenas. Como se mencionó, se exporta el modelo a TensorFlow Lite y luego se genera un archivo de encabezado .h que posee la matriz a utilizar, para luego hacer otro programa que incluya dicho archivo y genere la predicciones.

3.3. Repositorio de Git

Este proyecto se trabajó en un repositorio de Git para poder tener un control de las diversas versiones de todos los archivos que componen el proyecto. Se utilizó la plataforma de GitHub y el repositorio se encuentran en este link. Este laboratorio se encuentra en el directorio "Laboratorio 5".

3.4. Funcionalidad del circuito y del programa

Luego de cargar el programa en la placa, para visualizar los datos en la terminal es necesario correr el siguiente comando:

```
screen /dev/ttyACM0 115200
```

En las siguientes figuras es posible observar que se imprime en la terminal las lecturas del giroscopio, así como los resultados de la predicción. Es posible ver que ante la mayoría de movimientos realizados se predice con un 99 % de certeza que el movimiento corresponde a un golpe con un 0 % en los otros movimientos, lo cual muestra la deficiencia del modelo implementado. Sin embargo, al realizar movimientos en forma de círculo es posible observar que el porcentaje de predicción para un movimiento circular aumenta, por lo que sí se logra detectar.

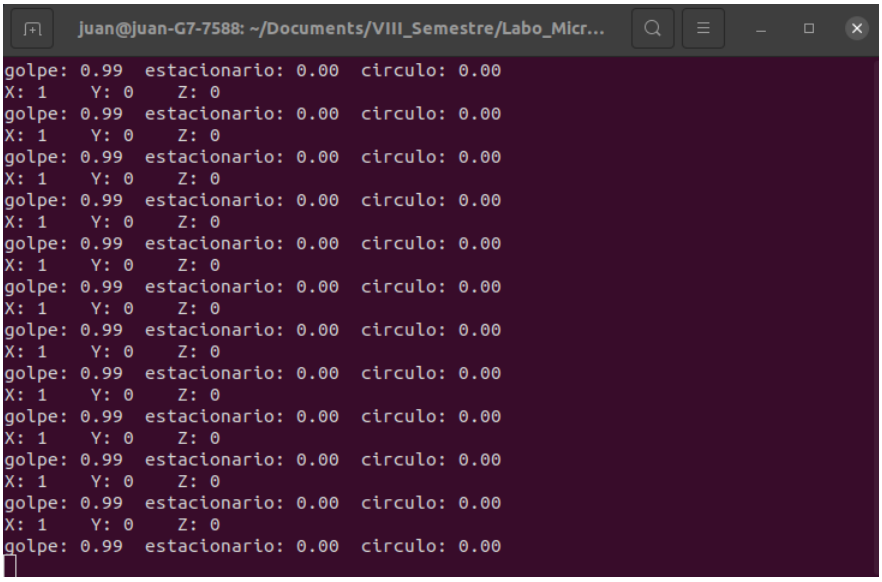


Figura 14: Predicciones obtenidas ante movimiento estacionario. Elaboración propia.

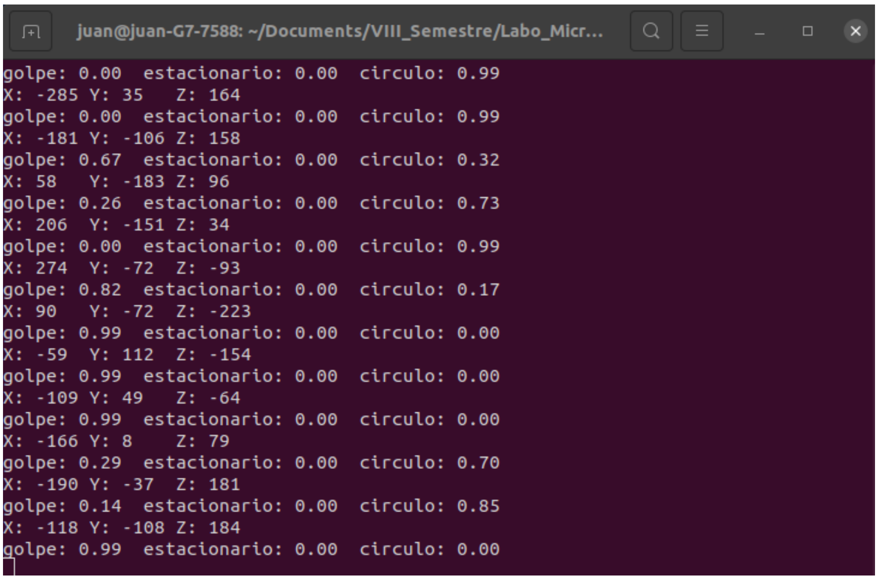


Figura 15: Predicciones obtenidas ante movimiento de golpe. Elaboración propia.

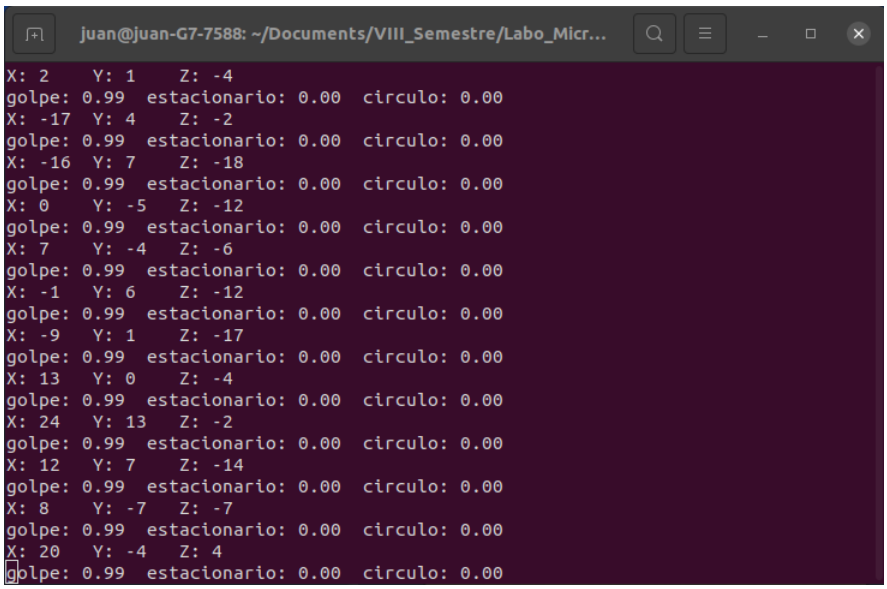


Figura 16: Predicciones obtenidas ante movimiento circular. Elaboración propia.

4. Conclusiones y Recomendaciones

En este laboratorio se exploró el área de la inteligencia artificial, específicamente el del aprendizaje automático, y como está puede ser implementada en microcontroladores con la ayuda de librerías externas como TensorFlow Lite en este caso. Antes de crear una red neuronal es importante contar con las muestras de datos que se quieren utilizar para entrenar el modelo y que los distintos grupos de muestras tengan la misma cantidad de datos. En ocasiones es necesario realizar lo que se conoce como una limpieza de datos para luego proceder a crear el modelo deseado. La creación de una red neuronal, a pesar de tener una composición de capas y neuronas fácil de comprender, es muy difícil ya que los criterios para determinar la cantidad óptima de capas y neuronas son más complejos. En el caso del presente laboratorio, estos criterios fueron los que afectaron en mayor medida el funcionamiento del modelo creado. La red desarrollada, a pesar de no brindar datos satisfactorios, sí muestra que se pueden detectar distintos movimientos, ya que al realizar un movimiento circular sí se logra detectar, a pesar de nunca llegar a detectar si se encuentra en un estado estacionario. Por esta razón se considera que se logró con cierto grado de éxito la implementación de una red neuronal que detecte movimientos humanos en un microcontrolador.

Como recomendaciones se tienen primero el estudio teórico de lo que se esté realizando, ya que para poder realizar un mejor modelo es necesario tener un mejor entendimiento del funcionamiento de las capas y neuronas de una red neuronal. Además, es necesario familiarizarse primero con las librerías que se van a utilizar para crear el modelo y entender su funcionamiento y manera de entrenamiento, para este caso se utilizó Tensor Flow Lite.

5. Referencias

[1] STMicroelectronics. 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 DocID024030 Rev 10, STM, jan 2018.

[2] STMicroelectronics. Discovery kit for stm32f429/439 lines. Datasheet DocID025175 Rev 1, STM, sep 2013.

[3] Matthew Stewart. Tiny machine learning: The next ai revolution. Disponible en: `https://towardsdatascience.com/tiny-machine-learning-the-next-ai-revolution-495c26463868`, 2016.

[4] Gaudenz Boesch. Tensorflow lite – real-time computer vision on edge devices. Disponible en: `https://viso.ai/edge-ai/tensorflow-lite/`, 2022.

[5] TinyML Foundation. About tinyml foundation. Disponible en: `https://www.tinyml.org/about/`, 2022.

6. Anexos

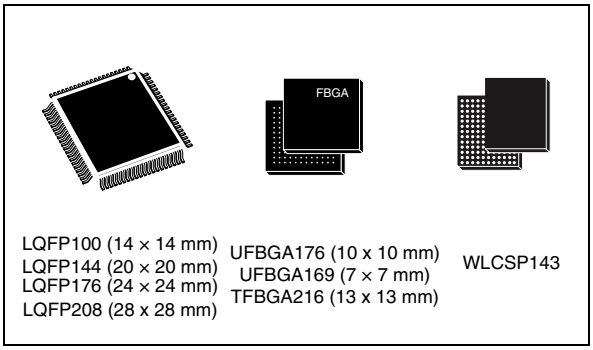
STM32F427xx STM32F429xx

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 × 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

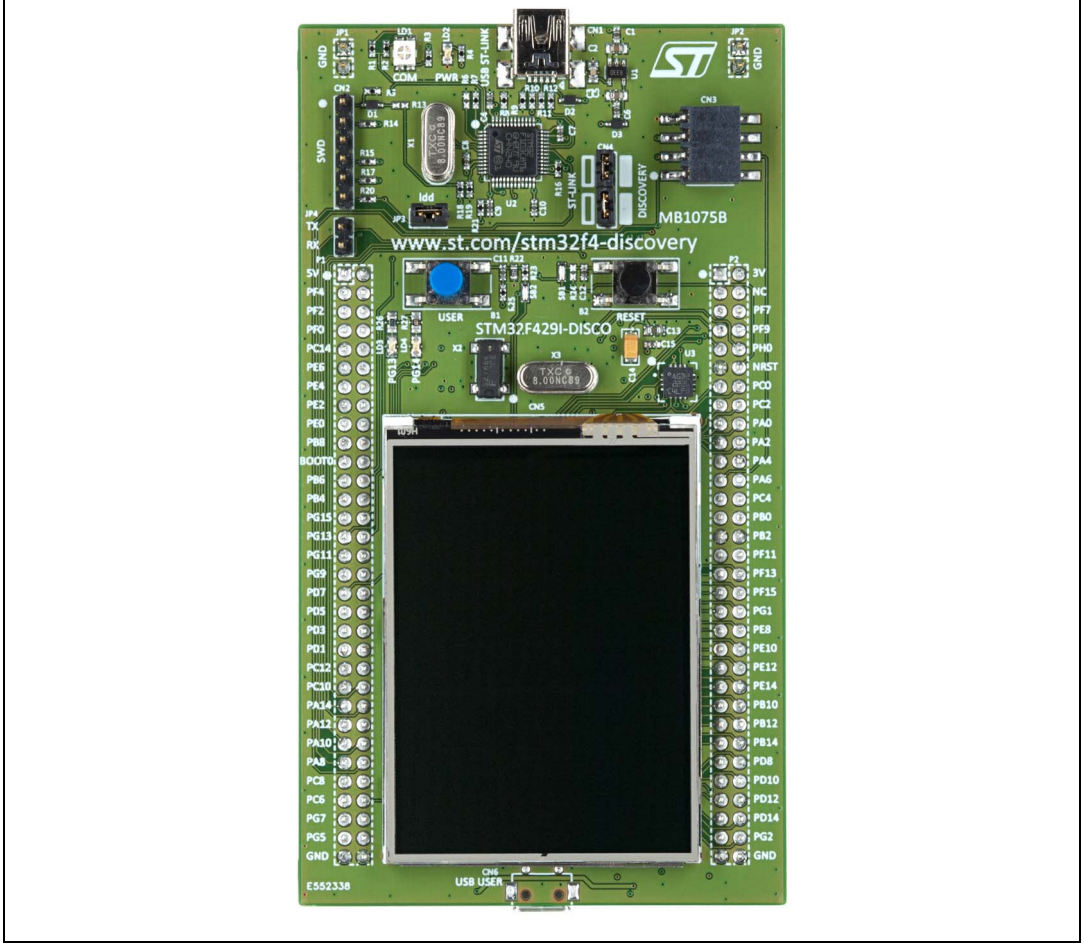
UM1670
User manual

Discovery kit for STM32F429/439 lines

Introduction

The STM32F429 Discovery kit (32F429IDISCOVERY) helps you to discover the high performance of the STM32F4 series and to develop your applications. It is based on an STM32F429ZIT6 and includes an ST-LINK/V2 embedded debug tool interface, 2.4" TFT LCD, SDRAM 64 Mbits, Gyroscope ST MEMS, LEDs, pushbuttons and a USB OTG micro-B connector.

Figure 1. STM32F429 Discovery board



3 Features

The STM32F429 Discovery board offers the following features:

- STM32F429ZIT6 microcontroller featuring 2 MB of Flash memory, 256 KB of RAM in an LQFP144 package
- On-board ST-LINK/V2 with selection mode switch to use the kit as a standalone ST-LINK/V2 (with SWD connector for programming and debugging)
- Board power supply: through the USB bus or from an external 3 V or 5 V supply voltage
- L3GD20, ST MEMS motion sensor, 3-axis digital output gyroscope
- TFT LCD (Thin-film-transistor liquid-crystal display) 2.4", 262K colors RGB, 240 x 320 dots
- SDRAM 64 Mbits (1 Mbit x 16-bit x 4-bank) including an AUTO REFRESH MODE, and a power-saving
- 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) VBUS and LD6 (red) OC (over-current)
- Two pushbuttons (user and reset)
- USB OTG with micro-AB connector
- Extension header for LQFP144 I/Os for a quick connection to the prototyping board and an easy probing

4 Hardware layout

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

Figure 1 illustrates the connections between the STM32F429ZIT6 and its peripherals (ST-LINK/V2, pushbutton, LED, USB OTG, Gyroscope ST MEMS, Accelerometer + Magnetometer ST MEMS, and connectors).

Figure 2 and *Figure 3* help you to locate these features on the STM32F429 Discovery board.

Figure 1. Hardware block diagram

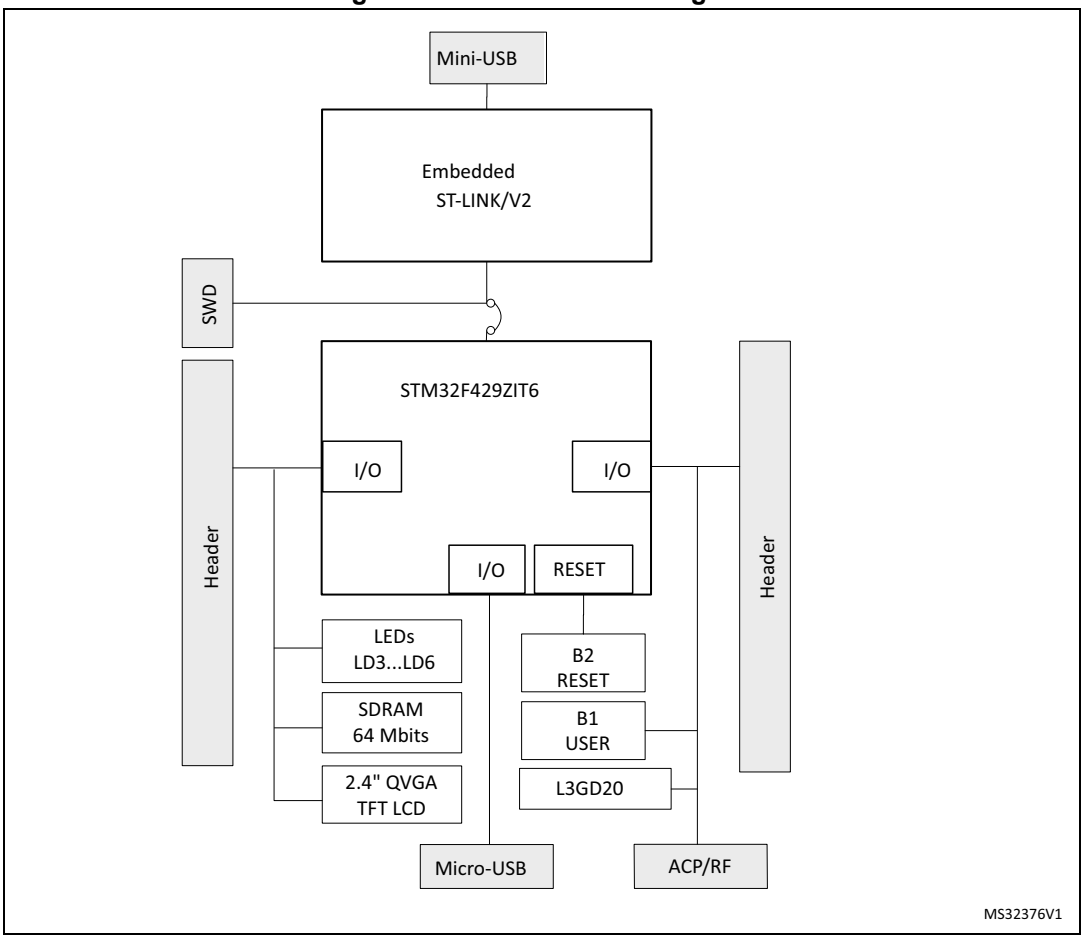
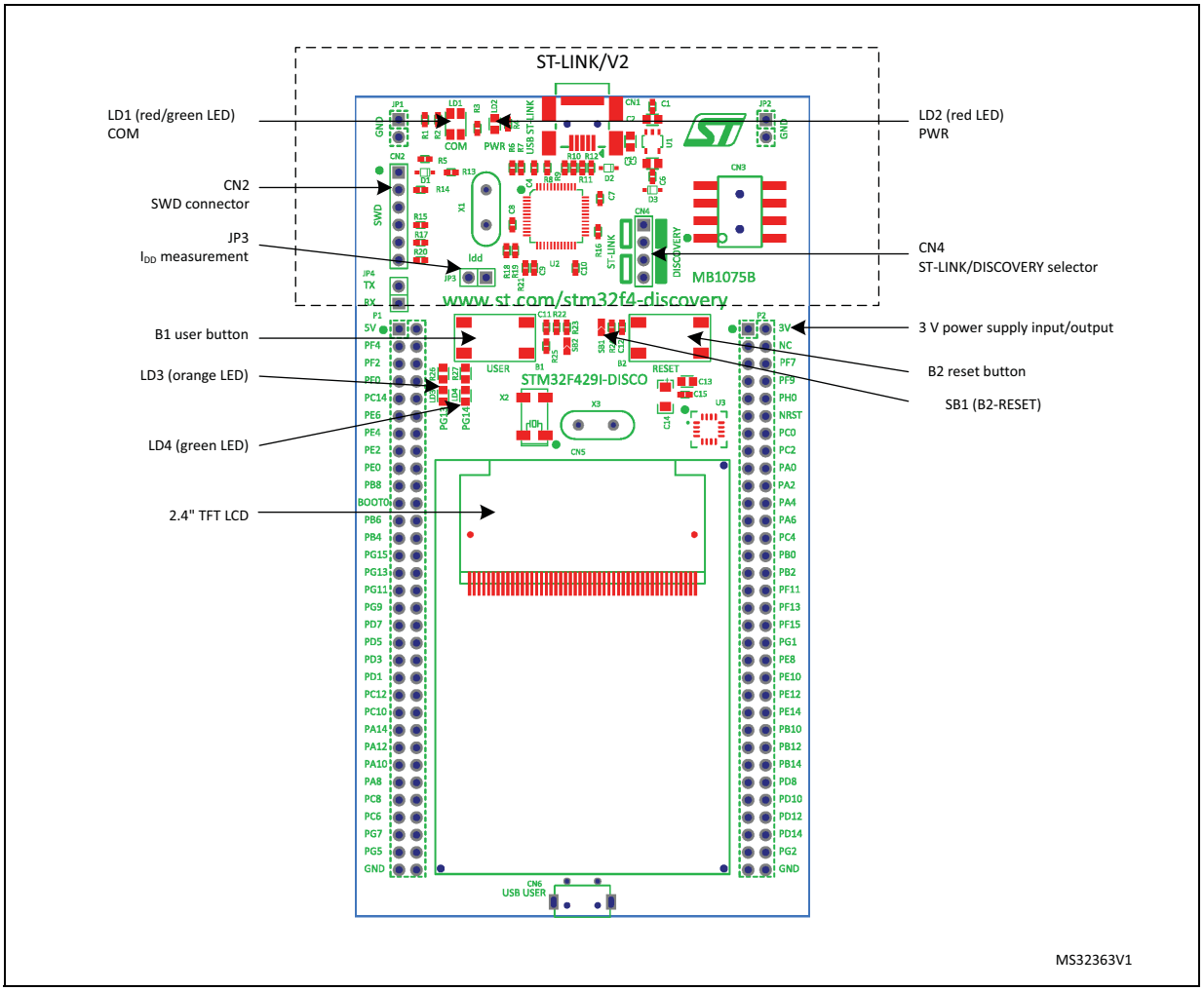


Figure 2. Top layout

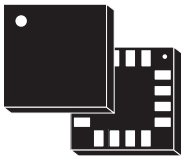


.

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