UNIVERSIDAD DE COSTA RICA
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

EIE
Escuela de
Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

Jorge Adán Mora Soto, B95222 jorgeadan.mora@ucr.ac.cr
Jafet David Gutiérrez Guevara, B73558 jafet.gutierrez@ucr.ac.cr

1 de diciembre de 2022

Laboratorio 5
Arduino: GPIO, Giroscopio, comunicaciones, TinyML

Resumen

En el siguiente laboratorio se desarrolla un reconocedor de actividad humana el cual utiliza un kit Arduino Nano 33 BLE. Para esto se realiza un programa para el microcontrolador que capture la informacion del giroscopio y se envie a la computadora por el puerto USB; esta información se procesa mediante un script de python que la guarde con etiqueta el tipo de movimiento que se efectua y la ordene en un archivo CSV. Se registran 3 movimientos: ≪Up-Down≫, ≪Circle≫ y ≪Stationary≫. Con esta información muestreada se pretende realizar un modelo de red neuronal para cargar en el Arduino, de modo que este pueda identificar el movimiento realizado según la red cargada con datos históricos y frecuentes para determinado movimiento.
Sobre el modelo: se aspira a una red de dos capas con varias neuronas y 3 salidas (una para cada movimiento). Se entrena el modelo con el 60 % de los datos, se valida con el 20 % y se prueba con el último 20 % de los datos. Este modelo se procesa/exporta al microcontrolador mediante y gracias a la librería de TensorFlow. Los practicantes se apoyan en la plataforma de EdgeImpulse para pefeccionar el modelo. El objetivo se cumplió dentro de los parámetos evaluativos. Se exporta un modelo que detectan los 3 movimientos. Con respecto a la red neuronal, no se cumplió perfectamente el objetivo con el movimiento "Stationary"pues este se predice parcialmente; pero este conocimiento de redes neuronales no corresponde a criterios evacuativos del curso, por lo que se concluye una práctica exitosa.

Link del Proyecto: https://github.com/Jams1001/IE0624/tree/main/L5

`ID del último commit de interés:75f7e2e`

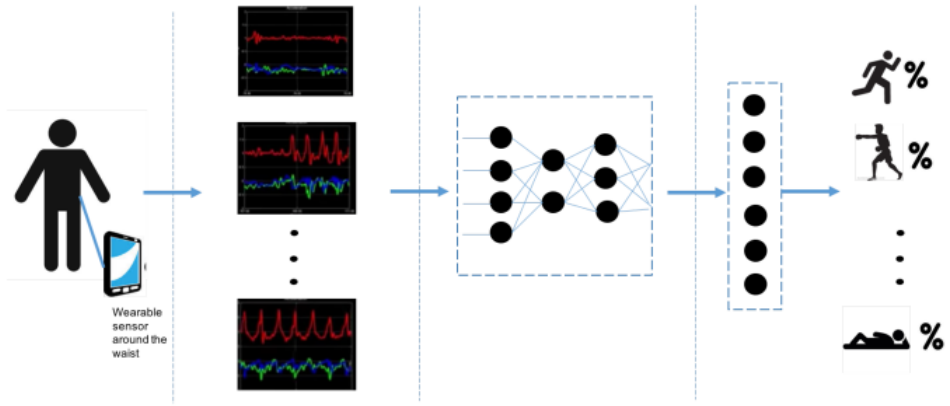


Figura 1: HAR.

1. Nota Teórica

1.1. Información general del MCU

El nRF52840 es el miembro más avanzado de la serie nRF52. Cumple con los desafíos de las aplicaciones sofisticadas que necesitan concurrencia de protocolos y un conjunto rico y variado de periféricos y funciones. Ofrece una generosa disponibilidad de memoria tanto para Flash como para RAM, que son requisitos previos para aplicaciones tan exigentes[1]. Se trata de un procesador ARM Cortex-M4 con unidad de coma flotante (FPU) que tiene un conjunto de instrucciones de 32 bits que implementa un superconjunto de instrucciones de 16 y 32 bits para maximizar la densidad del código y actuación [1].
Tiene numerosos periféricos e interfaces digitales, como SPI y QSPI de alta velocidad para conectarse a pantallas y flashes externos, PDM e I2S para micrófonos y audio digitales, y un dispositivo USB de alta velocidad para transferencia de datos y fuente de alimentación para recargar la batería[1]. A continuación en la tabla 1 se presenta las características eléctricas, en la figura 5 se presenta el diagrama de pines de la placa en donde se integra el MCU a utilizar, y en la figura 3 se presenta el diagrama de bloques.

MCU	nRF52840
Voltaje de operación	3.3V
Voltaje de entrada	21V
DC corriente por I/O Pin	15 mA
Frecuencia de reloj	64MHz
Memoria Flash	1MB (nRF52840)
SRAM	256KB (nRF52840)
EEPROM	none
I/O Digital Pins	14
PWM Pins	14 (Todos los pines digitales)
UART	1
SPI	1
I2C	1
Analog Input Pins	8 (ADC 12 bit 200 ksamples)
Analog Output Pins	Only through PWM (no DAC)
External Interrupts	all digital pins
LED_BUILTIN	13
USB	Native in the nRF52840 Processor

Tabla 1: Características generales y eléctricas

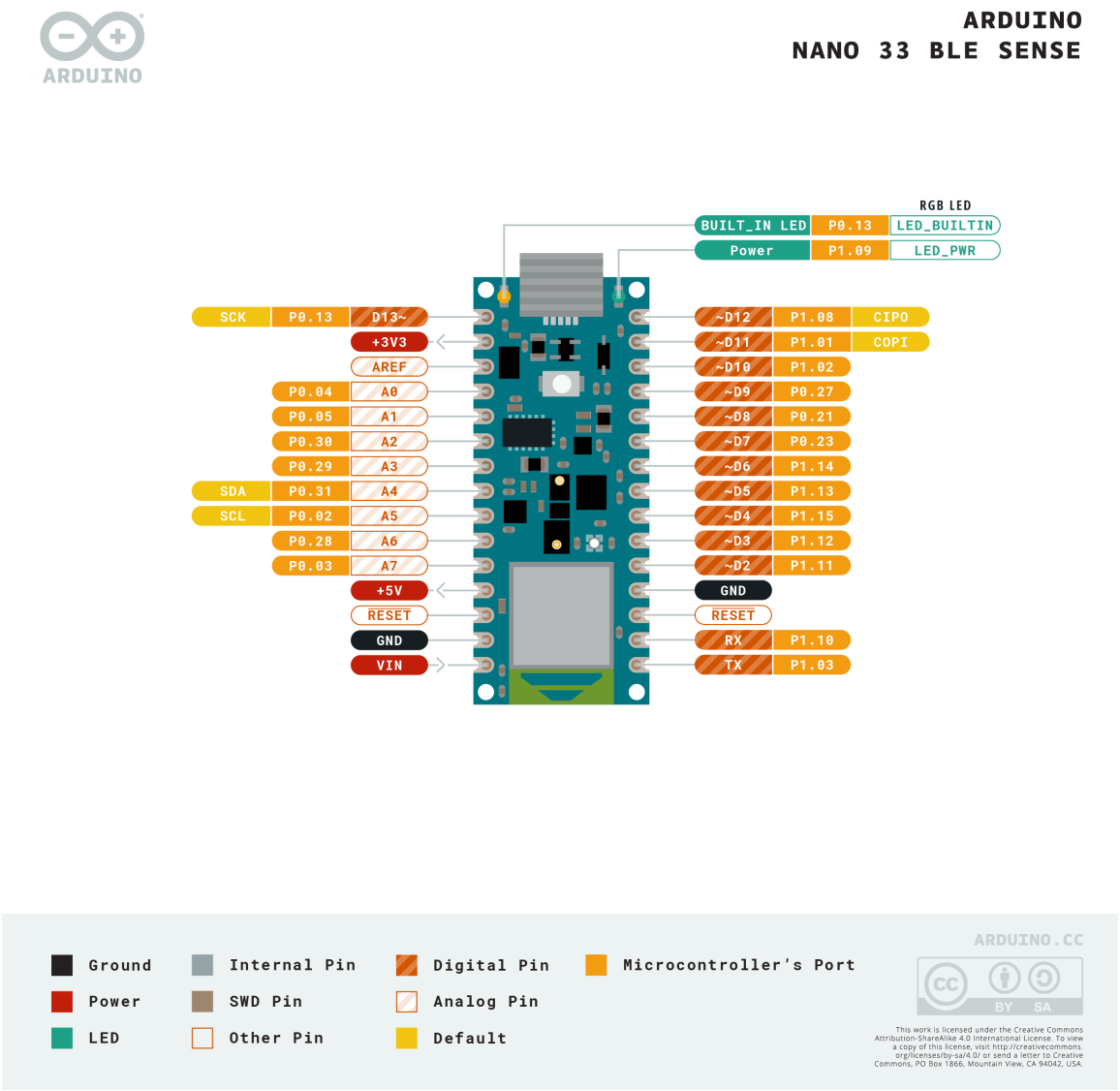


Figura 2: Diagrama de pines [2]

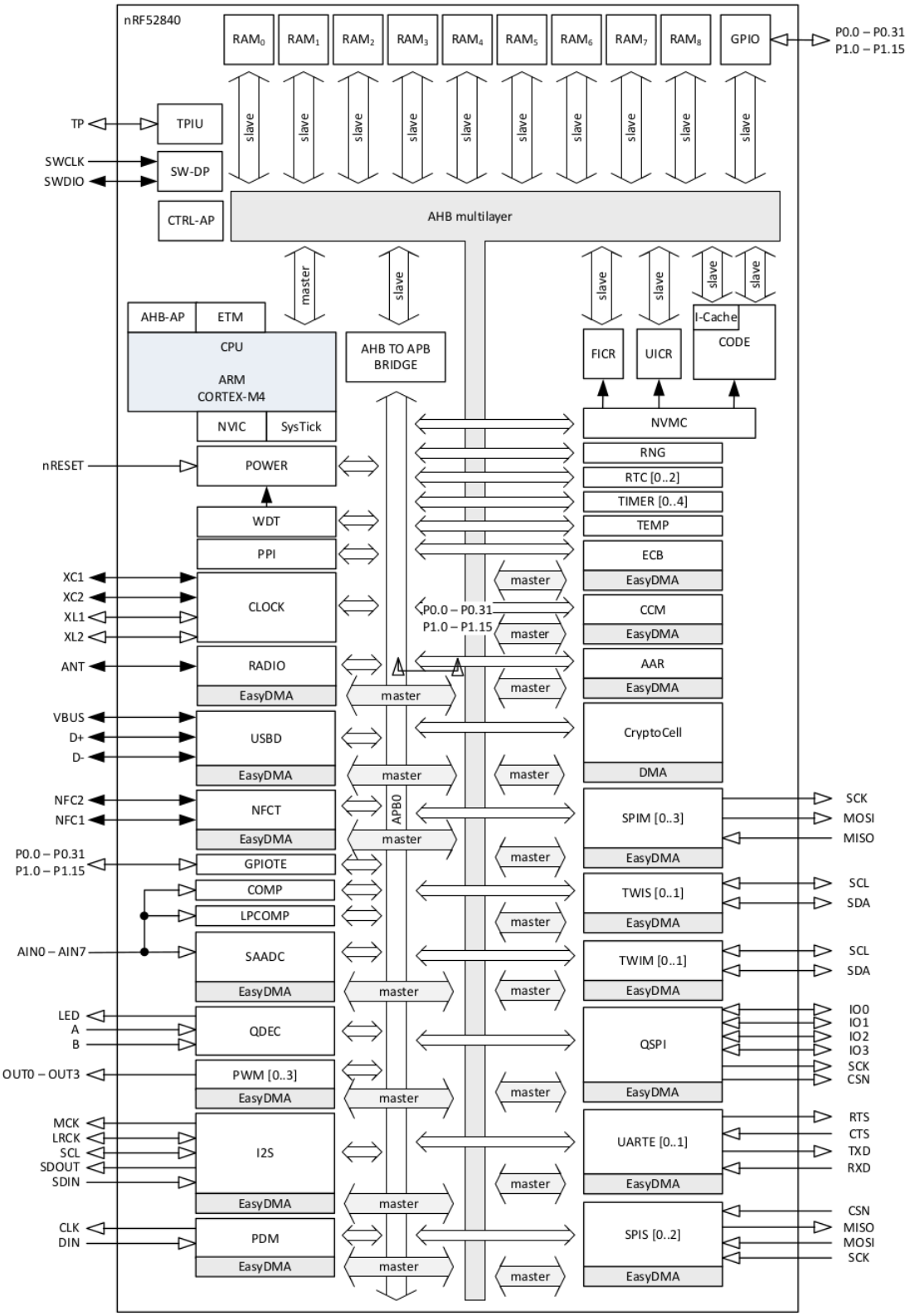


Figure 1: Block diagram

Figura 3: Diagrama de bloques [1]

.

1.2. Sobre el Nano 33 BLE Sense

El Arduino Nano 33 BLE Sense es una excelente opción para cualquier principiante, fabricante o profesional para comenzar con el aprendizaje automático integrado. Se basa en el microcontrolador nRF52840 y se ejecuta en el sistema operativo Arm Mbed. El Nano 33 BLE Sense no solo cuenta con la posibilidad de conectarse a través de Bluetooth Low Energy, sino que también viene equipado con sensores para detectar color, proximidad, movimiento, temperatura, humedad, audio y más [2]. Sobre sus periféricos y principales características:

- **u-blox NINA-B306:** Módulo Bluetooth 5 de bajo consumo de u-blox, con antena interna [2].
- **IMU for Motion Detection:** La unidad de medición inercial LSM9DS1 cuenta con un acelerómetro, un giroscopio y un magnetómetro 3D y le permite detectar la orientación, el movimiento o las vibraciones en su proyecto [2]. Se trata de un sistema en paquete que presenta un sensor de aceleración lineal digital 3D, un sensor digital 3D, un sensor de velocidad angular y un sensor magnético digital 3D [3].
  - 3 canales de aceleración, 3 de velocidad angular canales, 3 canales de campo magnético [3].
  - Interfaces seriales SPI / I^2C [3].
  - 16-bit de data-output [3].
- **Omnidirectional Digital Microphone:** El micrófono MP34DT05 permite capturar y analizar el sonido en tiempo real y puede usarse para crear una interfaz de voz para su proyecto[2].
- **Proximity and Gesture Detection:** El chip APDS-9960 permite medir la proximidad digital y la luz ambiental, así como detectar colores y gestos RGB [2].
- **Barometric Pressure Sensor:** El LPS22HB detecta la presión barométrica y permite una salida de datos de presión de 24 bits entre 260 y 1260 hPa. Estos datos también se pueden procesar para calcular la altura sobre el nivel del mar de la ubicación actual [2].
- **Temperature and Humidity Detection:** The HTS221 capacitive digital sensor measures relative humidity and temperature. It has a temperature accuracy of ± 0.5 °C (between 15-40 °C) and is thereby perfectly suited to detect ambient temperature [2].

1.3. Funciones de interés soportadas

- **pinMode():** Configura el pin especificado para que se comporte como una entrada o una salida [4].
- **analogRead():** Lee el valor del pin analógico especificado. Las placas Arduino contienen un convertidor analógico a digital multicanal de 10 bits. Esto significa que mapeará los voltajes de entrada entre 0 y el voltaje de funcionamiento (5 V o 3,3 V) en valores enteros entre 0 y 1023. En un Arduino UNO, por ejemplo, esto produce una resolución entre lecturas de: 5 voltios / 1024 unidades o 0,0049 voltios (4,9 mV) por unidad [4].
- **map():** Vuelve a asignar un número de un rango a otro. Es decir, un valor de fromLow se asignaría a toLow, un valor de fromHigh a toHigh, valores intermedios a valores intermedios, etc. Reasigna la escala anlógica a la escala digital en la resolución disponible [4].

- **delay():** Pausa el programa por la cantidad de tiempo (en milisegundos) especificado como parámetro. (Hay 1000 milisegundos en un segundo) [4].
- **analogWrite():** Escribe un valor analógico (onda PWM) en un pin. Se puede usar para encender un LED con diferentes brillos o impulsar un motor a varias velocidades. Después de una llamada a analogWrite(), el pin generará una onda rectangular constante del ciclo de trabajo especificado hasta la próxima llamada a analogWrite() (o una llamada a digitalRead() o digitalWrite()) en el mismo pin. Recibe como parámetros `analogWrite(pin, value)`, en donde `pin` corresponde el pin de Arduino para escribir y value al ciclo de trabajo que va entre 0 (siempre apagado) y 255 (siempre encendido) [4].

1.4. Diseño

Nuevamente para este proyecto no existe un diseño electrónico, fue suficiente con la utilización del Nano 33 BLE Sense. El fin del laboratorio es usar exitosamente los perifércos de la misma y es por esto que se detallaron en las secciones anteriores.

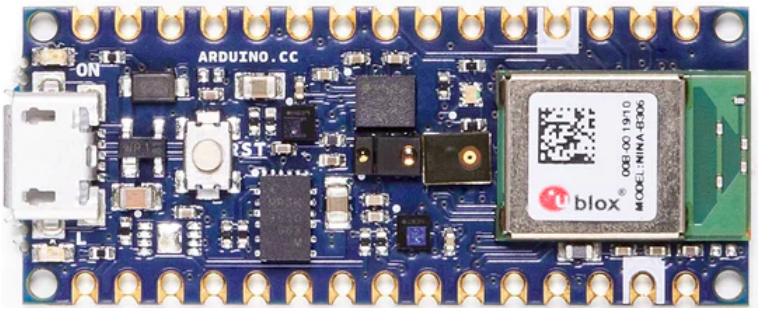


Figura 4: Arduino Nano 33 BLE Sense [2].

1.5. Componentes complementarios

Para está práctica únicamente se trabaja con la placa Nano 33 BLE Sense. Sus abundantes periféricos cumplen con todo lo necesario para la realización, por lo que únicamente se debe conectar a la alimnetación y comunicación con la computadora. El precio de esta placa ronda los $40.5USD$.

1.6. Conceptos

1.6.1. AI

La inteligencia artificial (o por sus siglas en inglés «AI»), la capacidad de una computadora digital o un robot controlado por computadora para realizar tareas comúnmente asociadas con seres inteligentes. El término se aplica con frecuencia al proyecto de desarrollar sistemas dotados de los procesos intelectuales característicos de los humanos, como la capacidad de razonar, descubrir significado, generalizar o aprender de experiencias pasadas [5].

1.6.2. Machine learning en microcontroladores

El aprendizaje automático, mejor conocido como *machine learning*, es una rama de la inteligencia artificial dirigida al desarrollo de herramientas que permiten que las máquinas aprendan sin estar explícitamente programadas para ello. Con dichas herramientas, se puede hacer que distintos dispositivos sean capaces identificar patrones entre un conjunto de datos, y a partir de ellos, realizar predicciones. El *machine learning* puede utilizarse para crear tecnologías inteligentes que faciliten la vida de sus usuarios, como Google Assistant, para dar un ejemplo [6].

Sin embargo, la aplicación del *machine learning* a menudo requiere de una gran cantidad de recursos, que pueden incluir un potente servidor en la nube o un computador con una considerable capacidad de procesamiento. Afortunadamente, ahora es posible ejecutar la inferencia de aprendizaje automático en hardware diminuto y de baja potencia, como los microcontroladores. Al traer el *machine learning* a estos dispositivos se puede potenciar la inteligencia de miles de millones de equipos que se utilizan cotidianamente, y sin depender de hardware costoso o conexiones robustas a Internet [6].

Una de las herramientas más populares para la implementación de *machine learning* es TensorFlow. Este es un framework de aprendizaje automático, de código abierto y de Google para entrenar y ejecutar modelos de reconocimiento de patrones. Como parte de los esfuerzos de TensorFlow, Google también desarrolló una versión optimizada, destinada a ejecutar modelos de TensorFlow en microcontroladores. Dicha versión tiene el nombre de TensorFlow Lite For Microcontrollers. Este se adhiere a las restricciones requeridas en entornos embebidos, es decir, tiene un tamaño binario pequeño, no requiere el soporte de ningún sistema operativo, ninguna librería estándar de C o C++, o la asignación memoria dinámica [6].

1.6.3. Red Neuronal artificial

Consiste en una metodología de inteligencia artificial que enseña a las computadoras a procesar datos de una manera inspirada en el cerebro humano, con patrones o modelos previamente cargados de esos mismos eventos. Es un tipo de proceso de aprendizaje automático, llamado aprendizaje profundo, que utiliza nodos o neuronas interconectados en una estructura en capas que se asemeja al cerebro humano. Crea un sistema adaptativo que las computadoras usan para aprender de sus errores y mejorar continuamente. De esta forma las redes neuronales artificiales intentan resolver problemas complicados, como resumir documentos o reconocer rostros, reconocer movimientos, y todo esto en búsqueda de una mayor precisión para conquistar los límites de las capacidades humanas [7].

2. Desarrollo / Análisis de Resultados

2.1. Análisis de SW

2.1.1. recoder.py

Este archivo entregado en la ruta `/IE0624/L5/src` , únicamente se encarga de recibir la información serial enviada por el microcontrolador. Recibe 6 variables: `header = ['aX', 'aY', 'aZ', 'gX', 'gY', 'gZ']` correspondencientes a las 3 componentes de la aceleración y el giroscopio del LSM9DS1. Con esta información escribe los archivos con los datos para el movimiento específico que se está haciendo en ese momento en formato de `csv`.

2.1.2. Modelo

A continuación se explica las secciones con las que cuenta el archivo que genera el modelo. Para esto referencie el archivo entregado como `arduino_tinyml_workshop.ipynb`, el cual corrresponde a un notebook de google colab.
Esta sección no es de principal interés para los fines evaluativos del laboratorio. Consiste primeramente en graficar los datos del csv procesado por el `recoder.py`. Este primer punto únicamente es con fines ilustrativos.
Seguidamente se debe entrenar la red neuronal del modelo con las muestras por cada uno de los 3 gestos deseados a predecir. De modo que se normalizan los datos de de 0 a 1 y se definen las entradas del modelo.
Luego se debe dividir las muestras para generar, validar y entrenar el modelo. Esta es la sección de Aleatorizar y dividir los pares de entrada y salida para el entrenamiento. dividiendo aleatoriamente los pares de entrada y salida en conjuntos de datos: 60 % para entrenamiento, 20 % para validación (para medir qué tan bien se está desempeñando el modelo durante el entrenamiento) y 20 % para prueba (para probar el modelo después del entrenamiento).
El siguiente punto importante es construir y entrenar la red neuronal, lo cual se hace mediante un modelo de TensorFlow con la API de alto nivel de Keras.
La siguiente sección de interés es la validación del desempeño del modelo mediante la gráfica de la pérdida (definida mediante el .^{er}ror cuadrático medioçomo la función de pérdida) para ver cuándo el modelo deja de mejorar. Esto se realiza de diferentes formas y entre esas el cálculo del error absoluto medio es otra métrica para juzgar el rendimiento del modelo.
Seguidamente viene la validación en donde se ponen los datos de prueba en el modelo y trazar las predicciones.
Una vez validado, se convierte el modelo al formato TensorFlow Lite sin cuantificación lo cual finaliza el protagonista del generador del modelo.
Para poder convertir el modelo anterior en una librería de arduino se debe ejecutar la siguiente sintaxis:

```
!echo "const unsigned char model[] = {" > /content/model.h
!cat gesture_model.tflite | xxd -i      >> /content/model.h
!echo "};"                              >> /content/model.h
```

A pesar de que esto no es de caracter principal en la evaluación, el generador del modelo cumple su función pues proporciona un modelo funcional para cargar en el microcontrolador.

2.1.3. Firmwares

data_senser.ino

El primer sketch, llamado `data_senser.ino`, fue un firmware que se desarrolló inicialmente para acceder a las lecturas del giroscopio y de aceleración que registra el Arduino Nano 33 BLE. Además de acceder a dichos registros, este firmware se encarga de enviar los datos leídos en cada ciclo al puerto serial, para que estos puedan ser recopilados desplegados y guardados por el script `recorder.py`.

imu.ino

El segundo sketch, llamado `imu.ino`, es el firmware en el cual se implementa el modelo generado para la detección de movimientos. En este sketch primero se incluye la librería `Arduino_LSM9DS1`, la cual permite leer los valores del acelerómetro, magnetómetro y giroscopio de la IMU LSM9DS1 en el Arduino Nano 33 BLE Sense. Después se incluyen los encabezados asociados a todas las funciones de TensorFlow Lite que se van a usar más adelante. El último archivo que se incluye es `model.h`, el cual contiene el modelo entrenado de la red neuronal que se generó anteriormente. Posteriormente, se crea un array para asignar el índice de gestos a un nombre, como se muestra a continuación:

```
const char* GESTURES[] = {
  "circle",
  "up-down",
  "stationary",
};
```

Si el Arduino detecta un movimiento válido accederá a uno de los elementos dentro del array `GESTURES` para escribirlo al puerto serial.
Dentro de la configuración inicial se inicializa el puerto serial, así como también el IMU. También se obtiene la representación TFL de la matriz de bytes del modelo, se crea un interprete para correr el modelo, se asigna memoria para sus tensors de entrada y salida y se obtienen punteros para estos últimos dos.
Por otro lado, en el lazo de ejecución se tienen variables de tipo flotante para almacenar los datos leídos de aceleración y el giroscopio. Después se espera a que la cantidad de muestras recolectadas sea la suficiente como para determinar que se produjo un movimiento significativo. Luego, se normalizan los datos recopilados de la IMU entre 0 y 1, y los resultados son almacenados en el tensor de entrada del modelo. Por último, se ejecuta la inferencia y se escribe en el puerto serial el movimiento detectado por el modelo, como se muestra a continuación:

```
for (int i = 0; i < NUM_GESTURES; i++) {
    Serial.print(GESTURES[i]);
    Serial.print(": ");
}
```

2.2. Análisis de HW

Nuevamente, no existen parámetros electrónicos por analizar pues la electrónica únicamente se basa en el Nano 33 BLE Sense; el cual tiene la calidad de un producto de la compañía Arduino. De igual forma, se presenta una imagen con una demostración de su funcionamiento en la figura 5.

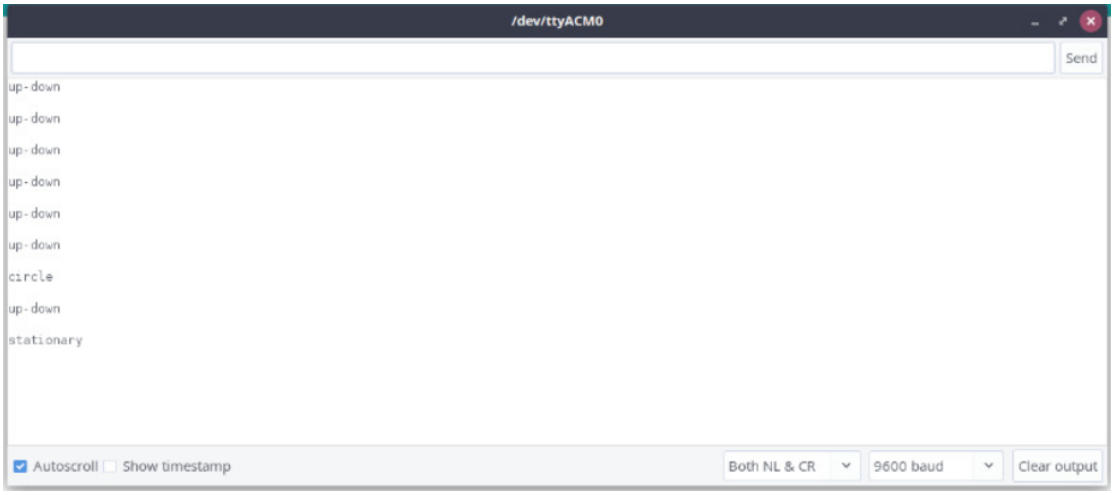


Figura 5: Salida serial del arduino posterior a la carga del firmware .

3. Conclusiones y Recomendaciones

- Se concluye que el mcu integrado en el arduino nano 33 BLE Sense es una poderosa herramienta para implementar modelos de inteligencia artificial y redes neoronales. Sus perifericos y funciones adicionales como que se puede programar con el lenguaje de programación Python a través de OpenMV IDE, son pluses que hacen una gran herramientaa para proyectos serios y de altura para proyectos con AI. Sumado a que el IDE que utiliza es de mayor comodidad con respecto a su antiguo predecesor.
- Se recomienda la utilización de Google Colab Notebooks para agilizar la generación del modelo con el fin de aprovechar los recursos existentes y no consumir la capacidad de los recursos de los practicantes. Esto debido a que en primera instancia se tuvo problemas por la tardanza en la generación del modelo, la capacidad de memoria de las computadoras locales, entre otras cosas.
- Se recomienda la futura implementación del mismo proyecto pero implementado con un STM32 para obtener el panorama completo y comparar las dificultades en la generación de modelos para diferentes microcontroladores.

Bibliografía

[1] N. Semiconductor, "nRF52840", Nordic Semiconductor ASA, Datasheet $4413_4 17v1.1$, feb. de 2019.

[2] Arduino, "Arduino® Nano 33 BLE Sense", Arduino, Product Reference Manual SKU: ABX00031, nov. de 2022.

[3] life.augmented - STMicroelectronics, "LSM9DS1", life.augmented - STMicroelectronics, Datasheet DocID025715 Rev 3, nov. de 2015.

[4] Arduino. "Language Reference". Visitado en noviembre 29 de 2022. (2022), dirección: https://www.arduino.cc/reference/en/.

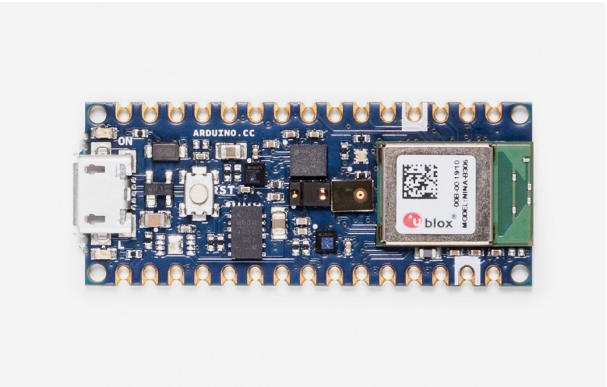
[5] B. Copeland. “artificial intelligence”. Visitado en noviembre 29 de 2022. (nov. de 2011), dirección: https://www.britannica.com/technology/artificial-intelligence.

[6] M. Natraj. “AI Magic Wand with TensorFlow Lite for Microcontrollers and Arduino”. Visitado en noviembre 23 de 2022. (mar. de 2022), dirección: https://codelabs.developers.google.com/magicwand#0.

[7] AWS. “What Is A Neural Network?” Visitado en noviembre 29 de 2022. (nov. de 2022), dirección: https://aws.amazon.com/what-is/neural-network/.

4. Apéndices

Product Reference Manual
SKU: ABX00031



Description

Nano 33 BLE Sense is a miniature sized module containing a NINA B306 module, based on Nordic nRF52480 and containing a Cortex M4F, a crypto chip which can securely store certificates and pre shared keys and a 9 axis IMU. The module can either be mounted as a DIP component (when mounting pin headers), or as a SMT component, directly soldering it via the castellated pads

Target areas:

Maker, enhancements, IoT application

1 The Board

As all Nano form factor boards, Nano 33 BLE Sense does not have a battery charger but can be powered through USB or headers.

NOTE: Arduino Nano 33 BLE Sense only supports 3.3V I/Os and is **NOT** 5V tolerant so please make sure you are not directly connecting 5V signals to this board or it will be damaged. Also, as opposed to Arduino Nano boards that support 5V operation, the 5V pin does NOT supply voltage but is rather connected, through a jumper, to the USB power input.

1.1 Ratings

1.1.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (40 °F)	85°C (185 °F)

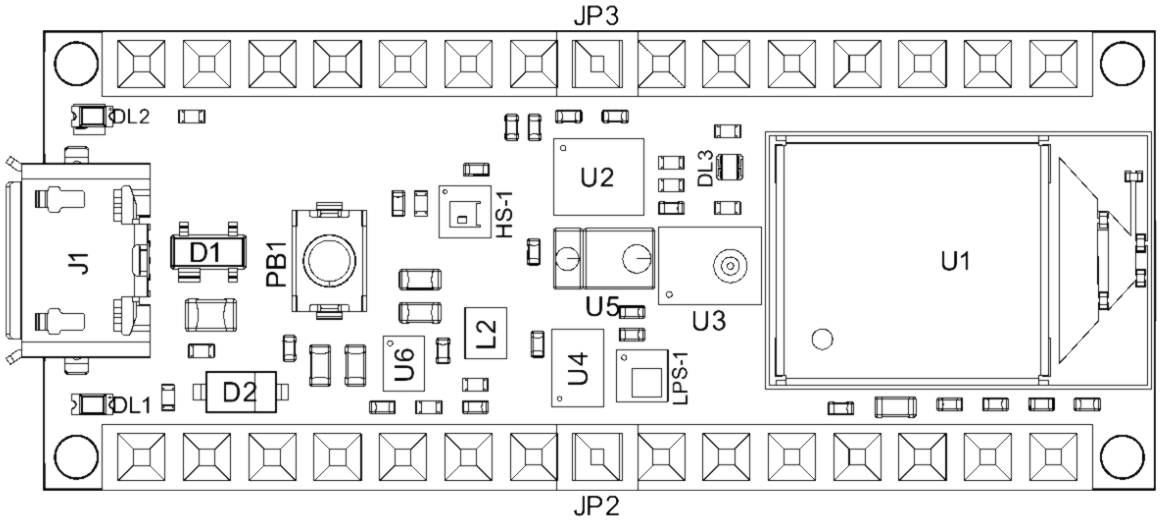
1.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
PBL	Power consumption with busy loop		TBC		mW
PLP	Power consumption in low power mode		TBC		mW
PMAX	Maximum Power Consumption		TBC		mW

2 Functional Overview

2.1 Board Topology

Top:

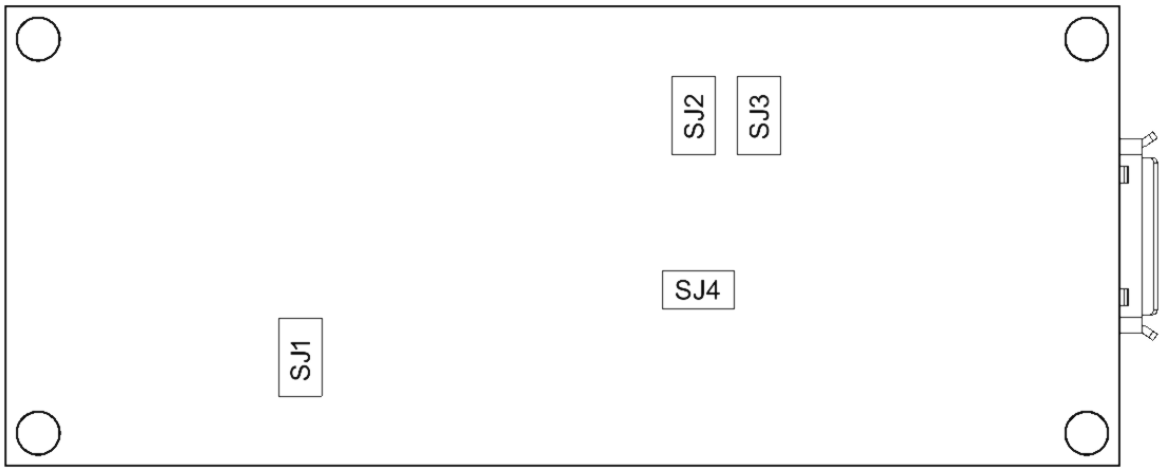


Board topology top

Ref.	Description	Ref.	Description
U1	NINA-B306 Module Bluetooth® Low Energy 5.0 Module	U6	MP2322GQH Step Down Converter
U2	LSM9DS1TR Sensor IMU	PB1	IT-1185AP1C-160G-GTR Push button
U3	MP34DT06JTR Mems Microphone	HS-1	HTS221 Humidity Sensor
U4	ATECC608A Crypto chip	DL1	Led L

Ref.	Description	Ref.	Description
U5	APDS-9660 Ambient Module	DL2	Led Power

Bottom:



Board topology bot

Ref.	Description	Ref.	Description
SJ1	VUSB Jumper	SJ2	D7 Jumper
SJ3	3v3 Jumper	SJ4	D8 Jumper

2.2 Processor

The Main Processor is a Cortex M4F running at up to 64MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the wireless module and the on-board internal I^2C peripherals (IMU and Crypto).

NOTE: As opposed to other Arduino Nano boards, pins A4 and A5 have an internal pull up and default to be used as an I^2C Bus so usage as analog inputs is not recommended.

2.3 Crypto

The crypto chip in Arduino IoT boards is what makes the difference with other less secure boards as it provides a secure way to store secrets (such as certificates) and accelerates secure protocols while never exposing secrets in plain text.

Source code for the Arduino Library that supports the Crypto is available **[8]**

2.4 IMU

Arduino Nano 33 BLE has an embedded 9 axis IMU which can be used to measure board orientation (by checking the gravity acceleration vector orientation or by using the 3D compass) or to measure shocks, vibration, acceleration and rotation speed.

Source code for the Arduino Library that supports the IMU is available **[9]**

2.5 Barometer and Temperature Sensor

The embedded Barometer and temperature sensor allow measuring ambient pressure. The temperature sensor integrated with the barometer can be used to compensate the pressure measurement.

Source code for the Arduino Library that supports the Barometer is available **[10]**

2.6 Relative Humidity and Temperature Sensor

Relative humidity sensor measures ambient relative humidity. As the Barometer this sensor has an integrated temperature sensor that can be used to compensate for the measurement.

Source code for the Arduino Library that supports the Humidity sensor is available **[11]**

2.7 Digital Proximity, Ambient Light, RGB and Gesture Sensor

Source code for the Arduino Library that supports the Proximity/gesture/ALS sensor is available **[12]**

2.7.1 Gesture Detection

Gesture detection utilizes four directional photodiodes to sense reflected IR energy (sourced by the integrated LED) to convert physical motion information (i.e. velocity, direction and distance) to a digital information. The architecture of the gesture engine features automatic activation (based on Proximity engine results), ambient light subtraction, cross-talk cancellation, dual 8-bit data converters, power saving inter-conversion delay, 32-dataset FIFO, and interrupt driven I2C communication. The gesture engine accommodates a wide range of mobile device gesturing requirements: simple UP-DOWN-RIGHT-LEFT gestures or more complex gestures can be accurately sensed. Power consumption and noise are minimized with adjustable IR LED timing.

2.7.2 Proximity Detection

The Proximity detection feature provides distance measurement (E.g. mobile device screen to user's ear) by photodiode detection of reflected IR energy (sourced by the integrated LED). Detect/release events are interrupt driven, and occur whenever proximity result crosses upper and/ or lower threshold settings. The proximity engine features offset adjustment registers to compensate for system offset caused by unwanted IR energy reflections appearing at the sensor. The IR LED intensity is factory trimmed to eliminate the need for end-equipment calibration due to component variations. Proximity results are further improved by automatic ambient light subtraction.

2.7.3 Color and ALS Detection

The Color and ALS detection feature provides red, green, blue and clear light intensity data. Each of the R, G, B, C channels have a UV and IR blocking filter and a dedicated data converter producing16-bit data simultaneously. This architecture allows applications to accurately measure ambient light and sense color which enables devices to calculate color temperature and control display backlight.

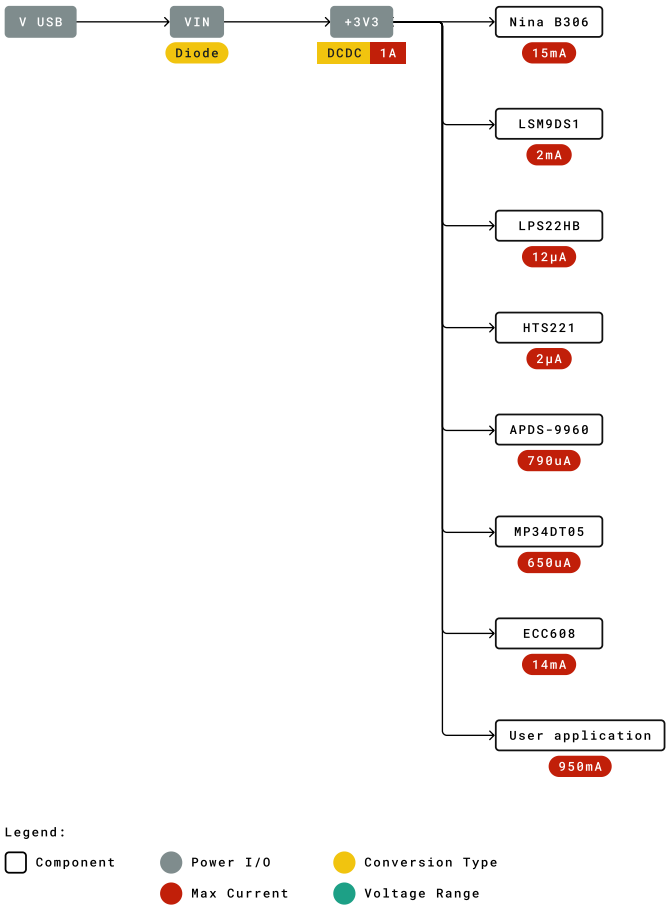
2.8 Digital Microphone

The MP34DT05 is an ultra-compact, low-power, omnidirectional, digital MEMS microphone built with a capacitive sensing element and an IC interface.

The sensing element, capable of detecting acoustic waves, is manufactured using a specialized silicon micromachining process dedicated to produce audio sensors

2.9 Power Tree

The board can be powered via USB connector, V_{IN} or V_{USB} pins on headers.



Power tree

NOTE: Since V_{USB} feeds V_{IN} via a Schottky diode and a DC-DC regulator specified minimum input voltage is 4.5V the minimum supply voltage from USB has to be increased to a voltage in the range between 4.8V to 4.96V depending on the current being drawn.

3 Board Operation

3.1 Getting Started - IDE

If you want to program your Arduino Nano 33 BLE while offline you need to install the Arduino Desktop IDE [1] To connect the Arduino Nano 33 BLE to your computer, you'll need a Micro-B USB cable. This also provides power to the board, as indicated by the LED.

3.2 Getting Started - Arduino Web Editor

All Arduino boards, including this one, work out-of-the-box on the Arduino Web Editor [2], by just installing a simple plugin.

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for all boards. Follow **[3]** to start coding on the browser and upload your sketches onto your board.

3.3 Getting Started - Arduino IoT Cloud

All Arduino IoT enabled products are supported on Arduino IoT Cloud which allows you to Log, graph and analyze sensor data, trigger events, and automate your home or business.

3.4 Sample Sketches

Sample sketches for the Arduino Nano 33 BLE can be found either in the "Examples" menu in the Arduino IDE or in the "Documentation" section of the Arduino Pro website [4]

3.5 Online Resources

Now that you have gone through the basics of what you can do with the board you can explore the endless possibilities it provides by checking exciting projects on ProjectHub **[13]**, the Arduino Library Reference **[14]** and the on line store **[15]** where you will be able to complement your board with sensors, actuators and more.

3.6 Board Recovery

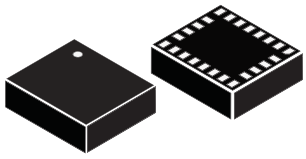
All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB it is possible to enter bootloader mode by double-tapping the reset button right after power up.

4 Connector Pinouts

LSM9DS1

iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer

Datasheet - production data



LGA-24L (3.5x3x1.0 mm)

Features

- 3 acceleration channels, 3 angular rate channels, 3 magnetic field channels
- ±2/±4/±8/±16 *g* linear acceleration full scale
- ±4/±8/±12/±16 gauss magnetic full scale
- ±245/±500/±2000 dps angular rate full scale
- 16-bit data output
- SPI / I^2C serial interfaces
- Analog supply voltage 1.9 V to 3.6 V
- “Always-on” eco power mode down to 1.9 mA
- Programmable interrupt generators
- Embedded temperature sensor
- Embedded FIFO
- Position and motion detection functions
- Click/double-click recognition
- Intelligent power saving for handheld devices
- ECOPACK®, RoHS and “Green” compliant

Applications

- Indoor navigation
- Smart user interfaces
- Advanced gesture recognition
- Gaming and virtual reality input devices
- Display/map orientation and browsing

Description

The LSM9DS1 is a system-in-package featuring a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor.

The LSM9DS1 has a linear acceleration full scale of ±2*g*/±4*g*/±8/±16 *g,* a magnetic field full scale of ±4/±8/±12/±16 gauss and an angular rate of ±245/±500/±2000 dps.

The LSM9DS1 includes an I^2C serial bus interface supporting standard and fast mode (100 kHz and 400 kHz) and an SPI serial standard interface.

Magnetic, accelerometer and gyroscope sensing can be enabled or set in power-down mode separately for smart power management.

The LSM9DS1 is available in a plastic land grid array package (LGA) and it is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

Table 1. Device summary

Part number	Temperature range [°C]	Package	Packing
LSM9DS1	-40 to +85	LGA-24L	Tray
LSM9DS1TR	-40 to +85	LGA-24L	Tape and reel

List of tables

List of figures

nRF52840

Product Specification

v1.1

NORDIC® SEMICONDUCTOR