Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0624 Laboratorio de Microcontroladores
II ciclo 2023

Laboratorio 05

Controlador por voz con Arduino Nano BLE 33 Sense & Edge Impulse

Estudiantes
Javier Solera Bolaños — B66963
Mike Mai Chen — B94487

Profesor
Marco Villalta Fallas

23 de noviembre de 2023

Índice

1. URL del Repositorio de GitHub

El repositorio de git donde se trabajó todo lo referente al laboratorio se encuentra en Github.

2. Resumen

En este laboratorio consistió en desarrollar un sistema de control por voz, mediante el uso del microcontrolador Arduino Nano 33 BLE y el programa Edge impulse, el cual es un programa para entrenar una red neuronal con palabras calves, en el cual se utilizaron 3, una para simular que se controlan unas luces, se utilizó "lumos", para simular que se controla un reproductor de música, se utilizó "música" y para simular que se controla un televisor, se utilizó "tele". Por medio del micrófono que esta incorporado en el Arduino Nano 33 BLE se da la instrucción de cual de las 3 palabras se quiere utilizar. También se implementó el uso de IOT (Intenet de las Cosas), y el motivo es que se conectó el dispositivo a la red, con el fin de transferir datos y visualizarlos en forma de grafica en la plataforma Thingsboard. En la parte de machine learning, se agregaron además de las 3 palabras calves, se agregó ruido y unknown, esto con el fin de que el entrenamiento sea mejor y mas certero a la hora de ejecutar el programa.

3. Nota teórica

3.1. Características del Arduino Nano 33 BLE

3.1.1. Características generales

El Arduino Nano 33 BLE Sense es una placa de desarrollo de baja potencia que trae el nRF52840 de Nordic Semiconductor. Este procesador es un ARM Cortex-M4 @ 64 MHz de 32 bits [1, 5]. Adicionalmente, la placa de desarrollo cuenta con

- 1 MB de memoria de almacenamiento,
- 256 kB de memoria RAM,
- UART,
- SPI,
- Comunicación I2C,
- Bluetooth Low Energy,
- módulo LSM9DS1 (acelerómetro, giroscopio y magnetómetro),
- módulo MP34DT05 (micrófono),
- módulo APDS9960 (sensor de proximidad, luz y gestos),
- módulo LPS22HB (barómetro) y
- módulo HTS221 (sensor de temperatura y humedad) [1].

3.1.2. Características eléctricas

Los rangos absolutos que se deben respetar para el microcontrolador de la placa de desarrollo son

- tensión operación máxima $V_{\text{DD max}}$: 3.3 V,
- tensión máxima de entrada $V_{\text{in max}}$: 21 V,
- corriente máxima admitida para aplicaciones del usuario: 950 mA [1].

3.1.3. Diagrama de bloques

La figura 1 ilustra el diagrama de bloques del Arduino Nano BLE 33 Sense. Los bloques de interés para este laboratorio son el propio CPU así como el micrófono (módulo MP34DT05) [1]. El resto de bloques de la placa no serán utilizados para este laboratorio.

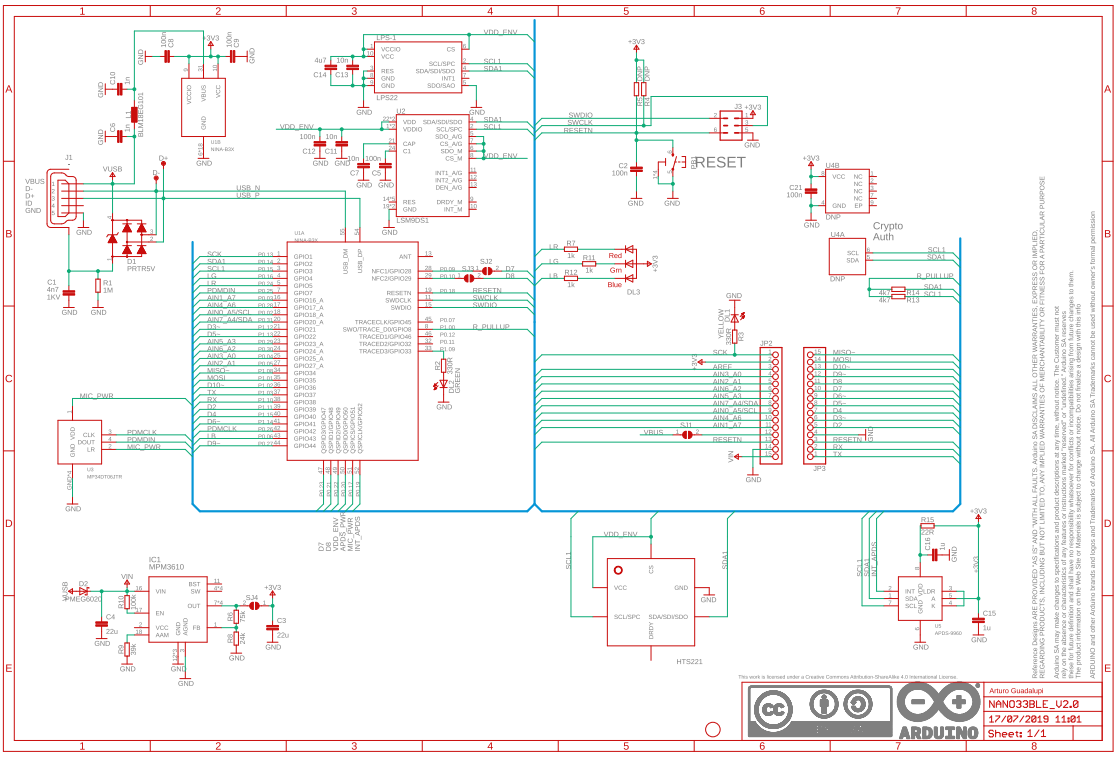


Figura 1: Diagrama de bloques del CPU nRF52840. Fuente y créditos: [1].

3.1.4. Diagrama de pines

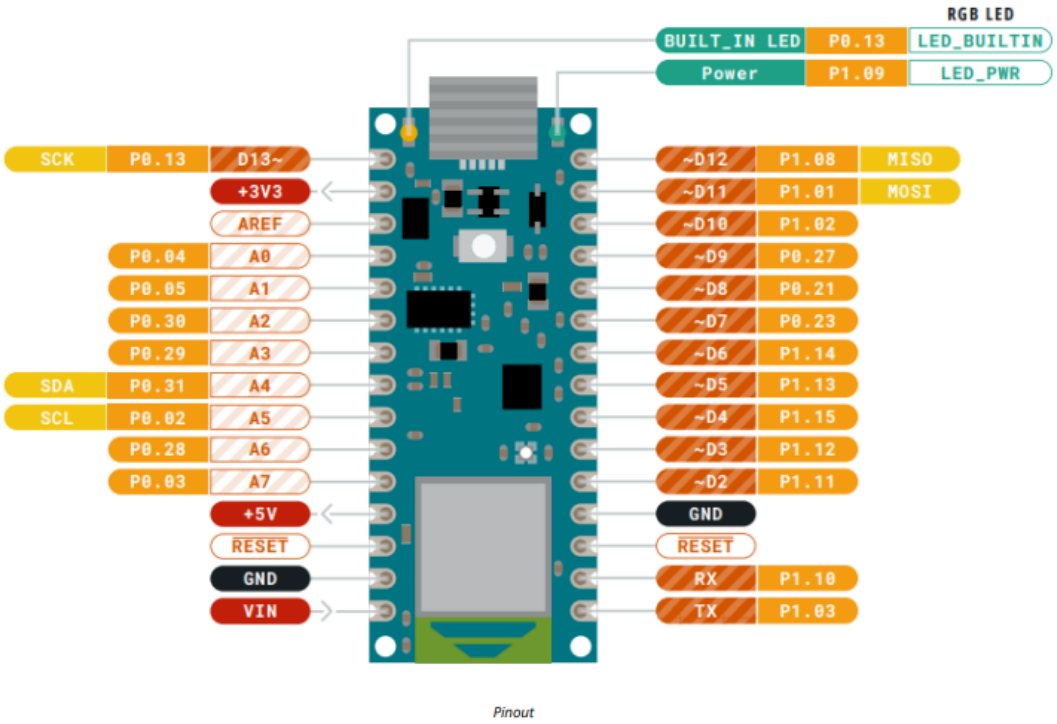


Figura 2: Diagrama de pines del Arduino Nano BLE 33 Sense. Fuente y créditos: [1].

4. Diseño del hardware

Para esta sección no se realizó ningún diseño adicional, ya que lo que se necesita para este laboratorio es un micrófono para detectar los comandos. El Arduino ya trae incorporado un micrófono, por lo que no es necesario diseñar el hardware para un micrófono.

5. Diseño de los programas

5.1. Red neuronal

Por un lado, se utilizó el tutorial de Edge Impulse para crear y entrenar una red neuronal que detecte los comandos por voz. Los comandos son `lumos`, `musica` y `tele`. Además, se siguió los pasos del mismo tutorial para flashear el modelo en el Arduino Nano BLE 33 Sense. Esto quiere decir que el código necesario para crear, entrenar y flashear al Arduino el modelo fue generado automáticamente por Edge Impulse, por lo que todos los créditos van hacia Edge Impulse [3], los autores de este reporte de laboratorio solamente hicieron uso de los scripts generado por Edge Impulse.

Además, los parámetros del modelo son las que Edge Impulse trae por defecto (figura 3) así como recomendaciones que el tutorial menciona, cuya arquitectura de la red neuronal es el preset `2D convolutional` (figura 4) [3]. El código que se encarga de crear y entrenar la red se encuentra en el repositorio en la carpeta `src/edge-impulse/` y el software necesario para flashear el modelo generado en el Arduino se encuentra en la carpeta `src/firmware-arduino-nano-33-ble-sense/`.

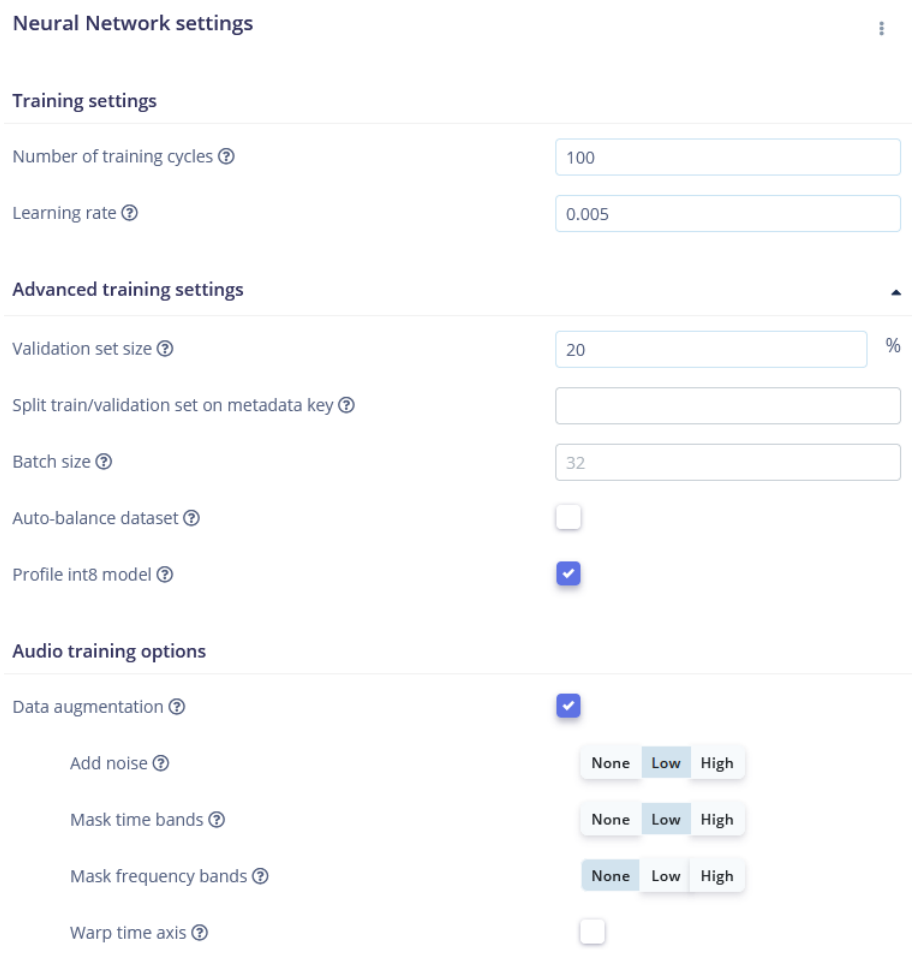


Figura 3: Captura de pantalla de los parámetros de la red neuronal.

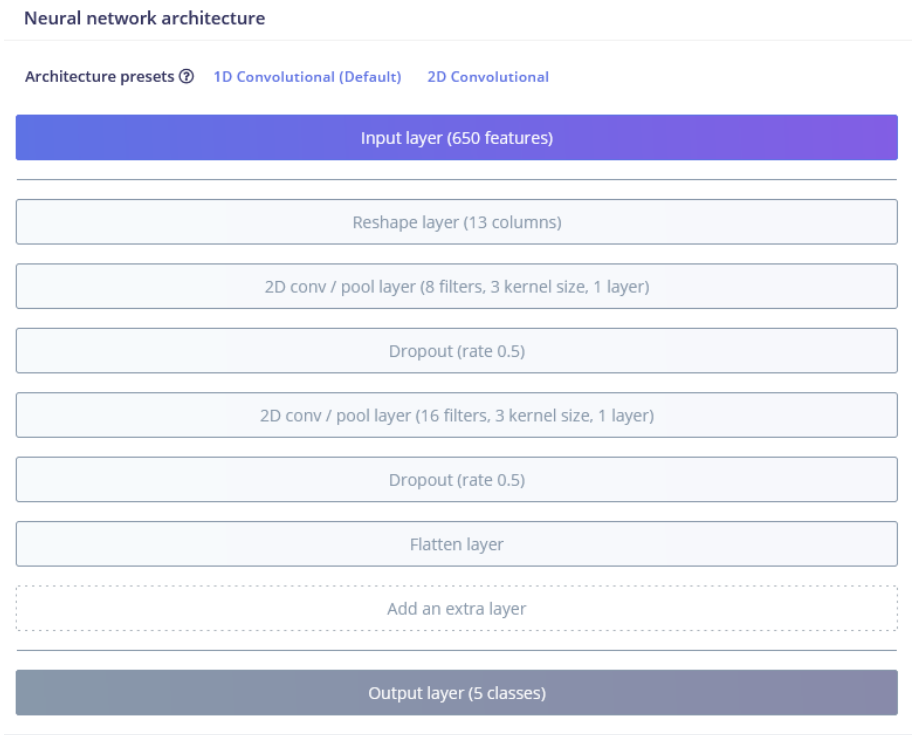


Figura 4: Captura de pantalla de la arquitectura de la red neuronal.

5.2. Diagrama de flujo del script de Python

Por otro lado, se creó un script de Python (`datos.py`) para poder recibir los datos del Arduino a la computadora. El diagrama de flujo se muestra en la figura 5. La idea de este script es aprovechar que Edge Impulse tiene una interfaz en la terminal, esto es, tiene comandos para comunicarse con el Arduino. Tiene un comando denominado `edge-impulse-run-impulse` que arroja en terminal las palabras, cuya cada palabra viene asociada un valor entre cero y uno. La palabra con el valor más alto es la palabra que el modelo cree lo que dijo el usuario [2].

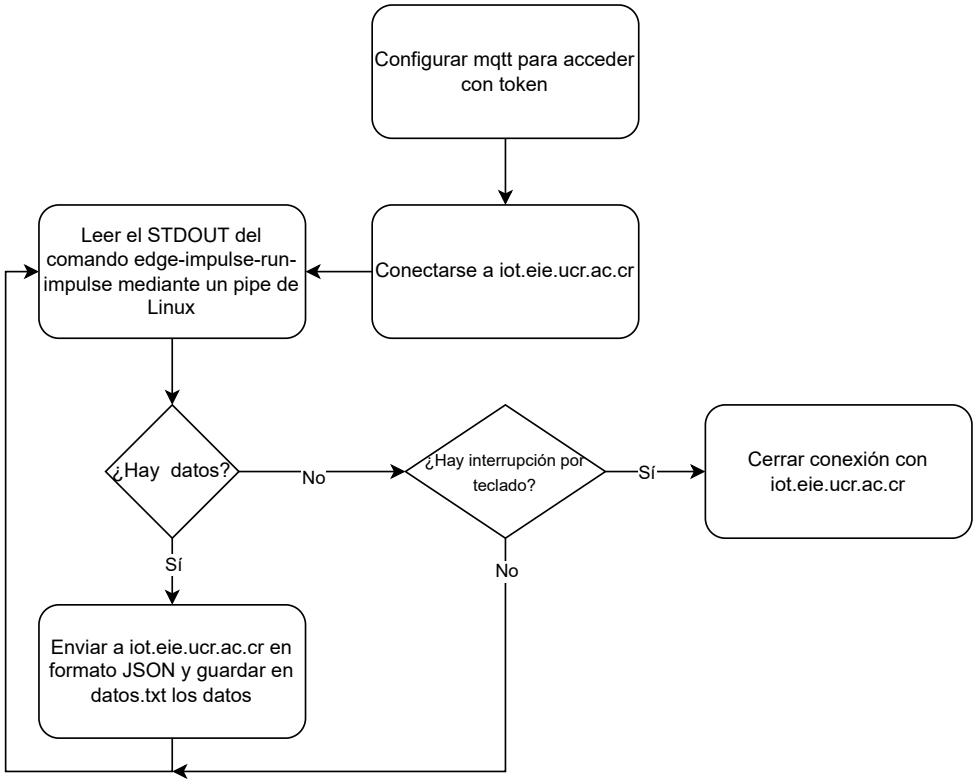


Figura 5: Diagrama de flujo del script para comunicar el Arduino con Thingsboard.

Ahora, mediante el pipe (|) de Linux [4] se puede redireccionar la salida del comando `edge-impulse-run-impulse` a un script de Python en vez de arrojarlo en la terminal. Por lo tanto, lo que tendría que hacer el script de Python es recibir lo que `edge-impulse-run-impulse` arroje y enviar a Thingsboard en formato JSON así como guardar en un archivo de texto las palabras con sus valores asociados. Por lo tanto, se debe correr `edge-impulse-run-impulse --continuous | python3 datos.py` para poder recibir los datos del Arduino y guardalo en un archivo de texto así como enviarlo a Thingsboard. Sin embargo, el comando es largo y tedioso, así que el repositorio proporciona el script `correr.sh` que ejecuta justamente ese comando para facilitar al usuario.

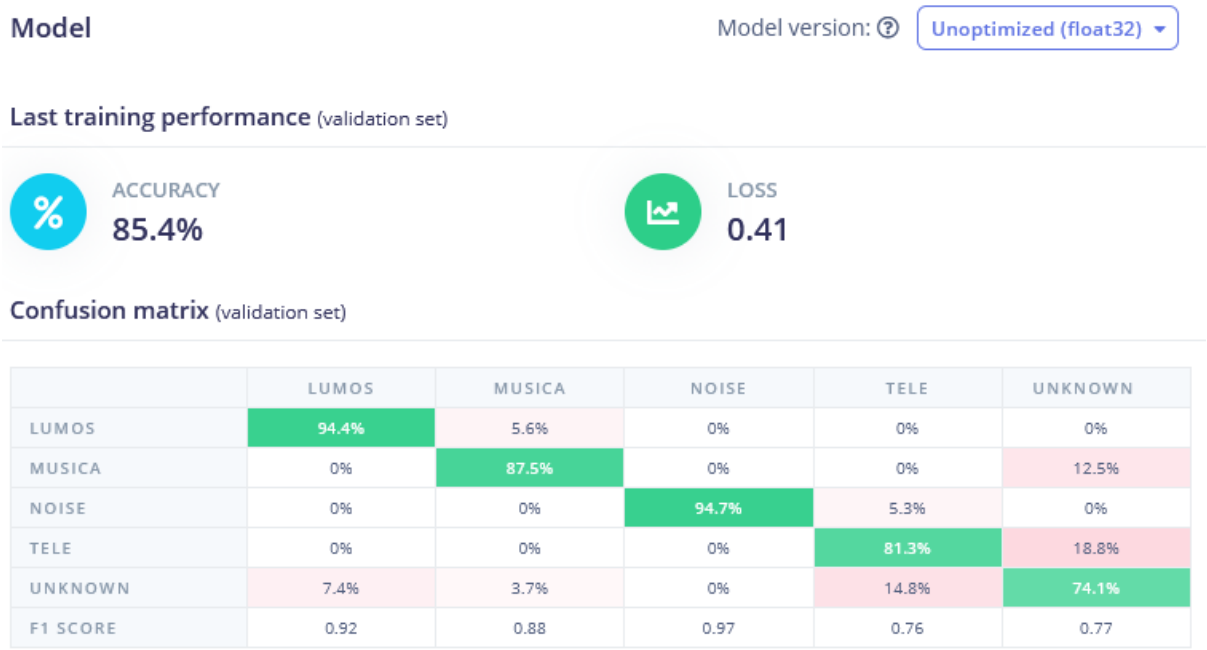
6. Lista de equipos

Componente	Valor Nominal	Cantidad	Precio en el Mercado
Arduino Nano BLE 33 Sense	-	1	$40.50

Tabla 1: Lista de Componentes [1].

7. Resultados y análisis

Se realizó una serie de entrenamientos con las palabras que se deseaban, en este caso son 1) "lumos", la cual pretende controlar las luces; 2) "musica", el cual pretende controlar un reproductor de música y por último 3) "tele", el cual es para controlar un televisor.



	LUMOS	MUSICA	NOISE	TELE	UNKNOWN
LUMOS	94.4%	5.6%	0%	0%	0%
MUSICA	0%	87.5%	0%	0%	12.5%
NOISE	0%	0%	94.7%	5.3%	0%
TELE	0%	0%	0%	81.3%	18.8%
UNKNOWN	7.4%	3.7%	0%	14.8%	74.1%
F1 SCORE	0.92	0.88	0.97	0.76	0.77

Figura 6: Métricas de rendimiento de la red neuronal con datos de validación.

Figura 7: Métricas de rendimiento de la red neuronal con datos de prueba.

Como se muestra en las figuras 6 y 7, se observa que la precisión del modelo es del 85.4 % y 80.83 % con los datos de validación y de prueba, respectivamente. Esta precisión es más bajo que la precisión que el tutorial de Edge Impulse obtuvo. Una principal razón se debe a que en el tutorial grabó 10 minutos de cada comando así como 10 minutos de ruido. En cambio, en este laboratorio se grabó 2 minutos de cada comando y ruido.

En la siguientes figuras se observa los resultados en el Thingsboard, es decir, haciendo uso de las palabras claves para realizar una tarea. En la primera prueba, se utilizó la palabra "lumos", con el fin de poder controlar las luces, como se puede observar en la figura 8.

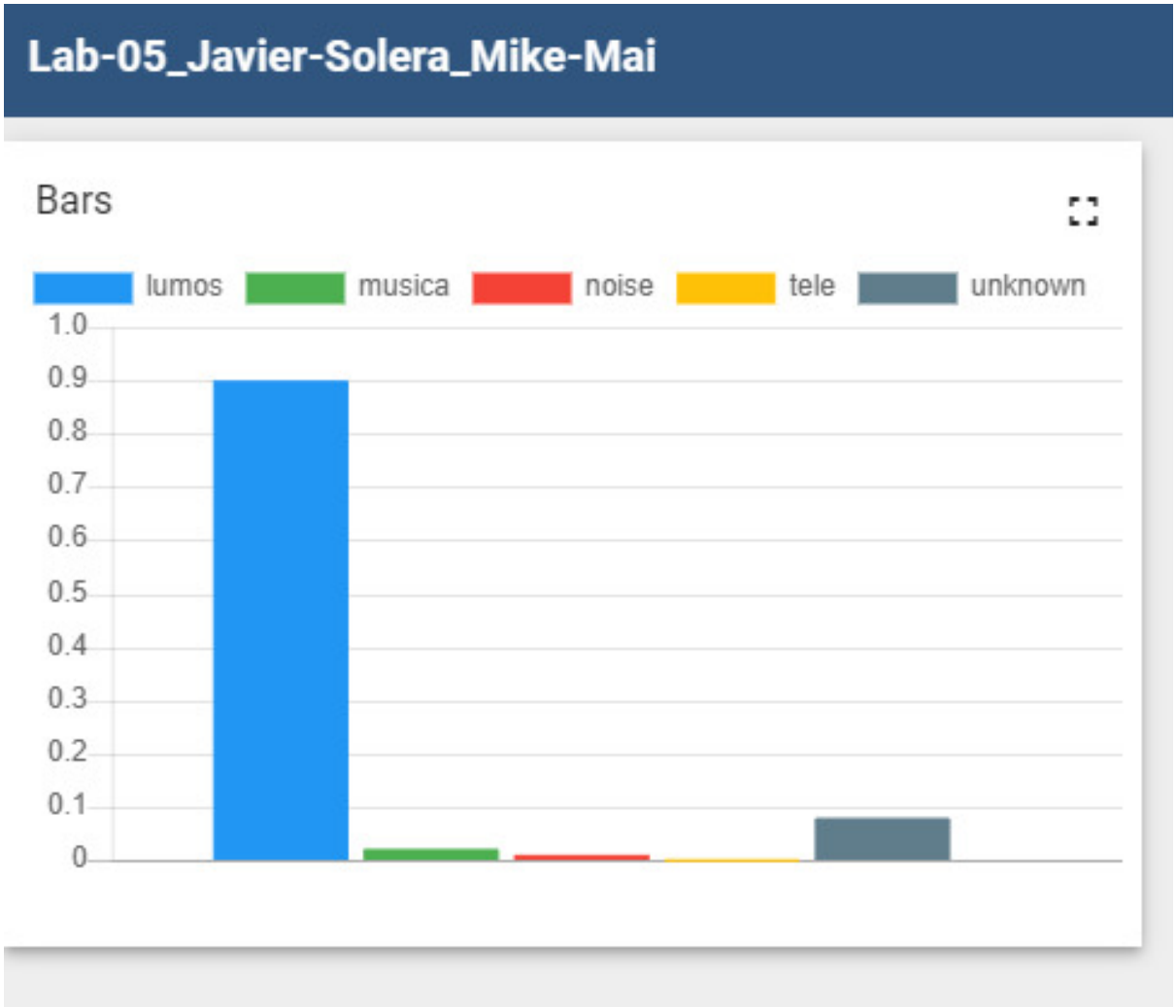


Figura 8: Prueba con la palabra “lumos”.

Como se puede observar, se obtuvo un respuesta del programa muy satisfactoria, ya que una vez que se dice la palabra lumos, el Arduino detecta que se esta diciendo esta palabra, mientras que las demás palabras y ruidos se mantienen bajos.

Luego se realizo una prueba con la palabra musica, con el fin de simular de encender un reproductor de musica, en la figura 9 se observa lo que sucede cuando uno le dice al Arduino la palabra clave “musica”.

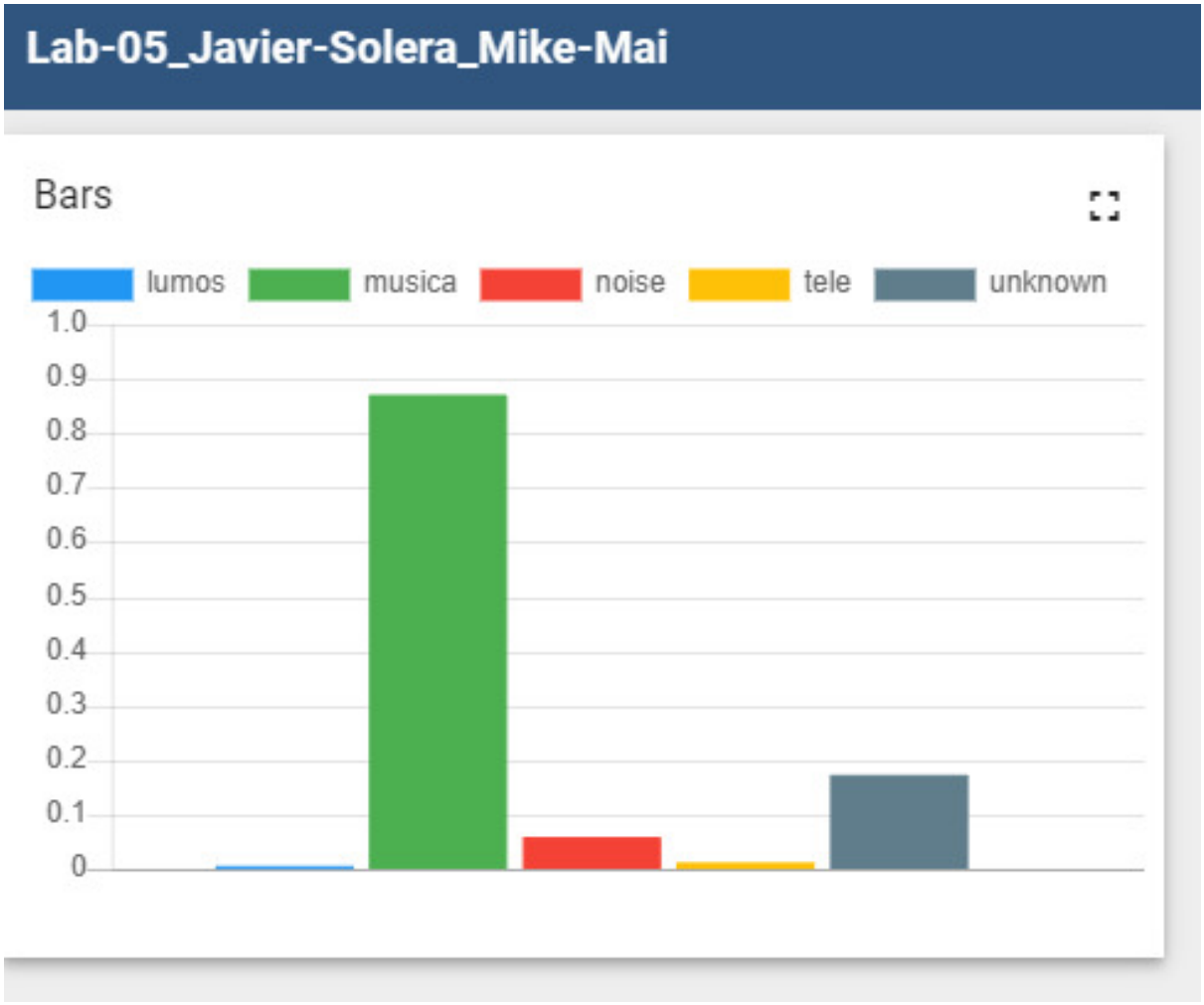


Figura 9: Prueba con la palabra “musica”.

Se puede observar que la señal con la etiqueta “Musica” se levanta y las demás se mantienen abajo, también se observa que la etiqueta de ruido se levanta un poco y esto se debe a que probablemente detecto un mínimo de ruido en el momento de la ejecución, lo mismo con el “unknown”.

Luego, se realizó una prueba con la palabra calve “tele”, el cual pretende simular el encendido de un televisor en una casa.

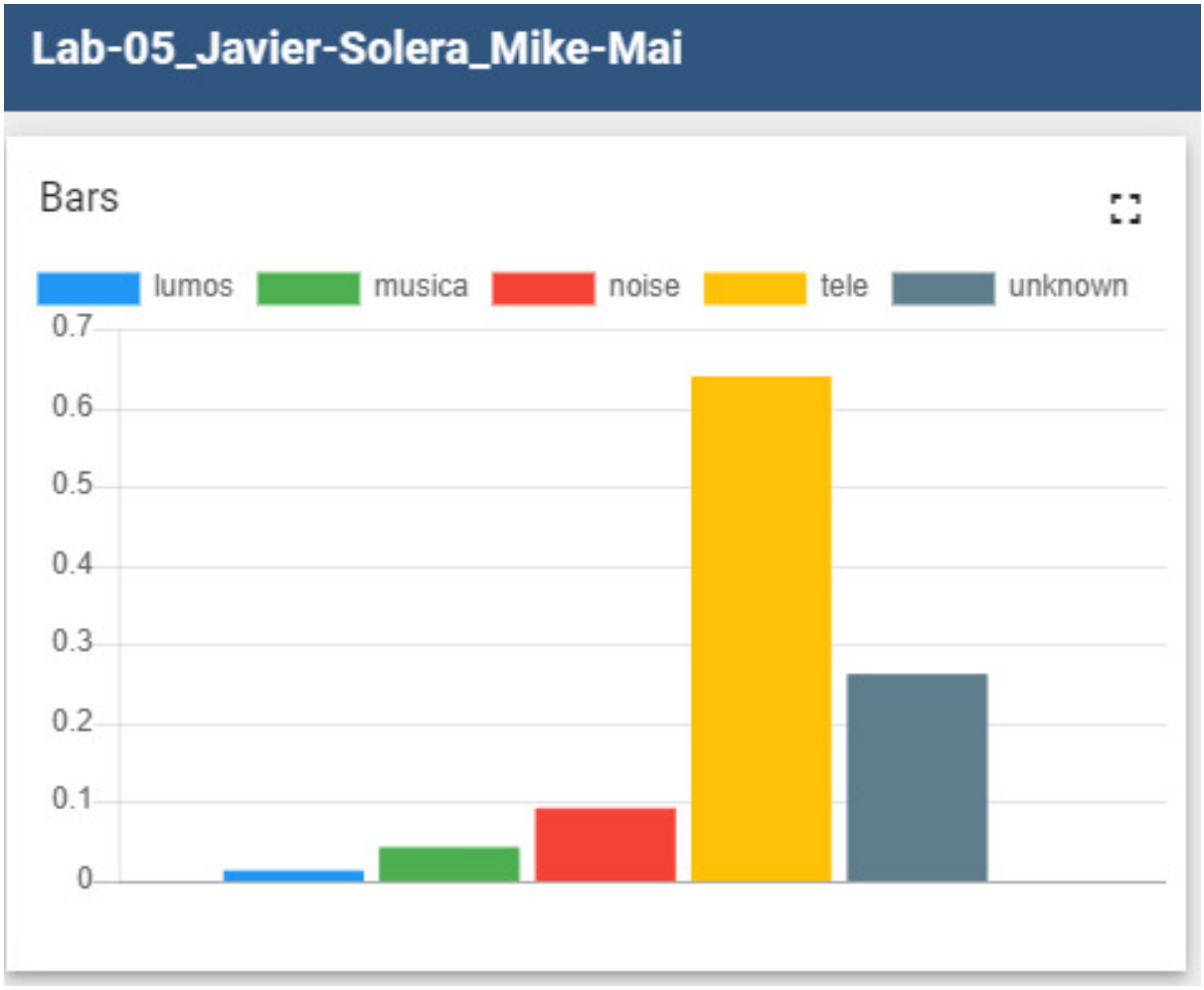


Figura 10: Prueba con la palabra “tele”.

Como se observa en la figura 10, la barra amarilla se levanta, dando a entender que el sistema comprendió muy bien la palabra “tele”, la cual era la que se deseaba utilizar, también se levanta el “unknown” y esto es porque probablemente detectó ciertas voces distintas a las nuestras.

Luego se midió el ruido, el cual se ve reflejado en la figura 11, el cual en esta medición se toma todo lo que se considera ruido ambiental, es decir, ruido de la calle, de electrodomésticos utilizándose, etc.

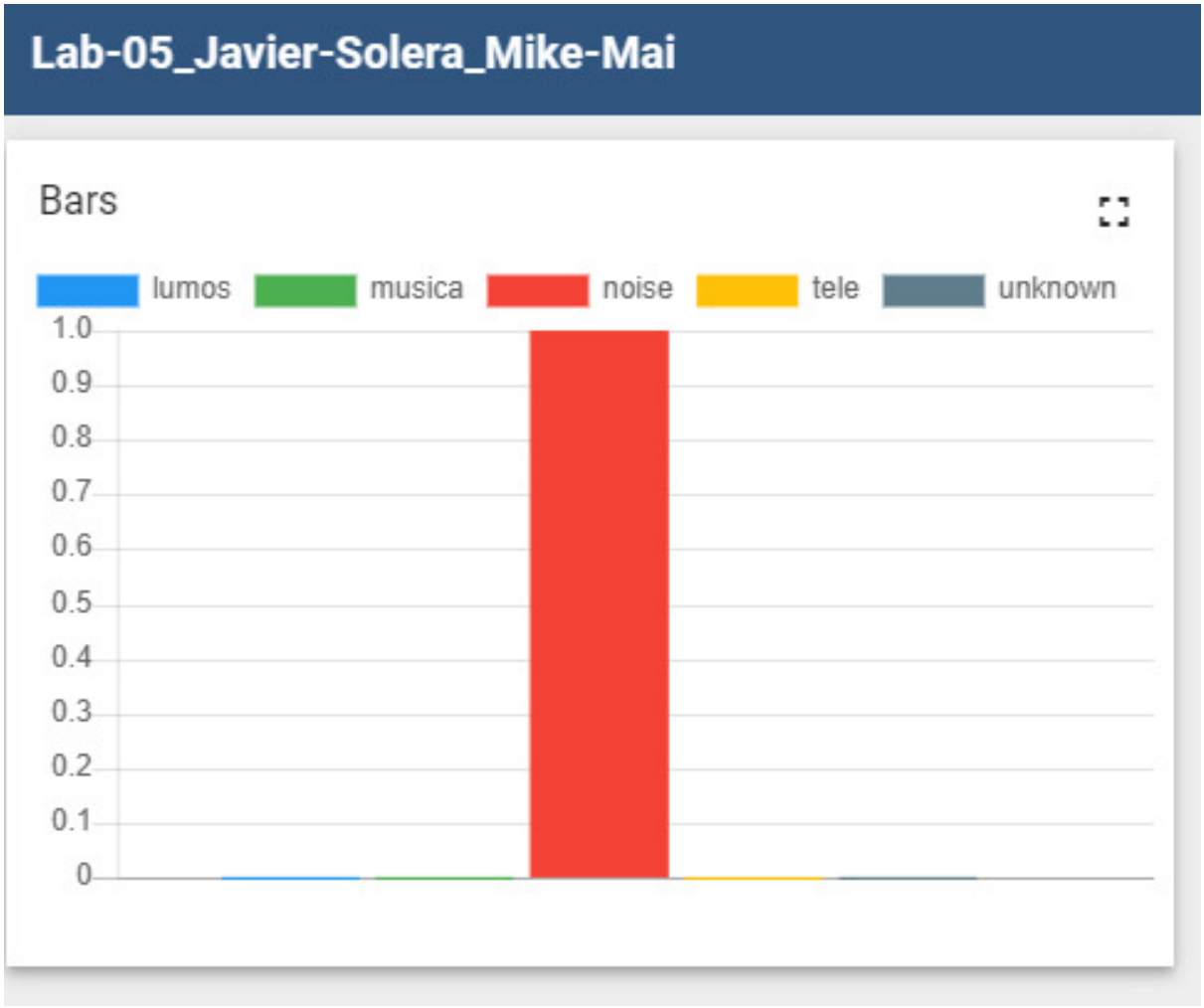


Figura 11: Prueba con ruido de ambiente.

Luego se realizo una medición de distintas voces, de manera para comprobar el funcionamiento y reconocimiento de voz del programa.

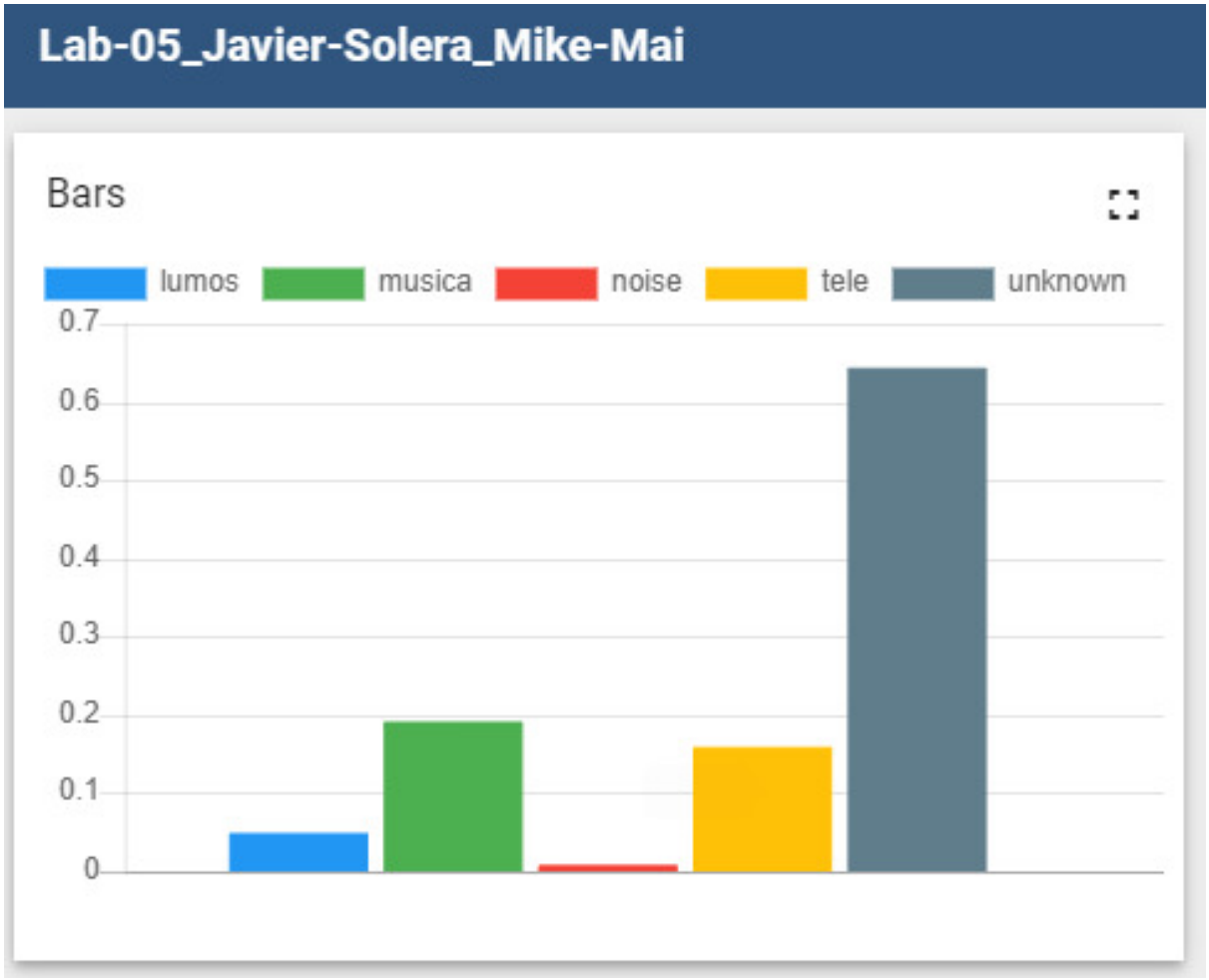


Figura 12: Prueba con voces o palabras diferentes.

Tanto los datos de ruido y los datos de “unknown” son necesarios para tener un entrenamiento bueno y satisfactorio y así tener un mejor funcionamiento a la hora de detectar las palabras claves, ya que así se puede entrenar la red neuronal y así se puedan ejecutar las acciones que se desean cuando se dicen las palabras claves (“ lumos”, “tele” y “musica”). Cabe destacar que el Arduino con el modelo nunca tuvo ni tendrá una precisión del 100 % al detectar palabras. Es más, durante las pruebas se tuvo que repetir varias veces cada comando para que eventualmente el modelo capture el comando y clasifique correctamente el comando. Las figuras 8, 9, 10, 11 y 12 solo muestra el caso cuando se clasifica correctamente.

8. Conclusiones y recomendaciones

- El laboratorio se concluyó de manera satisfactoria ya que se cumplió con todas las especificaciones del enunciado.
- Esto incluye crear un modelo de red neuronal capaz de clasificar los comandos deseados, así como cargar el modelo en el Arduino.
- Además, se pudo implementar la redirección de salida de los comandos con un valor numérico asociado a un script de Python y reenviarlos a `iot.eie.ucr.ac.cr`, así como guardarlo en un archivo de texto.

- A pesar de que se debe repetir varias veces los comandos, el Arduino eventualmente clasifica correctamente el comando y lo envía a Thingsboard.

Referencias

[1] Arduino, *Arduino Nano BLE 33 Sense*. [Online]. Available: https://store-usa.arduino.cc/products/arduino-nano-33-ble-sense

[2] Edge Impulse, "Impulse runner." [Online]. Available: https://docs.edgeimpulse.com/docs/tools/edge-impulse-cli/cli-run-impulse

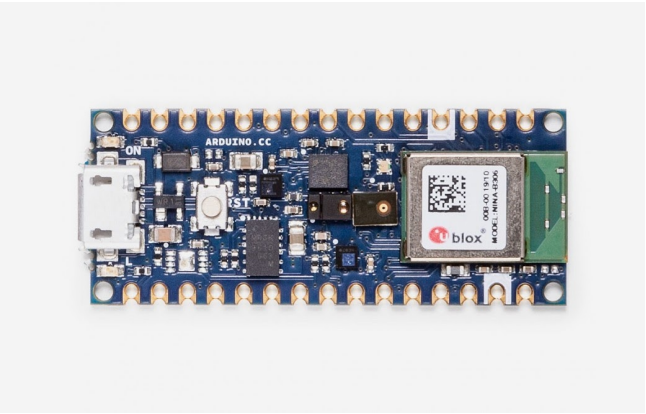
[3] ——, "Responding to your voice." [Online]. Available: https://docs.edgeimpulse.com/docs/tutorials/end-to-end-tutorials/responding-to-your-voice

[4] GeeksforGeeks, "Piping in Unix or Linux." [Online]. Available: https://www.geeksforgeeks.org/piping-in-unix-or-linux/

[5] Nordic Semiconductors, *nRF52840: product specification*. [Online]. Available: https://content.arduino.cc/assets/Nano_BLE_MCU-nRF52840_PS_v1.1.pdf

9. Anexos

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

Features

- **NINA B306 Module**
  - **Processor**
    - 64 MHz Arm® Cortex-M4F (with FPU)
    - 1 MB Flash + 256 KB RAM
  - **Bluetooth® 5 multiprotocol radio**
    - 2 Mbps
    - CSA #2
    - Advertising Extensions
    - Long Range
    - +8 dBm TX power
    - -95 dBm sensitivity
    - 4.8 mA in TX (0 dBm)
    - 4.6 mA in RX (1 Mbps)
    - Integrated balun with 50 Ω single-ended output
    - IEEE 802.15.4 radio support
    - Thread
    - Zigbee
  - **Peripherals**
    - Full-speed 12 Mbps USB
    - NFC-A tag
    - Arm CryptoCell CC310 security subsystem
    - QSPI/SPI/TWI/I²S/PDM/QDEC
    - High speed 32 MHz SPI
    - Quad SPI interface 32 MHz
    - EasyDMA for all digital interfaces
    - 12-bit 200 ksps ADC
    - 128 bit AES/ECB/CCM/AAR co-processor
- **LSM9DS1** (9 axis IMU)
  - 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
- **LPS22HB** (Barometer and temperature sensor)
  - 260 to 1260 hPa absolute pressure range with 24 bit precision
  - High overpressure capability: 20x full-scale
  - Embedded temperature compensation
  - 16-bit temperature data output
  - 1 Hz to 75 Hz output data rateInterrupt functions: Data Ready, FIFO flags, pressure thresholds
- **HTS221** (relative humidity sensor)
  - 0-100% relative humidity range
  - High rH sensitivity: 0.004% rH/LSB
  - Humidity accuracy: ± 3.5% rH, 20 to +80% rH
  - Temperature accuracy: ± 0.5 °C,15 to +40 °C
  - 16-bit humidity and temperature output data

- **APDS-9960** (Digital proximity, Ambient light, RGB and Gesture Sensor)
  - Ambient Light and RGB Color Sensing with UV and IR blocking filters
  - Very high sensitivity – Ideally suited for operation behind dark glass
  - Proximity Sensing with Ambient light rejection
  - Complex Gesture Sensing
- **MP34DT05** (Digital Microphone)
  - AOP = 122.5 dbSPL
  - 64 dB signal-to-noise ratio
  - Omnidirectional sensitivity
  - –26 dBFS ± 3 dB sensitivity
- **ATECC608A** (Crypto Chip)
  - Cryptographic co-processor with secure hardware based key storage
  - Protected storage for up to 16 keys, certificates or data
  - ECDH: FIPS SP800-56A Elliptic Curve Diffie-Hellman
  - NIST standard P256 elliptic curve support
  - SHA-256 & HMAC hash including off-chip context save/restore
  - AES-128 encrypt/decrypt, galois field multiply for GCM
- **MPM3610** DC-DC
  - Regulates input voltage from up to 21V with a minimum of 65% efficiency @minimum load
  - More than 85% efficiency @12V

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.

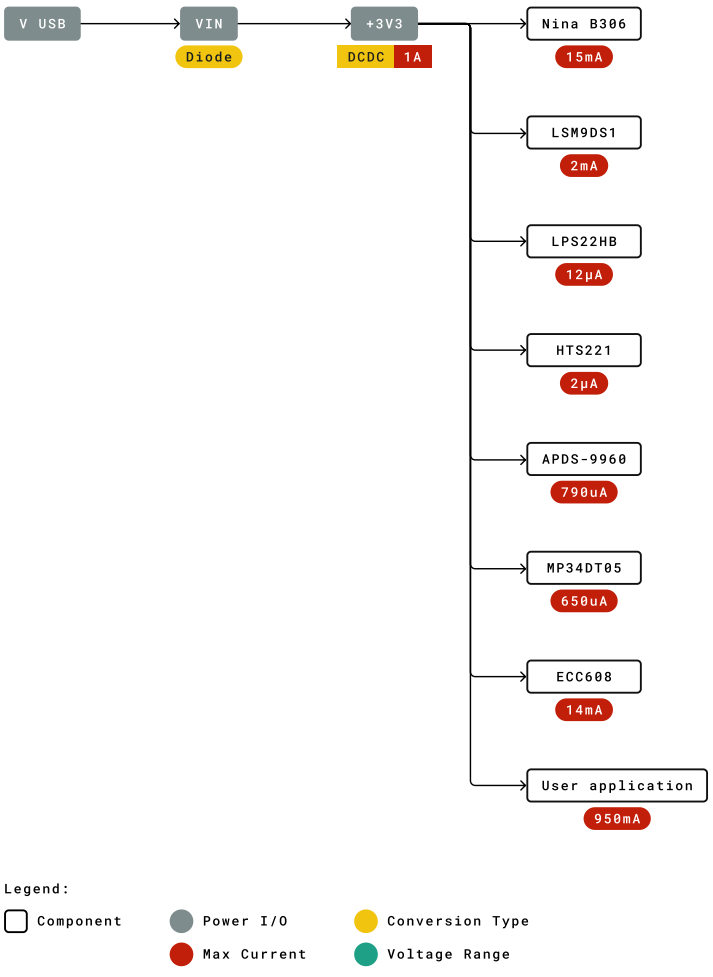
2.8 Digital Microphone

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

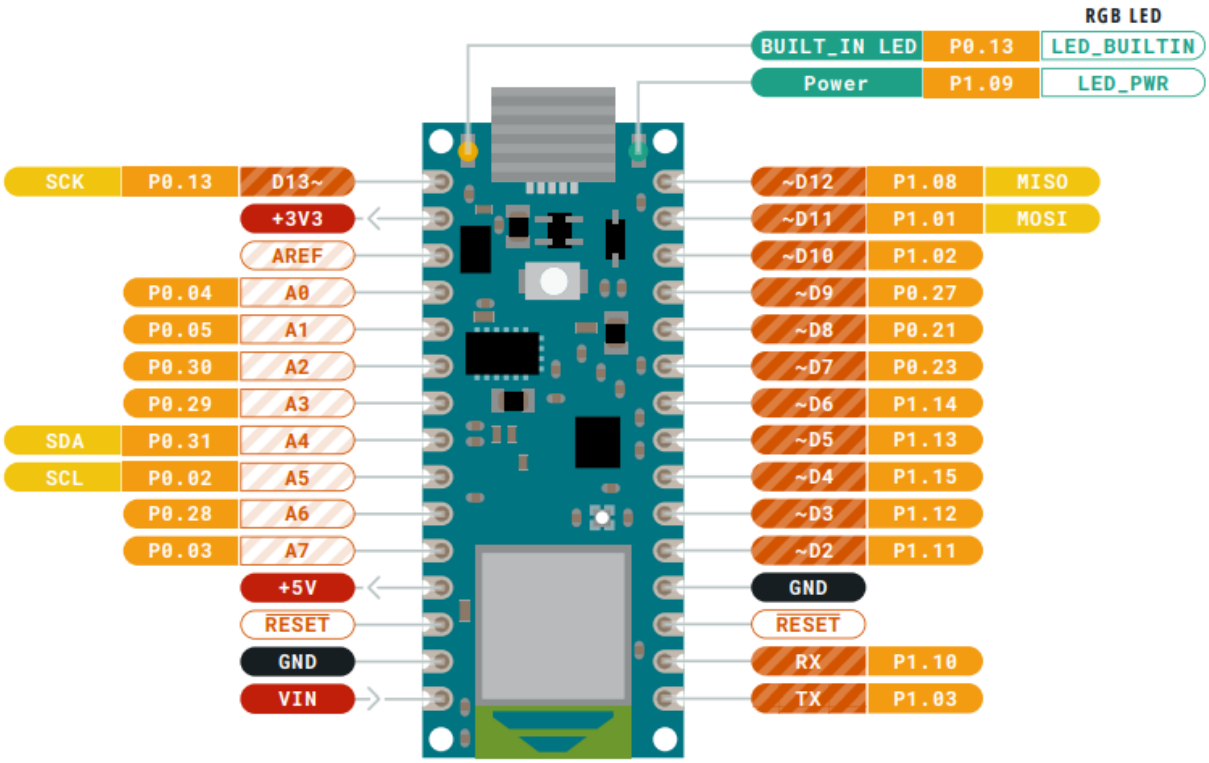
The board can be powered via USB connector, V_{IN} or V_{USB} pins on headers.



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.

4 Connector Pinouts



Pinout

4.1 USB

Pin	Function	Type	Description
1	VUSB	Power	Power Supply Input. If board is powered via VUSB from header this is an Output **(1)**
2	D-	Differential	USB differential data -
3	D+	Differential	USB differential data +
4	ID	Analog	Selects Host/Device functionality
5	GND	Power	Power Ground

4.2 Headers

The board exposes two 15 pin connectors which can either be assembled with pin headers or soldered through castellated vias.

Pin	Function	Type	Description
1	D13	Digital	GPIO
2	+3V3	Power Out	Internally generated power output to external devices
3	AREF	Analog	Analog Reference; can be used as GPIO
4	A0/DAC0	Analog	ADC in/DAC out; can be used as GPIO
5	A1	Analog	ADC in; can be used as GPIO
6	A2	Analog	ADC in; can be used as GPIO
7	A3	Analog	ADC in; can be used as GPIO
8	A4/SDA	Analog	ADC in; I2C SDA; Can be used as GPIO **(1)**
9	A5/SCL	Analog	ADC in; I2C SCL; Can be used as GPIO **(1)**
10	A6	Analog	ADC in; can be used as GPIO
11	A7	Analog	ADC in; can be used as GPIO
12	VUSB	Power In/Out	Normally NC; can be connected to VUSB pin of the USB connector by shorting a jumper
13	RST	Digital In	Active low reset input (duplicate of pin 18)
14	GND	Power	Power Ground
15	VIN	Power In	Vin Power input
16	TX	Digital	USART TX; can be used as GPIO
17	RX	Digital	USART RX; can be used as GPIO
18	RST	Digital	Active low reset input (duplicate of pin 13)
19	GND	Power	Power Ground
20	D2	Digital	GPIO
21	D3/PWM	Digital	GPIO; can be used as PWM
22	D4	Digital	GPIO
23	D5/PWM	Digital	GPIO; can be used as PWM
24	D6/PWM	Digital	GPIO, can be used as PWM
25	D7	Digital	GPIO
26	D8	Digital	GPIO
27	D9/PWM	Digital	GPIO; can be used as PWM
28	D10/PWM	Digital	GPIO; can be used as PWM
29	D11/MOSI	Digital	SPI MOSI; can be used as GPIO
30	D12/MISO	Digital	SPI MISO; can be used as GPIO

4.3 Debug

On the bottom side of the board, under the communication module, debug signals are arranged as 3x2 test pads with 100 mil pitch with pin 4 removed. Pin 1 is depicted in Figure 3 – Connector Positions

Pin	Function	Type	Description
1	+3V3	Power Out	Internally generated power output to be used as voltage reference
2	SWD	Digital	nRF52480 Single Wire Debug Data
3	SWCLK	Digital In	nRF52480 Single Wire Debug Clock
5	GND	Power	Power Ground
6	RST	Digital In	Active low reset input

(1) this device may not cause interference

(2) this device must accept any interference, including interference that may cause undesired operation of the device.

French: Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes :

(1) l' appareil nedoit pas produire de brouillage

(2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

IC SAR Warning:

English This equipment should be installed and operated with minimum distance 20 cm between the radiator and your body.

French: Lors de l' installation et de l' exploitation de ce dispositif, la distance entre le radiateur et le corps est d 'au moins 20 cm.

Important: The operating temperature of the EUT can't exceed 85°C and shouldn't be lower than -40°C.

Hereby, Arduino S.r.l. declares that this product is in compliance with essential requirements and other relevant provisions of Directive 2014/53/EU. This product is allowed to be used in all EU member states.

Frequency bands	Maximum output power (ERP)
863-870Mhz	5.47 dBm

8 Company Information

Company name	Arduino S.r.l
Company Address	Via Andrea Appiani 25 20900 MONZA Italy

9 Reference Documentation

Reference	Link
Arduino IDE (Desktop)	https://www.arduino.cc/en/software
Arduino IDE (Cloud)	https://create.arduino.cc/editor
Cloud IDE Getting Started	https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-arduino-web-editor-4b3e4a
Forum	http://forum.arduino.cc/
Nina B306	https://content.u-blox.com/sites/default/files/NINA-B3_DataSheet_UBX-17052099.pdf
ECC608	https://ww1.microchip.com/downloads/aemDocuments/documents/SCBU/ProductDocuments/DataSheets/ATECC608A-CryptoAuthentication-Device-Summary-Data-Sheet-DS40001977B.pdf
MPM3610	https://www.monolithicpower.com/pub/media/document/MPM3610_r1.01.pdf
ECC608 Library	https://github.com/arduino-libraries/ArduinoECCX08
LSM6DSL Library	https://github.com/adafruit/Adafruit_LSM9DS1
LPS22HB	https://github.com/stm32duino/LPS22HB
HTS221 Library	https://github.com/stm32duino/HTS221
APDS9960 Library	https://github.com/adafruit/Adafruit_APDS9960
ProjectHub	https://create.arduino.cc/projecthub?by=part&part_id=11332&sort=trending

Reference	Link
Library Reference	https://www.arduino.cc/reference/en/

10 Revision History

Date	Revision	Changes
03/08/2022	2	Reference documentation links updates
27/04/2021	1	General datasheet updates