

EIE

Escuela de
Ingeniería Eléctrica

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica



UNIVERSIDAD DE
COSTA RICA

IE-0624

Laboratorio de Microcontroladores

**Laboratorio #5:
STM32/Arduino: GPIO, Giroscopio,
comunicaciones, TinyML**

Fernando Jiménez Ureña B74020

Kristel Herrera Rodríguez C13769

II ciclo
Octubre 2023

1. Resumen

En el presente reporte, se exponen los resultados y el proceso seguido para realizar el Laboratorio 5 donde se desarrolla un controlador de voz utilizando un kit Arduino Nano 33 BLE, enfocándose en aplicaciones de Tiny Machine Learning.

Se describen las etapas iniciales que incluyeron la preparación del entorno de trabajo, como la configuración de las bibliotecas necesarias, y la utilización de la plataforma de Edge Impulse. Posteriormente, el desarrollo de un programa para el microcontrolador que captura y transmite la información del micrófono a la computadora, y un script en Python que procesa y almacena estos datos.

Además, se explica el proceso de registro y clasificación de comandos de voz, así como la creación y entrenamiento de un modelo de red neuronal en TensorFlow Lite para la identificación de dichos comandos. Finalmente, se utiliza la plataforma *Thingsboard* para implementar la comunicación entre el microcontrolador y una plataforma de IoT.

Link del repositorio:

<https://github.com/KrisKy02/IE0624-Laboratorio-de-microcontroladores.git>

2. Nota Teórica

El *Arduino Nano 33 BLE Sense* desempeña un papel crucial en nuestro proyecto de control por voz. Este módulo compacto alberga un módulo *NINA B306* basado en *Nordic nRF52840*, con características clave[1].

2.1. Módulo NINA B306

- Procesador Cortex M4F a 64 MHz con FPU.
- 1 MB de Flash y 256 KB de RAM.
- Radio Bluetooth 5 con alta velocidad y potencia.
- Chip criptográfico para almacenamiento seguro.

2.2. Periféricos y Sensores Integrados

El *Arduino Nano 33 BLE Sense* incluye una amplia gama de periféricos y sensores esenciales[1]:

- Puerto USB de alta velocidad.
- Etiqueta NFC-A.
- Subsistema de seguridad *Arm CryptoCell CC310*.
- Interfaces versátiles como SPI e I2C.
- Sensores integrados, incluyendo IMU de 9 ejes, barómetro, sensor de humedad, sensor de proximidad, luz ambiental, color RGB y micrófono digital.

El diagrama de pines se puede visualizar en la siguiente imagen (Figura1):

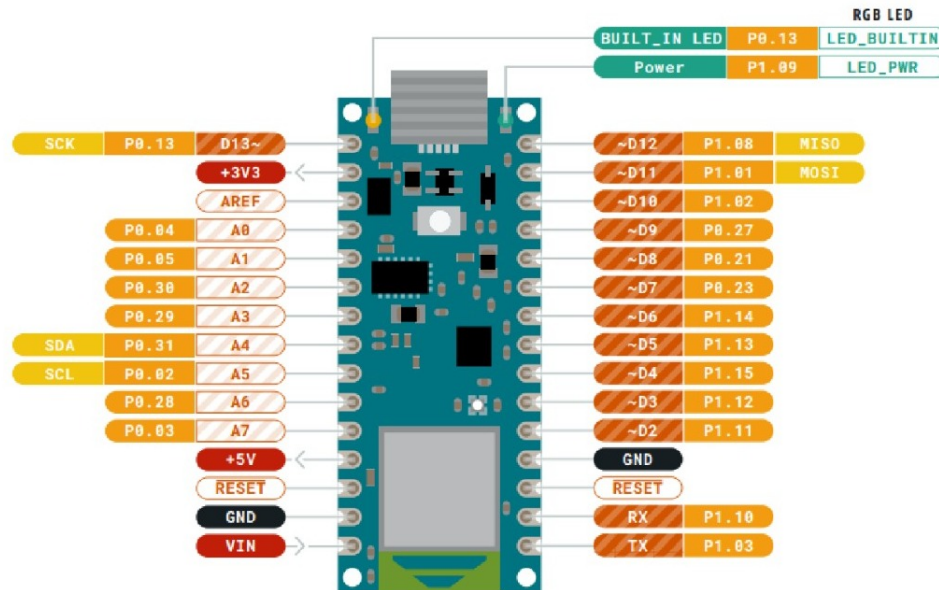


Figura 1: Diagrama de pines.[1]

Y la funcionalidad de cada pin se muestra en las siguientes tablas (Figuras 2 y 3).

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

Figura 2: Funcionalidad de Headers y USB.[1]

Pin	Function	Type	Description
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

Figura 3: Funcionalidad de Headers y Debug.[1]

2.3. Regulador DC-DC

El *Arduino Nano 33 BLE Sense* utiliza un regulador DC-DC *MPM3610* para mantener una operación eficiente y flexible con diferentes voltajes de entrada

2.4. Procesador y Criptografía

El procesador principal es un Cortex M4F que opera a 64 MHz y se comunica con el módulo inalámbrico y los sensores internos. Además, cuenta con un chip criptográfico que proporciona seguridad en el almacenamiento de secretos[1].

2.5. IMU (Unidad de Medición Inercial)

El *Arduino Nano 33 BLE Sense* incluye una IMU de 9 ejes integrada que se puede utilizar para medir la orientación de la placa, la aceleración debida a la gravedad o para medir impactos, vibración, aceleración y velocidad de rotación[1].

2.6. Sensores Adicionales

El módulo también cuenta con un barómetro y sensor de temperatura, un sensor de humedad relativa y temperatura, así como un sensor de proximidad, luz ambiental, color RGB y gestos digitales[1].

2.7. Micrófono Digital

El *Arduino Nano 33 BLE Sense* incorpora un micrófono digital MEMS ultra compacto y de bajo consumo que es omnidireccional y cuenta con una interfaz IC[1].

2.8. Captura de audio

La información del micrófono se capturará utilizando el *Arduino Nano 33 BLE Sense* mediante el proceso de conversión analógica a digital (ADC). En este contexto, el ADC del Arduino toma la señal analógica del micrófono y la convierte en una representación digital. Esto implica muestrear la señal analógica a intervalos regulares y asignar un valor digital correspondiente a cada muestra. La tasa de muestreo y la resolución del ADC son factores clave en la calidad de la captura de audio.

2.9. Entrenamiento del modelo de TensorFlow Lite o Edge Impulse

El proceso de desarrollo del modelo de detección de voz utilizando TensorFlow Lite o Edge Impulse es fundamental para la efectividad de nuestro proyecto. Estas tecnologías han sido desarrolladas para permitir la ejecución de modelos de aprendizaje automático altamente optimizados en microcontroladores de bajo consumo, como el *Arduino Nano 33 BLE Sense* que estamos utilizando.

Como se menciona en un artículo de TensorFlow Lite publicado por Daniel Situnayake, Founding TinyML Engineer de Edge Impulse, estos microcontroladores son componentes esenciales en numerosos dispositivos modernos, desde relojes inteligentes hasta electrodomésticos de cocina y equipos industriales. Los ingenieros de software embebido que escriben el software para estos dispositivos deben optimizar cada recurso, ya que los microcontroladores suelen tener recursos limitados, como un procesador de 32 bits a 64 MHz, 256 KB de RAM y 1 MB de memoria flash [2].

2.10. Exportación del modelo al microcontrolador

El modelo entrenado se exportará al *Arduino Nano 33 BLE Sense*. Este proceso implica consideraciones de memoria y recursos, ya que el microcontrolador debe ser capaz de cargar

y ejecutar el modelo. Se debe garantizar que el modelo sea lo suficientemente eficiente en cuanto a recursos para funcionar sin problemas en el microcontrolador.

2.11. Detección de comandos de voz

En el microcontrolador, se implementará un programa que utilizará el modelo entrenado para detectar comandos de voz en tiempo real. Cuando se detecte un comando válido, se comunicará a la PC a través del puerto USB. La implementación de este programa en el microcontrolador implica la carga del modelo, la configuración de la entrada de audio desde el micrófono y la lógica de detección de comandos.

2.12. Integración con plataformas IoT

Los comandos de voz detectados se enviarán a una plataforma IoT, como *Thingsboard*. Esto se logrará a través de la comunicación en red utilizando protocolos IoT estándar, como MQTT o HTTP. En la plataforma IoT, se activarán widgets que representarán las acciones correspondientes, permitiendo el control de dispositivos y la interacción con la IoT a través de comandos de voz.

3. Desarrollo

3.1. Entrenamiento del Modelo

Para el desarrollo del Laboratorio, el punto de inicio fue entrenar el modelo alimentándolo con datos. Para esto, se le pidió a varias personas grabar un audio con su voz diciendo diferentes palabras que cumplirían una función en específico más adelante en el programa.

Lograr un conjunto de datos bien equilibrado es fundamental para asegurar que el modelo de aprendizaje automático sea capaz de generalizar bien y no esté sesgado hacia una clase particular, por lo tanto la importancia de tener diferentes audios provenientes de diferentes personas. Se buscó evadir el desequilibrio en los conjuntos de datos, donde puede surgir debido a tener más muestras de una clase que de otra. Esto puede llevar a que el modelo se incline hacia la clase con más muestras, ignorando o clasificando incorrectamente las clases menos representadas. Por lo tanto se buscó entrenar el modelo con una cantidad equilibrada de audios.

Para esto, se utilizó la plataforma de Edge Impulse ya que la plataforma facilita el equilibrio de los conjuntos de datos. Edge Impulse proporciona herramientas y funcionalidades que permiten a los usuarios dividir fácilmente los datos en conjuntos de entrenamiento y prueba de manera que se mantenga un equilibrio proporcional entre las diferentes clases. Esto se logra mediante la selección de un porcentaje adecuado de datos para cada conjunto, asegurando que cada clase esté igualmente representada tanto en el conjunto de entrenamiento como en el de prueba.

Es importante notar que en este punto no se logró directamente grabar los audios con el microcontrolador debido a que no se consiguió la configuración necesaria para lograrlo. Por lo tanto se procedió a grabar los audios con las Notas de Voz de los teléfonos y dichos audios fueron subidos directamente a la plataforma de *Edge Impulse* para hacer el entrenamiento del modelo.

3.2. Parámetros para el procesamiento de Señales

Seguidamente, fue necesario el procesamiento de señales para poder transformar los datos crudos, en este caso los audios que fueron recolectados anteriormente, en un formato que es más adecuado y eficiente para el análisis y la clasificación por parte de los modelos de aprendizaje automático.

Se utilizaron técnicas que el programa de Edge Impulse contenía como fue el caso de MFCC (Mel Frequency Cepstral Coefficients). Ésta es una técnica para la extracción de características en el procesamiento de voz y audio. Los MFCCs son eficaces para capturar las propiedades básicas del sonido, como el timbre y el tono. [3]

Además el modelo de aprendizaje elegido fue el de Red Neuronal desde la interfaz de *Edge Impulse*. La Red Neuronal (NN Classifier) es un algoritmo de aprendizaje automático

avanzado utilizado para clasificar datos en categorías distintas. Utiliza múltiples capas de procesamiento para identificar patrones complejos en los datos, lo que lo hace ideal para tareas de clasificación que requieren un alto grado de precisión y generalización.

3.3. Resultados del Modelo

Luego de la alimentación del modelo y el respectivo procesamiento de las señales con las técnicas ya descritas, se obtuvieron los siguientes resultados en la precisión del modelo.

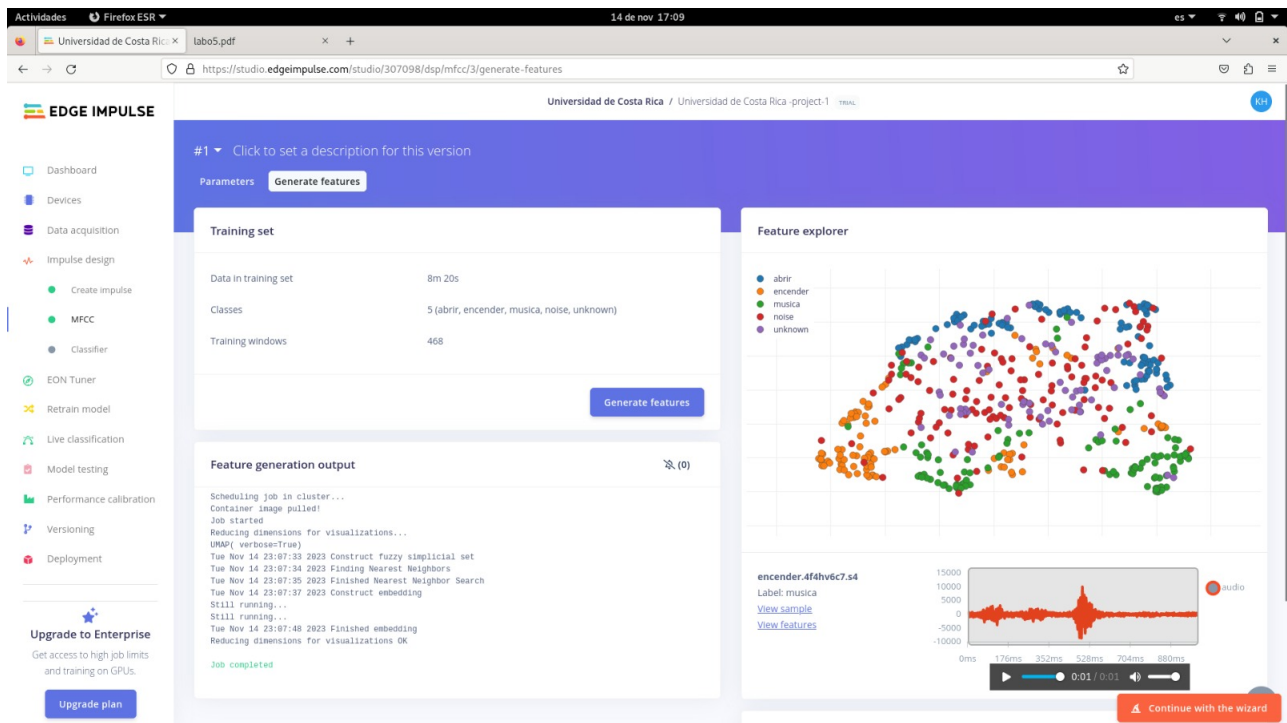


Figura 4: Resultados Obtenidos con el modelo implementado

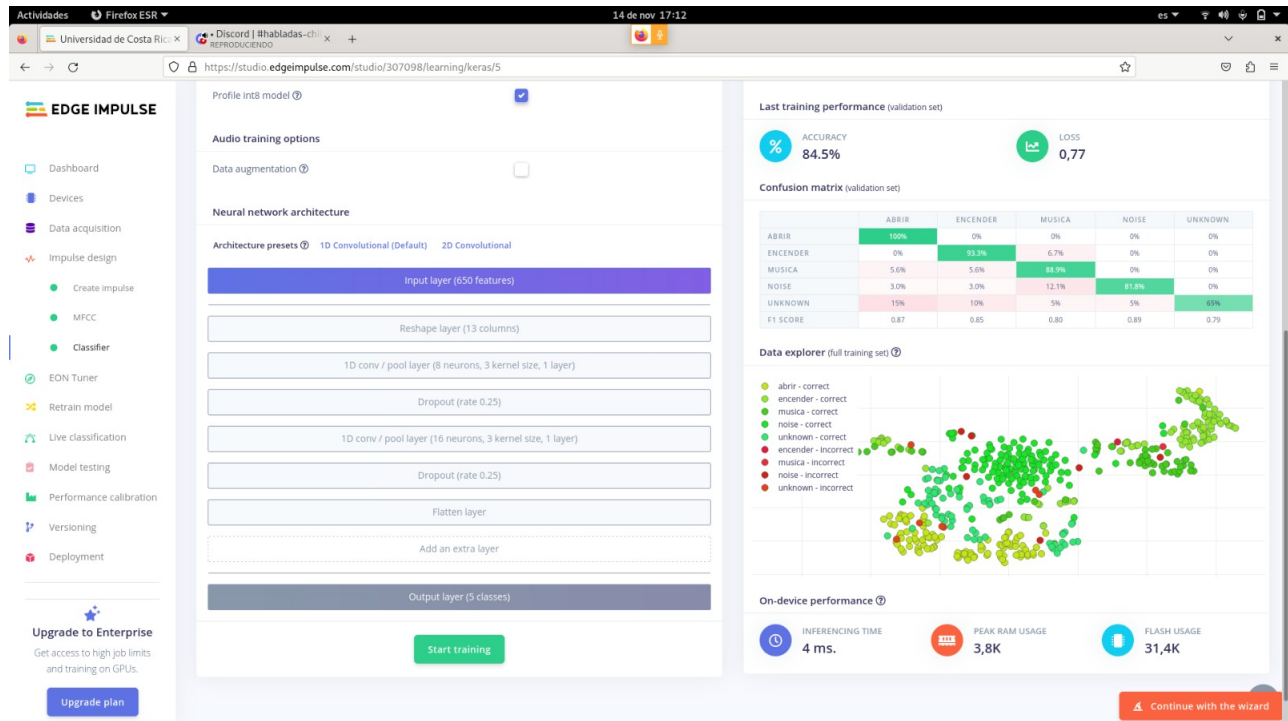


Figura 5: Resultados Obtenidos con el modelo implementado

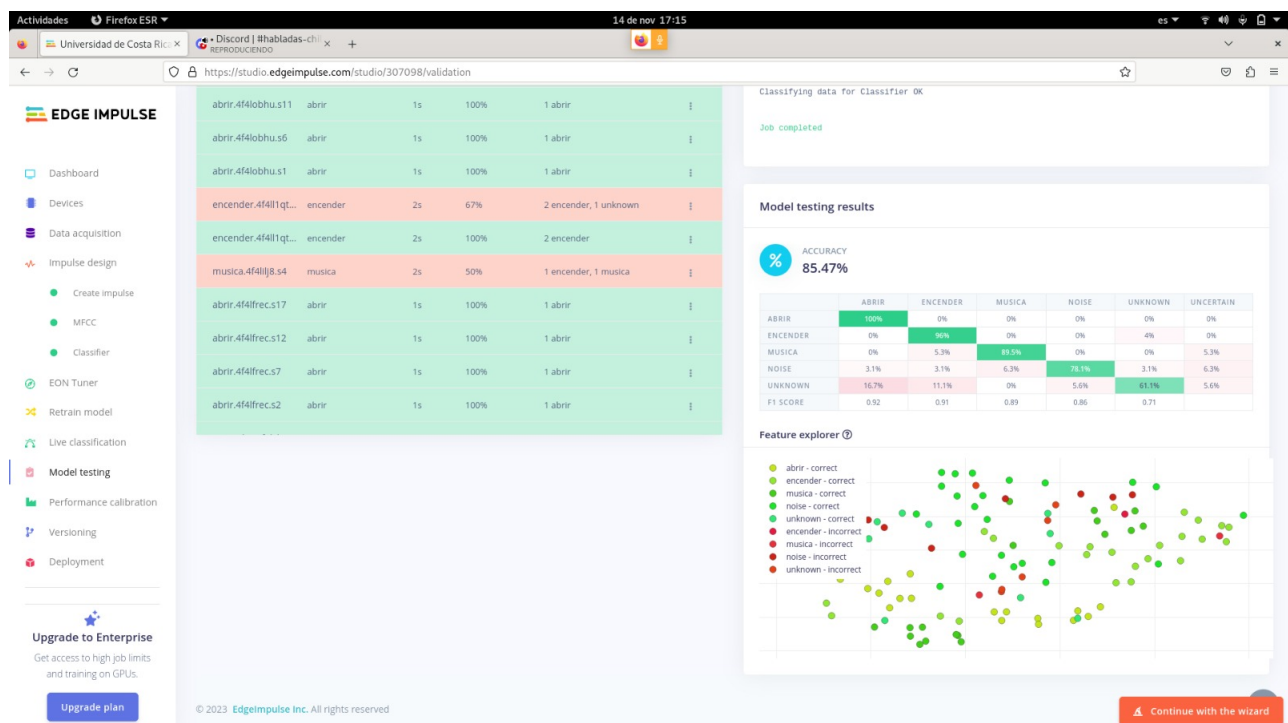


Figura 6: Resultados Obtenidos con el modelo implementado

Se puede observar que se logró alcanzar una precisión del 85 %. Este nivel de precisión indica una alta eficacia del modelo en clasificar correctamente los datos de entrada. El entrenamiento del modelo se realizó utilizando comandos de voz específicos: ".abrir", ".encenderz "música". Estas palabras fueron seleccionadas cuidadosamente para representar distintas categorías de comandos que el modelo debe reconocer y clasificar. Estos comando luego tendrán una funcionalidad mediante la plataforma de IoT *ThingsBoard*

Para mejorar la robustez y la precisión del modelo, se incorporó un conjunto de datos adicional de las librerías de Edge Impulse. Este conjunto de datos incluye una variedad de ruidos y sonidos desconocidos. La inclusión de estos datos ayuda al modelo a aprender a distinguir entre los comandos de voz deseados y otros sonidos ambientales o ruidos que podrían ocurrir en escenarios de uso real.

La adición de ruido y sonidos desconocidos en el proceso de entrenamiento es crucial para aumentar la robustez del modelo frente a variaciones y perturbaciones en los datos de entrada. Esto asegura que el modelo no solo sea preciso en un entorno controlado, sino también en situaciones del mundo real donde los ruidos de fondo y las variaciones acústicas son comunes.

3.4. Exportación del Modelo

Seguidamente, ya con el modelo en funcionamiento, se procedió a exportarlo como una librería. Esto se realizó con el fin de que el *Arduino Nano Sense 33 BLE* pueda cumplir su función inicial propuesta de un controlador por voz. Se tomó el ejemplo de la librería exportada llamado `nano_ble33_sense_microphone_continuous` y se ajustó para que imprimiera los datos deseados por consola.

Explicación del Código

El código comienza con un **Encabezado de Licencia**, que detalla los términos bajo los cuales se distribuye. Este es un comentario estándar para la licencia Apache 2.0.

En la sección de **Definiciones de Preprocesador**, encontramos varias macros clave. `EIDSP_QUANTIZE_FILTERBANK` controla la cuantización del banco de filtros, una técnica utilizada para optimizar el uso de la memoria. `EI_CLASSIFIER_SLICES_PER_MODEL_WINDOW` define el número de segmentos en una ventana de tiempo del modelo, lo que afecta cómo se procesan los datos de audio para la inferencia.

La **Inclusión de Bibliotecas** es esencial para el funcionamiento del código. `#include <PDM.h>` importa la biblioteca necesaria para manejar el micrófono digital PDM, mientras que `#include <Universidad_de_Costa_Rica-project-1_inferencing.h>` incluye los encabezados específicos para la inferencia del modelo de Edge Impulse.

En cuanto a la **Estructura y Variables Globales**, se declara la estructura `inference_t` para manejar los búferes de audio y los contadores relacionados. Además, se definen variables globales para gestionar el estado de la grabación y el procesamiento del audio.

La **Función setup** se encarga de configurar el puerto serie, mostrar mensajes iniciales y preparar todo para la inferencia y la grabación de audio.

El núcleo del código se encuentra en la **Función loop**, donde se realiza el bucle principal de grabación y procesamiento. Aquí, se ejecuta el modelo de inferencia y se manejan los resultados de la clasificación.

Las **Funciones de Manejo de Audio y PDM** incluyen `pdm_data_ready_inference_callback`, que maneja los datos de audio del micrófono PDM, `microphone_inference_start` para configurar los búferes de audio, `microphone_inference_record` que controla la grabación de audio, `microphone_audio_signal_get_data` para convertir los datos de audio al formato requerido por el modelo, y `microphone_inference_end` para liberar recursos al final.

Finalmente, la **Verificación del Sensor** asegura que el sensor configurado sea el adecuado para el modelo, en este caso, un micrófono. Esta verificación es crucial para el correcto funcionamiento del código con el modelo de Edge Impulse.

A continuación se muestra el código en funcionamiento ya corriendo dentro del Arduino.

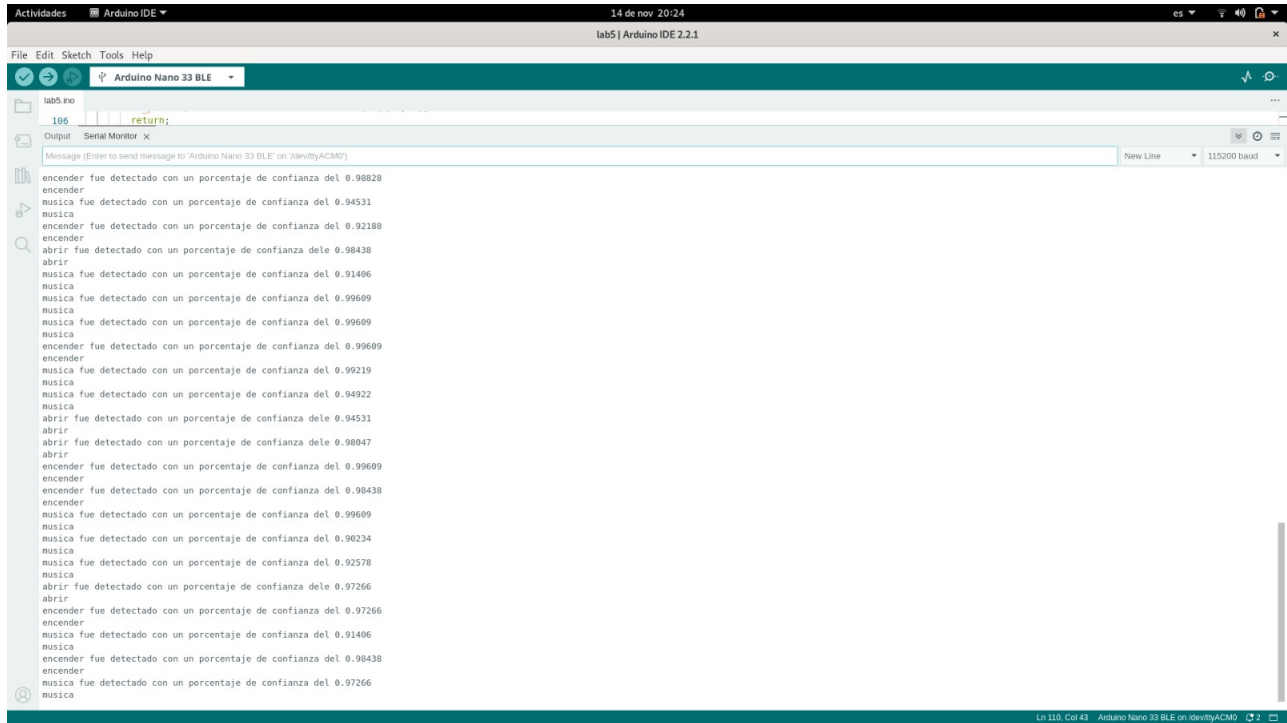


Figura 7: Librería instalada dentro del Arduino

Se puede apreciar que luego de realizar pruebas, diciendo las palabras con las que fue entrenado el modelo, se obtuvieron porcentajes de confianza por encima del 90 %. Este dato por lo tanto, confirma que el Arduino está detectando de manera correcta las palabras y que por lo tanto el entrenamiento del modelo funcionó con éxito.

3.5. IoT: Thingsboard y Código en Python

Finalmente, ya con el Arduino en funcionamiento y el modelo comportándose de la manera esperada, se procedió a implementar la última característica de poder conectar el Arduino en la plataforma de IoT: Thingsboard.

Para lograr esto, se utilizó un *script* desarrollado en el lenguaje de programación *Python*. Este código, tiene la finalidad de leer/escribir el puerto USB y que pueda enviar la información del modelo y lo que logra leer el Arduino para ser desplegados en un dashboard de la plataforma Iot thingsboard.

El código se estructura en una clase principal denominada *IOTClient*, la cual encapsula todas las funciones necesarias para establecer las conexiones, leer los datos del sensor y enviarlos al servidor. Hay 3 elementos importantes en el código para su funcionamiento:

- `time`: maneja las pausas entre envíos de datos
- `json`: convierte los datos a formato JSON

- serial: permite la comunicación a través del puerto serial
- paho.mqtt.client: logra la interacción con el protocolo MQTT.

La clase IOTClient mencionada anteriormente, maneja la configuración de las conexiones, la lectura de datos del sensor, y su envío al servidor. Al iniciar, establece las conexiones serial y MQTT, e inicia un bucle que lee los datos del sensor, los convierte a formato JSON, y los publica en el servidor MQTT con una pausa entre envíos. Si se ejecuta el script directamente, se crea una instancia de IOTClient y se inicia este proceso, permitiendo la transmisión continua de datos del sensor al servidor MQTT.

Iot Thingsboard

Luego del correcto funcionamiento del modelo en el Arduino y la implementación del código en Python para leer/escribir el puerto USB, se configuró la plataforma de Iot Thingsboard para poder manejar y visualizar los datos enviados y recibidos. Para la configuración en la plataforma, se utilizó la cuenta que fue creada por el profesor con el correo institucional.

A continuación se puede apreciar la data visualizada del Arduino mediante dicha plataforma:

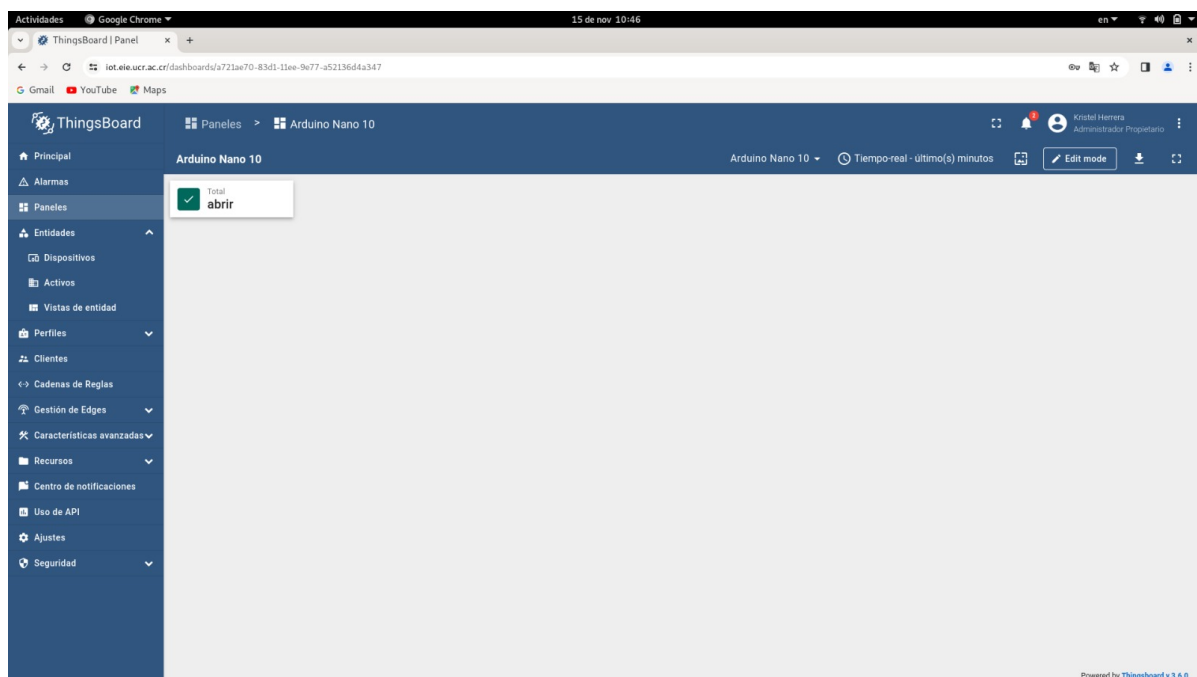
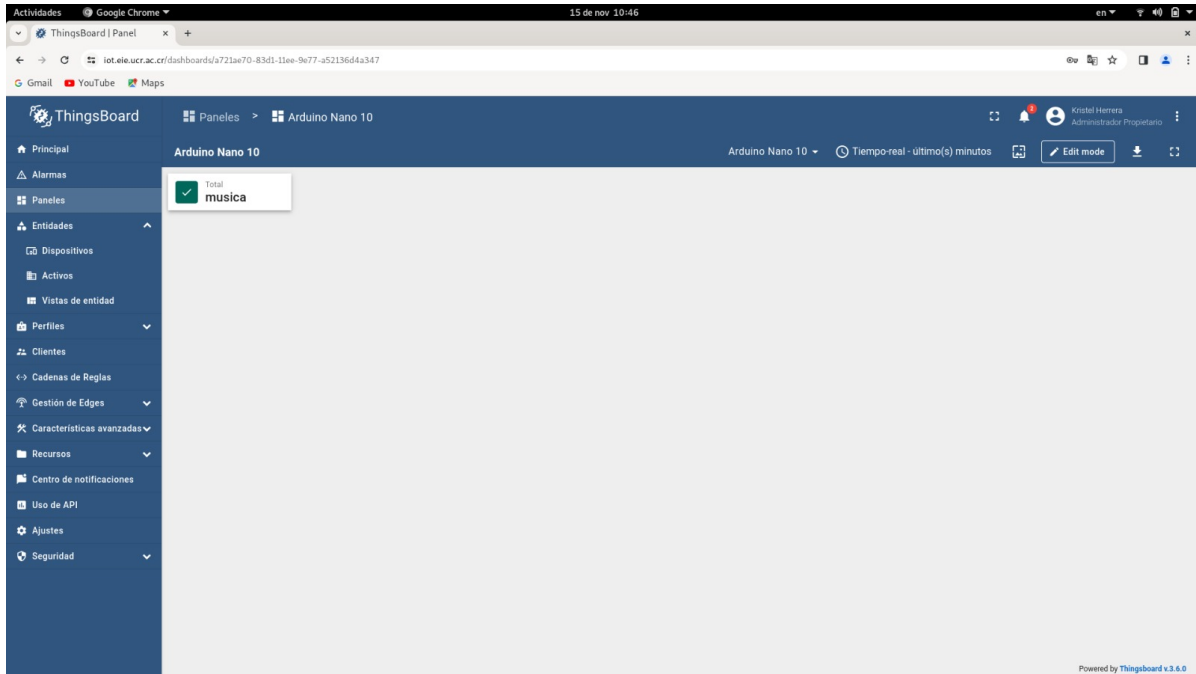
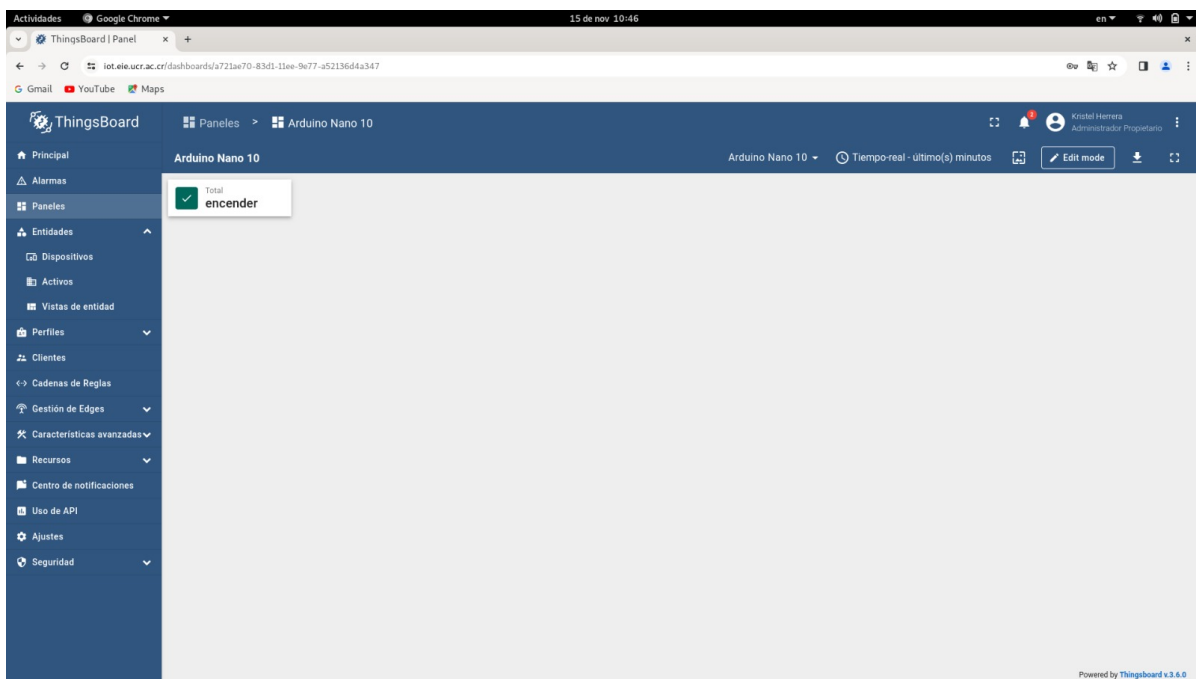


Figura 8: Comando *Abrir* en plataforma ThingsBoard

Figura 9: Comando *Musica* en plataforma ThingsBoardFigura 10: Comando *Encender* en plataforma ThingsBoard

4. Conclusiones y Recomendaciones

4.1. Conclusiones

4.2. Conclusiones

- Se cumplió de manera exitosa el objetivo del laboratorio de construir un controlador por voz con el Arduino Nano 33 BLE
- Se logró de manera correcta utilizar un modelo de Machine Learning para construir el controlador por voz mediante la plataforma de Edge Impulse
- Se alcanzó el objetivo de la manera esperada de mostrar los datos mediante IoT utilizando la plataforma ThingsBoard.

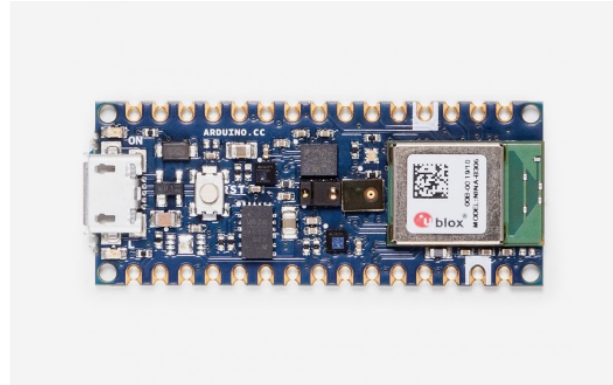
4.3. Recomendaciones

- Es de gran utilidad la plataforma Edge Impulse debido a que facilita de gran manera la construcción e implementación del modelo de Machine Learning en el Arduino ya que es bastante intuitiva y guía de gran manera al usuario en este proceso.
- Se recomienda recolectar diferentes grabaciones de diferentes personas para poder entrenar de mejor manera y hacer más robusto el modelo que se está implementando.
- Se recomienda reutilizar scripts que ya habían sido implementados en laboratorios pasados como fue el script de Python para IoT y así facilitar aún más el desarrollo del laboratorio.

Referencias

- [1] Arduino, *Arduino Nano 33 BLE Sense Product Reference Manual*, <https://content.arduino.cc/assets/NANO-33-BLE-SENSE-Product-Reference-Manual-v03.02.pdf>, Arduino, 02 2022, modified: 02/11/2022.
- [2] D. Situnayake, “How tensorflow helps edge impulse make ml accessible to embedded engineers,” *TensorFlow*, June 02 2021. [Online]. Available: <https://blog.tensorflow.org/2021/06/how-tensorflow-helps-edge-impulse-make-ml-accessible.html>
- [3] V. Tiwari, “Mfcc and its applications in speaker recognition,” *International journal on emerging technologies*, vol. 1, no. 1, pp. 19–22, 2010.

5. Anexos



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 ksp/s ADC
 - 128 bit AES/ECB/CCM/AAR co-processor
- **LSM9DS1** (9 axis IMU)
 - 3 acceleration channels, 3 angular rate channels, 3 magnetic field channels
 - $\pm 2/\pm 4/\pm 8/\pm 16$ g linear acceleration full scale
 - $\pm 4/\pm 8/\pm 12/\pm 16$ gauss magnetic full scale
 - $\pm 245/\pm 500/\pm 2000$ 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 rate/Interrupt 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: $\pm 3.5\%$ 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 dB SPL
 - 64 dB signal-to-noise ratio
 - Omnidirectional sensitivity
 - -26 dBFS \pm 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



Contents

1 The Board	5
1.1 Ratings	5
1.1.1 Recommended Operating Conditions	5
1.2 Power Consumption	5
2 Functional Overview	5
2.1 Board Topology	5
2.2 Processor	6
2.3 Crypto	6
2.4 IMU	7
2.5 Barometer and Temperature Sensor	7
2.6 Relative Humidity and Temperature Sensor	7
2.7 Digital Proximity, Ambient Light, RGB and Gesture Sensor	7
2.7.1 Gesture Detection	7
2.7.2 Proximity Detection	7
2.7.3 Color and ALS Detection	8
2.8 Digital Microphone	8
2.9 Power Tree	8
3 Board Operation	9
3.1 Getting Started - IDE	9
3.2 Getting Started - Arduino Web Editor	9
3.3 Getting Started - Arduino IoT Cloud	9
3.4 Sample Sketches	9
3.5 Online Resources	9
3.6 Board Recovery	9
4 Connector Pinouts	9
4.1 USB	10
4.2 Headers	10
4.3 Debug	11
5 Mechanical Information	11
5.1 Board Outline and Mounting Holes	11
6 Certifications	12
6.1 Declaration of Conformity CE DoC (EU)	12
6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021	12
6.3 Conflict Minerals Declaration	13
7 FCC Caution	13
8 Company Information	14
9 Reference Documentation	14
10 Revision History	14



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)

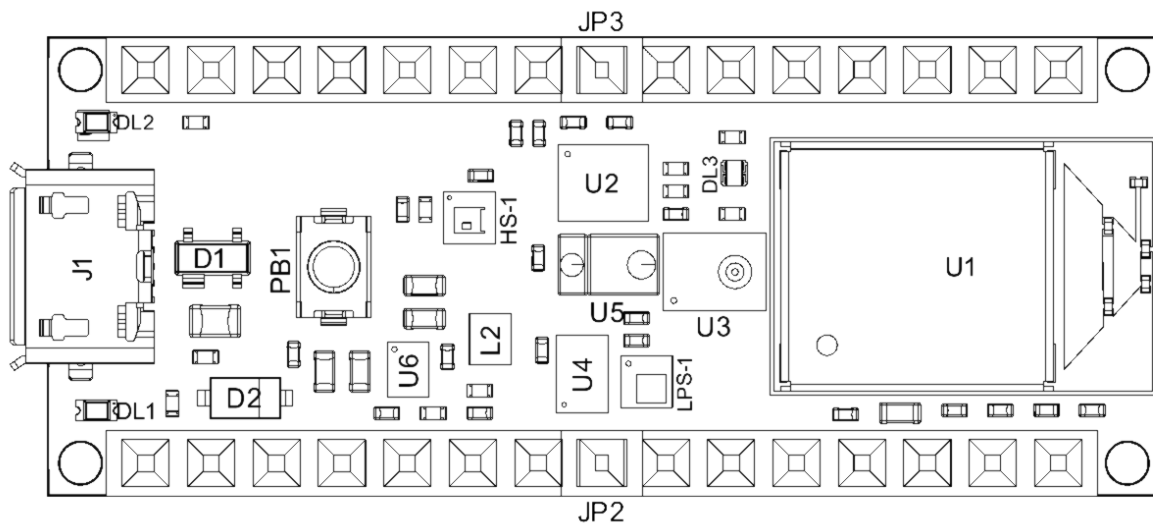
1.2 Power Consumption

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:



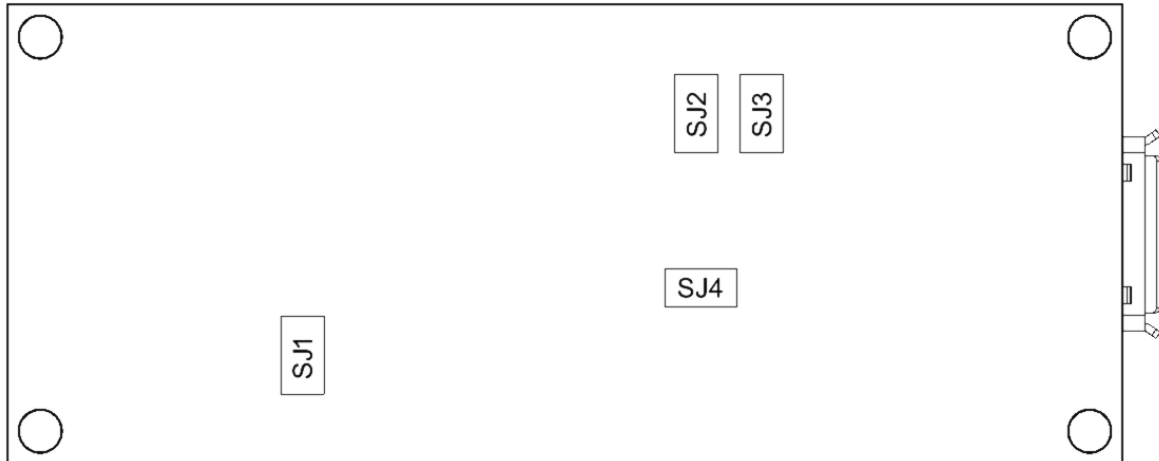
Board topology top

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 I²C 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 I²C 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 producing 16-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.

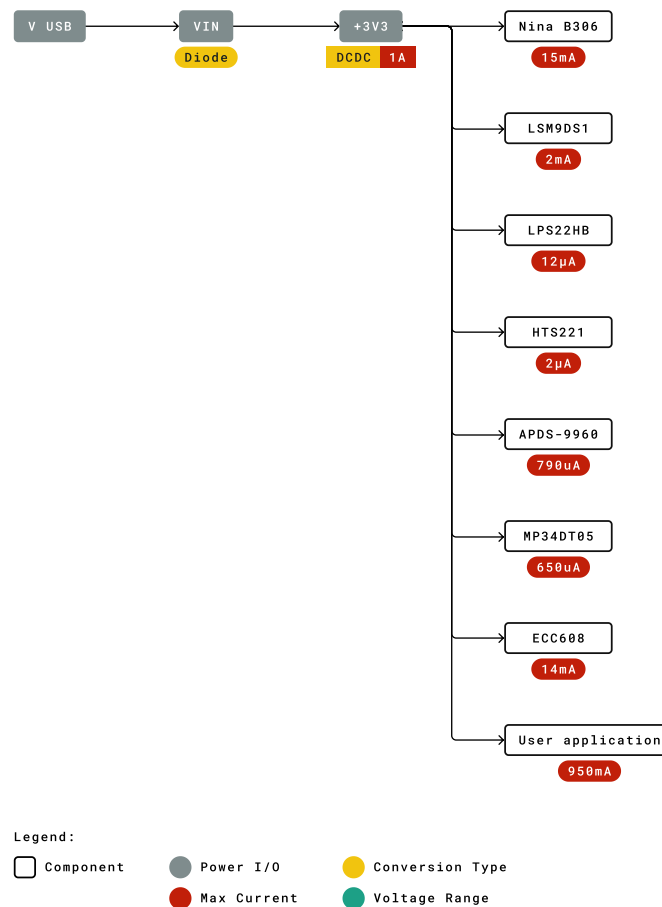
2.8 Digital Microphone

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]

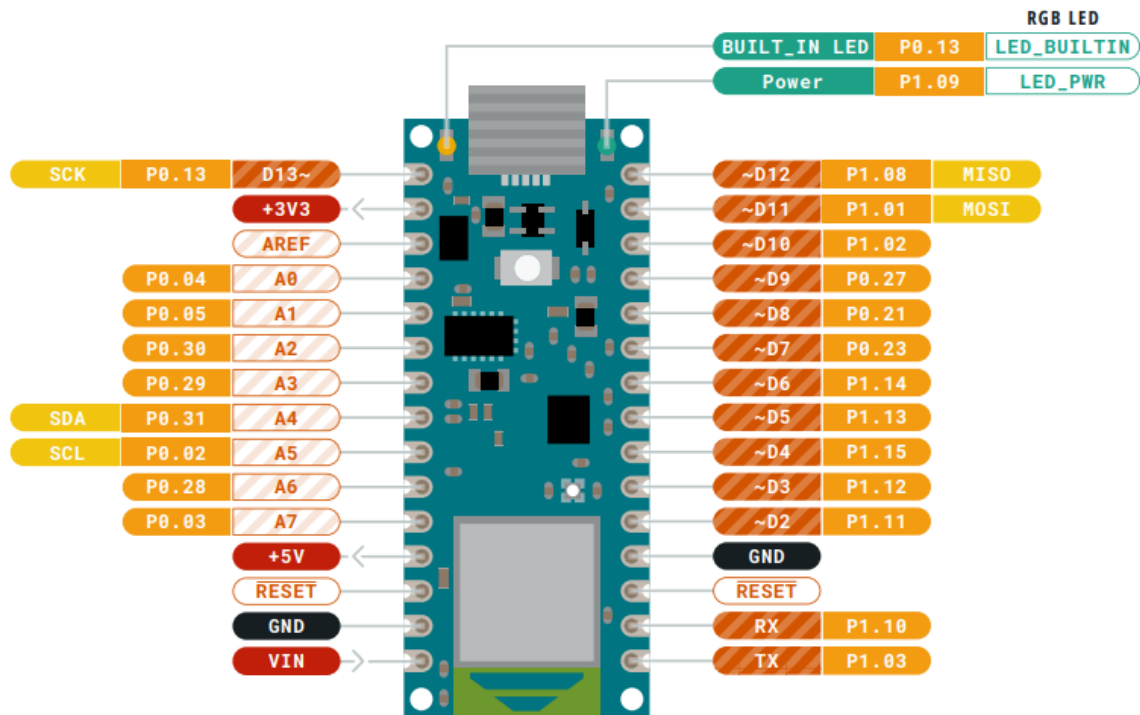
3.5 Online Resources

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



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



Pin	Function	Type	Description
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

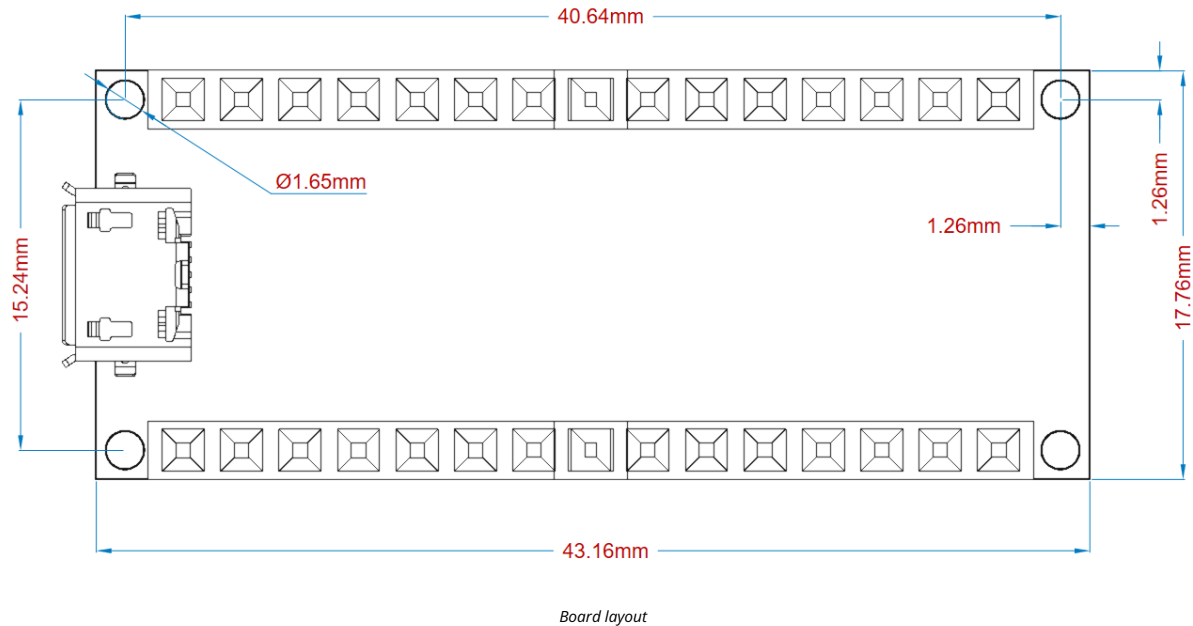
On the bottom side of the board, under the communication module, debug signals are arranged as 3x2 test pads with 100 mil pitch with pin 4 removed. Pin 1 is depicted in Figure 3 – Connector Positions

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

5 Mechanical Information

5.1 Board Outline and Mounting Holes

The board measures are mixed between metric and imperial. Imperial measures are used to maintain 100 mil pitch grid between pin rows to allow them to fit a breadboard whereas board length is Metric



6 Certifications

6.1 Declaration of Conformity CE DoC (EU)

We declare under our sole responsibility that the products above are in conformity with the essential requirements of the following EU Directives and therefore qualify for free movement within markets comprising the European Union (EU) and European Economic Area (EEA).

6.2 Declaration of Conformity to EU RoHS & REACH 211 01/19/2021

Arduino boards are in compliance with RoHS 2 Directive 2011/65/EU of the European Parliament and RoHS 3 Directive 2015/863/EU of the Council of 4 June 2015 on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

Substance	Maximum limit (ppm)
Lead (Pb)	1000
Cadmium (Cd)	100
Mercury (Hg)	1000
Hexavalent Chromium (Cr6+)	1000
Poly Brominated Biphenyls (PBB)	1000
Poly Brominated Diphenyl ethers (PBDE)	1000
Bis(2-Ethylhexyl) phthalate (DEHP)	1000
Benzyl butyl phthalate (BBP)	1000
Dibutyl phthalate (DBP)	1000
Diisobutyl phthalate (DIBP)	1000

Exemptions : No exemptions are claimed.

Arduino Boards are fully compliant with the related requirements of European Union Regulation (EC) 1907 /2006 concerning the Registration, Evaluation, Authorization and Restriction of Chemicals (REACH). We declare none of the SVHCs (<https://echa.europa.eu/web/guest/candidate-list-table>), the Candidate List of Substances of Very High Concern for authorization currently released by ECHA, is present in all products (and also package) in quantities totaling in a concentration equal or above 0.1%. To the best of our knowledge, we also declare that our products do not contain any of the substances listed on the "Authorization List"



(Annex XIV of the REACH regulations) and Substances of Very High Concern (SVHC) in any significant amounts as specified by the Annex XVII of Candidate list published by ECHA (European Chemical Agency) 1907 /2006/EC.

6.3 Conflict Minerals Declaration

As a global supplier of electronic and electrical components, Arduino is aware of our obligations with regards to laws and regulations regarding Conflict Minerals, specifically the Dodd-Frank Wall Street Reform and Consumer Protection Act, Section 1502. Arduino does not directly source or process conflict minerals such as Tin, Tantalum, Tungsten, or Gold. Conflict minerals are contained in our products in the form of solder, or as a component in metal alloys. As part of our reasonable due diligence Arduino has contacted component suppliers within our supply chain to verify their continued compliance with the regulations. Based on the information received thus far we declare that our products contain Conflict Minerals sourced from conflict-free areas.

7 FCC Caution

Any Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference
- (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC RF Radiation Exposure Statement:

1. This Transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2. This equipment complies with RF radiation exposure limits set forth for an uncontrolled environment.
3. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body.

English: User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both. This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

- (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 n' doit 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
Library Reference	https://www.arduino.cc/reference/en/

10 Revision History

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