Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0624 – Laboratorio de Microcontroladores
II ciclo 2024

Laboratorio # 5

STM32/Arduino: GPIO, Giroscopio, comunicaciones, TinyML

Leonardo Serrano Arias C17484
Lorena Solís Extteny B97657

Profesor: Marco Villalta

20 de Noviembre, 2024

Índice

Índice de figuras

Introducción

En este laboratorio se desarrolló un modelo para el reconocimiento de actividad humana utilizando un microcontrolador Arduino Nano 33 BLE Sense. Este modelo tiene como objetivo identificar tres movimientos principales: círculo, estacionario y .ªrriba-abajo", basándose en datos de aceleración, velocidad angular y orientación proporcionados por un sensor inercial (IMU LSM9DS1). A lo largo del proyecto, se implementaron diversas etapas, desde la captura de datos y su procesamiento, hasta la creación de un modelo de aprendizaje automático en la plataforma Edge Impulse, logrando su implementación en hardware embebido. Los resultados obtenidos demostraron un alto nivel de exactitud, superando el 99 % en las clasificaciones para los movimientos definidos, y una capacidad robusta para identificar anomalías. Finalmente, el modelo entrenado fue integrado al microcontrolador mediante una biblioteca compatible con Arduino IDE, garantizando su funcionamiento autónomo y eficiente.

El repositorio utilizado para el desarrollo de este laboratorio se encuentra en la siguiente dirección: `https://github.com/Leonardo-SA/IE-0624_II_2024_C17484_B97657.git`

Nota Teórica

Arduino Nano 33 BLE

El *Arduino Nano 33 BLE* [1] es una placa de desarrollo compacta, diseñada para aplicaciones de bajo consumo energético y comunicaciones inalámbricas mediante *Bluetooth Low Energy* (BLE). Esta placa está basada en el microcontrolador nRF52840 de Nordic Semiconductor, un procesador ARM Cortex-M4 de 32 bits con punto flotante, que ofrece una combinación de capacidades de procesamiento, eficiencia energética y conectividad. A continuación se presentan sus características técnicas:

- **Microcontrolador**:

  Modelo: nRF52840

  Arquitectura: Procesador ARM Cortex-M4F a 64 MHz, con soporte para punto flotante, facilitando el procesamiento de señales y la ejecución de algoritmos de aprendizaje automático ligero.

- **Voltaje de funcionamiento**: 3.3V.

- **Memoria**:

  Flash: 1 MB, para almacenamiento de programas y datos de sensores.

  SRAM: 256 KB, optimizada para operaciones de cálculo en tiempo real y almacenamiento de variables temporales.

- **Pines de E/S**:

  14 pines digitales (PWM habilitado).

  8 pines de entrada analógica. Permite la integración de múltiples dispositivos y sensores externos.

- **Sensores integrados**:

  IMU: LSM9DS1 de 9 ejes (acelerómetro, giroscopio y magnetómetro).

  Sensor de proximidad y luz: APDS9960 (detección de proximidad, color RGB, intensidad de luz y gestos).

Audio y Video: Micrófono digital MP34DT05 y cámara OV7675 (solo en la versión BLE Sense).

- **Comunicación**: Compatible con interfaces UART, SPI e I2C, facilitando la integración con dispositivos de comunicación y otros periféricos.
- **Dimensiones**: 45x18 mm, ideal para proyectos portátiles y aplicaciones IoT.

Adicionalmente, se agrega el diagrama de pines acontinuación:

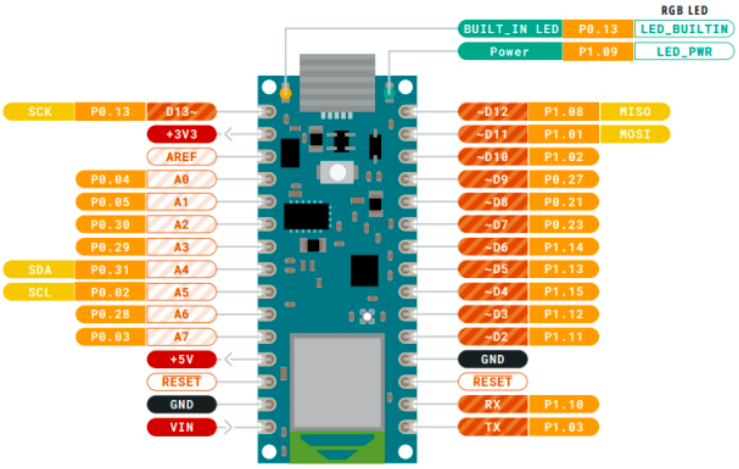


Figura 1: Diagrama de pines del Arduino Nano BLE 33 [1]

Seguidamente, se muestra la topología de la placa:

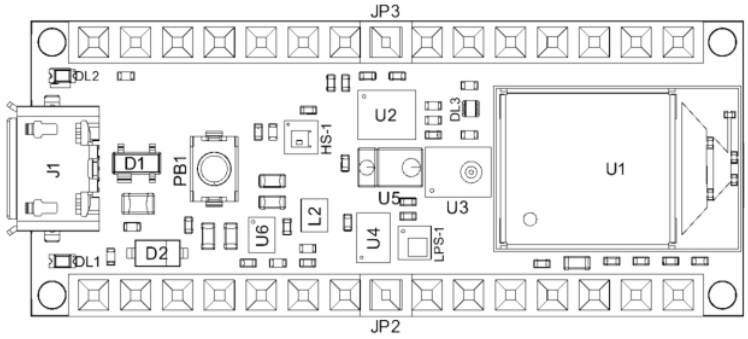


Figura 2: Topología del Arduino Nano BLE 33 [1]

A continuación, se muestra el árbol de potencia del microcontrolador:

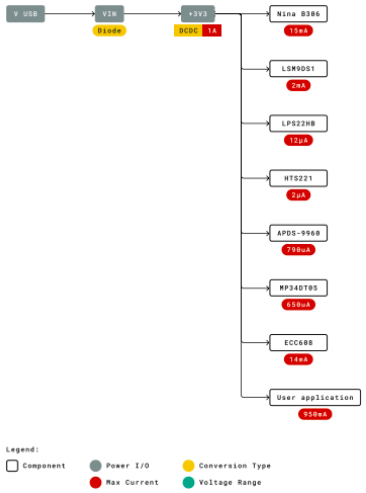


Figura 3: Arbol de potencia del Arduino Nano BLE 33 [1]

Por otro lado, en la siguiente imagen se puede observar el diagrama de bloques del microcontrolador nRF52840:

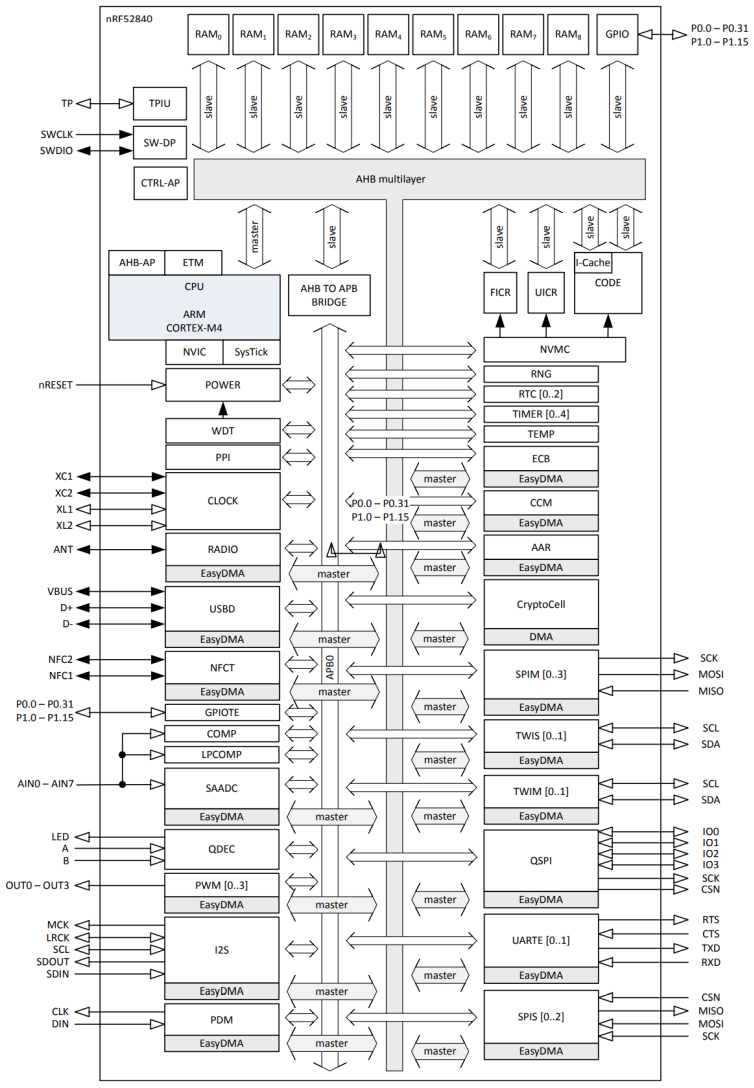


Figura 4: Diagrama de bloques del MCU nRF52840 [2]

Componentes del Arduino Nano 33 BLE

IMU LSM9DS1

El LSM9DS1 [3] es un sistema en paquete que combina sensores de aceleración, velocidad angular y magnetómetro. Este dispositivo es clave en aplicaciones de reconocimiento de actividad humana (HAR) ya que permite detectar la orientación, rotación y movimiento de la placa.

- **Acelerómetro**:

  Rango de medida: configuraciones ajustables de ±2g, ±4g, ±8g o ±16g, permitiendo capturar aceleraciones desde movimientos suaves hasta rápidos.

  Resolución: mide aceleraciones lineales en 3 ejes con una resolución que depende del rango configurado.

  Aplicaciones: útil para detectar cambios de posición, inclinación y patrones de movimiento en aplicaciones de HAR.

- **Giroscopio**:

  Rango de medida: configuraciones de ±245, ±500 y ±2000 dps (grados por segundo), adaptándose a diferentes tipos de rotación y velocidad.

  Resolución: mide velocidades angulares en 3 ejes, permitiendo detectar rotaciones en tiempo real.

  Aplicaciones: esencial para captar rotaciones rápidas, como giros, y movimientos de rotación, útiles en sistemas de reconocimiento de gestos y actividad física.

- **Magnetómetro**:

  Rango de medida: configuraciones de ±4, ±8, ±12 y ±16 gauss, ajustables para diferentes niveles de sensibilidad.

  Aplicaciones: permite la orientación mediante la detección del campo magnético terrestre, complementando las lecturas del acelerómetro y giroscopio para obtener datos completos de la orientación de la placa.

El LSM9DS1 utiliza interfaces I2C y SPI para comunicación, y permite configurar los sensores en modos de bajo consumo para reducir la demanda energética, ideal para aplicaciones portátiles y de IoT.

Sensor de proximidad, color, y gestos APDS9960

El sensor APDS9960 [6] es un dispositivo multifuncional que integra la detección de proximidad, medición de luz ambiente, y reconocimiento de gestos. Estas capacidades son útiles en proyectos de interacción y en aplicaciones de HAR donde es necesario detectar movimientos específicos de la mano u objetos cercanos. Sus características incluyen:

- **Detección de gestos**: Utiliza cuatro fotodiodos direccionales para medir la energía infrarroja (IR) reflejada y así detectar movimientos en direcciones específicas (arriba, abajo, izquierda, derecha).

- **Sensor de luz ambiental y color**: Capta niveles de luz RGB y detecta la intensidad de la luz ambiental, ajustándose automáticamente a la iluminación del entorno.

- **Sensor de proximidad**: Detecta objetos cercanos midiendo la intensidad de la luz IR reflejada. Puede configurarse para activar funciones cuando se detecta un objeto a cierta distancia.

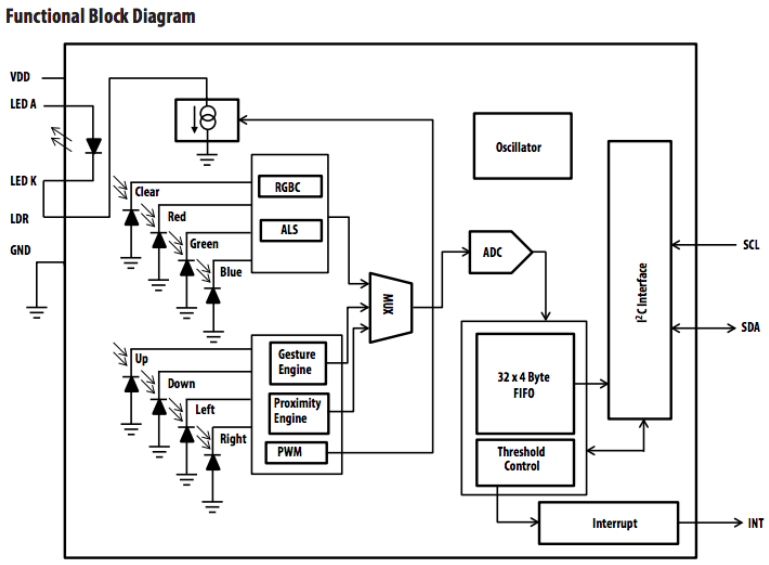


Figura 5: Diagrama de bloques del sensor de proximidad APDS9960 [6]

Micrófono digital MP34DT05 y cámara OV7675

El micrófono MP34DT05 [4] y la cámara OV7675 [5] añaden funcionalidades de captura de audio e imagen al Arduino Nano 33 BLE, abriendo posibilidades para aplicaciones de monitoreo en tiempo real y control mediante voz o imagen.

- **Micrófono MP34DT05**:

  Resolución: Capta sonidos del ambiente y permite realizar análisis de audio o comandos de voz.

  Aplicaciones: Usado en aplicaciones de reconocimiento de voz, monitoreo de audio o sistemas de activación por sonido.

- **Cámara OV7675**:

  Resolución máxima: 640x480 píxeles (VGA), proporcionando imágenes claras para capturas básicas.

  Aplicaciones: Ideal para proyectos de reconocimiento de gestos visuales, detección de movimiento, o cualquier aplicación que requiera entrada visual.

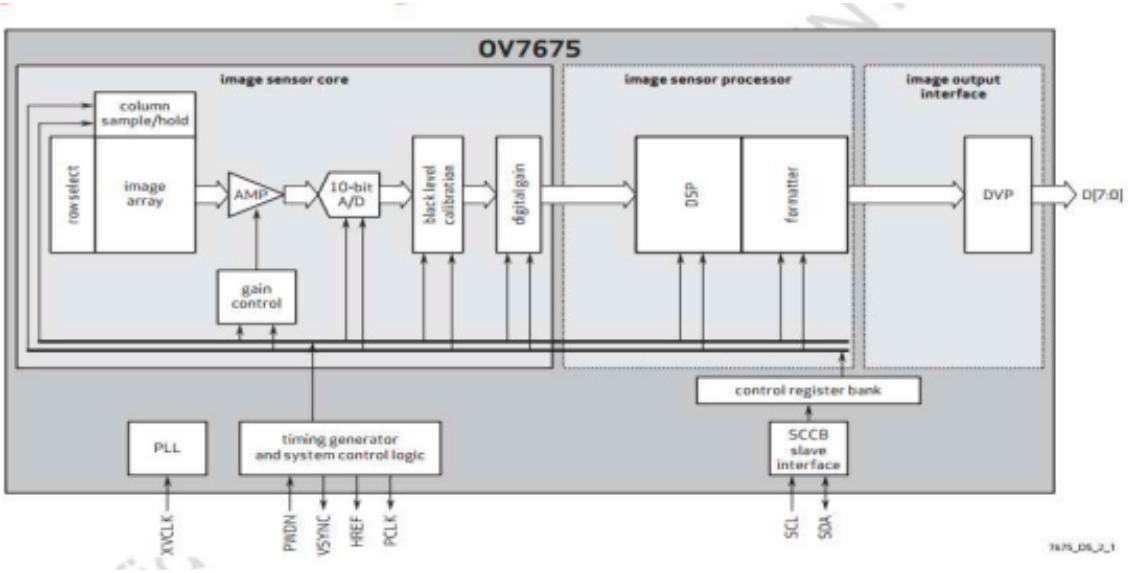


Figura 6: Diagrama de bloques del módulo de la cámara OV7675 [5]

Registros y periféricos

Los registros que utiliza este microcontrolador se muestran en la siguiente tabla [2]:

Tablas 1: Instancias

Dirección Base	Periférico	Instancia	Descripción	Configuración
0x50000000	GPIO	GPIO	Entrada y salida de propósito general	Obsoleto
0x50000000	GPIO	P0	Entrada y salida de propósito general, puerto 0	Implementado P0.00 a P0.31
0x50000300	GPIO	P1	Entrada y salida de propósito general, puerto 1	Implementado P1.00 a P1.15

Tablas 2: Registros

Registro	Desplazamiento	Descripción
OUT	0x504	Escribe en el puerto GPIO
OUTSET	0x508	Establece bits individuales en el puerto GPIO
OUTCLR	0x50C	Borra bits individuales en el puerto GPIO
IN	0x510	Lee el puerto GPIO
DIR	0x514	Dirección de los pines GPIO
DIRSET	0x518	Registro de configuración DIR
DIRCLR	0x51C	Registro para borrar configuración DIR
LATCH	0x520	Registro de retención que indica qué pines GPIO cumplen criterios
DETECTMODE	0x524	Selecciona entre comportamiento por defecto y modo LDETECT
PIN_CNF[0]	0x700	Configuración de los pines GPIO

Tablas 3: Continuación de registros

Registro	Desplazamiento	Descripción
OUT	0x504	Escribe en el puerto GPIO
OUTSET	0x508	Establece bits individuales en el puerto GPIO
OUTCLR	0x50C	Borra bits individuales en el puerto GPIO
IN	0x510	Lee el puerto GPIO
DIR	0x514	Dirección de los pines GPIO
DIRSET	0x518	Registro de configuración DIR
DIRCLR	0x51C	Registro para borrar configuración DIR
LATCH	0x520	Registro de retención que indica qué pines GPIO cumplen criterios
DETECTMODE	0x524	Selecciona entre comportamiento por defecto y modo LDETECT
PIN_CNF[0]	0x700	Configuración de los pines GPIO

Por otra parte, la instanciación de los periféricos se muestra a continuación:

Tablas 4: Instanciación de los periféricos

ID	Dirección Base	Periférico	Instancia	Descripción
0	0x40000000	CLOCK	CLOCK	Control de reloj
0	0x40000000	POWER	POWER	Control de energía
0	0x50000000	GPIO	GPIO	Entrada y salida de propósito general (obsoleto)
0	0x50000000	GPIO	P0	Entrada y salida de propósito general, puerto 0
0	0x50000300	GPIO	P1	Entrada y salida de propósito general, puerto 1
0	0x40001000	RADIO	RADIO	Radio de 2.4 GHz
2	0x40002000	UART	UART0	Transmisor/receptor asincrónico universal (obsoleto)

Conceptos importantes

Machine Learning

El aprendizaje automático (ML, por sus siglas en inglés) es una rama de la inteligencia artificial que se enfoca en el uso de datos y algoritmos para permitir que las máquinas simulen la forma en que los humanos adquieren conocimiento. Este proceso implica una mejora constante en la precisión de los resultados a medida que el modelo aprende de los datos proporcionados. ML tiene aplicaciones que van desde clasificaciones hasta predicciones, adaptándose a diversas áreas de la informática e industrias. [9]

El funcionamiento de los algoritmos de aprendizaje automático se divide en tres fases principales. En primer lugar, el proceso de decisión, donde el algoritmo analiza los datos de entrada para identificar patrones y realizar predicciones. Luego, se utiliza una función de error para medir la discrepancia entre las predicciones y los resultados esperados, evaluando así la precisión del modelo. Finalmente, un proceso de optimización ajusta los parámetros del modelo para mejorar su desempeño, iterando hasta alcanzar un nivel aceptable de precisión. Este ciclo de evaluación y mejora es esencial para el aprendizaje continuo del sistema. [9]

Edge Impulse

Edge Impulse es una plataforma para desarrollar algoritmos de aprendizaje máquina enfocados a implementarse en sistemas embebidos como microcontroladores o computadoras con recursos reducidos. Tiene disponibles diversas herramientas que la hacen adecuada tanto para principiantes como usuarios avanzados. La practicidad de esta herramienta es que no necesitas involucrarte demasiado con el código, puedes implementar tu algoritmo con ingresar tu base de datos, ajustas los hiperparámetros y entrenas el programa. [10]

Tensorflow

TensorFlow es una herramienta fundamental en el desarrollo de inteligencia artificial y aprendizaje automático. Es una biblioteca de código abierto diseñada para Machine Learning (ML), desarrollada por Google para abordar las necesidades que surgen al trabajar con redes neuronales artificiales. Esta plataforma permite construir y entrenar redes neuronales capaces de detectar patrones y replicar razonamientos propios del ser humano. Una de las principales ventajas de TensorFlow es su versatilidad. Además de su enfoque en redes neuronales, es una solución multiplataforma que puede ejecutarse en CPUs, GPUs e incluso en unidades de procesamiento especializadas como las TPUs (Tensor Processing Units), optimizando su rendimiento en distintos entornos. [11]

Librerías de Arduino IDE

A continuación se detallan las librerías utilizadas en el Arduino [12]:

`Laboratorio_5_inferencing.h`

Esta librería forma parte del SDK de Edge Impulse y se emplea para ejecutar inferencias de modelos de machine learning en dispositivos embebidos. Facilita la integración de modelos preentrenados generados por Edge Impulse, permitiendo la clasificación de datos en tiempo real y la generación de predicciones.

`Arduino_LSM9DS1.h`

Proporciona una interfaz para el sensor `LSM9DS1`, un módulo de 9 grados de libertad (DoF) que combina acelerómetro, giroscopio y magnetómetro. Este sensor permite medir:

- **Aceleración:** En los ejes X, Y y Z.
- **Velocidad angular:** Utilizando el giroscopio.
- **Campo magnético:** Con el magnetómetro.

`Arduino_LPS22HB.h`

Esta librería permite interactuar con el sensor barométrico `LPS22HB`, el cual mide la presión atmosférica y la convierte a kilopascales (`kPa`). Este sensor es útil para:

- Calcular la altitud.
- Detectar cambios ambientales relacionados con la presión.

`Arduino_HTS221.h`

Proporciona acceso al sensor `HTS221`, que mide:

- **Temperatura.**
- **Humedad relativa.**

Este sensor es útil en aplicaciones que requieren monitoreo de condiciones ambientales.

`Arduino_APDS9960.h`

Permite la interacción con el sensor `APDS9960`, el cual incluye las siguientes funcionalidades:

- **Color:** Mide intensidades de rojo, verde, azul y brillo.
- **Proximidad:** Detecta objetos cercanos al sensor.
- **Gestos:** Identifica movimientos como deslizamientos hacia la izquierda, derecha, arriba y abajo.

Hardware

Tabla de componentes electrónicos

A continuación, se muestra una tabla con los componentes utilizados para el desarrollo del proyecto, el cual fue únicamente el Arduino Tiny Machine Learning Kit con el Arduino Nano BLE 33:

Componente	Descripción	Cantidad	Precio ($)
Arduino Tiny Machine Learning Kit	Microcontrolador	1	60

Tablas 5: Componentes utilizados

Desarrollo

Diagrama de flujo

A continuación se muestra el diagrama de flujo que sigue este proyecto.

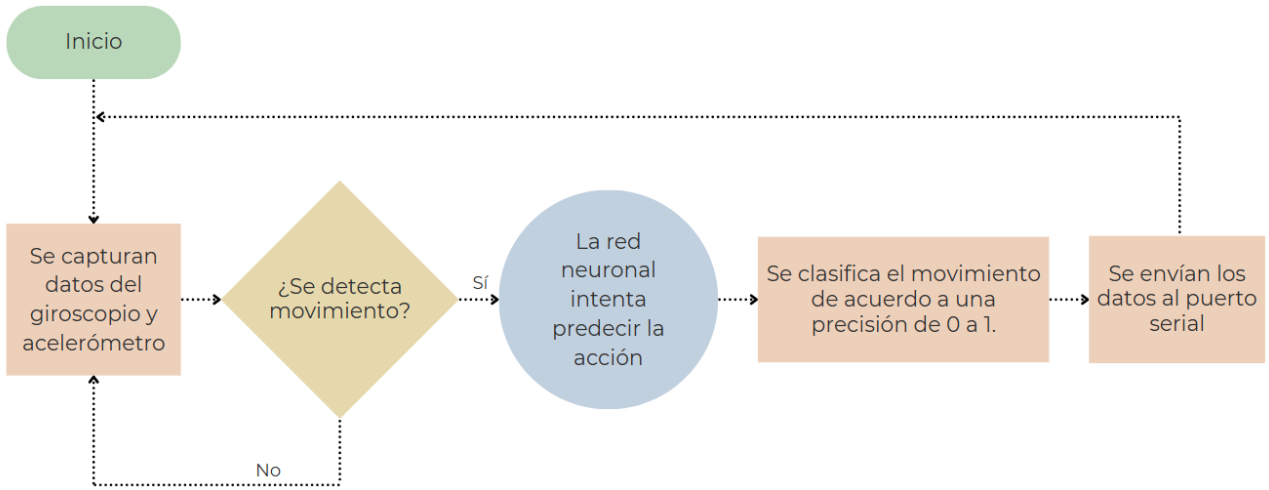


Figura 7: Diagrama de flujo

Creación del modelo

Para la creación del modelo se utilizó la plataforma *Edge Impulse*, lo que requirió la instalación de las herramientas **arduino-cli** y **edge-impulse-cli**. Una vez completada la instalación de los programas y paquetes necesarios, se procedió a crear un proyecto en Edge Impulse, nombrado *Laboratorio_5*.

El primer paso consistió en conectar el dispositivo Arduino Nano 33 BLE Sense al proyecto. Para lograrlo, se siguieron las instrucciones en la documentación oficial del microcontrolador proporcionada por Edge Impulse [8]. Tras establecer la conexión correctamente, se pudo comenzar con la recolección de datos. En la Figura 8 se muestra el Arduino Nano conectado correctamente a la plataforma.

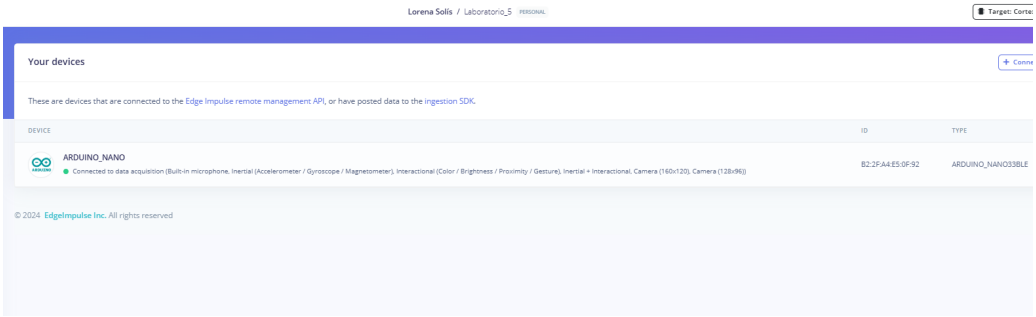


Figura 8: Arduino Nano 33 BLE Sense conectado a Edge Impulse.

Adquisición de datos

En la sección de **Data Acquisition** se procedió a grabar los tres movimientos que se desean implementar en el modelo: *arriba-abajo*, *círculo* y *estacionario*. En la Figura 9, se observa que el modelo fue entrenado durante un tiempo total de 20 minutos y 37 segundos, incluyendo tanto la fase de pruebas como la de entrenamiento. Además, se utilizó el sensor inercial que utiliza el acelerómetro, giroscopio y magnetómetro del IMU LSM9DS1.

Cada *sample* registrado tuvo una duración de 10 segundos. Se realizaron un total de 40 muestras para los movimientos *círculo* y *arriba-abajo*, mientras que para el movimiento *estacionario* se tomaron 24 muestras, dado que, al ser un movimiento más simple, no requiere tantos datos. Con esta información recopilada, fue posible generar los impulsos necesarios para entrenar el modelo de manera completa.

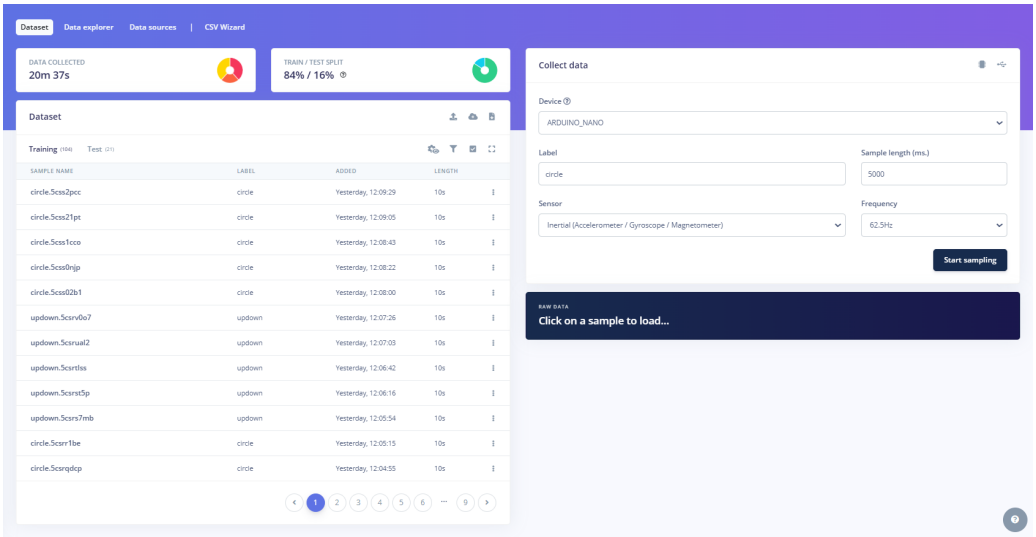


Figura 9: Set de datos para entrenar y testear el modelo

Impulsos

Con el conjunto de datos de entrenamiento listo, se procedió a diseñar un impulso. Un impulso toma los datos crudos, los divide en ventanas más pequeñas, aplica bloques de procesamiento de señales para extraer características, y utiliza un bloque de aprendizaje para clasificar nuevos datos. En el entorno de Edge Impulse, dentro de la sección *Create Impulse*, se configuraron los siguientes parámetros:
Tamaño de la ventana: **2000 ms**.
Incremento de ventana: **80 ms**.
Bloques añadidos: *Spectral Analysis*, *Classification* y *Anomaly Detection*.
Finalmente, se guardó el diseño del impulso seleccionando la opción *Save impulse*.

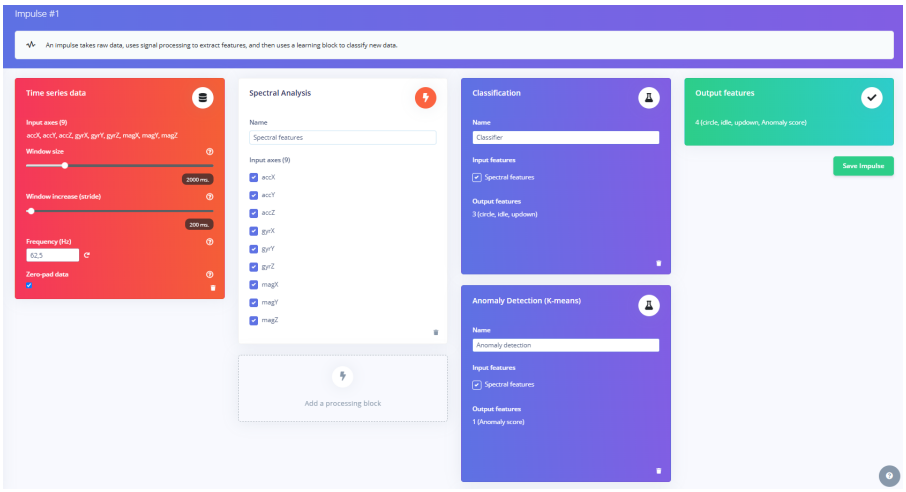


Figura 10: Impulsos para el entrenamiento del modelo

- **Spectral Features**: El bloque de procesamiento de señales *Spectral Analysis*, el cual aplica un filtro, realiza un análisis espectral sobre la señal y extrae datos relacionados con la frecuencia y la potencia espectral.

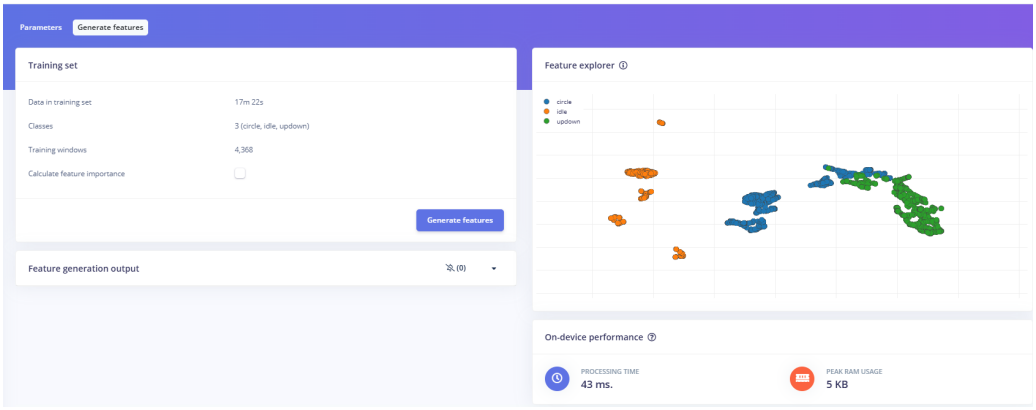


Figura 11: Impulso de características espectrales

Como se observa en la figura 11, al generar las funciones se muestra un gráfico y se observan clusters de cada uno de los movimientos lo que indica que el modelo de ML será capaz de verlo también.

- **Classifier**: El bloque de aprendizaje *Classifier* toma las características espectrales y aprende a distinguir entre las clases definidas (*idle*, *circle*, y *updown*). Para este proceso, se utilizan los siguientes parámetros de configuración:

Número de ciclos de entrenamiento: Se configuró para 100 ciclos, lo que implica que el modelo ajusta sus parámetros en 100 iteraciones completas sobre el conjunto de datos de entrenamiento.

Tasa de aprendizaje: Se estableció en 0.0005, un valor pequeño que permite una actualización gradual de los pesos de la red neuronal durante el entrenamiento, evitando ajustes demasiado grandes que podrían llevar a un sobreajuste o a una convergencia inestable.

Tamaño del conjunto de validación: Se asignó un 60 % de los datos para la validación, lo que permite evaluar el desempeño del modelo en datos no vistos durante el entrenamiento y así detectar posibles problemas de sobreajuste.

Tamaño del lote (Batch size): Se configuró el tamaño del lote en 32, lo que significa que el modelo actualiza sus parámetros después de procesar 32 muestras a la vez.

En la figura 12, se muestra la clasificación realizada con esta configuración de entrenamiento con un resultado del 99.8 % de exactitud y un 0.02 de perdidas.

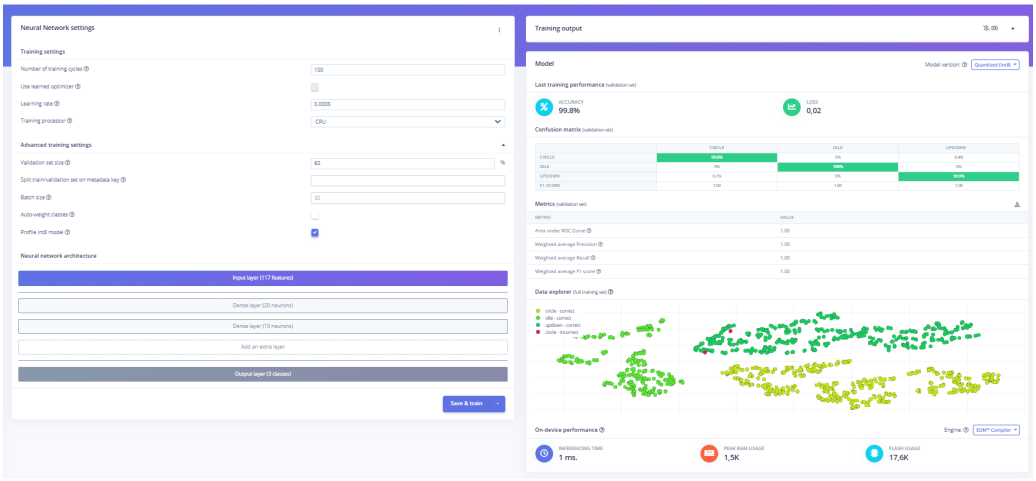
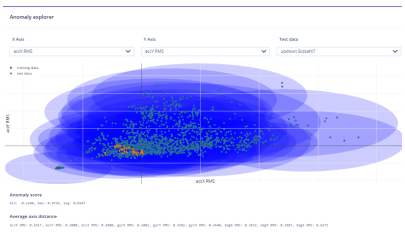
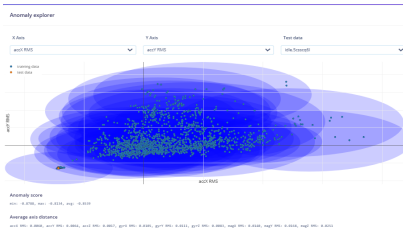


Figura 12: Impulso de clasificación

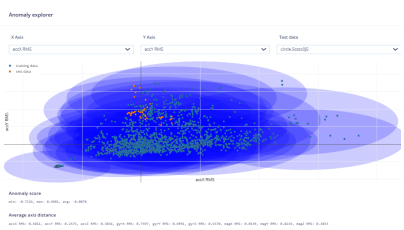
- **Anomaly Detection** El *anomaly detection* ayuda a identificar datos que son significativamente diferentes de los que el modelo ha visto durante el entrenamiento. En lugar de clasificar estos datos como uno de los grupos predefinidos, se utiliza un segundo modelo que crea clústeres alrededor de los datos conocidos. Si un nuevo dato se encuentra demasiado alejado de estos clústeres, se considera una anomalía y se marca como no confiable, evitando que el modelo realice una clasificación errónea. En las imágenes de la Figura 13 se muestran las anomalías para tres movimientos: En la subfigura 13a, correspondiente al movimiento *updown*, se observan puntos alejados del clúster principal, indicando anomalías.Seguidamente, la subfigura 13b, para el movimiento *idle*, las anomalías se presentan como variaciones fuera del patrón estacionario. Finalmente, en la subfigura 13c, para el movimiento *circle*, se destacan algunas muestras que no siguen el comportamiento esperado de un movimiento circular. Estas gráficas permiten visualizar cómo el modelo detecta comportamientos anómalos en los diferentes movimientos.



(a) Anomalías para movimiento updown.



(b) Anomalías para movimiento idle.



(c) Anomalías para movimiento circle.

Figura 13: Anomalías detectadas para diferentes movimientos.

Clasificación en tiempo real desde Edge Impulse

En la clasificación en tiempo real, se evaluó cómo el modelo clasifica datos en escenarios nuevos, no vistos durante el entrenamiento. Este análisis permite verificar tanto la precisión como la robustez del modelo frente a datos reales. Para esta evaluación, se tomaron muestras de 10 segundos y 5 segundos, explorando así diferentes modalidades del modelo.

En primer lugar, para el movimiento updown, mostrado en la figura 14, se observa que el modelo clasifica correctamente la mayoría de las muestras, obteniendo un resultado de 39/42 clasificaciones correctas. Esto corresponde a una exactitud aproximada del $92,8\,\%$, lo que indica que el modelo no solo identifica adecuadamente el movimiento, sino que también clasifica correctamente las anomalías.

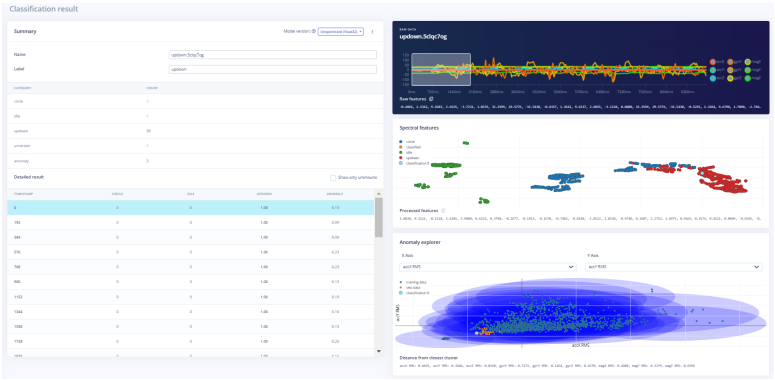


Figura 14: Clasificación en tiempo real del movimiento updown

Por otra parte, en la prueba del movimiento idle, representado en la figura 15, el modelo logró una exactitud del $100\,\%$, clasificando correctamente las 42 instancias evaluadas. Este resultado demuestra la capacidad del modelo para identificar este tipo de movimiento con total precisión bajo las condiciones de prueba en tiempo real.

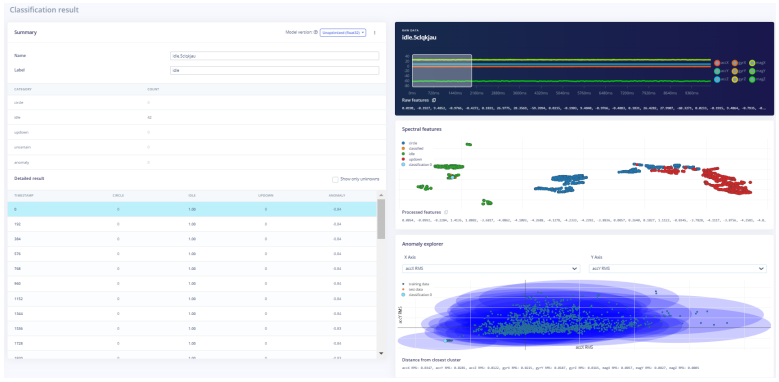


Figura 15: Clasificación en tiempo real del movimiento idle

En el caso del movimiento circle, representado en la figura 16, el modelo alcanzó una exactitud del 97,6 %, clasificando correctamente 41 de las 42 instancias evaluadas y presentando una única anomalía. Estos resultados reflejan un desempeño sólido en la clasificación en tiempo real de este tipo de movimiento.

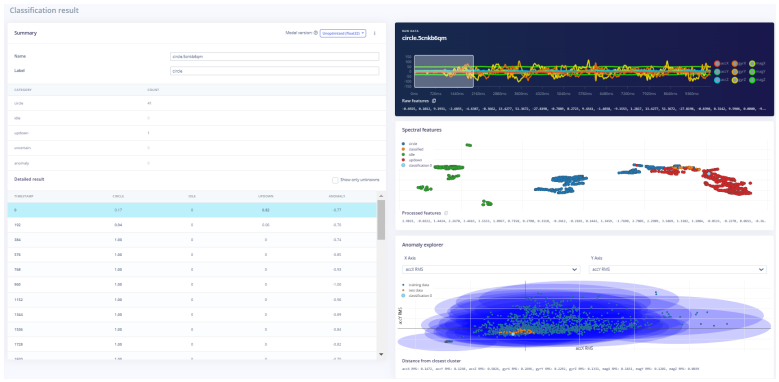


Figura 16: Clasificación en tiempo real del movimiento circle

Prueba de modelo

Se realizó una prueba final de clasificación antes del deployment, en la cual el modelo evaluó todos los datos obtenidos tanto en la etapa de adquisición de datos (data acquisition) como en la clasificación en tiempo real (live classification) con el propósito de prueba. Esta prueba involucró la ejecución de múltiples clasificaciones de manera simultánea. Como se observa en la figura 17, el modelo mostró un desempeño exitoso, alcanzando una exactitud del 94,65 %. Se clasificaron correctamente las anomalías y, en un número reducido de casos, se observaron errores de clasificación.

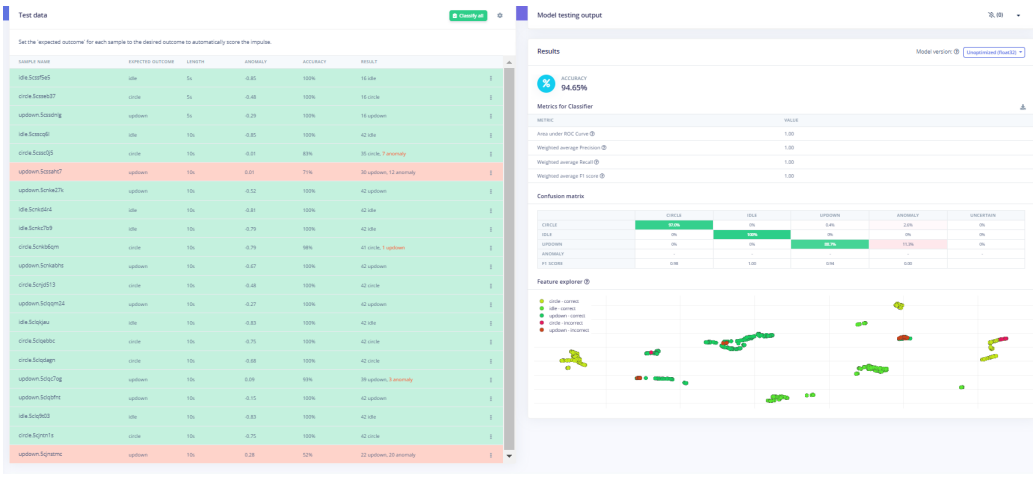


Figura 17: Model testing

Deployment

Finalmente, se llevó a cabo el deployment en la plataforma Edge Impulse, donde se generó una biblioteca compatible con el entorno de desarrollo Arduino IDE. Esto permitió implementar el modelo en un dispositivo Arduino Nano, logrando obtener resultados de clasificación directamente en el dispositivo sin necesidad de conexión a Internet, garantizando así un funcionamiento autónomo y eficiente.

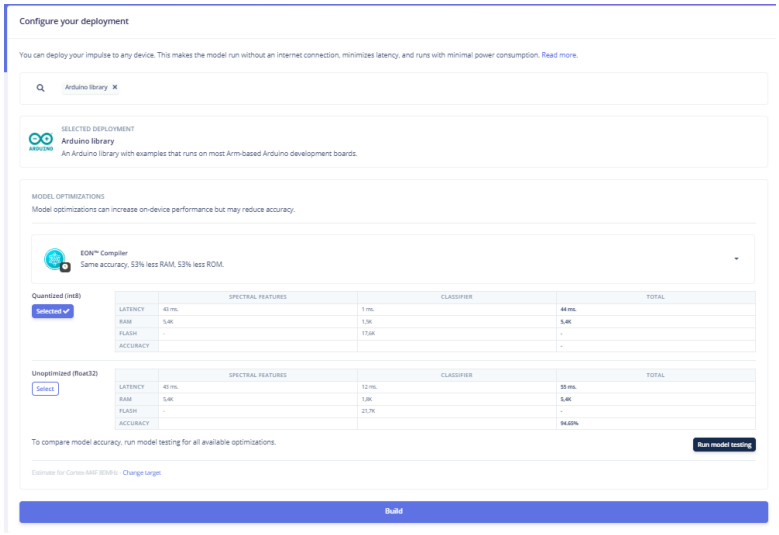


Figura 18: Creación de la biblioteca para Arduino IDE

Resultados

En la Figura 19 se muestran los resultados obtenidos al ejecutar el modelo de clasificación directamente en el microcontrolador Arduino Nano utilizando la biblioteca generada a partir de Edge Impulse. Durante la inferencia en tiempo real, se realizaron múltiples pruebas para clasificar los movimientos circle, idle y updown, obteniendo las siguientes observaciones:

- El modelo clasificó correctamente el movimiento como idle con una confianza del 99.2 %. Aunque se detectó un puntaje de anomalía de -0.622, este valor está dentro de los rangos esperados para el comportamiento normal

- En este caso, el modelo clasificó correctamente el movimiento circle con una confianza del 99.6 %. El puntaje de anomalía, de -0.449, confirma que el modelo interpreta adecuadamente este movimiento.
- Finalmente, el modelo identificó el movimiento updown con una confianza del 99.6 %. El puntaje de anomalía fue positivo (0.242), pero sigue siendo un valor que valida la robustez del sistema.

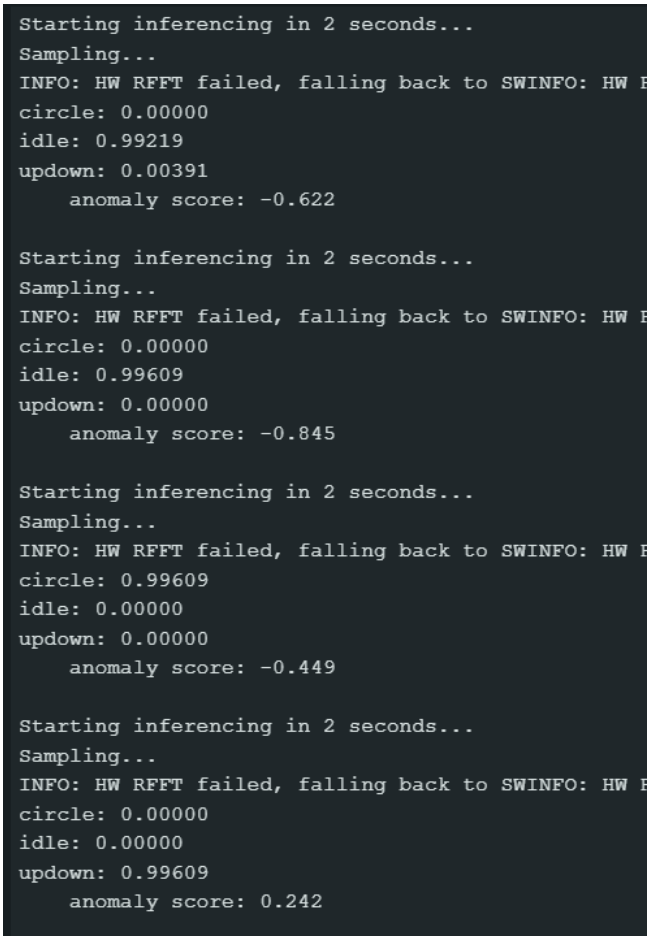


Figura 19: Resultados con Arduino IDE

Conclusiones y recomendaciones

Conclusiones

- El modelo alcanzó altos niveles de precisión en la clasificación de los movimientos *circle*, *idle* y *updown*, con exactitudes superiores al 99 % en cada escenario.
- Los puntajes de anomalía obtenidos indicaron un comportamiento consistente, permitiendo identificar con claridad movimientos anómalos o fuera de los patrones esperados.
- La implementación del modelo directamente en el hardware demostró ser eficiente, eliminando la necesidad de conexión a internet para la inferencia y destacando la adaptabilidad del sistema en entornos autónomos.

Recomendaciones

- **Preparación del entorno de trabajo:** Es fundamental instalar con anticipación todos los programas, controladores y paquetes necesarios para el desarrollo del proyecto, incluyendo **Edge Impulse CLI**, **Arduino IDE**, y cualquier otra dependencia requerida. Esto es especialmente importante debido a la complejidad y las posibles incompatibilidades que pueden surgir.
- **Gestión de datos en Edge Impulse:** Se recomienda planificar cuidadosamente el proceso de adquisición de datos (*sampling*), ya que ocasionalmente **Edge Impulse** puede presentar dificultades al registrar muestras, lo que podría afectar la calidad y cantidad de datos recolectados.

Referencias

[1] Arduino. (2022). *Arduino Nano 33 BLE Sense Product Reference Manual*. SKU: ABX00031.

[2] Nordic Semiconductor. (2019). *nRF52840 Product Specification*. Version 1.1.

[3] STMicroelectronics. (2024). *LSM9DS1 - iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer*.https://www.st.com/en/mems-and-sensors/lsm9ds1.html

[4] STMicroelectronics. (2024). *MP34DT05 - MEMS Audio Sensor Omnidirectional Digital Microphone*. https://www.datasheet-pdf.com/datasheet-pdf/view/929317/STMICROELECTRONICS/MP34DT05.html

[5] Arduino. (2024). *Arducam Camera Module*. https://store-usa.arduino.cc/products/arducam-camera-module?srsltid=AfmBOoodj96T_OyxRV8OikVgn4KPTsDMR_OBRClfQGnVa8ZW9Zvh_h5P

[6] Avago Technologies. (2024). *APDS-9960 - Digital Proximity, Ambient Light, RGB and Gesture Sensor*.https://www.alldatasheet.com/datasheet-pdf/pdf/918047/AVAGO/APDS-9960.html

[7] Nano BLE Sense. (2024). *Nano BLE Sense Digital Proximity, Ambient Light, RGB and Gesture Sensor APDS-9960*.

[8] Edge Impulse (2024), *Arduino Nano 33 BLE Sense - Documentation*. https://docs.edgeimpulse.com/docs/edge-ai-hardware/mcu/arduino-nano-33-ble-sense.

[9] IBM (s.f.), *¿Qué es machine learning (ML)?*. https://www.ibm.com/mx-es/topics/machine-learning.

[10] 330ohms (2024), *¿Qué es Edge Impulse?*. https://www.330ohms.com/blogs/blog/que-es-edge-impulse?srsltid=AfmBOorLj5YIWroPbyPYL0d0jBbZskmB3wZKJQkvdZWus_f-05pUvxIC.

[11] incentro (s.f.), *¿Qué es TensorFlow y para qué sirve?*. https://www.incentro.com/es-ES/blog/que-es-tensorflow.

[12] Arduino Docs (s.f.), *Libraries*. https://docs.arduino.cc/libraries/.

Apéndices

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

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)

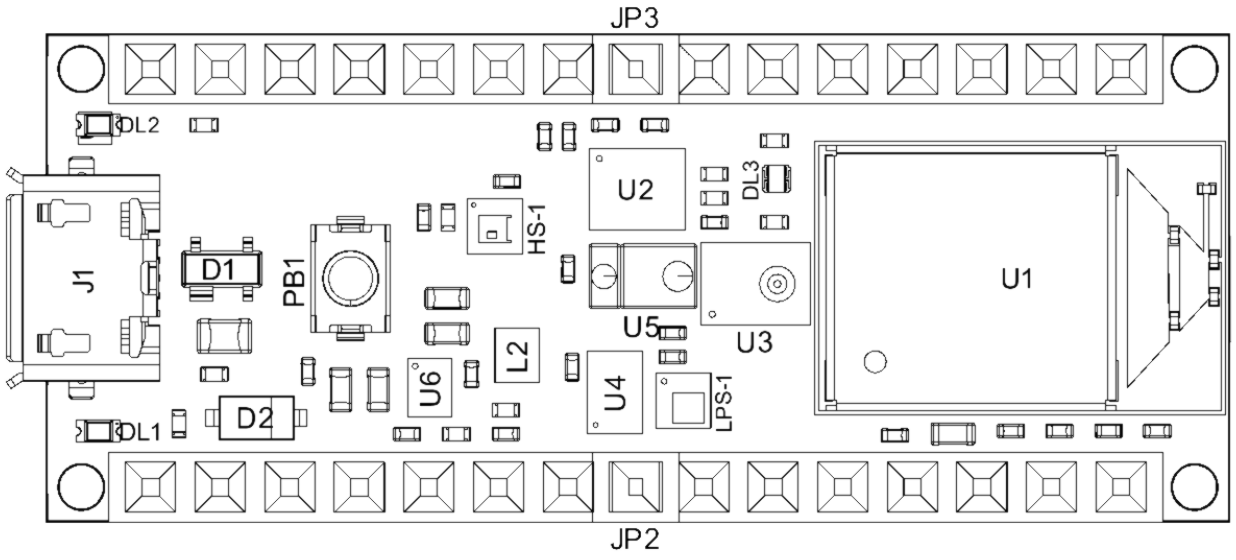
1.2 Power Consumption

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:

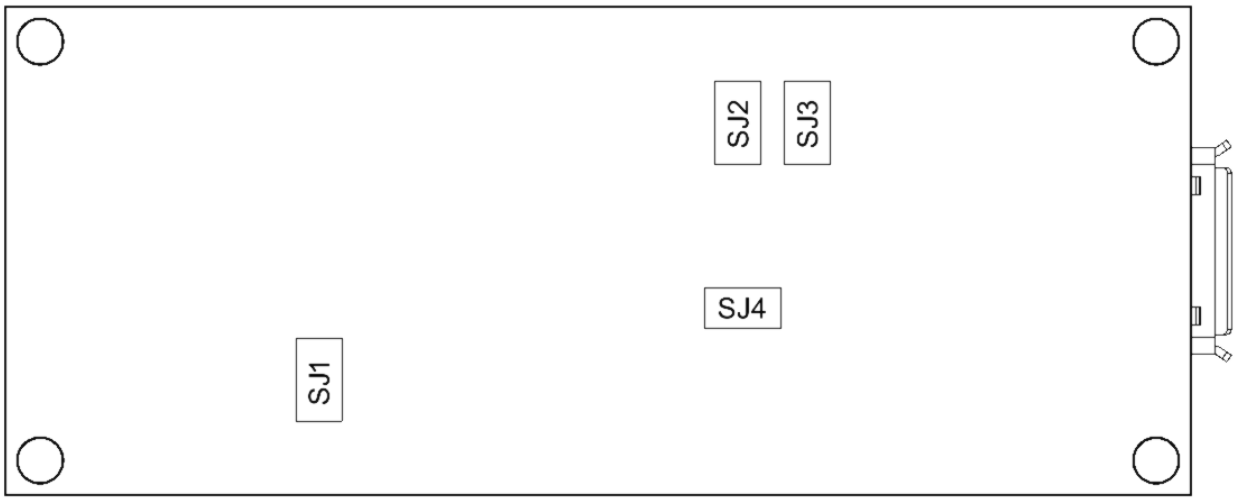


Board topology top

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.

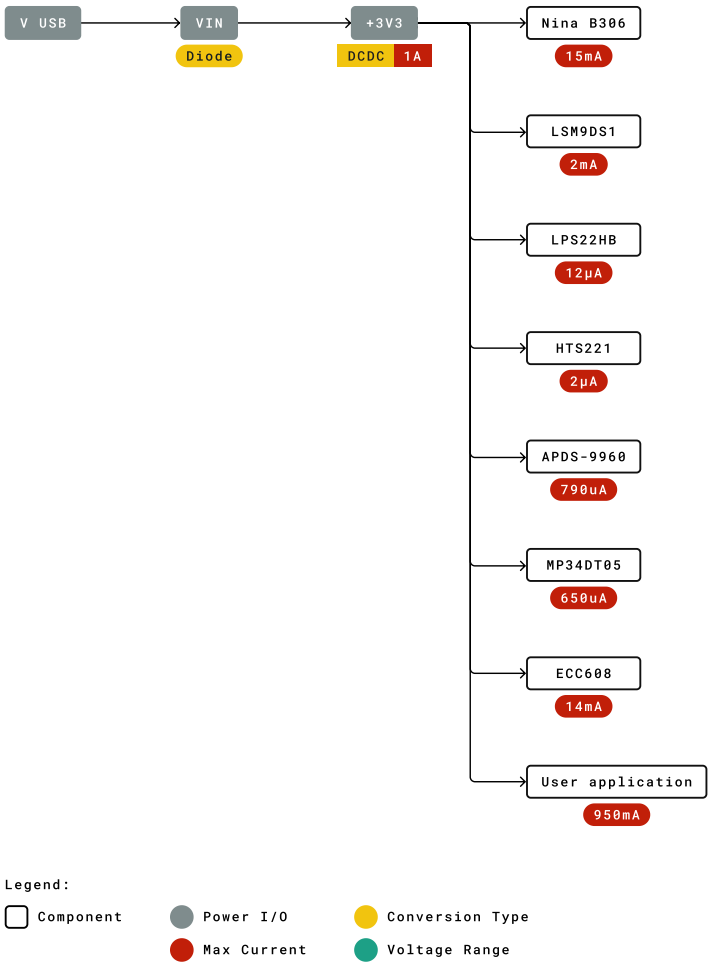
2.8 Digital Microphone

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

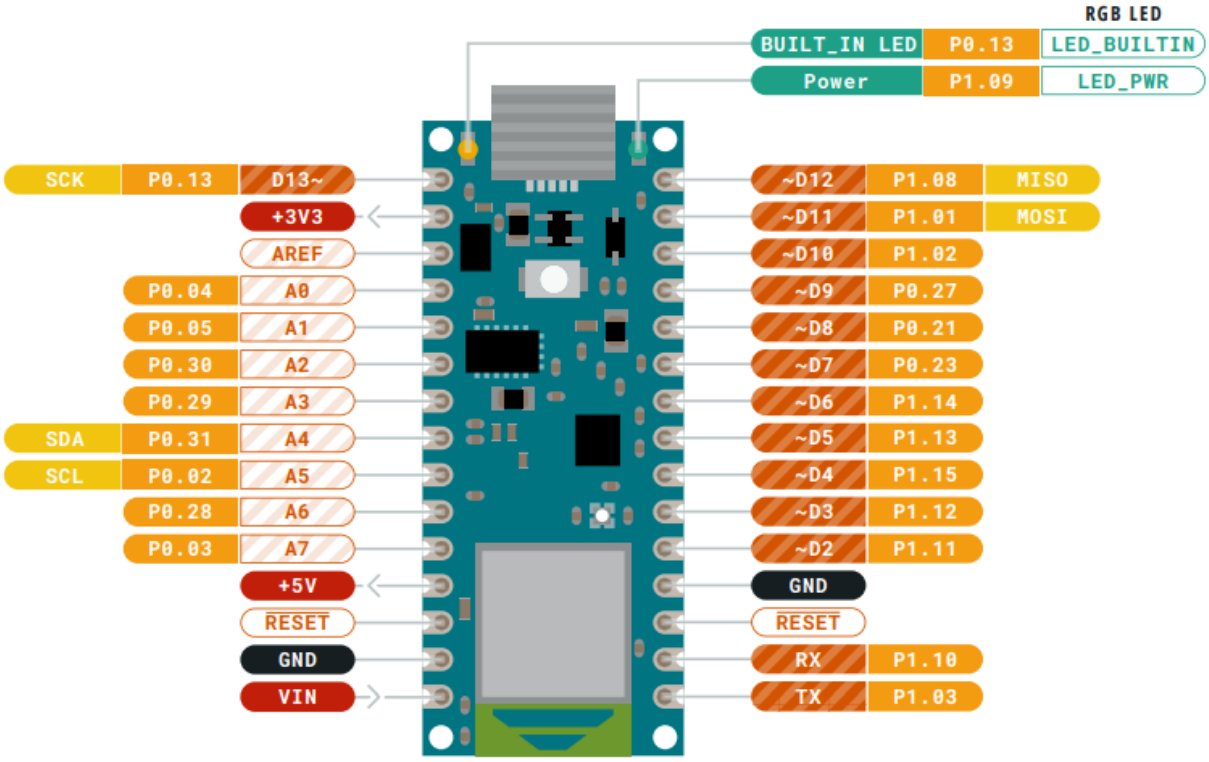
2.9 Power Tree

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.



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

APDS-9960

Digital Proximity, Ambient Light, RGB and Gesture Sensor

AVAGO TECHNOLOGIES

Data Sheet

Lead (Pb) Free
RoHS 6 fully
compliant

Description

The APDS-9960 device features advanced Gesture detection, Proximity detection, Digital Ambient Light Sense (ALS) and Color Sense (RGBC). The slim modular package, L 3.94 × W 2.36 × H 1.35 mm, incorporates an IR LED and factory calibrated LED driver for drop-in compatibility with existing footprints.

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 cancelation, dual 8-bit data converters, power saving inter-conversion delay, 32-dataset FIFO, and interrupt-driven I^2C-bus 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.

Description continued on next page...

Applications

- Gesture Detection
- Color Sense
- Ambient Light Sensing
- Cell Phone Touch Screen Disable
- Mechanical Switch Replacement

Ordering Information

Part Number	Packaging	Quantity
APDS-9960	Tape & Reel	5000 per reel

Features

- Ambient Light and RGB Color Sensing, Proximity Sensing, and Gesture Detection in an Optical Module
- Ambient Light and RGB Color Sensing
  - UV and IR blocking filters
  - Programmable gain and integration time
  - Very high sensitivity – Ideally suited for operation behind dark glass
- Proximity Sensing
  - Trimmed to provide consistent reading
  - Ambient light rejection
  - Offset compensation
  - Programmable driver for IR LED current
  - Saturation indicator bit
- Complex Gesture Sensing
  - Four separate diodes sensitive to different directions
  - Ambient light rejection
  - Offset compensation
  - Programmable driver for IR LED current
  - 32 dataset storage FIFO
  - Interrupt driven I^2C-bus communication
- I^2C-bus Fast Mode Compatible Interface
  - Data Rates up to 400 kHz
  - Dedicated Interrupt Pin
- Small Package L 3.94 × W 2.36 × H 1.35 mm

Description (Cont.)

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.

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.

Functional Block Diagram