UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

IE0624
Laboratorio de Microcontroladores

Laboratorio V

Arduino Nano 33 BLE: GPIO y TinyML

Estudiantes:
Kendall Saborio Picado - B87103
Alexander Rojas Brenes - B86869

Profesor:
Ing. Marco Villalta Fallas

Fecha de entrega: 26/06/2024

Índice

1. INTRODUCCIÓN

Este trabajo se centra en el desarrollo de un clasificador y detector de mascarillas faciales utilizando el kit Arduino Nano 33 BLE con la cámara OV7675, con el objetivo de diseñar y entrenar un modelo que pueda detectar si una persona está usando una mascarilla, ofreciendo una solución eficiente y de bajo costo para reforzar las medidas de salud pública. El desarrollo del proyecto comenzó con la construcción de un conjunto de datos, capturando 20 imágenes de personas con mascarillas, 20 imágenes de personas sin mascarillas y 20 imágenes sin ninguna persona frente a la cámara, variando parámetros como el ángulo de la cabeza y el color de la mascarilla. Posteriormente, se diseñó el impulso en Edge Impulse, utilizando un bloque de preprocesamiento de imagen configurado para trabajar con imágenes en RGB o en escala de grises, generando vectores de características y empleando un bloque de modelo de transferencia configurado con parámetros específicos como el número de ciclos de entrenamiento, la tasa de aprendizaje y el aumento de datos. El modelo se entrenó considerando las características del microcontrolador, como la memoria ROM y RAM, y se validó exhaustivamente con pruebas para afinar su precisión y capacidad de generalización. La implementación en el microcontrolador incluyó una lógica de respuesta utilizando un LED RGB: azul cuando no se detectaba una cara, rojo para una cara sin mascarilla y verde para una cara con mascarilla, proporcionando una respuesta visual clara y efectiva. Los resultados mostraron una precisión del 92.3 % en el entrenamiento y del 80.77 % en la validación, indicando una alta eficacia en la identificación de las clases definidas. Aunque los resultados fueron satisfactorios, se recomendó explorar técnicas adicionales para mejorar el rendimiento del modelo, como diferentes arquitecturas de redes neuronales y el aumento del conjunto de datos, así como implementar medidas para mitigar desafíos como el ruido en los datos o las variaciones ambientales, para asegurar un rendimiento consistente en condiciones del mundo real. Este trabajo demuestra que es posible desarrollar un clasificador eficiente de mascarillas faciales utilizando herramientas accesibles y de bajo costo, ofreciendo una solución potencialmente valiosa para entornos donde la salud pública es una prioridad.

Para consultar el trabajo realizado, puede consultarse el siguiente repositorio de trabajo, en la rama llamada “Lab5”: https://github.com/SaboEIE/IE-0624_IS_2024_B87103.git. El código principal en C (archivo .ino), puede consultarse en la ruta `Laboratorio_05/src`.

2. NOTA TEÓRICA

Antes de iniciar con el uso y manipulación del lenguaje de programación C para la creación de los algoritmos objetivo del presente laboratorio, es necesario tener a mano una serie de conceptos importantes, los cuales se explican a continuación:

2.1. Microcontrolador nRF52840

Tal como se detalla en la hoja del fabricante [1], el nRF52840 es un microcontrolador avanzado basado en un sistema en chip (SoC) de Nordic Semiconductor, diseñado principalmente para aplicaciones de Internet de las Cosas (IoT) que requieren conectividad inalámbrica de baja energía y alta capacidad.

Este microcontrolador incluye una memoria flash con capacidad de 1 MB y una RAM de 256Kb. Asimismo, cuenta con hasta 48 GPIO programables, 5 timers de 32 bits, un convertidor Analógico a Digital de 12 bits, 4 módulos de PWM, High-speed 32 MHz SPI, Sensor de temperatura, Interfaces UART, TWI/I2C, Decodificador de cuadratura para interfaces rotatorias, entre muchas otras características. El nRF52840 es un microcontrolador muy versátil y potente, ideal para aplicaciones que requieren conectividad inalámbrica robusta, de bajo consumo y con capacidades de procesamiento avanzadas.

En la Figura 1 puede observarse la distribución de pines del microcontrolador y sus respectivas funciones:

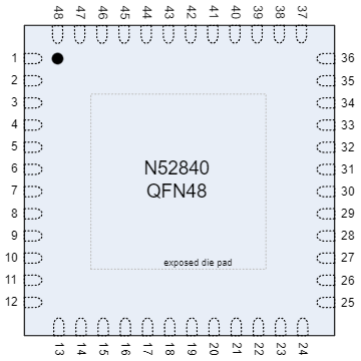


Figura 1: Diagrama de pines. (Fuente: Imagen tomada de [1])

A continuación la siguiente tabla muestra las funciones de cada uno de los pines:

Pin	Name	Function
Left side of the chip		
1	DEC1	Power
2	P0.00 XL1	Digital I/O, Analog input
3	P0.01 XL2	Digital I/O, Analog input
4	P0.04 AIN2	Digital I/O, Analog input
5	P0.03 AIN3	Digital I/O, Analog input
6	P0.07 TRACECLK	Digital I/O, Trace clock
7	P0.08	Digital I/O
8	P1.08	Digital I/O
9	P0.09 TRACEDATA0	Digital I/O, Trace data
10	P0.10 TRACEDATA1	Digital I/O, Trace data
11	P0.11 TRACEDATA2	Digital I/O, Trace data
12	VDD	Power
Bottom side of the chip		
13	P0.13	Digital I/O
14	P0.14	Digital I/O
15	P0.17	Digital I/O
16	P0.18 nRESET	Digital I/O
17	VDD	Power
18	P0.19	Digital I/O
19	P0.20	Digital I/O
20	P0.21	Digital I/O
21	P0.22	Digital I/O
22	P0.23	Digital I/O
23	P0.24	Digital I/O
24	P0.00 TRACEDATA3	Digital I/O, Trace data
Right side of the chip		
25	VDD	Power
26	SWDIO	Debug
27	SWDCLK	Debug
28	NC	-
29	P0.09 NFC1	Digital I/O, NFC input
30	P0.10 NFC2	Digital I/O, NFC input
31	ANT	RF
-		
32	VSS_PA	Power
33	DEC5	Power
34	DEC3	Power

Tabla 1: Pinout of the nRF52840 chip

Pin	Name	Function
35	XC1	Analog input
36	XC2	Analog input
Top side of the chip		
37	VDD	Digital I/O
38	P1.15	Digital I/O
39	P0.03 AIN1	Digital I/O, Analog input
40	P0.02 AIN0	Digital I/O, Analog input
41	P0.28 AIN4	Digital I/O, Analog input
42	P0.29 AIN5	Digital I/O, Analog input
43	P0.30 AIN6	Digital I/O, Analog input
44	P0.31 AIN7	Digital I/O, Analog input
45	VSS	Power
46	DEC4	Power
47	DCC	Power
48	VDD	Power

Tabla 2: Pinout of the nRF52840 chip (Continuation)

Para más información general del dispositivo, observar el Anexo 5.1. Por otro lado, en la Figura 2 puede consultarse el diagrama de bloques del microcontrolador en el cual puede consultarse a alto nivel la conexión interna de este (no incluye el CPU):

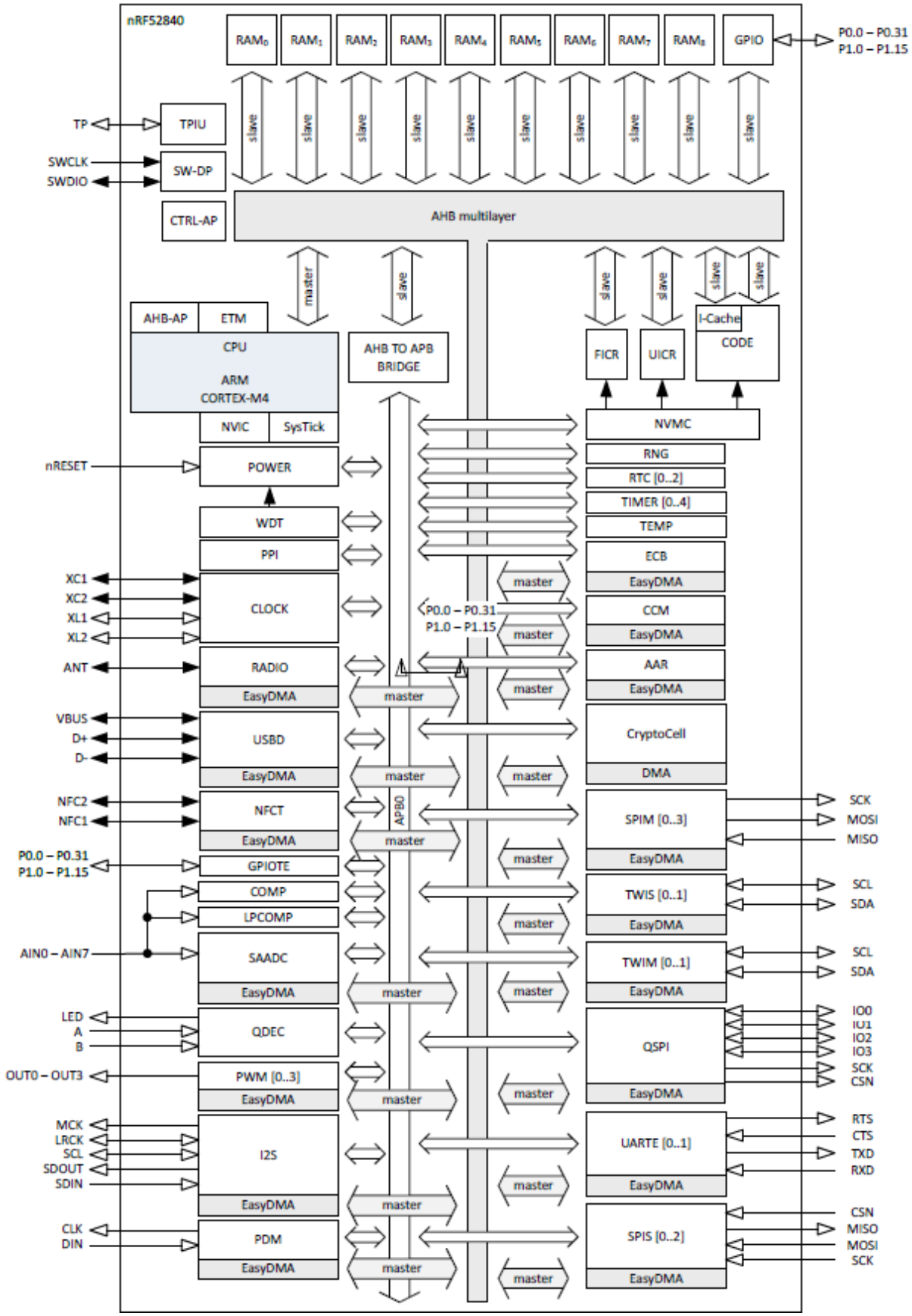


Figura 2: Diagrama de bloques del nRF52840. (Fuente: Imagen tomada de [1])

2.1.1. Registros de importancia

Según la hoja del fabricante [1], el mapa de memoria del nRF52840 está dividido en varias regiones, cada una asignada a diferentes tipos de periféricos y funcionalidades. A continuación, se muestran los rangos de direcciones de registros de importancia:

ID	Base address	Peripheral	Instance	Description
0	0x40000000	CLOCK	CLOCK	Clock control
0	0x40000000	POWER	POWER	Power control
0	0x50000000	GPIO	GPIO	General purpose input and output
0	0x50000000	GPIO	P0	General purpose input and output, port 0
0	0x50000300	GPIO	P1	General purpose input and output, port 1
1	0x40001000	RADIO	RADIO	2.4 GHz radio
1	0x40002000	UART	UART0	Universal asynchronous receiver/transmitter
2	0x40028000	UARTE	UARTE0	Universal asynchronous receiver/transmitter with EasyDMA, unit 0
3	0x40003000	SPI	SPI0	SPI master 0
3	0x40003000	SPIM	SPIM0	SPI master 0
3	0x40003000	SPIS	SPIS0	SPI slave 0
3	0x40003000	TWI	TWI0	Two-wire interface master 0
3	0x40003000	TWIM	TWIM0	Two-wire interface master 0
3	0x40003000	TWIS	TWIS0	Two-wire interface slave 0
4	0x40004000	SPI	SPI1	SPI master 1
4	0x40004000	SPIM	SPIM1	SPI master 1
4	0x40004000	SPIS	SPIS1	SPI slave 1
4	0x40004000	TWI	TWI1	Two-wire interface master 1
4	0x40004000	TWIM	TWIM1	Two-wire interface master 1
4	0x40004000	TWIS	TWIS1	Two-wire interface slave 1
5	0x40005000	NFCT	NFCT	Near field communication tag
6	0x40006000	GPIOTE	GPIOTE	GPIO tasks and events
7	0x40007000	SAADC	SAADC	Analog to digital converter
8	0x40008000	TIMER	TIMER0	Timer 0
9	0x40009000	TIMER	TIMER1	Timer 1
10	0x4000A000	TIMER	TIMER2	Timer 2
11	0x4000B000	RTC	RTC0	Real-time counter 0
12	0x4000C000	TEMP	TEMP	Temperature sensor
13	0x4000D000	RNG	RNG	Random number generator
14	0x4000E000	ECB	ECB	AES electronic code book (ECB) mode block encryption
15	0x4000F000	AAR	AAR	Accelerated address resolver
16	0x4000F000	CCM	CCM	AES counter with CBC-MAC (CCM) mode block encryption
17	0x40010000	WDT	WDT	Watchdog timer
18	0x40011000	RTC	RTC1	Real-time counter 1
19	0x40012000	QDEC	QDEC	Quadrature decoder
19	0x40013000	COMP	COMP	General purpose comparator

Tabla 3: Rangos direcciones de registros.[1]

ID	Base address	Peripheral	Instance	Description
19	0x40013000	LPCOMP	LPCOMP	Low power comparator
20	0x40014000	EGU	EGU0	Event generator unit 0
20	0x40014000	SWI	SWI0	Software interrupt 0
21	0x40015000	EGU	EGU1	Event generator unit 1
21	0x40015000	SWI	SWI1	Software interrupt 1
22	0x40016000	EGU	EGU2	Event generator unit 2
22	0x40016000	SWI	SWI2	Software interrupt 2
23	0x40017000	EGU	EGU3	Event generator unit 3
23	0x40017000	SWI	SWI3	Software interrupt 3
24	0x40018000	EGU	EGU4	Event generator unit 4
24	0x40018000	SWI	SWI4	Software interrupt 4
25	0x40019000	EGU	EGU5	Event generator unit 5
25	0x40019000	SWI	SWI5	Software interrupt 5
26	0x4001A000	TIMER	TIMER3	Timer 3
27	0x4001B000	TIMER	TIMER4	Timer 4
28	0x4001C000	PWM	PWM0	Pulse width modulation unit 0
29	0x4001D000	PDM	PDM	Pulse Density modulation (digital microphone) interface
30	0x4001E000	ACL	ACL	Access control lists
30	0x4001E000	NVMC	NVMC	Non-volatile memory controller
31	0x4001F000	PPI	PPI	Programmable peripheral interconnect
32	0x40020000	MWU	MWU	Memory watch unit
33	0x40021000	PWM	PWM1	Pulse width modulation unit 1
34	0x40022000	PWM	PWM2	Pulse width modulation unit 2
35	0x40023000	SPI	SPI2	SPI master 2
35	0x40023000	SPIM	SPIM2	SPI master 2
35	0x40023000	SPIS	SPIS2	SPI slave 2
36	0x40024000	RTC	RTC2	Real-time counter 2
37	0x40025000	I2S	I2S	Inter-IC sound interface
38	0x40026000	FPU	FPU	FPU interrupt
39	0x40027000	USBD	USBD	Universal serial bus device
40	0x40028000	UARTE	UARTE1	Universal asynchronous receiver/transmitter with EasyDMA, unit 1
41	0x40029000	QSPI	QSPI	External memory interface
42	0x5002A000	CC_HOST_RGF	CC_HOST_RGF	Host platform interface
43	0x5002A000	CRYPTOCELL	CRYPTOCELL	CryptoCell subsystem control interface
44	0x4002D000	PWM	PWM3	Pulse width modulation unit 3
45	0x4002F000	SPIM	SPIM3	SPI master 3

Tabla 4: Rangos direcciones de registros(Continuación).[1]

2.1.2. Características eléctricas

A continuación se muestra las especificaciones eléctricas del microcontrolador utilizado:

	Note	Min.	Max.	Unit
Supply voltages				
VDD		-0.3	+3.9	V
VDDH		-0.3	+5.8	V
VBUS		-0.3	+5.8	V
VSS			0	V
I/O pin voltage				
$V_{I/O}$, VDD ≤ 3.6 V		-0.3	VDD + 0.3	V
$V_{I/O}$, VDD > 3.6 V		-0.3	3.9	V
NFC antenna pin current				
$I_{NFC1/2}$			80	mA
Radio				
RF input level			10	dBm

Figura 3: Características Eléctricas. (Fuente: Imagen tomada de [1])

Symbol	Description	Min.	Typ.	Max.	Units
V_{IH}	Input high voltage	0.7 x VDD		VDD	V
V_{IL}	Input low voltage	VSS		0.3 x VDD	V
$V_{OH,SD}$	Output high voltage, standard drive, 0.5 mA, VDD ≥1.7	VDD - 0.4		VDD	V
$V_{OH,HDH}$	Output high voltage, high drive, 5 mA, VDD >= 2.7 V	VDD - 0.4		VDD	V
$V_{OH,HDL}$	Output high voltage, high drive, 3 mA, VDD >= 1.7 V	VDD - 0.4		VDD	V
$V_{OL,SD}$	Output low voltage, standard drive, 0.5 mA, VDD ≥1.7	VSS		VSS + 0.4	V
$V_{OL,HDH}$	Output low voltage, high drive, 5 mA, VDD >= 2.7 V	VSS		VSS + 0.4	V
$V_{OL,HDL}$	Output low voltage, high drive, 3 mA, VDD >= 1.7 V	VSS		VSS + 0.4	V
$I_{OL,SD}$	Current at VSS+0.4 V, output set low, standard drive, VDD ≥1.7	1	2	4	mA
$I_{OL,HDH}$	Current at VSS+0.4 V, output set low, high drive, VDD >= 2.7 V	6	10	15	mA
$I_{OL,HDL}$	Current at VSS+0.4 V, output set low, high drive, VDD >= 1.7 V	3			mA
$I_{OH,SD}$	Current at VDD-0.4 V, output set high, standard drive, VDD ≥1.7	1	2	4	mA
$I_{OH,HDH}$	Current at VDD-0.4 V, output set high, high drive, VDD >= 2.7 V	6	9	14	mA

Figura 4: Características electricas GPIO. (Fuente: Imagen tomada de [1])

2.2. Arduino Nano 33 BLE sense

Es una placa de desarrollo de la familia Arduino diseñada para proyectos de Internet de las Cosas (IoT), aplicaciones de aprendizaje automático y sistemas embebidos avanzados. Que utiliza el nRF52840 de Nordic Semiconductor, un microcontrolador ARM® Cortex®-M4 a 64 MHz con Bluetooth. A continuación se muestra la topología de la placa:

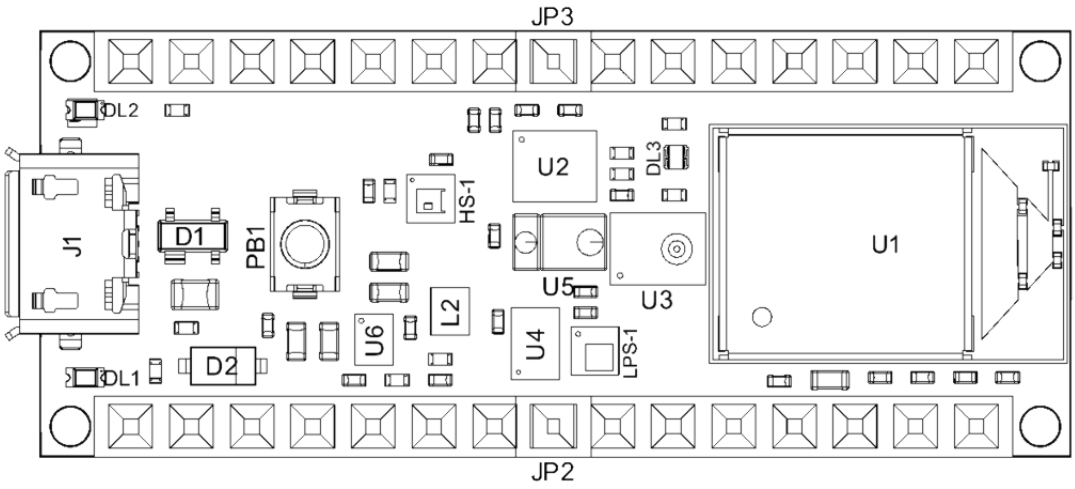


Figura 5: Topología de Nano 33 BLE Sense. (Fuente: Imagen tomada de [2])

A continuación se muestra el diagrama de la placa:

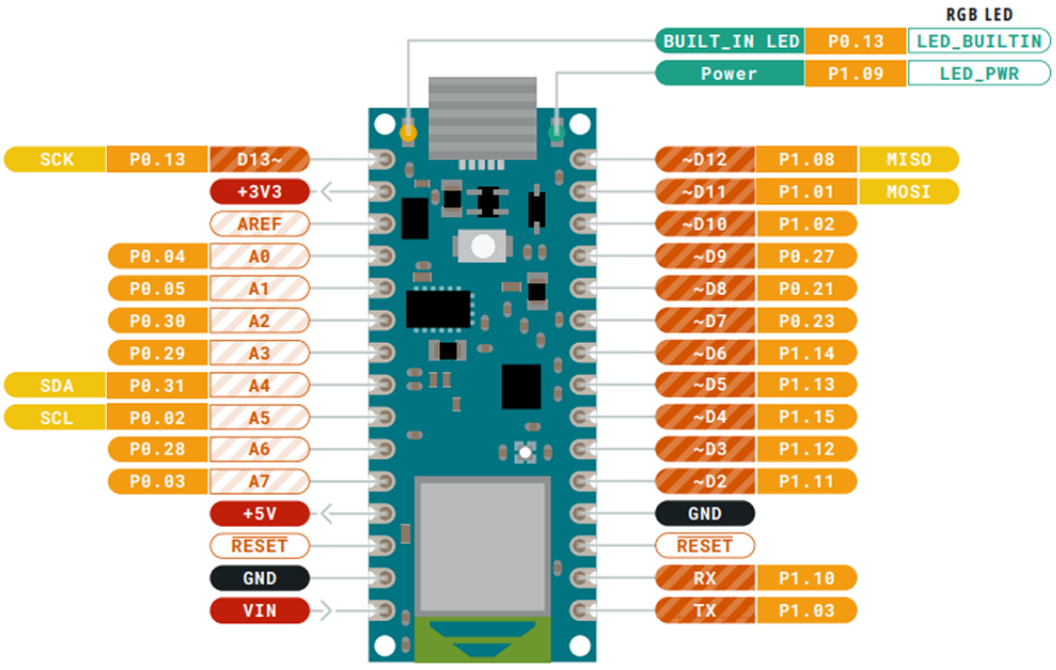


Figura 6: Diagrama pines de Nano 33 BLE Sense. (Fuente: Imagen tomada de [2])

A continuación se muestra una tabla con la descripción de pines de la placa:

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

Tabla 5: Pinout description of Arduino Nano 33 BLE Sense

2.3. Módulo de cámara OV7675

Tal como se describe en [3], la cámara Arducam OV7675 proporciona soporte para una resolución de hasta 640x480 píxeles, lo cual la hace ideal para una variedad de proyectos basados en Arduino, incluyendo sistemas de automatización del hogar y dispositivos inteligentes de monitoreo que necesitan entrada visual precisa. La OV7675 ofrece una interfaz fácil de usar y es capaz de generar salida en formatos RAW, YUV y RGB. Sus especificaciones incluyen un tamaño de píxel de 2.5 μm, una relación señal/ruido de 38 dB y un rango de temperatura operativa amplio, desde

-30℃ hasta 70℃. Con un tamaño compacto de placa de 30.5mm x 30.5mm, esta cámara destaca por su versatilidad y calidad de imagen en diversas condiciones ambientales. En la Figura 7 puede observarse el módulo de cámara utilizado para este proyecto:

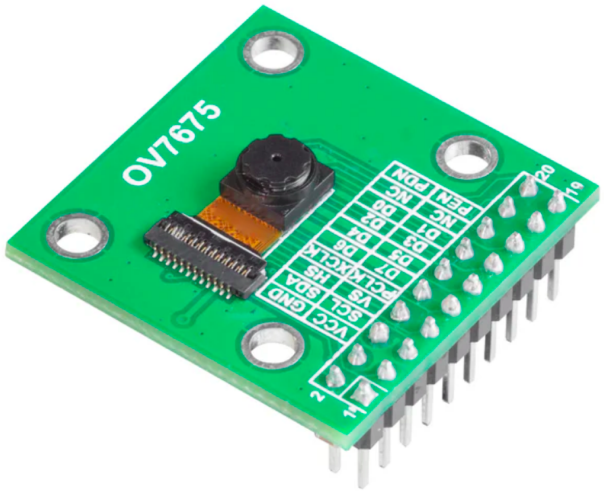


Figura 7: Módulo de cámara OV7675 utilizado en el laboratorio. (Fuente: Imagen tomada de [3]).

2.4. Librerías de importancia

A continuación se detallan las librerías que fueron utilizadas en el laboratorio:

2.4.1. Arduino_OV767X

La librería Arduino_OV767X proporciona una interfaz para utilizar la cámara OV7670/OV7675 con placas Arduino. Esta librería simplifica la captura y procesamiento de imágenes directamente desde la cámara, permitiendo a los desarrolladores incorporar funcionalidades de visión por computadora en sus proyectos de Arduino. Entre sus funcionalidades, se incluyen la configuración de la resolución de imagen, la captura de imágenes en formato RGB y la gestión de buffers de imagen.

2.4.2. stdint.h

La librería stdint.h es una parte estándar de la biblioteca de C y C++ que define tipos de datos enteros con tamaños específicos. Esta librería es fundamental en programación de sistemas embebidos, donde es crucial manejar tipos de datos con tamaños bien definidos para garantizar el uso eficiente de la memoria y el rendimiento del sistema.

2.4.3. stdlib.h

La librería stdlib.h es una parte estándar de la biblioteca de C que proporciona funciones para la gestión de memoria, generación de números aleatorios, conversión de tipos y otros utilitarios generales. Es una librería fundamental en la programación en C y C++, proporcionando herramientas esenciales para el desarrollo de aplicaciones.

2.5. Machine Learning

Machine Learning o Aprendizaje de maquina, es un subcampo de la inteligencia artificial. Este ayuda a los ordenadores a aprender y actuar como seres humanos mediante la ayuda de algoritmos y datos. Dado un conjunto de datos, un algoritmo de aprendizaje maquina aprende diferentes propiedades de los datos e infiera que propiedades de los datos se pueden presentar en el futuro. A diferencia de los sistemas tradicionales que siguen instrucciones preprogramadas, los sistemas de Machine Learning mejoran su rendimiento a medida que se les proporciona más datos [4]. Algunos conceptos clave para desenvolverse en el mundo de *machine learning* son:

- **Datos de entrenamiento:** Conforman los datos que se incorporan al algoritmo de aprendizaje maquina para entrenar el modelo.
- **Datos de prueba:** Constituyen los datos utilizados para validar la presición del modelo.
- **Sobreentrenamiento:** Ocurre cuando un modelo aprende los datos de entrenamiento de manera tan precisa o exacta que pierde la capacidad de generalizar.
- **Subentrenamiento:** Ocurre cuando el modelo no a aprendido lo suficiente la estructura de los datos.
- **Aprendizaje Supervisado:** El modelo se entrena con datos etiquetados, lo que significa que cada dato de entrenamiento está asociado a una etiqueta o resultado deseado.
- **Aprendizaje No Supervisado:** El modelo se entrena con datos que no están etiquetados. El objetivo es encontrar patrones o estructuras en los datos.

2.6. TinyML

Según se describe en [5], TinyML es un campo especializado centrado en desplegar modelos de aprendizaje automático en dispositivos pequeños y con recursos limitados, como microcontroladores y sensores. Esta tecnología es fundamental para habilitar el concepto de *edge computing*, donde los datos pueden procesarse localmente en los dispositivos sin necesidad de enviarlos a un servidor centralizado o a la nube. Al aprovechar algoritmos ligeros y técnicas de optimización de modelos, TinyML asegura que las capacidades de aprendizaje automático puedan operar eficientemente dentro de los recursos computacionales limitados y las restricciones de energía de estos dispositivos.

Una de las principales ventajas de TinyML es su capacidad para reducir la latencia y permitir el procesamiento en tiempo real. Al realizar el análisis de datos localmente, las aplicaciones de TinyML pueden lograr tiempos de respuesta más rápidos en comparación con las soluciones tradicionales basadas en la nube, lo cual es crucial para aplicaciones que requieren decisiones inmediatas. Además, TinyML mejora la eficiencia energética al minimizar la carga computacional en los dispositivos. En cuanto a las aplicaciones, TinyML tiene una amplia gama de usos. Puede ser utilizado en el sector de la salud para monitoreo continuo y diagnóstico y en vehículos autónomos para mejorar las capacidades de navegación y toma de decisiones.

2.7. TensorFlow Lite

TensorFlow Lite es un conjunto de herramientas diseñado para el aprendizaje automático en dispositivos, optimizado para ejecutarse en dispositivos celulares, empotrados y *edge* [6]. Ofrece soporte multiplataforma (Android, iOS, Linux empotrado, microcontroladores) y es compatible con varios lenguajes (Java, Swift, Objective-C, C++, Python). Sus características clave incluyen alta eficiencia en términos de latencia, privacidad (los datos no salen del dispositivo), conectividad (no requiere conexión a internet), tamaño reducido de modelos y consumo eficiente de energía. Facilita la generación de modelos mediante la conversión de modelos existentes o la creación de nuevos, y permite la ejecución de inferencias utilizando APIs adaptadas para modelos con o sin metadatos.

2.8. Edge Impulse

Es una plataforma especializada en el desarrollo y despliegue de modelos de Machine Learning (ML) y de inteligencia artificial (IA) en dispositivos con capacidades limitadas, comúnmente conocidos como dispositivos edge. Esta plataforma facilita la creación, el entrenamiento, la optimización y el despliegue de modelos ML en dispositivos de bajo consumo, tales como microcontroladores y sistemas integrados, sin necesidad de tener una profunda experiencia en ML o IA. Esta plataforma se caracteriza por tener una interfaz amigable con el usuario, que proporciona guías para facilitar el proceso de creación de modelos. Además es compatible con una amplia gama de hardware incluyendo microcontroladores de distintos fabricantes como Arduino, STMicroelectronics, Nordic Semiconductor entre otros. Una característica importante de Edge impulse es que permite el entrenamiento de modelos en la nube, aprovechando recursos computacionales potentes sin necesidad de hardware especializado por parte del usuario [7].

2.9. Configuración y parámetros de la red neuronal en Edge Impulse

Una vez que el proceso de diseño de las partes del impulso fue concluido, debe procederse con el *Transfer Learning*, para el cual deben ser seleccionados un grupo de parámetros. Una breve explicación de los parámetros que pueden ser modificados para ajustar el entrenamiento del modelo pueden observarse a continuación, según se explica en [8]:

- ***Number of training cycles***: corresponde a la cantidad de veces que el algoritmo de entrenamiento realiza un paso completo a través de todos los datos de entrenamiento con retropropagación y actualiza los parámetros del modelo mientras avanza.
- ***Learning rate***: maneja cuánto se actualizan los parámetros internos del modelo durante cada paso del proceso de entrenamiento. También se puede entender como la velocidad a la que la red neuronal aprenderá. Si la red se sobre-ajusta rápidamente, se puede reducir la tasa de aprendizaje.
- ***Data augmentation***: habilita la transformación aleatoria de los datos durante el entrenamiento. Permite ejecutar más ciclos de entrenamiento sin sobre-ajustarlo, lo que puede mejorar la precisión.
- ***Validation set size***: es el porcentaje del conjunto de entrenamiento reservado para validación, siendo un buen valor por defecto el 20 %.

- ***Split train/validation set on metadata key***: permite prevenir la fuga de datos entre conjuntos de entrenamiento y validación agrupando datos mediante metadatos. Se debe dejar vacío para deshabilitar esta opción.
- ***Batch size***: define el tamaño del *batch* (se refiere al n[umero de ejemplos de entrenamiento utilizados en cada iteración). Si no se establece, se utilizará el valor predeterminado. El entrenamiento podría fallar si el tamaño del *batch* es demasiado grande.
- ***Auto-weight classes***: habilita que durante el entrenamiento se preste más atención a las muestras de clases menos representadas. Puede ayudar a que el modelo sea más robusto contra el sobre-ajuste si se dispone de pocos datos para algunas clases.
- ***Profile int8 model***: analizar el modelos cuantizados puede llevar mucho tiempo en conjuntos de datos grandes. Deshabilitar esta opción para omitir el perfilado.

Otra sección de importancia es la de selección de la **arquitectura de la red neuronal**. Tal como se explica en [8], la arquitectura de red neuronal toma como entradas las características extraídas, y pasa las características a cada capa de su arquitectura. En el caso de la clasificación, la última capa utilizada es una capa *softmax* (toma las puntuaciones de salida brutas de la capa anterior y las transforma en probabilidades sumadas da como resultado 1). Es esta última capa la que da la probabilidad de pertenecer a una de las clases predefinidas.

2.10. Lista de componentes y precios

Para el presente laboratorio sólo fue necesario el *Tiny Machine Learning Kit* de Arduino, el cual puede obtener a través de la tienda de Arduino por $60.00 (consultar https://store-usa.arduino.cc/products/arduino-tiny-machine-learning-kit?selectedStore=us). Para realizar este proyecto son necesarios ₡31,403.70, según el valor del dólar al momento de escribir el presente informe. El contenido de este kit puede observarse en la Tabla 6 a continuación:

Tabla 6: Lista de componentes.

Componente	Cantidad
Arduino Nano 33 BLE Sense board	1
OV7675 Camera	1
Arduino Tiny Machine Learning Shield	1
USB A to Micro USB Cable	1

3. DESARROLLO Y ANÁLISIS DE RESULTADOS

A continuación se muestran los métodos de desarrollo utilizados para resolver cada uno de los problemas planteados, así como la funciones utilizadas, el método de operación a partir de diagramas de flujo y los resultados obtenidos:

3.1. Desarrollo de la solución

A continuación podrá consultarse una descripción de alto nivel de cada una de las funciones implementadas en el código .ino generado por Edge Impulse que contiene el modelo entrenado y optimizado. Además se incluye la descripción del código relacionado con el control del RGB, agregado al primero:

3.1.1. Función `OV7675::begin()`

Esta función inicializa la cámara OV7675 con los parámetros especificados de resolución, formato y velocidad de fotogramas. La cámara debe ser configurada correctamente para asegurar que los datos de imagen capturados sean coherentes con los requisitos de la aplicación.

3.1.2. Función `OV7675::readFrame()`

Esta función es responsable de leer un fotograma desde la cámara OV7675 y almacenarlo en el búfer proporcionado como argumento. Esto implica la transferencia de datos desde el sensor de la cámara a la memoria del microcontrolador o procesador.

3.1.3. Función `setup()`

La función setup es parte del ciclo de vida básico de un programa Arduino. En este contexto, configura el entorno inicial del programa, establece la comunicación serial y proporciona información sobre la configuración actual de la inferencia de Edge Impulse. Esto incluye la configuración de pines, inicialización de periféricos y otros ajustes necesarios antes de la ejecución principal del programa.

3.1.4. Función `loop()`

La función loop se ejecuta de forma repetitiva después de setup. En este caso específico, gestiona la captura de imagen, el procesamiento y la inferencia utilizando Edge Impulse. Controla la lógica de inicio y parada de la inferencia basada en la entrada del usuario o condiciones predefinidas.

3.1.5. Función `ei_camera_init()`

Es una función que inicializa la cámara para su uso. Esto podría incluir la configuración de los registros de la cámara para establecer la resolución, el formato de salida y otros parámetros necesarios para capturar imágenes correctamente. La función devuelve true si la inicialización es exitosa y false si hay algún problema.

3.1.6. Función `ei_camera_deinit()`

Es una función complementaria que desinicializa la cámara, liberando recursos utilizados y preparándola para su apagado o reinicialización. Esto puede ser útil para gestionar eficientemente los recursos del sistema cuando la cámara ya no es necesaria.

3.1.7. Función `ei_camera_capture()`

La función es responsable de capturar una imagen utilizando la cámara configurada previamente. Toma dimensiones de imagen deseadas como parámetros (img_width y img_height) y almacena los datos de imagen capturados en el búfer out_buf. Esta función encapsula la lógica necesaria para controlar el proceso de captura y la transferencia de datos desde la cámara al búfer de salida.

3.1.8. Función `ei_camera_cutout_get_data()`

Procesa los datos de imagen capturados para prepararlos para la inferencia. Esto puede incluir la conversión de datos de píxeles crudos en un formato adecuado para el modelo de aprendizaje automático. Los datos procesados se almacenan en el búfer out_ptr, comenzando desde el offset especificado y abarcando la length de datos.

3.1.9. Función `ei_get_serial_available()`

Esta función devuelve el número de bytes disponibles en el puerto serial. Esto es útil para la comunicación serie bidireccional, donde el programa puede verificar si hay datos disponibles antes de intentar leerlos.

3.1.10. Función `ei_get_serial_byte()`

La función lee y devuelve un byte del puerto serial. Facilita la recepción de datos serie del host conectado, lo que puede ser utilizado para recibir comandos o configuraciones adicionales durante la ejecución del programa.

3.1.11. Función `rgb_led_control()`

La función rgb_led_control controla un LED RGB basado en los resultados del modelo de clasificación. La función toma una matriz de valores de resultados (probabilidades) y enciende diferentes combinaciones de los componentes del LED RGB (rojo, verde y azul) según cuál de los valores (MASC,NO MASC, EMPTY) en la matriz es el mayor.

3.2. Proceso de entrenamiento del modelo en Edge Impulse

A continuación se detallará el proceso seguido para la construcción del impulso y el entrenamiento del modelo:

3.2.1. Adquisición de datos

Para realizar los pasos en esta sección se tomó como referencia un tutorial disponible en Edge Impulse llamado `Adding sight to your sensors` disponible en [9]. Se inicia capturando las imágenes y cargándolas en la sección de adquisición de datos. Ahí se puede mover las fotos entre los datos de entrenamiento y los datos de prueba. En este caso luego de distintas pruebas, la distribución 70 % para training y 30 % para testing resultó la más óptima. Otro aspecto importante mencionar es que en esta sección se le asigna el label deseado a cada una de las imagenes del dataset.

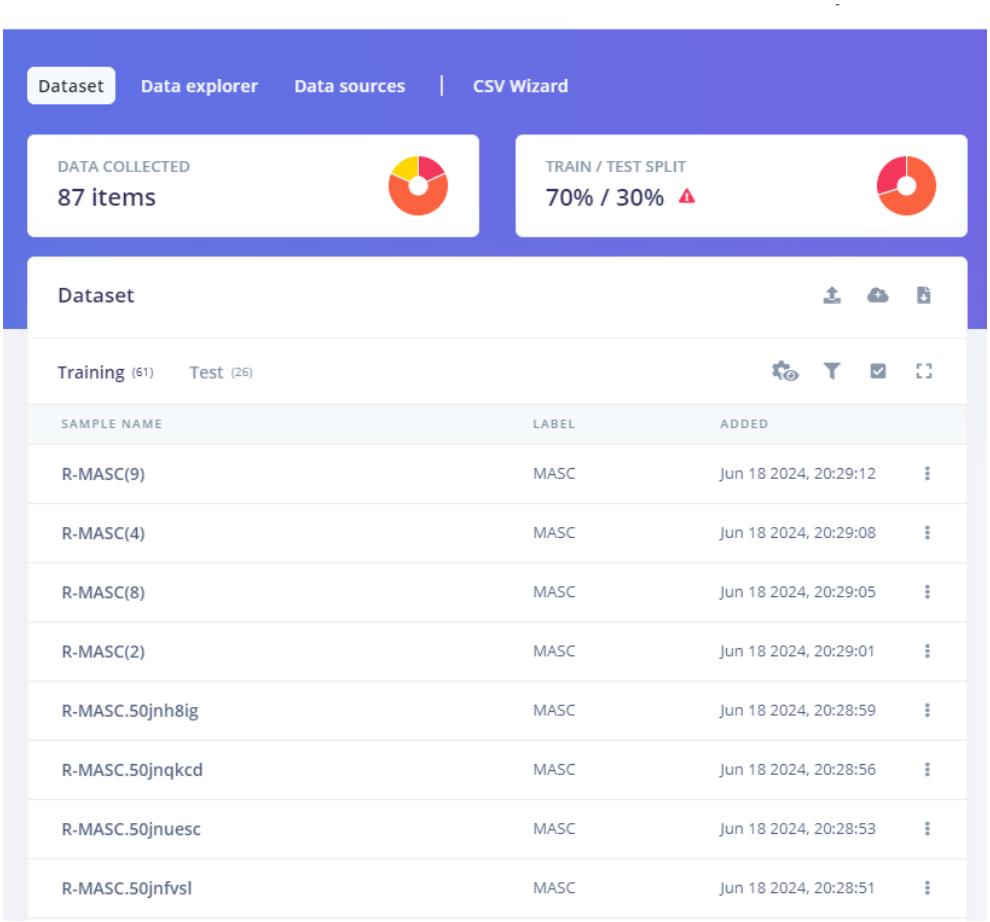


Figura 8: Bloque de adquisición de datos.

3.2.2. Bloque de pre-procesamiento

Seguidamente se entra a la etapa de pre-procesamiento de los datos, en una primera etapa se seleccionan características importantes como el ajuste a las dimensiones de las imágenes con las que se entrenará el modelo. En este caso se seleccionó 68x68 como las dimensiones óptimas tomando en consideración el uso de memoria adecuado para la tarjeta.

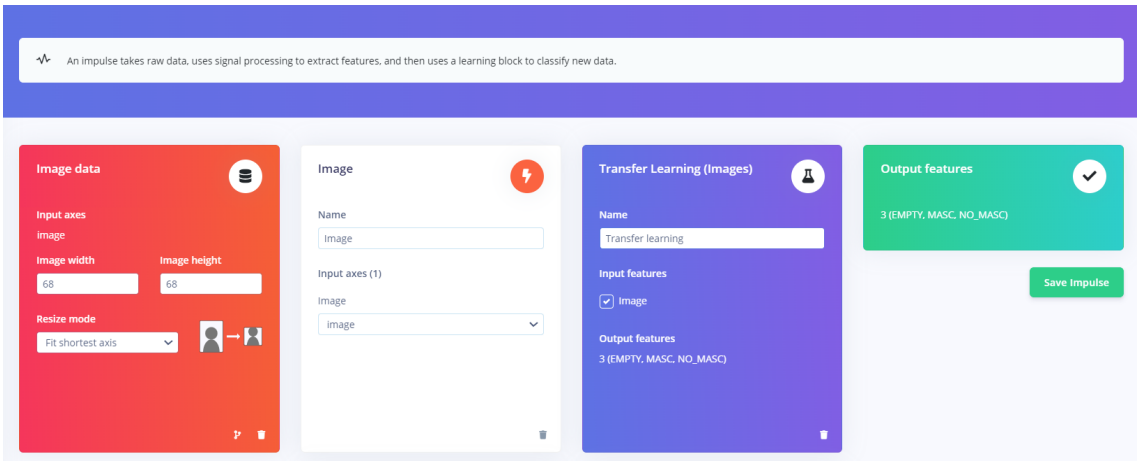


Figura 9: Bloque de adquisición de datos.

Luego en la sección de image se pueden escoger parámetros como el tipo de color de las imágenes (RGB o Gray scale).

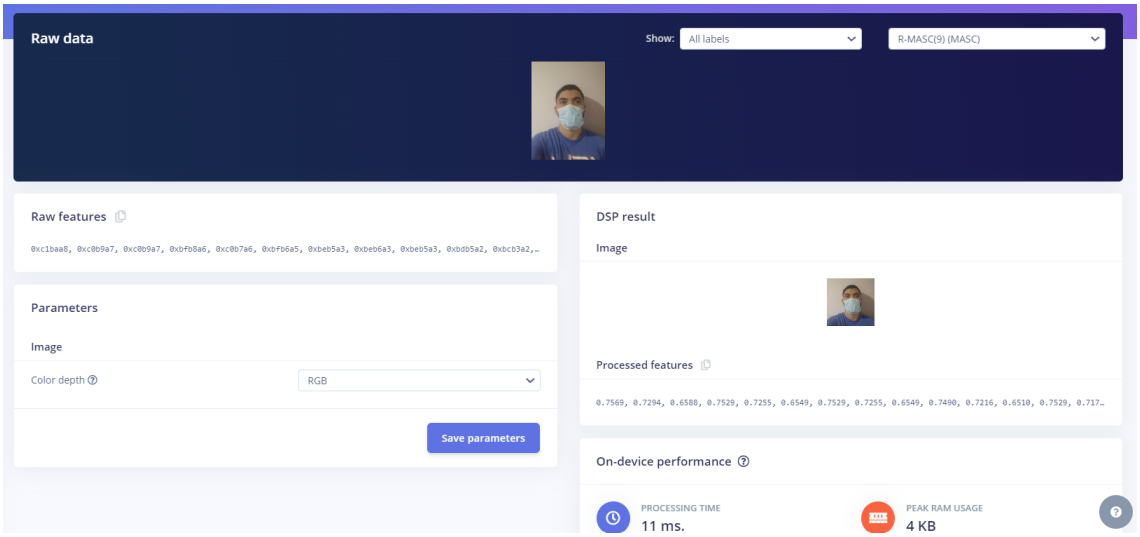


Figura 10: Bloque de adquisición de datos.

3.2.3. Selección de parámetros para el *Transfer learning* del modelo

En la Figura 11, puede consultarse la selección de los parámetros utilizados para el entrenamiento del modelo en Edge Impulse:

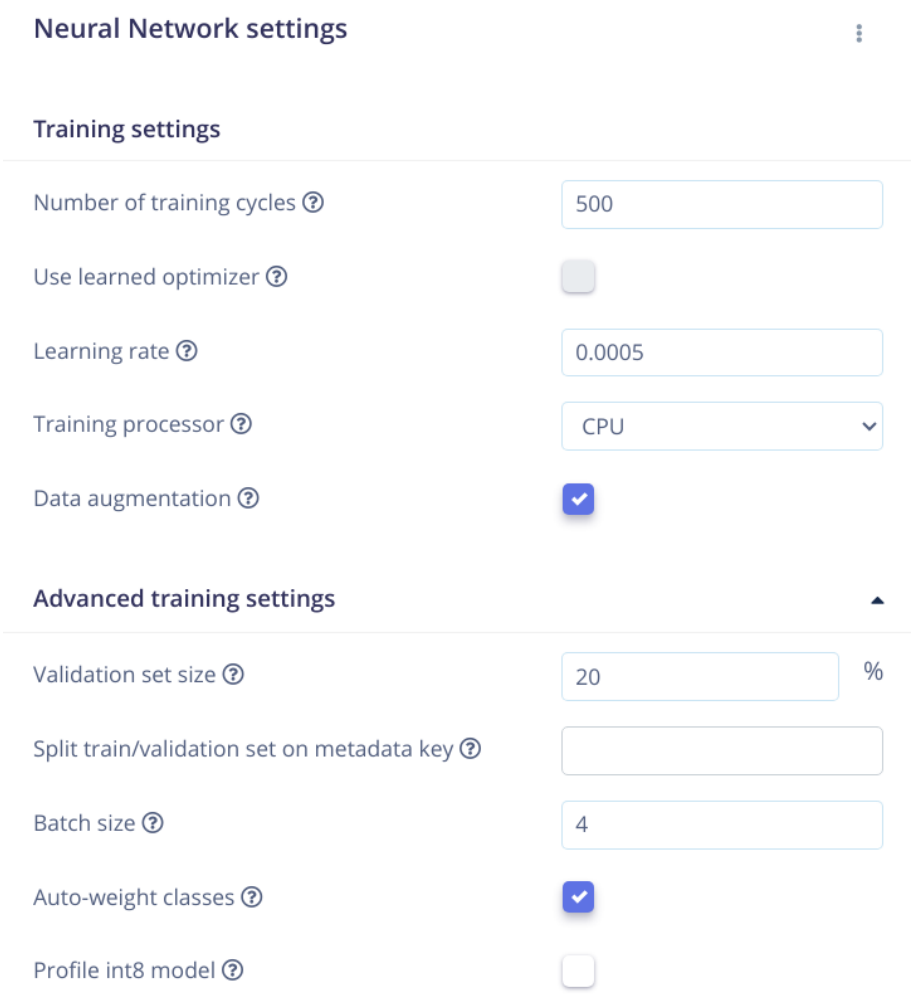


Figura 11: Ajuste de los parámetros utilizados para ajustar el modelo en Edge Impulse.

A continuación se detalla de manera selección el porqué de la selección de este parámetro:

- ***Number of training cycles***: se seleccionó un valor grande (cercano al límite superior permitido por el plan gratuito de la plataforma) para asegurarse de que hubiera un buen margen para obtener una presición alta.
- ***Learning rate***: se tomó un valor muy bajo para evitar un sobre-ajuste muy rápido del modelo.
- ***Data augmentation***: se habilitó, ya que permite ejecutar más ciclos de entrenamiento, aumentando la precisión.
- ***Validation set size***: se toma la recomendación de la plataforma de utlizar un 20 % de conjunto de datos para validación.
- ***Batch size***: se tomó un valor pequeño para permitir permite actualizaciones más frecuentes de los parámetros durante el entrenamiento, lo que ayuda a la convergencia del modelo y evita el sobreajuste de este.

- ***Auto-weight classes***: se habilitó, ya que no de todas las clase se tenía la misma cantidad de datos.

Por otro lado, fue necesario seleccionar la arquitectura de la red neuronal, en la Figura 12:

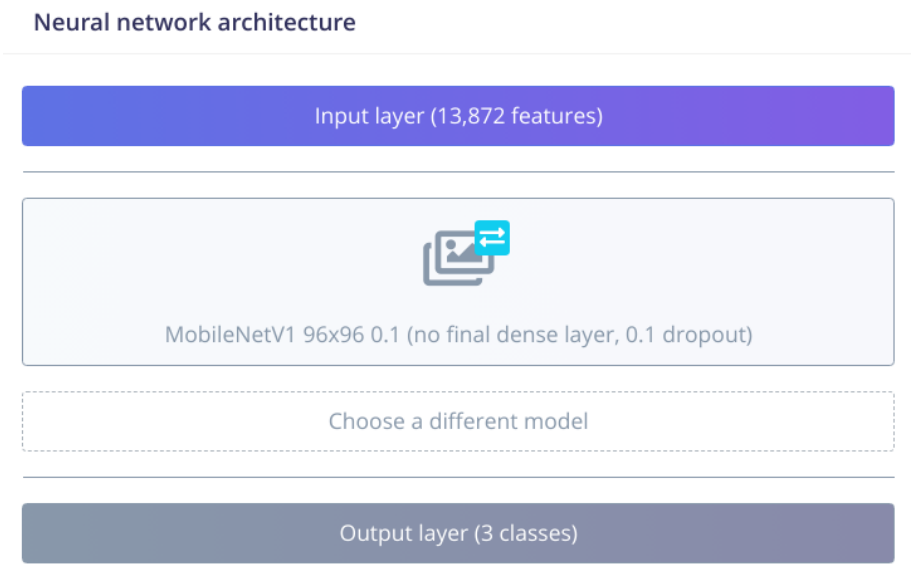


Figura 12: Arquitectura seleccionada para entrenar el modelo en Edge Impulse.

Como puede verse en la Figura 12, la arquitectura seleccionada fue la MobileNetV1 96x96 0.1, la cual utiliza aproximadamente 53.2 KB de RAM y 101 KB de ROM, utilizando las configuraciones por defecto y optimizaciones. Esta trabaja mejor con *inputs* de 96x96 píxeles y soporta tanto la escala de grises como RGB. Se seleccionó esta, ya que es la utilizaba la menor cantidad de RAM y ROM, permitiendo cargar el modelo sin problemas al Arduino Nano 33 BLE.

3.3. Análisis de los resultados

A continuación se mostrarán los resultados para cada una de las funcionalidades solicitadas por el enunciado:

3.4. Rendimiento del entrenamiento del modelo

En la Figura 13, puede observarse el rendimiento del entrenamiento del modelo, utilizando los datos cargados y los parámetros y arquitectura seleccionados:

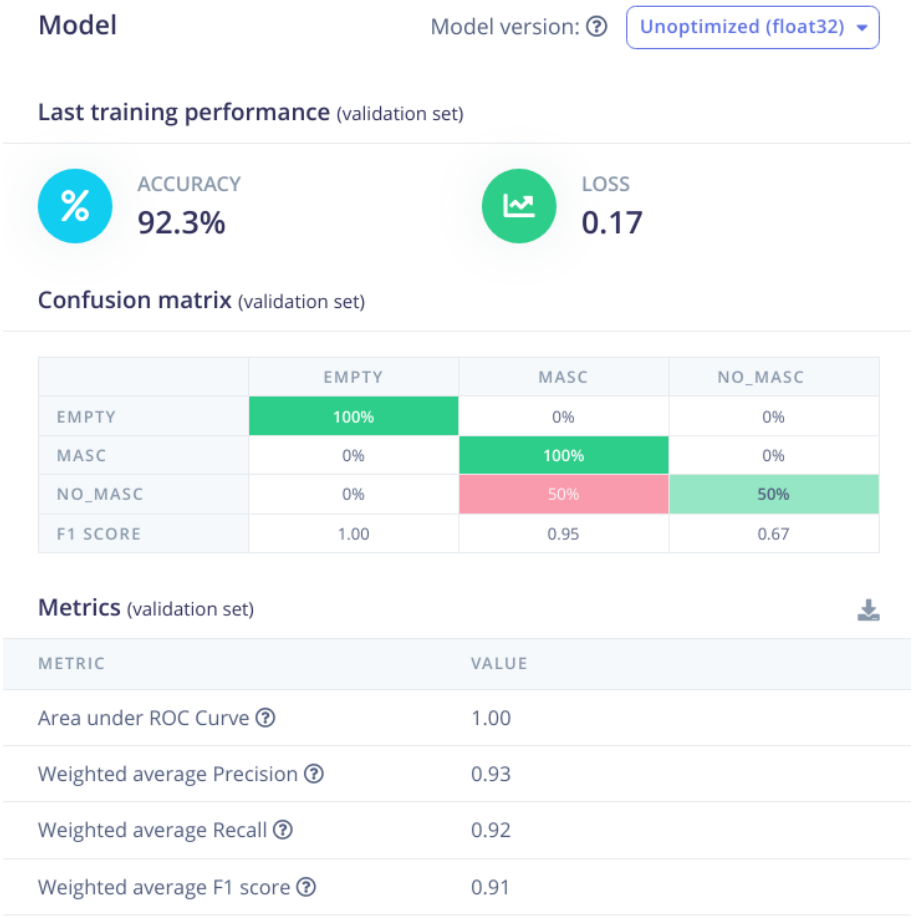


Figura 13: Rendimiento del entrenamiento del modelo tras el entrenamiento bajo las condiciones ya mencionadas.

Como puede verse en la Figura 13, se tiene una precisión bastante alta del 92.3 %, lo que indica que al menos utilizando el porcentaje de validación indicado, el modelo se comporta bastante bien. Además se puede observar que la pérdida del modelo es bastante baja.

3.5. Rendimiento de la comprobación del modelo

En la Figura 14, puede observarse el rendimiento de la comprobación del modelo tras ser entrenado y utilizando para esta los datos apartados para este propósito:

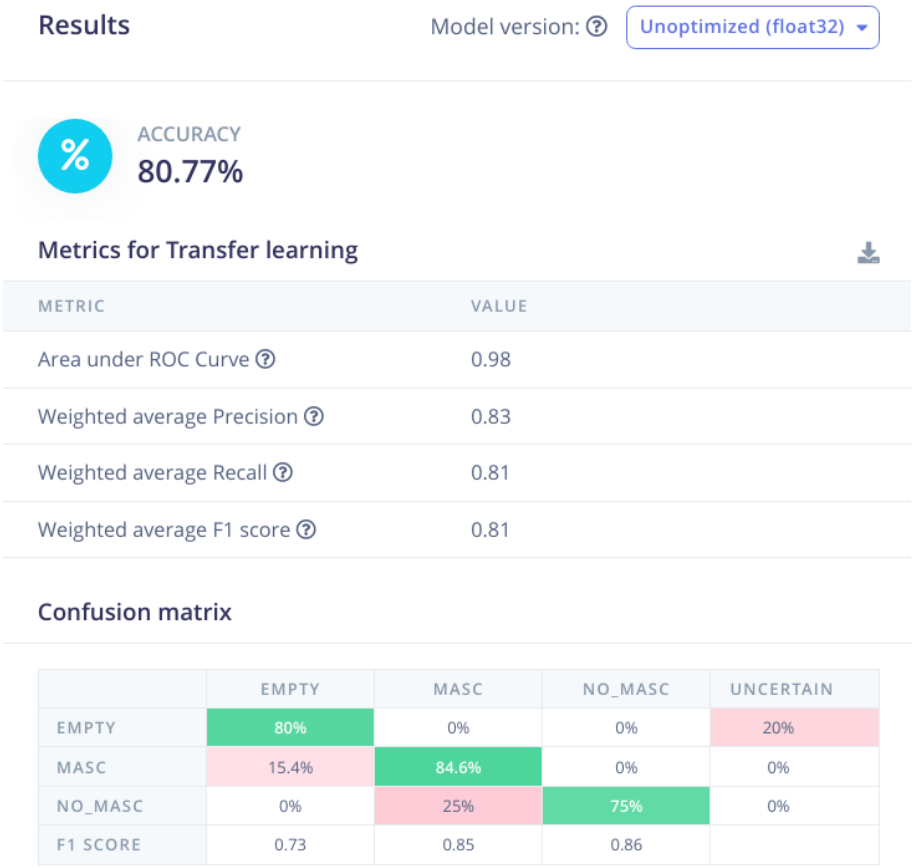


Figura 14: Rendimiento de la comprobación del modelo bajo las condiciones ya mencionadas.

Como puede verse en la Figura 14, la comprobación tiene una precisión bastante alta del 80.77 %. Lo que indica que el modelo se comporta lo suficientemente bien para efectuar una identificación adecuada de un espacio vacío, una persona sin mascarilla y una con mascarilla.

3.6. Despliegue de información en la línea de comandos

Una vez que el modelo es compilado y cargado en el microcontrolador, puede utilizarse la opción “Serial Monitor” del Arduino IDE para observaren en la línea de comandos los datos escritos en el puerto serial sobre el porcentaje de identificación según sea el caso. En la Figura 15 puede observarse un ejemplo de esto:

```
Predictions (DSP: 7 ms., Classification: 2062 ms., Anomaly: 0 ms.):
Predictions:
  EMPTY: 0.97585
  MASC: 0.01654
  NO_MASC: 0.00761

Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 7 ms., Classification: 2062 ms., Anomaly: 0 ms.):
Predictions:
  EMPTY: 0.98232
  MASC: 0.01124
  NO_MASC: 0.00645

Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 7 ms., Classification: 2062 ms., Anomaly: 0 ms.):
Predictions:
  EMPTY: 0.98254
  MASC: 0.01011
  NO_MASC: 0.00734
```

Figura 15: Impresión del porcentaje de identificación según sea el caso en el puerto serial.

Como puede verse en la Figura 15, pueden observase las tres clases definidas para el modelo de identificación (EMPTY, MASC y NO_MASC) y al lado el porcentaje asignado por este para cada caso. En el momento en el que se tomó esta captura, la cámara apuntaba hacia el techo de una habitación, es por esto que el porcentaje identificación de EMPTY es mayor que las otras dos clases.

3.7. Identificación de clases y notificación con el LED RGB

Para demostrar el funcionamiento del modelo de identificación es correcta y que la notificación de esta a través del LED RGB, puede consultarse en el siguiente video: https://youtu.be/gyIAK8Mi2a0. En este pueden observarse tres estados:

1. Al inicio del video puede observarse como mientras la cámara apunta hacia el vacío, el color azul es el que se mantiene en el LED RGB, indicando que la clase identificada es EMPTY.

2. Una vez que la cámara es apuntada a la cara de una persona sin mascarilla, el LED RGB cambia a rojo, indicando que la clase identificada es NO_MASC.

3. Al final del video, puede verse como al apuntarse la cámara a la cada de una persona con mascarilla, el LED RGB cambia a verde, indicando que la clase identificada es MASC.

Estos tres estados confirman la identificación adecuada de las tres clases y especialmente la identificación de una persona con mascarilla y lo notifica cambiando el color del LED RGB, por lo que se cumple de manera exitosa con lo solicitado en el enunciado del laboratorio.

4. CONCLUSIONES Y RECOMENDACIONES

A lo largo del desarrollo del informe se pusieron a prueba una serie de conceptos y prácticas importantes para el manejo básico del Arduino Nano 33 BLE, para la construcción de un algoritmo de reconocimiento de mascarailla utilziando *Machine Learning*. A continuación algunas conclusiones y recomendaciones recolectadas a lo largo de trabajo realizado:

- El modelo de identificación muestra una precisión considerable tanto en el entrenamiento como en la validación. Con una precisión del 92.3 % en el entrenamiento y del 80.77 % en la comprobación, se evidencia que el modelo es efectivo en reconocer las clases definidas (espacio vacío, persona sin mascarilla y persona con mascarilla). Esto también sugiere que el modelo está bien ajustado para su propósito y puede realizar identificaciones precisas bajo condiciones controladas.

- La validación del modelo mediante el cambio de color en un LED RGB proporciona una respuesta visual clara y efectiva sobre la identificación de las clases. Esto no solo confirma la precisión del modelo al identificar las clases esperadas, sino que también demuestra la capacidad de integración con dispositivos físicos para notificaciones en tiempo real.

- A pesar de los buenos resultados obtenidos, es recomendable explorar técnicas adicionales para mejorar aún más el rendimiento del modelo, además de exploración de diferentes arquitecturas de redes neuronales, o incluso el aumento de datos para fortalecer la identificación del modelo frente a diferentes escenarios y condiciones ambientales.

- Se recomienda implementar medidas para mitigar posibles desafíos como ruido en los datos o variaciones ambientales, para garantizar un rendimiento consistente del modelo en situaciones del mundo real.

REFERENCIAS BIBLIOGRÁFICAS

[1] Nordic:Semiconductor, *nRF52840 Product Specification*. 2019. Disponible en: https://docs.nordicsemi.com/bundle/nRF52840_PS_v1.1/resource/nRF52840_PS_v1.1.pdf.

[2] "Arduino nano 33 ble sense datasheet." https://docs.arduino.cc/resources/datasheets/ABX00031-datasheet.pdf.

[3] Arduino, "Arducam 0.3MP OV7675 20-pin DVP Camera Module." https://store-usa.arduino.cc/products/arducam-camera-module?selectedStore=us, s.f.

[4] E. Cuevas-Jiménez, *Introducción al machine learning con MATLAB*. Marcombo, 2021.

[5] Baeldung, "What Is TinyML?." https://www.baeldung.com/cs/tinyml, 2024.

[6] TensorFlow, "TensorFlow Lite." https://www.tensorflow.org/lite/guide, s.f.

[7] Edge Impulse, "What is Edge Impulse and tinyML?." https://www.youtube.com/watch?v=_1Pl6TmJugE, Nov. 2020.

[8] Edge Impulse, "Learning blocks." https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks#neural-network-settings, s.f.

[9] Edge Impulse, "Adding sight to your sensors." https://docs.edgeimpulse.com/docs/tutorials/end-to-end-tutorials/image-classification.

5. Anexos

5.1. Hoja del fabricante del nRF52840: información general

nRF52840

Product Specification

v1.1

NORDIC® SEMICONDUCTOR

Feature list

Features:

- *Bluetooth*® 5, IEEE 802.15.4-2006, 2.4 GHz transceiver
  - -95 dBm sensitivity in 1 Mbps *Bluetooth*® low energy mode
  - -103 dBm sensitivity in 125 kbps *Bluetooth*® low energy mode (long range)
  - -20 to +8 dBm TX power, configurable in 4 dB steps
  - On-air compatible with nRF52, nRF51, nRF24L, and nRF24AP Series
  - Supported data rates:
    - *Bluetooth*® 5: 2 Mbps, 1 Mbps, 500 kbps, and 125 kbps
    - IEEE 802.15.4-2006: 250 kbps
    - Proprietary 2.4 GHz: 2 Mbps, 1 Mbps
  - Single-ended antenna output (on-chip balun)
  - 128-bit AES/ECB/CCM/AAR co-processor (on-the-fly packet encryption)
  - 4.8 mA peak current in TX (0 dBm)
  - 4.6 mA peak current in RX
  - RSSI (1 dB resolution)
- ARM® Cortex®-M4 32-bit processor with FPU, 64 MHz
  - 212 EEMBC CoreMark score running from flash memory
  - 52 µA/MHz running CoreMark from flash memory
  - Watchpoint and trace debug modules (DWT, ETM, and ITM)
  - Serial wire debug (SWD)
- Rich set of security features
  - ARM® TrustZone® Cryptocell 310 security subsystem
    - NIST SP800-90A and SP800-90B compliant random number generator
    - AES-128: ECB, CBC, CMAC/CBC-MAC, CTR, CCM/CCM*
    - Chacha20/Poly1305 AEAD supporting 128- and 256-bit key size
    - SHA-1, SHA-2 up to 256 bits
    - Keyed-hash message authentication code (HMAC)
    - RSA up to 2048-bit key size
    - SRP up to 3072-bit key size
    - ECC support for most used curves, among others P-256 (secp256r1) and Ed25519/Curve25519
    - Application key management using derived key model
  - Secure boot ready
    - Flash access control list (ACL)
    - Root-of-trust (RoT)
    - Debug control and configuration
    - Access port protection (CTRL-AP)
  - Secure erase
- Flexible power management
  - 1.7 V to 5.5 V supply voltage range
  - On-chip DC/DC and LDO regulators with automated low current modes
  - 1.8 V to 3.3 V regulated supply for external components
  - Automated peripheral power management
  - Fast wake-up using 64 MHz internal oscillator
  - 0.4 µA at 3 V in System OFF mode, no RAM retention
  - 1.5 µA at 3 V in System ON mode, no RAM retention, wake on RTC
- 1 MB flash and 256 kB RAM
- Advanced on-chip interfaces
  - USB 2.0 full speed (12 Mbps) controller
  - QSPI 32 MHz interface
  - High-speed 32 MHz SPI
  - Type 2 near field communication (NFC-A) tag with wake-on field
    - Touch-to-pair support
  - Programmable peripheral interconnect (PPI)
  - 48 general purpose I/O pins
  - EasyDMA automated data transfer between memory and peripherals
- Nordic SoftDevice ready with support for concurrent multi-protocol
- 12-bit, 200 ksps ADC - 8 configurable channels with programmable gain
- 64 level comparator
- 15 level low-power comparator with wake-up from System OFF mode
- Temperature sensor
- 4x 4-channel pulse width modulator (PWM) unit with EasyDMA
- Audio peripherals: I2S, digital microphone interface (PDM)
- 5x 32-bit timer with counter mode
- Up to 4x SPI master/3x SPI slave with EasyDMA
- Up to 2x I2C compatible 2-wire master/slave
- 2x UART (CTS/RTS) with EasyDMA
- Quadrature decoder (QDEC)
- 3x real-time counter (RTC)
- Single crystal operation
- Package variants
  - aQFN™ 73 package, 7 x 7 mm
  - WLCSP93 package, 3.544 x 3.607 mm

Applications:

- Advanced computer peripherals and I/O devices
  - Mouse
  - Keyboard
  - Multi-touch trackpad
- Advanced wearables
  - Health/fitness sensor and monitor devices
  - Wireless payment enabled devices
- Internet of things (IoT)
  - Smart home sensors and controllers
  - Industrial IoT sensors and controllers
- Interactive entertainment devices
  - Remote controls
  - Gaming controllers

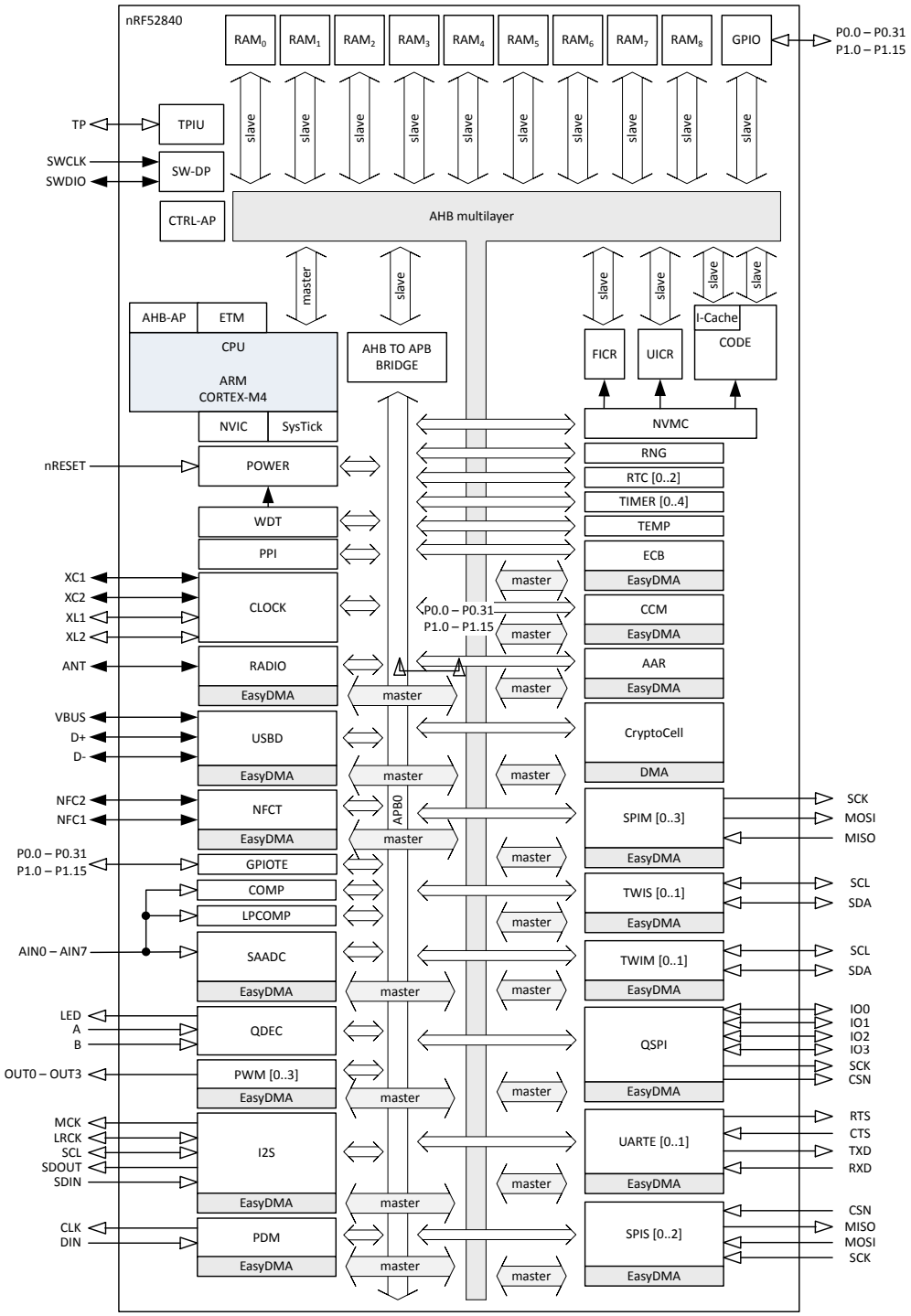


Figure 1: Block diagram

4.2.4 Instantiation

ID	Base address	Peripheral	Instance	Description	
0	0x40000000	CLOCK	CLOCK	Clock control	
0	0x40000000	POWER	POWER	Power control	
0	0x50000000	GPIO	GPIO	General purpose input and output	Deprecated
0	0x50000000	GPIO	P0	General purpose input and output, port 0	
0	0x50000300	GPIO	P1	General purpose input and output, port 1	
1	0x40001000	RADIO	RADIO	2.4 GHz radio	
2	0x40002000	UART	UART0	Universal asynchronous receiver/transmitter	Deprecated
2	0x40002000	UARTE	UARTE0	Universal asynchronous receiver/transmitter with EasyDMA, unit 0	
3	0x40003000	SPI	SPI0	SPI master 0	Deprecated
3	0x40003000	SPIM	SPIM0	SPI master 0	
3	0x40003000	SPIS	SPIS0	SPI slave 0	
3	0x40003000	TWI	TWI0	Two-wire interface master 0	Deprecated
3	0x40003000	TWIM	TWIM0	Two-wire interface master 0	
3	0x40003000	TWIS	TWIS0	Two-wire interface slave 0	
4	0x40004000	SPI	SPI1	SPI master 1	Deprecated
4	0x40004000	SPIM	SPIM1	SPI master 1	
4	0x40004000	SPIS	SPIS1	SPI slave 1	
4	0x40004000	TWI	TWI1	Two-wire interface master 1	Deprecated
4	0x40004000	TWIM	TWIM1	Two-wire interface master 1	
4	0x40004000	TWIS	TWIS1	Two-wire interface slave 1	
5	0x40005000	NFCT	NFCT	Near field communication tag	
6	0x40006000	GPIOTE	GPIOTE	GPIO tasks and events	
7	0x40007000	SAADC	SAADC	Analog to digital converter	
8	0x40008000	TIMER	TIMER0	Timer 0	
9	0x40009000	TIMER	TIMER1	Timer 1	
10	0x4000A000	TIMER	TIMER2	Timer 2	
11	0x4000B000	RTC	RTC0	Real-time counter 0	
12	0x4000C000	TEMP	TEMP	Temperature sensor	
13	0x4000D000	RNG	RNG	Random number generator	
14	0x4000E000	ECB	ECB	AES electronic code book (ECB) mode block encryption	
15	0x4000F000	AAR	AAR	Accelerated address resolver	
15	0x4000F000	CCM	CCM	AES counter with CBC-MAC (CCM) mode block encryption	
16	0x40010000	WDT	WDT	Watchdog timer	
17	0x40011000	RTC	RTC1	Real-time counter 1	
18	0x40012000	QDEC	QDEC	Quadrature decoder	
19	0x40013000	COMP	COMP	General purpose comparator	
19	0x40013000	LPCOMP	LPCOMP	Low power comparator	
20	0x40014000	EGU	EGU0	Event generator unit 0	
20	0x40014000	SWI	SWI0	Software interrupt 0	
21	0x40015000	EGU	EGU1	Event generator unit 1	
21	0x40015000	SWI	SWI1	Software interrupt 1	
22	0x40016000	EGU	EGU2	Event generator unit 2	
22	0x40016000	SWI	SWI2	Software interrupt 2	
23	0x40017000	EGU	EGU3	Event generator unit 3	
23	0x40017000	SWI	SWI3	Software interrupt 3	
24	0x40018000	EGU	EGU4	Event generator unit 4	
24	0x40018000	SWI	SWI4	Software interrupt 4	
25	0x40019000	EGU	EGU5	Event generator unit 5	
25	0x40019000	SWI	SWI5	Software interrupt 5	

ID	Base address	Peripheral	Instance	Description	
26	0x4001A000	TIMER	TIMER3	Timer 3	
27	0x4001B000	TIMER	TIMER4	Timer 4	
28	0x4001C000	PWM	PWM0	Pulse width modulation unit 0	
29	0x4001D000	PDM	PDM	Pulse Density modulation (digital microphone) interface	
30	0x4001E000	ACL	ACL	Access control lists	
30	0x4001E000	NVMC	NVMC	Non-volatile memory controller	
31	0x4001F000	PPI	PPI	Programmable peripheral interconnect	
32	0x40020000	MWU	MWU	Memory watch unit	
33	0x40021000	PWM	PWM1	Pulse width modulation unit 1	
34	0x40022000	PWM	PWM2	Pulse width modulation unit 2	
35	0x40023000	SPI	SPI2	SPI master 2	Deprecated
35	0x40023000	SPIM	SPIM2	SPI master 2	
35	0x40023000	SPIS	SPIS2	SPI slave 2	
36	0x40024000	RTC	RTC2	Real-time counter 2	
37	0x40025000	I2S	I2S	Inter-IC sound interface	
38	0x40026000	FPU	FPU	FPU interrupt	
39	0x40027000	USBD	USBD	Universal serial bus device	
40	0x40028000	UARTE	UARTE1	Universal asynchronous receiver/transmitter with EasyDMA, unit 1	
41	0x40029000	QSPI	QSPI	External memory interface	
42	0x5002A000	CC_HOST_RGF	CC_HOST_RGF	Host platform interface	
42	0x5002A000	CRYPTOCELL	CRYPTOCELL	CryptoCell subsystem control interface	
45	0x4002D000	PWM	PWM3	Pulse width modulation unit 3	
47	0x4002F000	SPIM	SPIM3	SPI master 3	
N/A	0x10000000	FICR	FICR	Factory information configuration	
N/A	0x10001000	UICR	UICR	User information configuration	

Table 3: Instantiation table

4.3 NVMC — Non-volatile memory controller

The non-volatile memory controller (NVMC) is used for writing and erasing of the internal flash memory and the UICR (user information configuration registers).

The CONFIG on page 27 is used to enable the NVMC for writing (CONFIG.WEN = Wen) and erasing (CONFIG.WEN = Een). The user must make sure that writing and erasing are not enabled at the same time. Having both enabled at the same time may result in unpredictable behavior.

The CPU must be halted before initiating a NVMC operation from the debug system.

4.3.1 Writing to flash

When write is enabled, full 32-bit words can be written to word-aligned addresses in the flash.

As illustrated in Memory on page 20, the flash is divided into multiple pages. The same 32-bit word in the flash can only be written n_{WRITE} number of times before a page erase must be performed.

The NVMC is only able to write `0` to bits in the flash that are erased (set to `1`). It cannot rewrite a bit back to `1`. Only full 32-bit words can be written to flash using the NVMC interface. To write less than 32 bits, write the data as a full 32-bit word and set all the bits that should remain unchanged in the word to `1`. Note that the restriction on the number of writes (n_{WRITE}) still applies in this case.

Only word-aligned writes are allowed. Byte or half-word-aligned writes will result in a hard fault.

The time it takes to write a word to flash is specified by t_{WRITE}. The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the flash.

9 Absolute maximum ratings

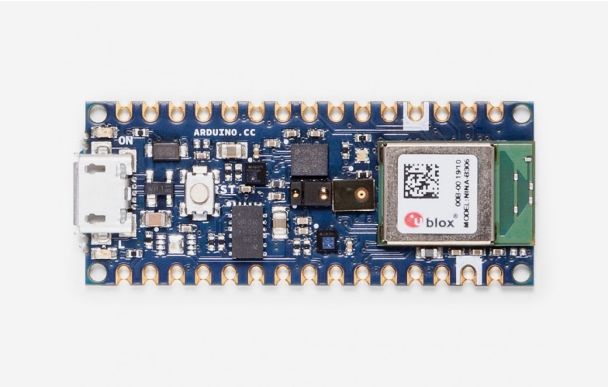
Maximum ratings are the extreme limits to which the chip can be exposed for a limited amount of time without permanently damaging it. Exposure to absolute maximum ratings for prolonged periods of time may affect the reliability of the device.

	Note	Min.	Max.	Unit
Supply voltages				
VDD		-0.3	+3.9	V
VDDH		-0.3	+5.8	V
VBUS		-0.3	+5.8	V
VSS			0	V
I/O pin voltage				
$V_{I/O}$, VDD ≤ 3.6 V		-0.3	VDD + 0.3	V
$V_{I/O}$, VDD > 3.6 V		-0.3	3.9	V
NFC antenna pin current				
$I_{NFC1/2}$			80	mA
Radio				
RF input level			10	dBm
Environmental aQFN™ package				
Storage temperature		-40	+125	°C
MSL	Moisture Sensitivity Level		2	
ESD HBM	Human Body Model		2	kV
ESD HBM Class	Human Body Model Class		2	
ESD CDM	Charged Device Model		750	V
Environmental WLCSP 3.544 x 3.607 mm package				
Storage temperature		-40	+125	°C
MSL	Moisture Sensitivity Level		1	
ESD HBM	Human Body Model		1	kV
ESD HBM Class	Human Body Model Class		1C	
ESD CDM	Charged Device Model		500	V
Flash memory				
Endurance		10 000		Write/erase cycles
Retention		10 years at 40°C		

Table 173: Absolute maximum ratings

5.2. Hoja del fabricante del Arduino Nano 33 BLE: información general

Product Reference Manual
SKU: ABX00031



Description

Arduino® Nano 33 BLE Sense is a miniature sized module containing a NINA B306 module, based on Nordic nRF52480 and containing an Arm® 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

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: 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)

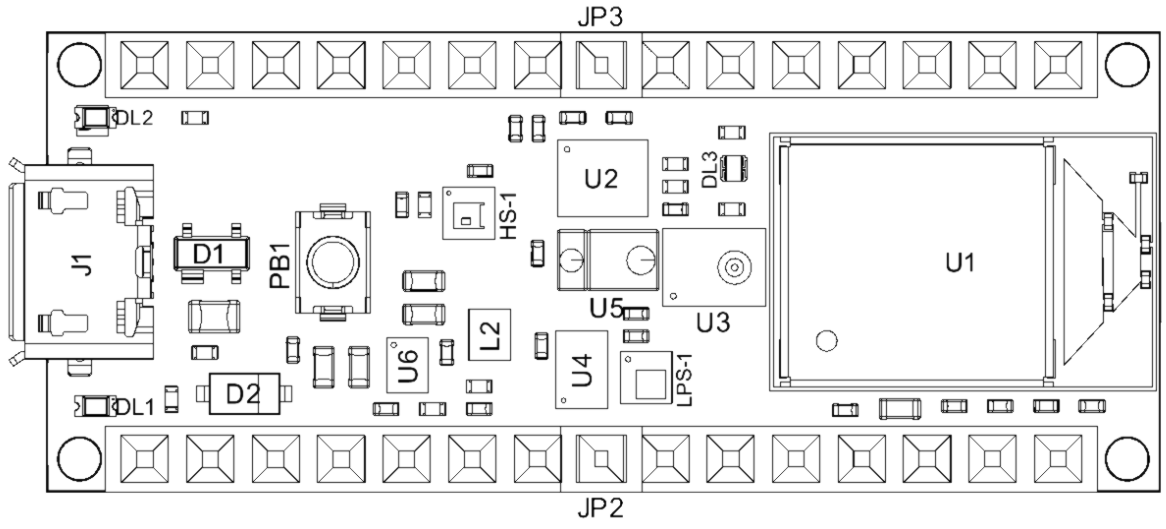
1.2 Power Consumption

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:

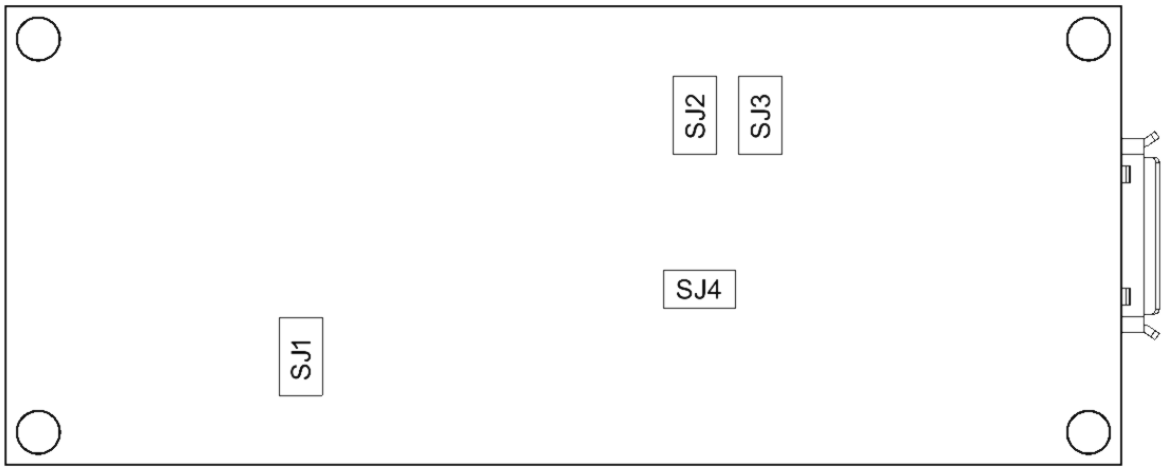


Board topology top

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 Arm® Cortex®-M4F running at up to 64 MHz. 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]**.

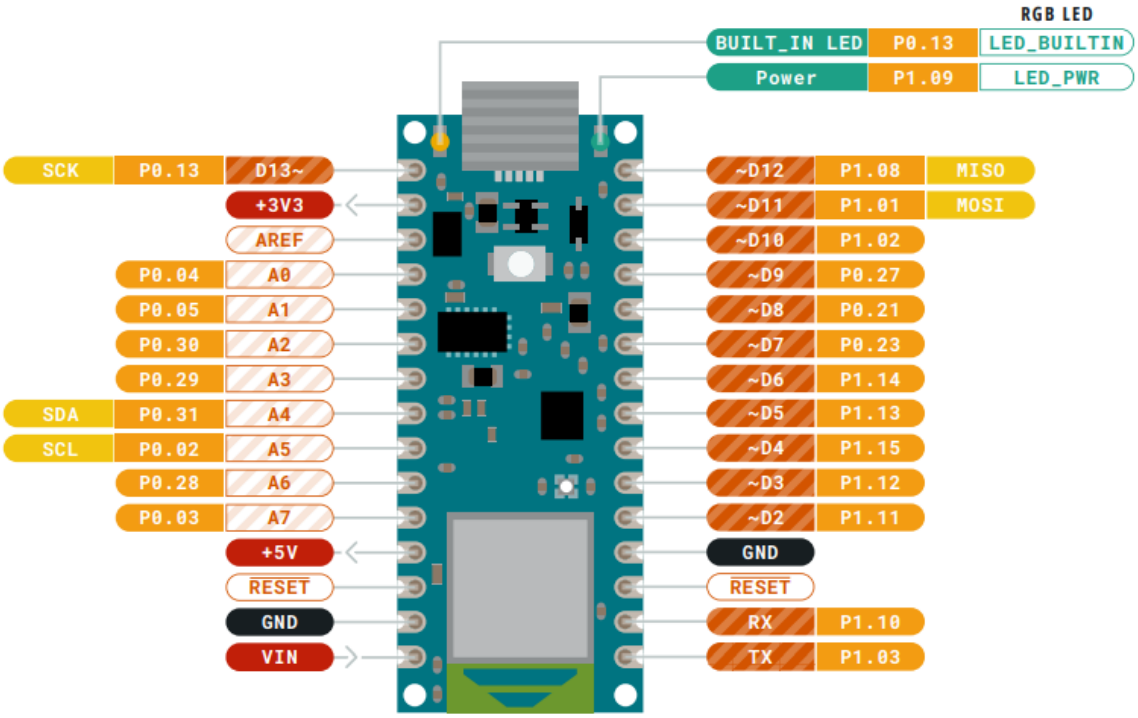
2.7.1 Gesture Detection

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.

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

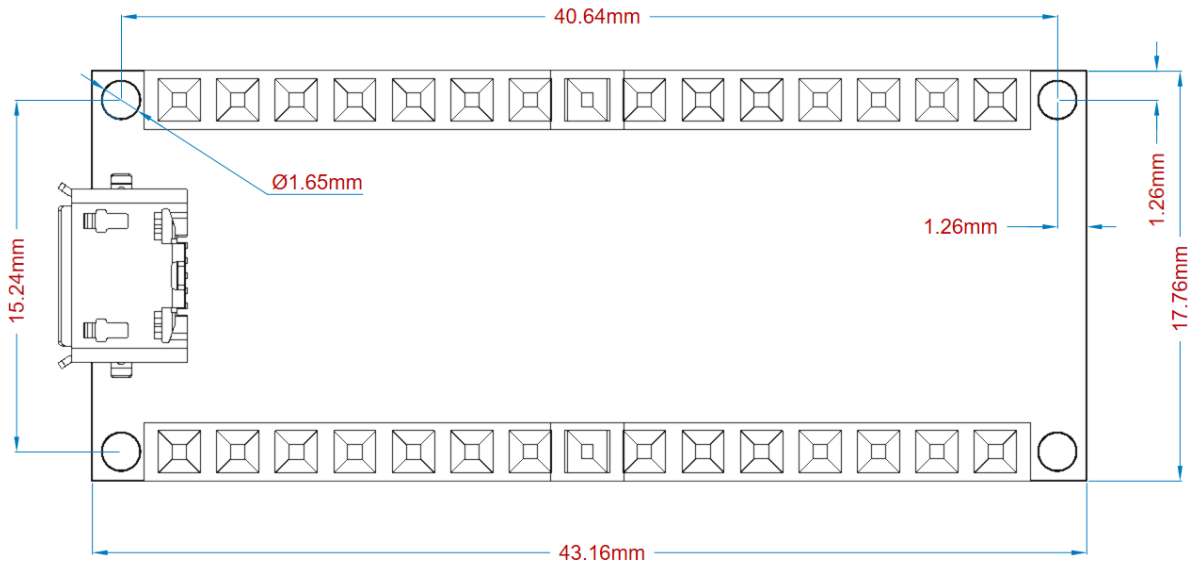
On the bottom side of the board, under the communication module, debug signals are arranged as 3x2 test pads with 100 mil pitch with pin 4 removed. Pin 1 is depicted in Figure 3 – Connector Positions

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

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



Board layout

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).