

Proyecto: Detector de incendios

Katherine Vargas, B88198
Susan Jiménez Hernández, B94031
Profesor: MSc. Marco Villalta Fallas

Resumen—En este documento se presenta el desarrollo de un sistema de detección de incendios utilizando un microcontrolador Arduino Nano 33 BLE, técnicas de Machine Learning y la plataforma Edge Impulse. La aplicación presente hace uso del valor RGB, la temperatura y la presión para detectar el fuego. Para la toma de datos primeros se tomaron datos sin la presencia de fuego, luego se procedió a generar un pequeño incendio para tomar el otro set de datos (tomando las medidas de precaución necesarias). Posteriormente se entreno un modelo de maching learnign en la plataforma de Edge Impulse, para luego exportar el modelo, crear un código basado en los ejemplos de la librería y finalmente ejecutarlo en el arduino.

Palabras clave—Arduino Nano 33 BLE, Machine learning, Detector de incendios, Edge Impulse.

I. JUSTIFICACIÓN

El desarrollo del proyecto de Detector de Incendios en el curso de Laboratorio de Microcontroladores cobra especial relevancia debido a la importancia crucial de detectar incendios en su fase inicial para minimizar los daños tanto humanos como materiales. Los detectores de incendios son esenciales en las instalaciones contra incendios, ya que representan la forma más rápida y eficaz de identificar un incendio en sus etapas iniciales. Esto posibilita la activación inmediata de medidas de control y facilita la evacuación oportuna de la zona afectada, reduciendo así el riesgo de pérdidas significativas.

II. OBJETIVO GENERAL

- Crear un sistema de detección de incendios mediante un Arduino Nano y técnicas de Machine Learning para anticipar la presencia de fuego.

III. OBJETIVOS Y ALCANCES

- Recopilar y procesar datos de detección de fuego para entrenar e implementar un modelo en *Edge Impulse*.
- Exportar el modelo desarrollado por medio de *Edge Impulse*, para su ejecución en el Arduino Nano 33 BLE.
- Desarrollar un programa en Arduino que utilice la biblioteca del modelo para detectar la presencia de fuego y activar una alerta.

Reporte entregado el 3 de julio del 2024.

IV. NOTA TEÓRICA

IV-A. Información general MCU

El Arduino Nano 33 BLE cuenta con un núcleo del módulo NINA B306 se basa en el Nordic nRF52840 y cuenta con un robusto procesador ARM Cortex-M4F. Esta combinación de hardware hace que Arduino Nano 33 BLE sea una opción excepcional para proyectos donde las limitaciones de tamaño y la eficiencia energética son fundamentales. Su diseño es particularmente beneficioso para la tecnología portátil, aplicaciones de IoT y cualquier proyecto donde la compacidad sea primordial. La integración Bluetooth Low Energy (BLE) de la placa agrega una capa de conectividad, lo que permite la comunicación inalámbrica y facilita su uso en ecosistemas de IoT modernos. [1]

Los sensores integrados, incluida una unidad de medición inercial (IMU) de 9 ejes, proporcionan datos valiosos para el seguimiento de movimiento y la detección ambiental, ampliando aún más el alcance de aplicación del Arduino Nano 33 BLE. Sus entradas y salidas digitales y analógicas integradas y una gama de interfaces periféricas la convierten en una herramienta increíblemente versátil para una amplia gama de proyectos. [1]

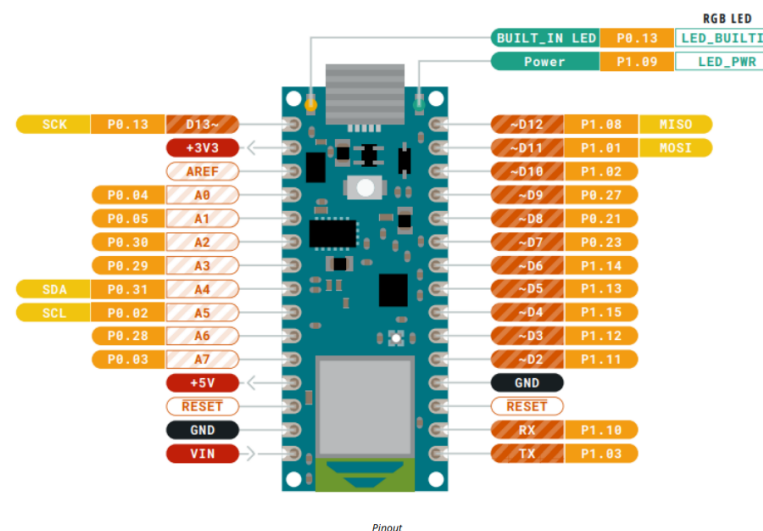


Figura 1: Diagrama de pines del Arduino Nano 33 BLE. [1]

IV-B. Periféricos

- Full-speed 12 Mbps USB

- NFC-A tag
- Arm CryptoCell CC310 security subsystem
- QSPI/SPI/TWI/I²S/PDM/QDEC
- High speed 32 MHz SPI
- Quad SPI interface 32 MHz
- EasyDMA for all digital interfaces
- 12-bit 200 ksp/s ADC
- 128 bit AES/ECB/CCM/AAR co-processor

IV-C. Sensor LPS22HB

El LPS22HB es un sensor piezorresistivo de presión absoluta ultra-compacto que funciona como un barómetro de salida digital, con un rango de presión absoluta de 260 a 1260 hPa. El dispositivo comprende un elemento de detección y una interfaz IC que se comunica a través de I2C o SPI desde el elemento de detección hasta la aplicación. Está garantizado para operar en un rango de temperatura que se extiende desde -40 °C hasta +85 °C. El paquete tiene orificios para permitir que la presión externa alcance el elemento de detección.[2]

IV-D. Sensor APDS9960

El APDS-9960 es un sensor de gestos, sensor de proximidad, sensor de color y sensor de luz ambiental. El APDS-9960 permite medir el color, medir la luz ambiental y detección de gestos. La detección de gestos sin contacto permite detectar un objeto moviéndose a 5-20cm, detectando gestos de izquierda, derecha, arriba, abajo, cerca y lejos. El APDS-9960 integra un emisor de infrarrojos, cuatro fotodiodos direccionales y cuatro fotodiodos para la medición de color. Además, incorpora la electrónica necesaria para la medición, con rechazo de luz ambiental y compensación de offset. [3]

IV-E. Edge Impulse

Edge Impulse es la plataforma de IA en el borde para la recopilación de datos, el entrenamiento de modelos y su implementación en tus dispositivos de computación en el borde. Proporciona un marco de trabajo de extremo a extremo que se integra fácilmente en tu flujo de trabajo de MLOps en el borde. Es una herramienta basada en la web con una interfaz gráfica que te ayuda a recopilar datos, construir un impulso e implementarlo en un dispositivo final.[4]

Edge Impulse ayuda en cada paso del ciclo de vida de la IA en el borde, desde la recopilación de datos, la extracción de características, el diseño de modelos de aprendizaje automático (ML), el entrenamiento y la prueba de esos modelos, y la implementación de los modelos en dispositivos finales, como se muestra en la Figura 2. [4]

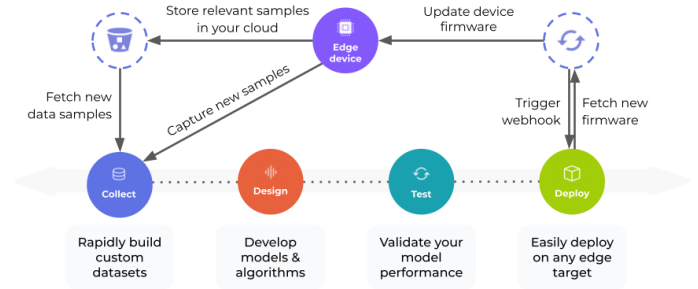


Figura 2: *Edge Impulse* para el ciclo de vida de la IA en el borde.[4]

IV-F. TinyML y Tensorflow

- **TinyML:** Tiny Machine Learning (o TinyML) es una técnica de aprendizaje automático que integra aplicaciones de aprendizaje automático reducidas y optimizadas que requieren soluciones de "pila completa" (hardware, sistema, software y aplicaciones), incluidas arquitecturas, técnicas, herramientas y enfoques de aprendizaje automático. capaz de realizar análisis en el dispositivo en el mismo borde de la nube. TinyML se puede implementar en sistemas de baja energía, como sensores o microcontroladores, para realizar tareas automatizadas. [5]
- **Tensorflow:** Se trata de una librería de código libre para Machine Learning (ML). Fue desarrollado por Google para satisfacer las necesidades a partir de redes neuronales artificiales. TensorFlow te permite construir y entrenar redes neuronales para detectar patrones y razonamientos usados por los humanos. [6]
TensorFlow es una plataforma que facilita la creación e implementación de modelos de aprendizaje automático. Lo que buscamos hoy en día es automatizar cientos de procesos y TensorFlow nos permite llegar a esta automatización. Nos permite crear y entrenar modelos de Aprendizaje Automático con facilidad mediante APIs intuitivas.[6]

IV-G. Análisis espectral

El análisis espectral investiga señales en el dominio de la frecuencia utilizando la Transformada de Fourier, a partir de una señal en el dominio del tiempo. La forma de onda se representa mediante una suma de senos y cosenos de diferentes frecuencias y amplitudes, generando un espectro que revela frecuencias dominantes y la concentración de energía en bandas específicas. Los sensores recogen datos en tiempo real. Estos datos se pueden transformar al dominio de la frecuencia para identificar patrones anómalos que podrían indicar un incendio. [7]

IV-H. Aprendizaje Supervisado, Clasificación y Redes Neuronales

El aprendizaje supervisado es una subcategoría del machine learning y la inteligencia artificial que utiliza conjuntos de

datos etiquetados para entrenar algoritmos, con el objetivo de clasificar datos o predecir resultados con precisión. [8]

En el aprendizaje supervisado, la clasificación utiliza algoritmos para asignar datos a categorías específicas, y las redes neuronales son una herramienta poderosa para esta tarea. Las redes neuronales imitan la estructura del cerebro humano con capas de nodos que procesan datos, ajustando ponderaciones durante el entrenamiento para minimizar errores. Pueden manejar grandes cantidades de datos y son efectivos en el aprendizaje de patrones complejos. Una vez que se han entrenado, pueden clasificar nuevos datos con gran precisión. [?]

IV-1. Buzzer

Un zumbador o mejor conocido como buzzer (en inglés) es un pequeño transductor capaz de convertir la energía eléctrica en sonido. Para hacerlos funcionar solo basta conectar el positivo con el + y la tierra o negativo con el – de una batería o cualquier fuente de corriente directa.[9]

El funcionamiento se basa en el efecto piezoeléctrico de los materiales. Este efecto funciona de tal manera que cuando aplicamos un voltaje el volumen del material cambia ligeramente. Los zumbadores están contruidos con dos pequeñas placas una metálica y una cerámica, las cuales aprovechan este efecto pero solo generan un click ya que los materiales cambiaron de forma pero no regresan a su estado natural hasta que se les quita el voltaje. Para que se pueda emitir un sonido continuo las placas necesitan vibrar constantemente, para eso se instala un oscilador que hace que los materiales cambien de estado una y otra vez, y así puedan cambiar miles de veces para poder alcanzar un audio perceptible.[9]

IV-J. Transistor 2N2222

El transistor 2N2222 es un NPN BJT común y se usa principalmente en aplicaciones de conmutación y amplificación con menos potencia. Este transistor está diseñado principalmente para baja potencia, corriente baja a media, voltaje medio y funciona a velocidades bastante altas. En el transistor NPN 2N2222, una sola capa dopada con P está integrada entre dos capas dopadas con N. Este transistor incluye tres terminales, como la base, el emisor y el colector. Este transistor tiene tres pines como se muestra en la Figura 3.[10]

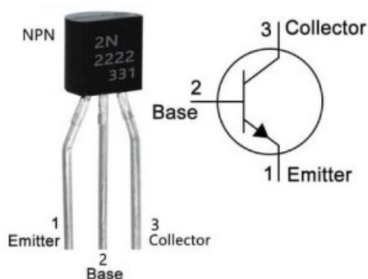


Figura 3: Configuración y símbolo de pines 2N22222. [10]

V. LISTA DE COMPONENTES Y PRECIOS

Componentes	Cantidad	Precio unitario	Tienda
Arduino Nano 33 BLE	1	€19 878	Mouser electronics
Buzzer	1	€350	RobotShop
Transistor 2N2222	1	€350	Centronik
Jumpers	4	€160	Steren
Resistor	1	€69,01	Steren
Total:		€20.807,01	

Cuadro I: tabla_precios

VI. DESARROLLO Y ANÁLISIS DE RESULTADOS

VI-A. Diseño del circuito

Para la alarma, se implementó un diseño de circuito simple, utilizando solo tres componentes externos a la placa principal. Estos incluyen un buzzer para emitir el sonido de alerta en caso de detección de fuego, un transistor 2n2222a y una resistencia para regular la corriente que circula por el circuito. La configuración de este circuito se puede observar en la figura 4.

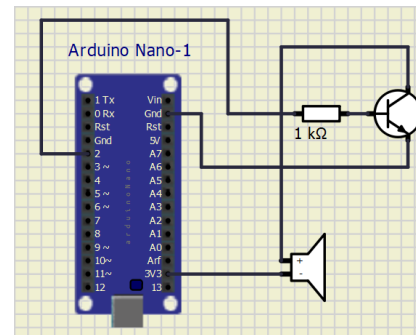


Figura 4: Circuito

Para saturar el transistor según la hoja de datos del fabricante, se necesita una corriente de base de $I_B = 5\text{mA}$. Dado que la tensión de salida del Arduino Nano es de 3.3V en estado alto, se tiene la siguiente ecuación:

$$R = \frac{V}{I} = \frac{3,3V}{5mA} = 660\Omega \quad (1)$$

Se eligió una resistencia de $1k\Omega$ (debido a que es un valor comercial, y ya teníamos una de este valor) con la que también se consigue una corriente mínima de saturación, proporcionando una corriente equivalente a:

$$I = \frac{V}{R} = \frac{3,3V}{1000} = 3,3mA \quad (2)$$

VI-B. Adquisición de datos

Para la adquisición de los datos de los dos diferentes sensores (LPS22H y APDS9960) se realizó por medio de *Edge Impulse*, pero para ello se trabajó con una herramienta que se llama *Data Forwarder*, donde se creó el siguiente código para poder obtener los datos en *Edge Impulse*, para ello se deben mandar los datos en una línea separadas por una coma (.). Luego al conectar el Arduino se utiliza el comando *edge-impulse-data-forwarder* se le coloca las etiquetas de los datos en el orden que se están mandando en este caso se mandan en el orden

rojo, verde, azul, presión, temperatura, como se muestra en la Figura 5.

```
21:48:21.285 -> 0.370,0.333,0.296,89.42,30.41
21:48:21.374 -> 0.370,0.333,0.296,89.42,30.42
21:48:21.420 -> 0.370,0.333,0.296,89.42,30.42
21:48:21.466 -> 0.370,0.333,0.296,89.42,30.41
21:48:21.502 -> 0.370,0.333,0.296,89.42,30.42
21:48:21.589 -> 0.370,0.333,0.296,89.42,30.42
21:48:21.617 -> 0.370,0.333,0.296,89.42,30.41
```

Figura 5: Datos a recolectar en *Edge Impulse*

```
1      #include <Arduino_LPS22HB.h>
2      #include <Arduino_APDS9960.h> //RGB
3
4      void setup() {
5          Serial.begin(115200); // Cambiado
                                   a 115200
6          while (!Serial);
7
8          if (!BARO.begin() ||
9              !APDS.begin()) {
10             Serial.println("Error al
11                 inicializar");
12         }
13
14         void loop() {
15
16             ///RGB
17             int rojo, verde, azul, lum, prox;
18             float sum;
19             // se espera hasta que existan
20                 datos de color y
21                 proximidad en el
22                 sensor
23
24             while (!APDS.colorAvailable()
25                 || !APDS.proximityAvailable()) {}
26
27             // Lee los valores de color y
28                 luminosidad del
29                 sensor
30
31             APDS.readColor(rojo, verde, azul,
32                 lum);
33
34             prox = APDS.readProximity();
35                 //Lee el valor de
36                 proximidad del
37                 sensor
38
39             sum = rojo + verde + azul;
40             float redRatio = rojo / sum;
41             float greenRatio = verde / sum;
42             float blueRatio = azul / sum;
```

```
34     ///Calculo de presi[on y
35         temperatura]
36
37     float pressure =
38         BARO.readPressure();
39         // LPS22HB usa hPa
40         por defecto
41
42     float temperature =
43         BARO.readTemperature();
44
45     // Enviar los datos en formato
46         CSV
47
48     Serial.print(redRatio, 3);
49     Serial.print(',');
50     Serial.print(greenRatio, 3);
51     Serial.print(',');
52     Serial.print(blueRatio, 3);
53     Serial.print(',');
54     Serial.print(pressure);
55     Serial.print(",");
56     Serial.println(temperature);
57 }
```

Una vez que se logra conectar el código desarrollado al *Edge impulse*, se toman los datos de pequeños incendios creados como se muestra en la Figura 6, asimismo, se toman los datos sin fuego en el mismo lugar antes de encender el fuego de prueba como se muestra en la Figura 7.

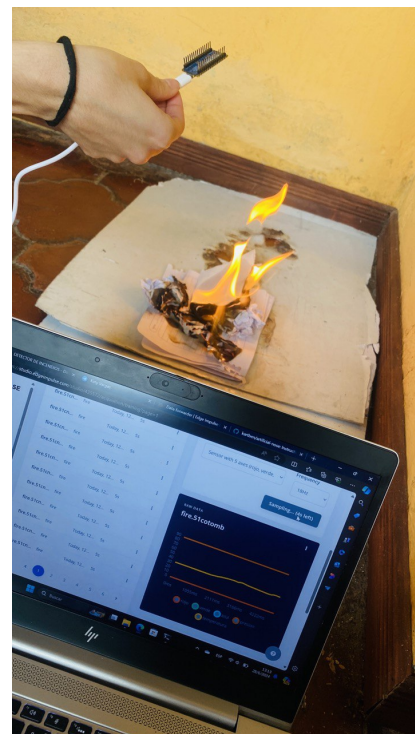


Figura 6: Datos con fuego controlado



Figura 7: Datos sin fuego.

Para visualizar mejor los datos recolectados decidimos multiplicar por 100 los porcentajes de colores, para esto tuvimos que exportar los datos y luego escribir un *script* en *Python* que editara, solo los 3 primeros valores de los vector, y que ejecutara esta acción para cada archivo, ya que editar esto de forma manual hubiera sido un proceso muy lento y e ineficiente.

```

1  import json
2  import os
3
4  # Definir el rango de archivos
5  num_files = 57
6
7  # Ruta base de los archivos
8  input_base_path = 'no_fire
      ({}).json'
9
10 output_base_path =
11
12 'no_fire.{}.json'
13
14 for i in range(1, num_files + 1):
15
16     input_file_name =
17     input_base_path.format(i)
18     output_file_name =
19     output_base_path.format(i)
20

```

```

21     # Verificar si el archivo de
           entrada existe
22
23     if not
24     os.path.exists(input_file_name):
25         print(f"Archivo
           {input_file_name}
26         no encontrado")
27         continue
28
29     # Cargar el archivo JSON
30     with open(input_file_name, 'r')
           as file:
31         data = json.load(file)
32
33     # Recorrer los vectores en el
           campo `values` y
           multiplicar los
           primeros tres
           números por 100
34
35     for vector in
36
37     data['payload']['values']:
38
39         for j in range(3):
40
41             vector[j] =
42             round(vector[j] * 100,
43                   2)
44
45     # Guardar los cambios en el
           archivo JSON
46     with open(output_file_name, 'w')
           as file:
47         json.dump(data, file,
48                 indent=4)
49
50     print(f"Archivo modificado y
           guardado como
           '{output_file_name}'")

```

VI-C. Etiquetado de Datos

Para el etiquetado de los datos se utilizó dos clases de datos "fire" y "no_fire", en la primera clase se agregaron los datos tomados en el momento que se estaba presentado el pequeño incendio realizado, mientras que en la segunda se agregaron los datos tomados antes de realizar el pequeño incendio, con esta clasificación se entrena el modelo.

VI-D. Creación del impulso y entrenamiento del modelo

Para la generación del impulso, se agregó un bloque de análisis espectral y un bloque de 'Classificatio' como se muestra en la Figura 8, el bloque de análisis espectral se encarga de convertir los datos recolectados con los sensores al dominio de la frecuencia con el fin de poder identificar patrones para lograr

detectar en el momento que exista un incendio, mientras que, el bloque classification se utiliza el conjunto de datos etiquetados para entrenar el algoritmo y así poder generar resultados más precisos.

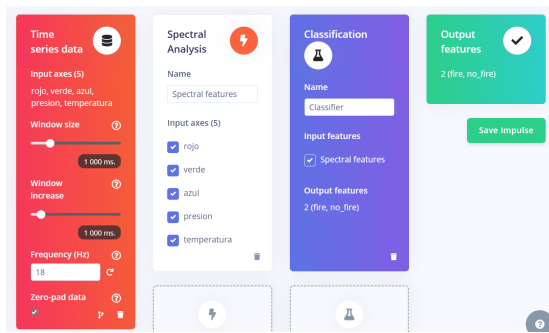


Figura 8: Creación del impulso.

Además, en el bloque classification hay ciertos parámetros como número de ciclos de entrenamiento, razón de aprendizaje, data augmentation y selección de modelo que se editaron para la configuración del modelo como se muestra en la Figura 9.

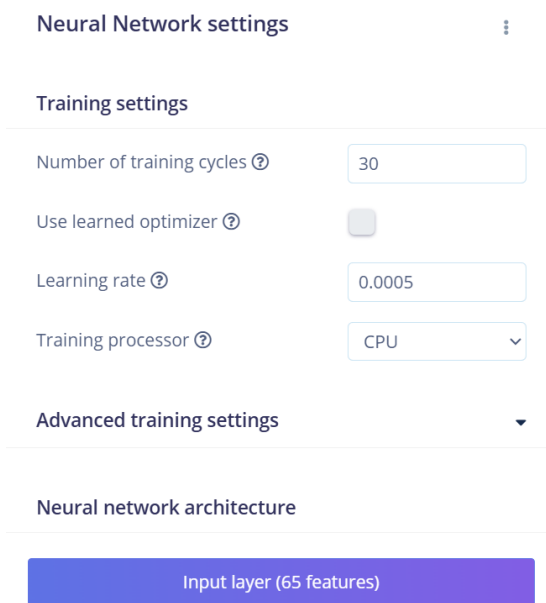


Figura 9: Neural Network settings.

VI-E. Resultados del modelo

Al realizar el entrenamiento del modelo se obtienen los resultados que se muestran en las Figuras 10 y 11, en la primera Figura se observa la matriz de confusión y el porcentaje de accuracy que es de 88.5% que es un valor mejorable pero aceptable para la cantidad y calidad de datos recolectados, sin embargo, en la matriz de confusión se puede observar que lo que reduce el accuracy es porque no detecta bien los datos de "no_fire", mientras que los datos con fire los detecta muy bien, esto nos indica que probablemente hicieron falta recolectar una mayor cantidad de datos de no fuego. Luego se puede observar

en el data explorer que los datos que no detecta bien son muy pocos en comparación con los que detecta de manera correcta.



Figura 10: Matriz de confusión obtenida al entrenar el modelo

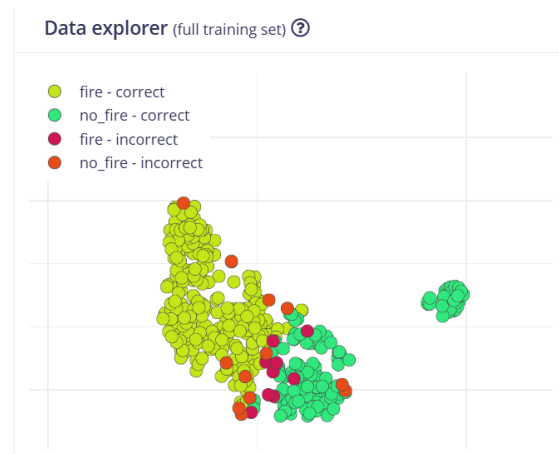


Figura 11: Clasificación de datos por el modelo

VI-F. Testeo del modelo

Para el testeo del modelo se utilizaron un 20% de los datos recolectados, y se obtuvo una precisión del 93%. El área bajo la curva ROC es considerablemente alta, alcanzando 0.98, lo que indica una excelente capacidad del modelo para distinguir entre clases. Tanto la precisión ponderada promedio como el recall ponderado promedio son del 98%, reflejando una buena capacidad de predicción y detección de casos positivos (fuego). La matriz de confusión revela una buena precisión en la clasificación de "FIRE" (fuego), con un F1 Score de 0.97, y aunque la precisión para "NO_FIRE" (sin fuego) es menor con 0.92, lo cual podría dar muchos falsos negativos, también se puede notar que hay una cantidad de datos considerables que el modelo no sabe como clasificar, pensamos que esto se podría solucionar entrenando el modelo con mayores datos.



Figura 12: Testeo del modelo

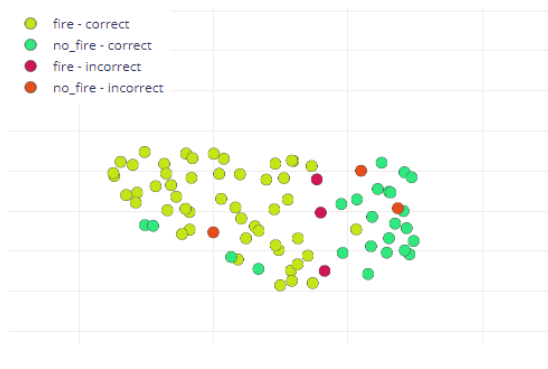


Figura 13: Testeo del modelo

VI-G. Exportación del modelo

Para exportar el modelo como un librería y poder utilizarlo en el Arduino Nano 33 BLE, se tenía dos opciones de optimizaciones que se muestran en la Figura 14, se selecciona la optimización Quantized, sin embargo, se podía seleccionar cualquiera de las dos, ya que, el desempeño de ambas optimizaciones es el mismo y de igual manera la RAM que consumen es muy similar.

Quantized (int8) <input checked="" type="checkbox"/>		SPECTRAL FEATURES		CLASSIFIER	TOTAL
	LATENCY	6 ms.		1 ms.	7 ms.
	RAM	1,2K		1,4K	1,4K
	FLASH	-		16,4K	-
	ACCURACY				93.02%
Unoptimized (float32) <input type="checkbox"/>		SPECTRAL FEATURES		CLASSIFIER	TOTAL
	LATENCY	6 ms.		54 ms.	60 ms.
	RAM	1,2K		1,6K	1,6K
	FLASH	-		17,4K	-
	ACCURACY				93.02%

Figura 14: Optimizaciones

VI-H. Detector de incendios

Después de la exportación del modelo, tuvimos que basarnos en uno de los ejemplos de la librería para poder ejecutar el código. En este caso, utilizamos el de 'nano_ble33_sense_fusion.ion' debido a que este código fusionaba varios sensores y era lo más cercano a lo que necesitábamos. Para esta parte fue fundamental leer y entender el código para eliminar las partes que incluían sensores que no necesitábamos y, al mismo tiempo, añadir los que sí, cuidando el orden en el que inicialmente tomamos los datos, para mantener la congruencia.

Finalmente pudimos ejecutar el código y evaluar su funcionalidad. A continuación, se presenta la figura 15 que muestra la situación cuando no hay fuego. En esta figura, podemos observar valores bastante altos para la clase 'no_fire', mientras que los valores para la clase 'fire' se mantienen bajos.

```
00:33:27.666 -> Starting inferencing in 2 seconds...
00:33:29.651 -> Sampling...
00:33:30.638 -> Predictions (DSP: 18 ms., Classification: 0 ms., Anomaly: 0 ms.):
00:33:30.638 -> fire: 0.00391
00:33:30.638 -> no_fire: 0.99609
00:33:30.638 ->
00:33:30.638 -> Starting inferencing in 2 seconds...
00:33:32.659 -> Sampling...
00:33:33.650 -> Predictions (DSP: 18 ms., Classification: 0 ms., Anomaly: 0 ms.):
00:33:33.650 -> fire: 0.00000
00:33:33.650 -> no_fire: 0.99609
00:33:33.650 ->
00:33:33.650 -> Starting inferencing in 2 seconds...
00:33:35.648 -> Sampling...
00:33:36.690 -> Predictions (DSP: 18 ms., Classification: 0 ms., Anomaly: 0 ms.):
00:33:36.690 -> fire: 0.00000
00:33:36.690 -> no_fire: 0.99609
00:33:36.690 ->
```

Figura 15: Captura del monitor serial

En la figura 15 se muestra el caso opuesto, donde se detecta la presencia de fuego con valores bastante altos (fire), mientras que la clase 'no_fire' se mantiene en valores bajos.

```
00:35:00.971 -> Starting inferencing in 2 seconds...
00:35:02.975 -> Sampling...
00:35:03.971 -> Predictions (DSP: 18 ms., Classification: 0 ms., Anomaly: 0 ms.):
00:35:03.971 -> fire: 0.99219
00:35:03.971 -> no_fire: 0.00781
00:35:03.971 ->
00:35:03.971 -> Starting inferencing in 2 seconds...
00:35:05.994 -> Sampling...
00:35:06.958 -> Predictions (DSP: 18 ms., Classification: 0 ms., Anomaly: 0 ms.):
00:35:06.958 -> fire: 0.99609
00:35:06.958 -> no_fire: 0.00000
00:35:06.958 ->
00:35:06.958 -> Starting inferencing in 2 seconds...
00:35:08.994 -> Sampling...
00:35:09.966 -> Predictions (DSP: 18 ms., Classification: 0 ms., Anomaly: 0 ms.):
00:35:09.966 -> fire: 0.99609
00:35:10.001 -> no_fire: 0.00391
```

Figura 16: Captura del monitor serial

VII. DIFICULTADES

- Obtención de los datos: Como en Edge impulse no permitía la obtención directa de dos diferentes sensores del Arduino se debe realizar la importación de archivos CSV o el uso de un data forwarder para transmitir los datos desde Arduino a Edge Impulse.
- Edición de los datos recolectados: *Edge Impulse* no cuenta con una opción para editar los datos, por lo que, se tuvo que realizar un script en *Python* para la automatización y edición de los datos.
- Se presentaron algunas dificultades con el Arduino, lo que retrasó el desarrollo del proyecto.

VIII. CONCLUSIONES

- Se logra realizar una conexión exitosa con *Edge Impulse* para la adquisición de los datos.
- Se entrena e implementa correctamente un modelo para la detección de incendios.
- Se logra exportar el modelo como una biblioteca para que pueda ser usada en el Arduino Nano 33 BLE.
- Se desarrolla un código en el Arduino Nano 33 BLE capaz de detectar los incendios y activar una alarma.

IX. RECOMENDACIONES

- Variar los parámetros de entrenamiento para lograr obtener mejores resultados del modelo.
- Agregar más datos con fuego y sin fuego para hacer más preciso el modelo.
- Implementar un módulo de WiFi al circuito para enviar un mensaje a un dispositivo cuando se detecte un posible incendio.

REFERENCIAS

- [1] S. Handi, "Explorando el arduino nano 33 ble: el poder de la innovación compacta," 2024, <https://full-skills.com/es/arduino-uno-projects/arduino-nano-33-ble/>.
- [2] "LPS22HB - STMicroelectronics." [Online]. Available: <https://www.st.com/en/mems-and-sensors/lps22hb.html>
- [3] L. Llamas, "Detectar gestos con Arduino y sensor APDS-9960," 1 2018. [Online]. Available: <https://www.luisllamas.es/detectar-gestos-con-arduino-y-sensor-apds-9960/>
- [4] "What is Edge Impulse? — Edge Impulse Documentation." [Online]. Available: <https://docs.edgeimpulse.com/docs/concepts/edge-ai/what-is-edge-impulse>
- [5] "¿Qué es TinyML y por qué es importante?" 12 2020. [Online]. Available: <https://ichi.pro/es/que-es-tinyml-y-por-que-es-importante-270142243498102>
- [6] "¿Qué es TensorFlow y para qué sirve?" 4 2024. [Online]. Available: <https://www.incentro.com/es-ES/blog/que-es-tensorflow>
- [7] D. S.A., "Cómo identificar fallas mediante el análisis espectral," 1 2022. [Online]. Available: <https://dynamox.net/es/blog/como-identificar-fallas-utilizando-el-analisis-espectral>
- [8] IBM, "¿Qué es el aprendizaje supervisado?" 7 2024. [Online]. Available: <https://www.ibm.com/es-es/topics/supervised-learning>
- [9] I. Mecafenix, "Que es el buzzer y como funciona (zumbador)," 12 2023. [Online]. Available: <https://www.ingmecafenix.com/electronica/componentes/el-buzzer/>
- [10] Javired and Javired, "¿Qué es un transistor 2N2222A y cómo funciona?" 7 2022. [Online]. Available: <https://electrositio.com/que-es-un-transistor-2n2222a-y-como-funciona/>

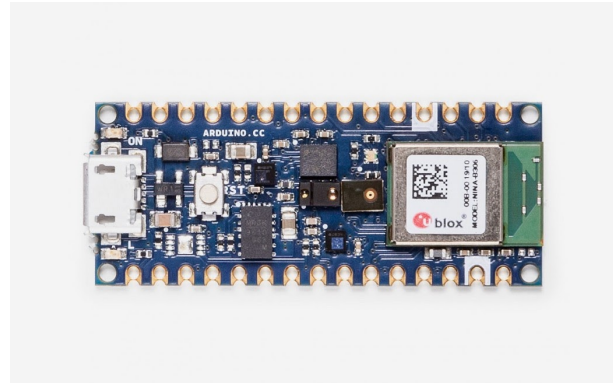
X. LINK DEL GITHUB

https://github.com/katyvc29/Laboratorio_de_microcontroladores

XI. LINK DEL EDGE IMPULSE

<https://studio.edgeimpulse.com/studio/436149>

XII. APÉNDICE



Description

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

Target areas:

Maker, enhancements, IoT application



Features

- **NINA B306 Module**
 - **Processor**
 - 64 MHz Arm® Cortex-M4F (with FPU)
 - 1 MB Flash + 256 KB RAM
 - **Bluetooth® 5 multiprotocol radio**
 - 2 Mbps
 - CSA #2
 - Advertising Extensions
 - Long Range
 - +8 dBm TX power
 - -95 dBm sensitivity
 - 4.8 mA in TX (0 dBm)
 - 4.6 mA in RX (1 Mbps)
 - Integrated balun with 50 Ω single-ended output
 - IEEE 802.15.4 radio support
 - Thread
 - Zigbee
 - **Peripherals**
 - Full-speed 12 Mbps USB
 - NFC-A tag
 - Arm CryptoCell CC310 security subsystem
 - QSPI/SPI/TWI/I²S/PDM/QDEC
 - High speed 32 MHz SPI
 - Quad SPI interface 32 MHz
 - EasyDMA for all digital interfaces
 - 12-bit 200 ksp/s ADC
 - 128 bit AES/ECB/CCM/AAR co-processor
- **LSM9DS1** (9 axis IMU)
 - 3 acceleration channels, 3 angular rate channels, 3 magnetic field channels
 - $\pm 2/\pm 4/\pm 8/\pm 16$ g linear acceleration full scale
 - $\pm 4/\pm 8/\pm 12/\pm 16$ gauss magnetic full scale
 - $\pm 245/\pm 500/\pm 2000$ dps angular rate full scale
 - 16-bit data output
- **LPS22HB** (Barometer and temperature sensor)
 - 260 to 1260 hPa absolute pressure range with 24 bit precision
 - High overpressure capability: 20x full-scale
 - Embedded temperature compensation
 - 16-bit temperature data output
 - 1 Hz to 75 Hz output data rate/Interrupt functions: Data Ready, FIFO flags, pressure thresholds
- **HTS221** (relative humidity sensor)
 - 0-100% relative humidity range
 - High rH sensitivity: 0.004% rH/LSB
 - Humidity accuracy: $\pm 3.5\%$ rH, 20 to +80% rH
 - Temperature accuracy: ± 0.5 °C, 15 to +40 °C
 - 16-bit humidity and temperature output data



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



Contents

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



1 The Board

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

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

1.1 Ratings

1.1.1 Recommended Operating Conditions

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

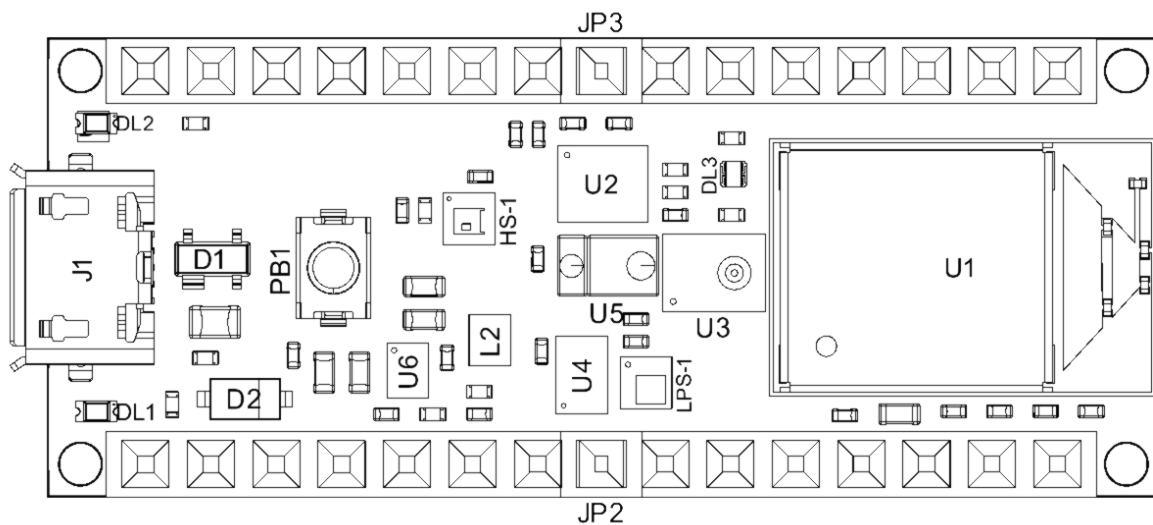
1.2 Power Consumption

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

2 Functional Overview

2.1 Board Topology

Top:



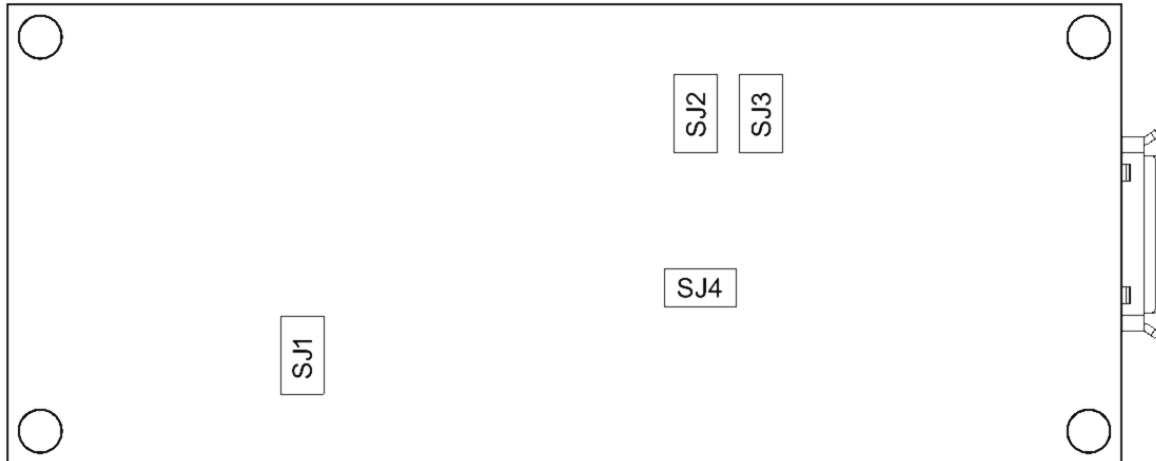
Board topology top

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



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

Bottom:



Board topology bot

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

2.2 Processor

The Main Processor is a Cortex M4F running at up to 64MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the wireless module and the on-board internal I²C peripherals (IMU and Crypto).

NOTE: As opposed to other Arduino Nano boards, pins A4 and A5 have an internal pull up and default to be used as an I²C Bus so usage as analog inputs is not recommended.

2.3 Crypto

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

Source code for the Arduino Library that supports the Crypto is available [\[8\]](#)



2.4 IMU

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

Source code for the Arduino Library that supports the IMU is available [\[9\]](#)

2.5 Barometer and Temperature Sensor

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

Source code for the Arduino Library that supports the Barometer is available [\[10\]](#)

2.6 Relative Humidity and Temperature Sensor

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

Source code for the Arduino Library that supports the Humidity sensor is available [\[11\]](#)

2.7 Digital Proximity, Ambient Light, RGB and Gesture Sensor

Source code for the Arduino Library that supports the Proximity/gesture/ALS sensor is available [\[12\]](#)

2.7.1 Gesture Detection

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

2.7.2 Proximity Detection

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



2.7.3 Color and ALS Detection

The Color and ALS detection feature provides red, green, blue and clear light intensity data. Each of the R, G, B, C channels have a UV and IR blocking filter and a dedicated data converter producing 16-bit data simultaneously. This architecture allows applications to accurately measure ambient light and sense color which enables devices to calculate color temperature and control display backlight.

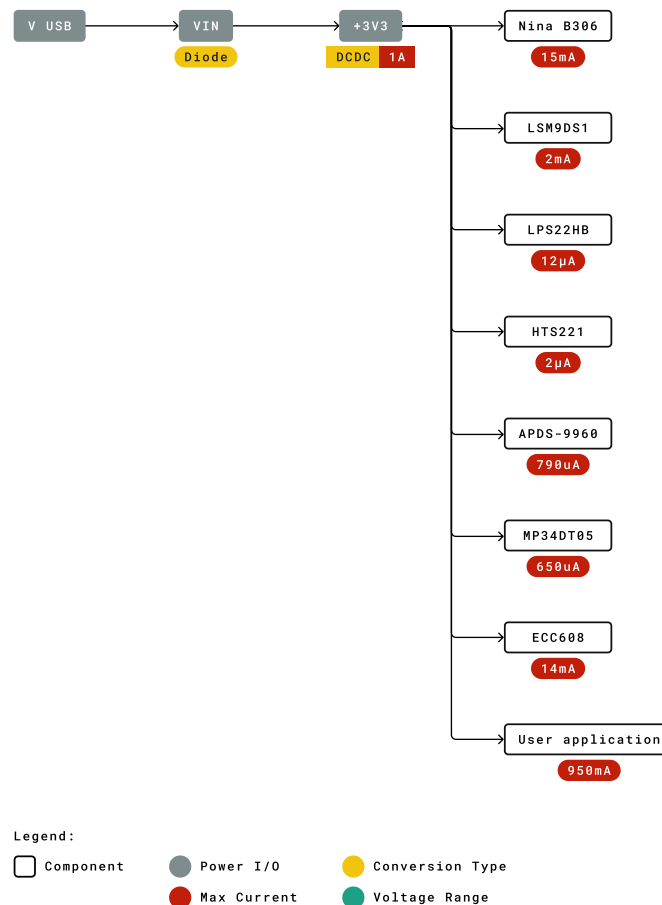
2.8 Digital Microphone

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

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

2.9 Power Tree

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



Power tree

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



3 Board Operation

3.1 Getting Started - IDE

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

3.2 Getting Started - Arduino Web Editor

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

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

3.3 Getting Started - Arduino IoT Cloud

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

3.4 Sample Sketches

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

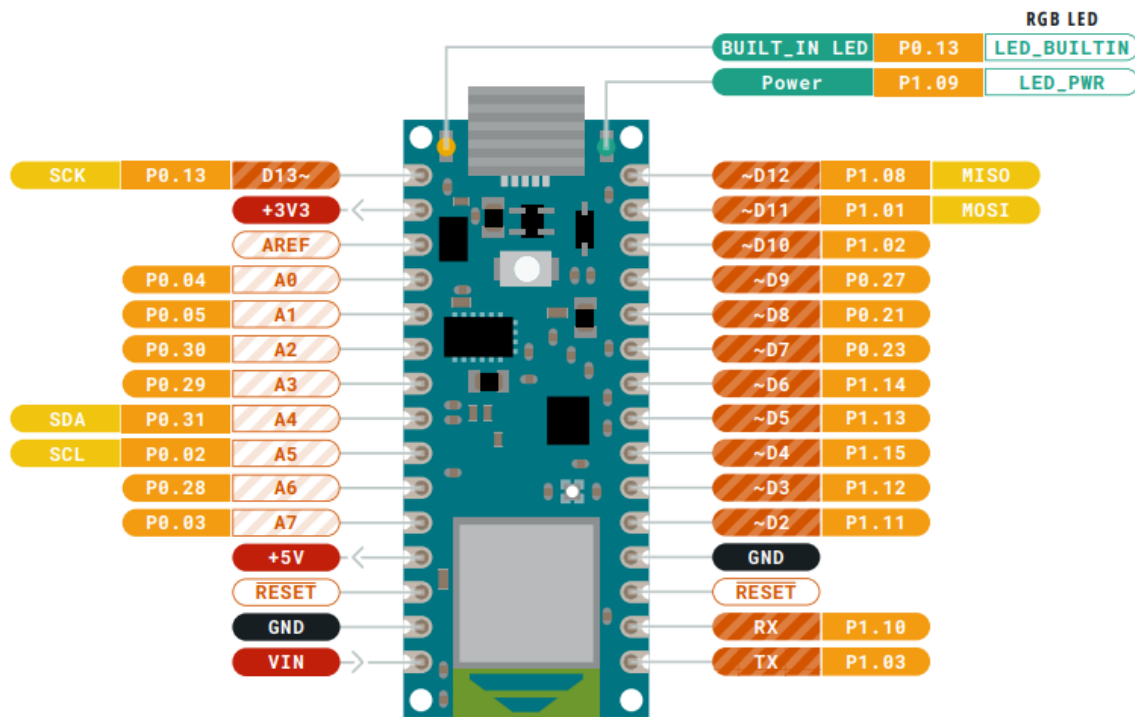
3.5 Online Resources

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

3.6 Board Recovery

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

4 Connector Pinouts



Pinout

4.1 USB

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

4.2 Headers

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

Pin	Function	Type	Description
1	D13	Digital	GPIO
2	+3V3	Power Out	Internally generated power output to external devices
3	AREF	Analog	Analog Reference; can be used as GPIO
4	A0/DAC0	Analog	ADC in/DAC out; can be used as GPIO
5	A1	Analog	ADC in; can be used as GPIO
6	A2	Analog	ADC in; can be used as GPIO
7	A3	Analog	ADC in; can be used as GPIO
8	A4/SDA	Analog	ADC in; I2C SDA; Can be used as GPIO (1)
9	A5/SCL	Analog	ADC in; I2C SCL; Can be used as GPIO (1)
10	A6	Analog	ADC in; can be used as GPIO
11	A7	Analog	ADC in; can be used as GPIO
12	VUSB	Power In/Out	Normally NC; can be connected to VUSB pin of the USB connector by shorting a jumper
13	RST	Digital In	Active low reset input (duplicate of pin 18)
14	GND	Power	Power Ground



Pin	Function	Type	Description
15	VIN	Power In	Vin Power input
16	TX	Digital	USART TX; can be used as GPIO
17	RX	Digital	USART RX; can be used as GPIO
18	RST	Digital	Active low reset input (duplicate of pin 13)
19	GND	Power	Power Ground
20	D2	Digital	GPIO
21	D3/PWM	Digital	GPIO; can be used as PWM
22	D4	Digital	GPIO
23	D5/PWM	Digital	GPIO; can be used as PWM
24	D6/PWM	Digital	GPIO, can be used as PWM
25	D7	Digital	GPIO
26	D8	Digital	GPIO
27	D9/PWM	Digital	GPIO; can be used as PWM
28	D10/PWM	Digital	GPIO; can be used as PWM
29	D11/MOSI	Digital	SPI MOSI; can be used as GPIO
30	D12/MISO	Digital	SPI MISO; can be used as GPIO

4.3 Debug

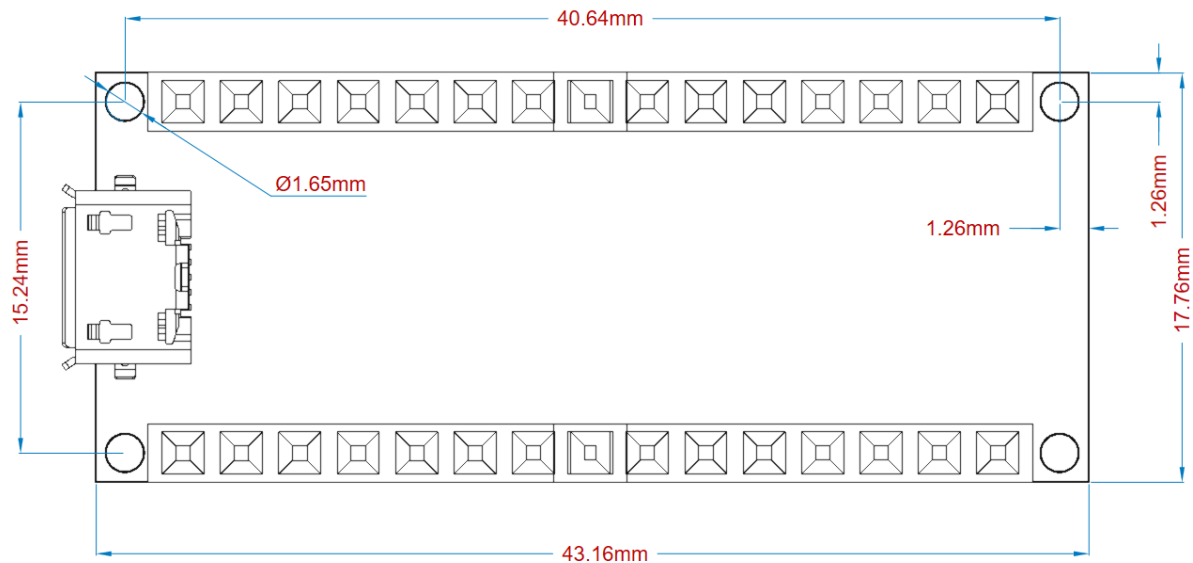
On the bottom side of the board, under the communication module, debug signals are arranged as 3x2 test pads with 100 mil pitch with pin 4 removed. Pin 1 is depicted in Figure 3 – Connector Positions

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

5 Mechanical Information

5.1 Board Outline and Mounting Holes

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



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).

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

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

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

Exemptions : No exemptions are claimed.

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



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

6.3 Conflict Minerals Declaration

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

7 FCC Caution

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

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

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

FCC RF Radiation Exposure Statement:

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

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

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

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

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

IC SAR Warning:

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

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

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

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

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



8 Company Information

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

9 Reference Documentation

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

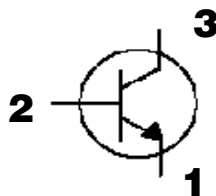
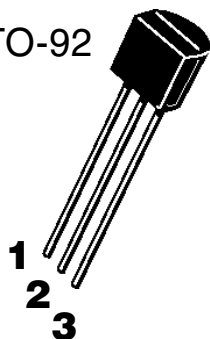
10 Revision History

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

Description

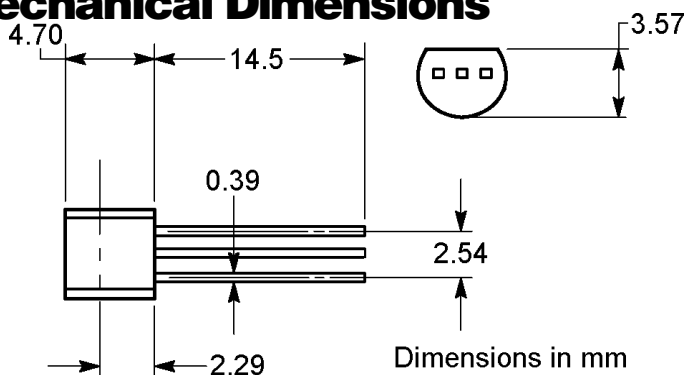
2N2222A

TO-92



NPN General Purpose Transistor

Mechanical Dimensions



Maximum Ratings

Ratings

	Symbol	Value	Units
Collector - Emitter Voltage	V_{CEO}	40	V
Collector - Base Voltage	V_{CBO}	75	V
Emitter - Base Voltage	V_{EBO}	6.0	V
Collector Current (Continuous)	I_C	600	mA
Total Device Dissipation $T_A = 25^\circ\text{C}$	P_D	625	mW
Junction and Storage Temperature	T_J, T_{STG}	-55 to 150	$^\circ\text{C}$

Electrical Characteristics

Characteristic

Characteristic	Symbol	Min	Max	Unit
Collector - Emitter Breakdown Voltage (Note 3) ($I_C = 10\text{mA}$)	$V_{BR(CEO)}$	40	---	V
Collector - Base Breakdown Voltage ($I_C = 10\mu\text{A}$)	$V_{BR(CBO)}$	75	---	V
Emitter - Base Breakdown Voltage ($I_E = 10\mu\text{A}$)	$V_{BR(EBO)}$	6.0	---	V
Base Cutoff Current ($V_{CB} = 60\text{V}$)	I_{CBO}	---	10	nA
Collector Cutoff Current ($V_{CE} = 60\text{V}, V_{EB(OFF)} = 3.0\text{V}$)	I_{CEX}	---	10	nA
Emitter Cutoff Current ($V_{EB} = 3.0\text{V}$)	I_{EBO}	---	10	nA
DC Current Gain ($I_C = 0.1\text{ mA}, V_{CE} = 10\text{ V}$) ($I_C = 1.0\text{ mA}, V_{CE} = 10\text{ V}$) ($I_C = 10\text{ mA}, V_{CE} = 10\text{ V}$) ($I_C = 150\text{ mA}, V_{CE} = 10\text{ V}$) ($I_C = 500\text{ mA}, V_{CE} = 10\text{ V}$)	H_{FE}	35 50 75 100 40	---	---
Collector - Emitter Saturation Voltage ($I_C = 150\text{ mA}, I_B = 15\text{ mA}$) ($I_C = 500\text{ mA}, I_B = 50\text{ mA}$)	$V_{CE(sat)}$	---	0.3 1.0	V
Base - Emitter Saturation Voltage ($I_C = 150\text{ mA}, I_B = 15\text{ mA}$) ($I_C = 500\text{ mA}, I_B = 50\text{ mA}$)	$V_{BE(sat)}$	---	1.2 2.0	V
Current - Gain - Bandwidth Product (Note 4) ($I_C = 20\text{ mA}, V_{CB} = 20\text{ V}, f = 100\text{ MHz}$)	f_T	300	---	MHz

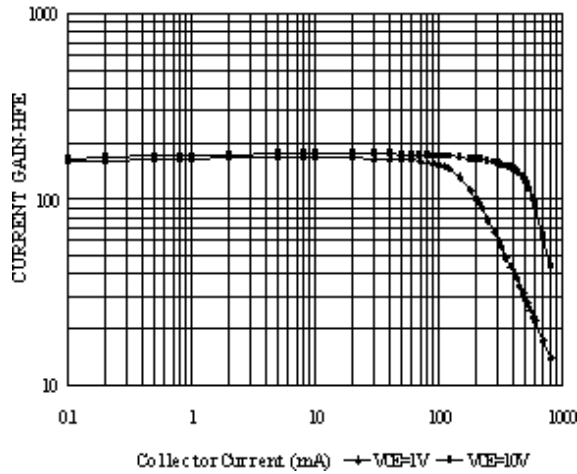
Classification of h_{FE4}

Rank	A	B
Range	100-210	190-300

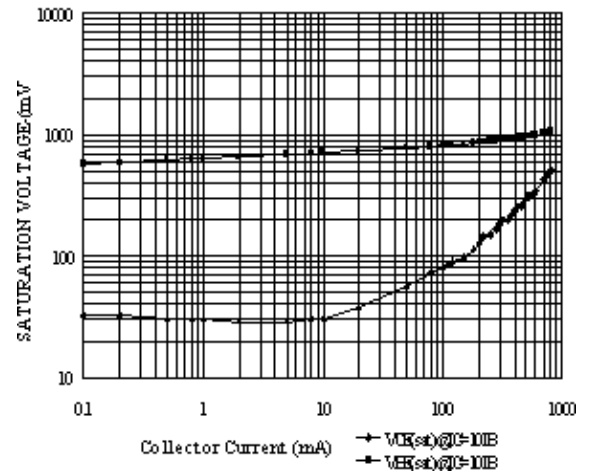


2N2222A NPN General Purpose Transistor

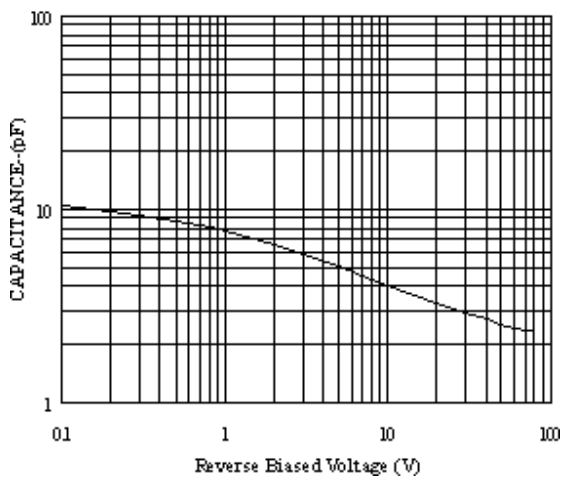
CURRENT GAIN VS. COLLECTOR CURRENT



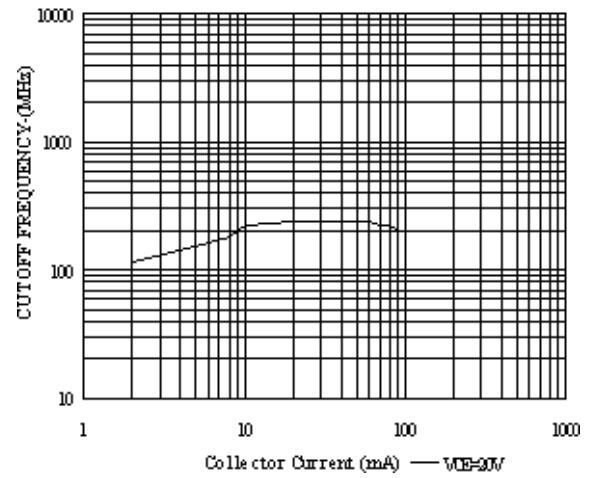
SATURATION VOLTAGE VS. COLLECTOR CURRENT



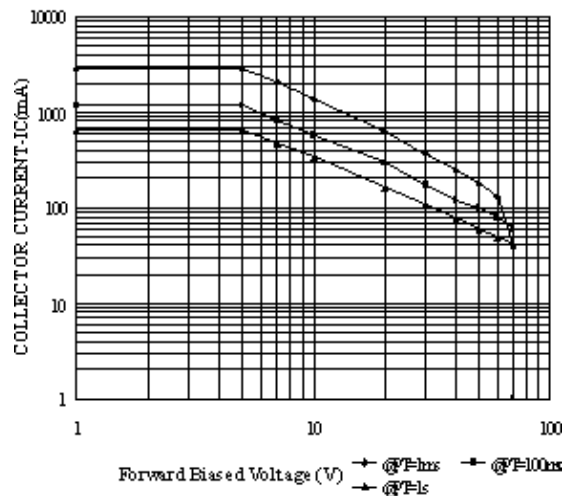
CAPACITANCE VS. REVERSE BIASED VOLTAGE



CUTOFF FREQUENCY VS. COLLECTOR CURRENT



SAFE OPERATING AREA



LTE12

Active Buzzer



SPECIFICATIONS:

Type	Unit	LTE12-03	LTE12-05	LTE12-12
Rated Voltage	V	3	5	12
Operating Voltage	V	2-5	4-8	8-15
*Rated Current(MAX)	mA	30	30	30
*Min Sound Output at 10cm	dB	80	85	85
*Resonant Frequency	Hz	2300±300		
Operating Temperature	°C	-20 ~ +70		
Storage temperature	°C	-30 ~ +105		

FREQUENCY RESPONSE

