

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

Informe Técnico Final:

Alarma láser con Arduino Nano BLE 33

Carlos Alberto Caravaca Mora - B61512.

Erick Sancho Alvarado - B87388.

Profesor: MSc. Marco Villalta Fallas.

4 de octubre, 2023

Índice

| | |
|--|-----------|
| 1. Introducción. | 1 |
| 2. Objetivos y alcances | 1 |
| 2.1. Objetivo General | 1 |
| 2.2. Objetivos Específicos | 1 |
| 2.3. Alcances. | 1 |
| 3. Justificación | 2 |
| 4. Marco Teórico. | 3 |
| 4.1. Arduino Nano 33 BLE | 3 |
| 5. Descripción de la aplicación. | 4 |
| 5.1. Láser de Baja Potencia: | 4 |
| 5.2. Espejos Reflectores: | 4 |
| 5.3. Sensor de Interrupción de Láser: | 4 |
| 5.4. Arduino Nano BLE 33: | 5 |
| 5.5. Buzzer de 5V: | 5 |
| 5.6. Batería de Bajo Consumo: | 5 |
| 5.7. Precios | 6 |
| 5.8. Funcionamiento: | 6 |
| 6. Análisis de resultados. | 7 |
| 7. Metodología | 8 |
| 8. Cronograma | 9 |
| 9. Conclusiones, observaciones y recomendaciones. | 9 |
| 10. Anexos. | 11 |

Índice de figuras

| | | |
|----|--|---|
| 1. | Topología de la placa. | 3 |
| 2. | Laser de baja potencia [1] | 4 |
| 3. | Espejos [1] | 4 |
| 4. | Sensor de Interrupción de Láser. Tomado de [1] | 4 |
| 5. | Sensor de Interrupción de Láser. Tomado de [1] | 5 |
| 6. | Buzzer. Tomado de [1] | 5 |
| 7. | Batería. Tomado de [1] | 5 |
| 8. | Circuito realizado. | 7 |

1. Introducción.

Esta propuesta busca desarrollar un proyecto enfocado en la creación de una alarma con láser de baja potencia que envíe notificaciones a telegram y whatsapp, utilizando el Arduino Nano BLE 33, un buzzer de 5V, LEDs y una batería de bajo consumo. Esta aplicación tiene el potencial de ejemplificar un método de seguridad real. A lo largo de la propuesta, se abordarán aspectos como la descripción de la aplicación, su justificación, objetivos y alcances, la metodología a emplear, incluyendo el lenguaje de programación y se presentará un cronograma para la ejecución del proyecto. Este laboratorio no solo promoverá el aprendizaje y la investigación de microcontroladores, sino que también tiene aplicaciones prácticas en el campo de la seguridad.

Nota: el proceso de este proyecto se puede ver evidenciado en el siguiente repositorio de git: https://github.com/Carlos-Caravaca/IE0624_B61512.

2. Objetivos y alcances

2.1. Objetivo General

- Desarrollar una alarma con láser de baja potencia utilizando componentes electrónicos y un microcontrolador llamado Arduino.

2.2. Objetivos Específicos

- Implementar un circuito funcional que garantice la seguridad y buen uso de cada uno de los componentes.
- Desarrollar un código en C++ que permita comunicar la señales del microcontrolador Arduino Nano BLE 33 con cada uno de los componentes físicos utilizados en el circuito y que el mismo envíe notificaciones a telegram y whatsapp cuando se activa la alarma.
- Demostrar una funcionalidad exitosa de la alarma láser mediante el diseño del circuito comunicándose con el microcontrolador, mediante los conocimientos y destrezas adquiridas en el curso.

2.3. Alcances.

El alcance principal de este proyecto se enfoca en crear una alarma de seguridad utilizando tecnología láser de baja potencia y el Arduino Nano BLE 33 como componente central. Por otro lado, se busca crear un diseño único mediante la utilización de diferentes componentes electrónicos que promuevan la innovación en la seguridad real utilizando microcontroladores.

Ademas, los alcances de este proyecto tienden a plasmar en la vida real los conceptos o principios estudiados en el curso, como lo son la programación de microcontroladores, diseño de circuitos e implementación de la combinación de ambos que ha sido el tema base durante el curso. Por otro lado, se aspira a desarrollar una alarma que permita enviar notificaciones al whatsapp y telegram del usuario cada vez que la alarma es activada.

3. Justificación

La justificación de esta aplicación se basa en la necesidad de contar con sistemas de seguridad asequibles y personalizables que utilicen tecnología accesible para una amplia gama de usuarios. El uso de un láser de baja potencia garantiza la seguridad al evitar riesgos para la vista humana, mientras que el Arduino Nano BLE 33 y otros componentes permiten una programación flexible y una integración sencilla en diferentes contextos.

Los objetivos de la aplicación incluyen la creación de un dispositivo de alarma que pueda detectar interrupciones en el haz de láser y activar una alarma sonora a través del buzzer, así como una señal visual a través de los LEDs. Además, se busca maximizar la eficiencia energética mediante el uso de una batería de bajo consumo, lo que garantiza una vida útil prolongada del sistema sin necesidad de frecuentes recargas o reemplazos de batería.

El alcance de la aplicación abarca desde la conceptualización y diseño del sistema hasta la programación y prueba en situaciones de la vida real. Se espera que el resultado final sea un dispositivo funcional y versátil que pueda ser utilizado para mejorar la seguridad en una variedad de escenarios, como la protección de objetos de valor, la detección de intrusiones en áreas específicas o la creación de un sistema de alerta temprana, una aplicación sencilla es la utilización de esta alarma en la entrada de un cuarto o habitación, por ejemplo.

La metodología a emplear en el desarrollo de esta aplicación incluye el uso de programación en el lenguaje de Arduino (C++), que es ampliamente accesible y bien documentado. Además, se llevarán a cabo pruebas exhaustivas para asegurar el correcto funcionamiento del sistema y se realizarán ajustes según sea necesario.

El cronograma para la ejecución del proyecto se establecerá de acuerdo con las fases de desarrollo, pruebas y ajustes necesarios, con el objetivo de lograr un producto final confiable y eficaz.

5. Descripción de la aplicación.

Esta innovadora aplicación tiene como objetivo desarrollar un sistema de seguridad altamente efectivo que utiliza la tecnología láser de baja potencia, Arduino Nano BLE 33, y espejos para proporcionar una amplia cobertura de detección en diversas situaciones de seguridad. Aquí presentamos los componentes clave y su funcionalidad:

5.1. Láser de Baja Potencia:

Emite un haz de láser de baja potencia hacia los espejos, su diseño compacto y eficiente es ideal para aplicaciones de seguridad.



Figura 2: Láser de baja potencia [1]

5.2. Espejos Reflectores:

Redirigen el haz láser a diferentes áreas, creando múltiples líneas de detección. Aumentan significativamente la cobertura de detección del sistema de seguridad.

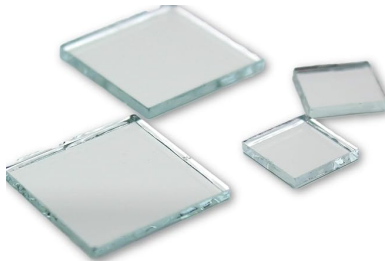


Figura 3: Espejos [1]

5.3. Sensor de Interrupción de Láser:

Detecta interrupciones en cualquiera de las líneas de láser reflejadas por los espejos. Envía una señal al Arduino Nano BLE 33 cuando se interrumpe cualquiera de las líneas láser.



Figura 4: Sensor de Interrupción de Láser. Tomado de [1]

5.4. Arduino Nano BLE 33:

El cerebro central del sistema de seguridad. Controla y coordina todas las operaciones del sistema. Ofrece conectividad Bluetooth Low Energy (BLE) para supervisión y control remotos.

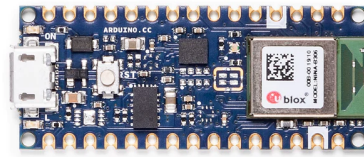


Figura 5: Sensor de Interrupción de Láser. Tomado de [1]

5.5. Buzzer de 5V:

Emite una alarma sonora cuando se detecta una intrusión en cualquiera de las líneas láser. Proporciona una alerta audible efectiva.



Figura 6: Buzzer. Tomado de [1]

5.6. Batería de Bajo Consumo:

Suministra energía al sistema de seguridad de manera eficiente. Asegura un funcionamiento prolongado sin necesidad de recargas frecuentes.



Figura 7: Batería. Tomado de [1]

5.7. Precios

| Componente | Precio en colones |
|------------|-------------------|
| Láser | 3727,14 |
| Buzzer | 1582,02 |
| Espejos | 8044,19 |
| Sensor | 1190 |
| Batería | 2654 |
| Conector | 600 |
| Arduino | 23596,30 |

5.8. Funcionamiento:

El sistema de seguridad emplea múltiples espejos reflectores para crear líneas de láser en diferentes áreas. Cuando cualquiera de estas líneas de láser se interrumpe debido a movimientos no deseados, el sensor de interrupción de láser envía una señal al Arduino Nano BLE 33, que activa una respuesta de alarma. El buzzer de 5V emite una alarma sonora y los LEDs indicadores proporcionan una señal visual para alertar al usuario sobre la intrusión.

Este sistema es altamente versátil y se adapta a una amplia gama de situaciones de seguridad. Los espejos reflectores permiten una cobertura eficiente y personalizable, mientras que el diseño de baja potencia garantiza una larga vida útil de la batería.

6. Análisis de resultados.

Luego de realizar el circuito del sistema, se obtiene una alarma simple pero efectiva, el circuito desarrollado se muestra de la siguiente manera:

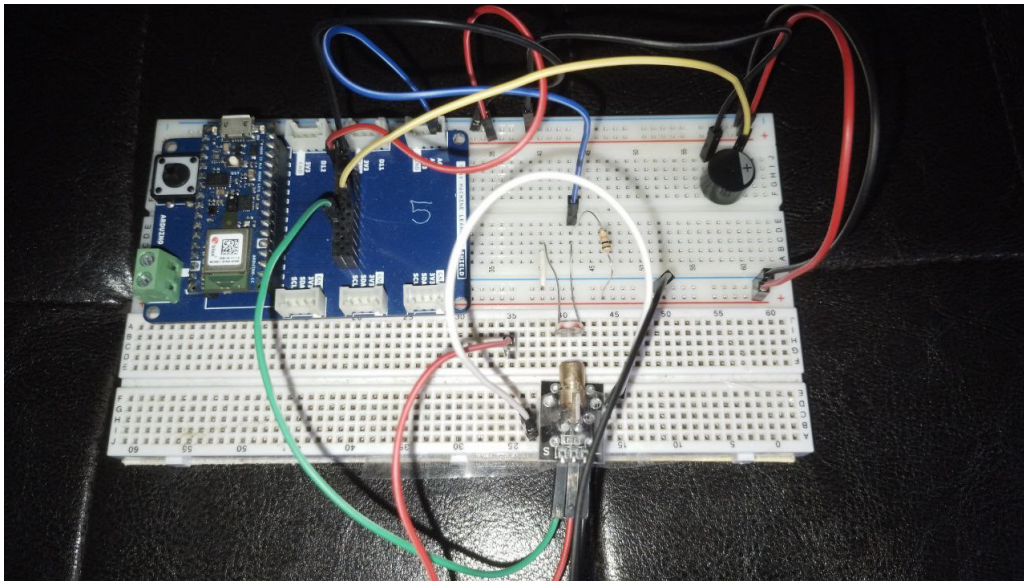


Figura 8: Circuito realizado.

Se observa que el láser de baja potencia emite un haz de luz que se divide en múltiples líneas de detección, ubicado frente al sensor de interrupción de láser que básicamente es una fotoresistencias que detecta cuando una de las líneas de detección se interrumpe. Por otro lado, el cerebro de todo el circuito es el Arduino Nano BLE 33 que permite controlar y coordinar el funcionamiento del sistema de alarma. El mismo se encarga de activar sonoramente el buzzer de 5V el cual emite una señal sonora cuando se activa la alarma. La comprobación de este funcionamiento se ve demostrado en el vídeo incluido en el repositorio de GitHub.

7. Metodología

Durante el desarrollo del código para el Arduino Nano 33 BLE se va a utilizar el lenguaje de programación de Arduino, que está basado en C/C++. Más específicamente, el Arduino Nano 33 BLE utiliza un dialecto simplificado de C/C++ que incluye algunas bibliotecas y funciones específicas para interactuar con el hardware de la placa, como sensores, actuadores y comunicación inalámbrica. Investigando un poco, habitualmente este lenguaje de Arduino utiliza las siguientes funciones básicas:

Funciones `setup()` y `loop()`: como es de costumbre, en general los programas de Arduino incluyen al menos dos funciones: `setup()` y `loop()`. La función `setup()` es ejecutada una vez al inicio del programa y se utiliza para realizar configuraciones iniciales, como la inicialización de pines o la configuración de parámetros. La función `loop()` es ejecutada cíclicamente y es el lugar donde se coloca el código principal del programa.

Por otro lado, sobre las **bibliotecas de Arduino**, este proporciona una serie de bibliotecas predefinidas que facilitan la interacción con sensores, actuadores y otros componentes de hardware comunes. Las cuales cuentan con una sintáxis simplificada, donde se facilitan ciertas complejidades de programación con el lenguaje C/C++ estándar, especialmente para principiantes. Por ejemplo, no es necesario declarar explícitamente el tipo de función `setup()` y `loop()`.

Por otro lado, las hay librerías específicas [2] para el Arduino Nano 33 BLE que permiten el acceso a las capacidades Bluetooth Low Energy (BLE) del microcontrolador nRF52840 que integra esta placa. Estas librerías facilitan la comunicación inalámbrica BLE y la interacción con otros dispositivos compatibles con BLE. Se va a trabajar de la mano con el entorno Arduino donde se puede ampliar mediante el uso de bibliotecas, al igual que la mayoría de las plataformas de programación. Las bibliotecas proporcionan funcionalidad adicional para usar en bocetos, por ejemplo, trabajar con hardware o manipular datos. Además el entorno arduino, tiene un software que facilita cargarle el firmware al microcontrolador, entonces será de gran utilidad para el desarrollo del proyecto y su funcionamiento.

8. Cronograma

| Fecha. | Actividad. | Responsables. |
|-----------|---|-----------------------------------|
| 30/09/23 | Inicio e investigación sobre el proyecto a realizar. Escogencia del tema de proyecto. | Erick Sancho Carlos Caravaca |
| 04/10/23 | Presentación del avance de proyecto. | Carlos Caravaca |
| 14/10/23 | Investigar sobre costos y precios de los componentes. Conseguir componentes que estén a nuestro alcance. | Carlos Caravaca Erick Sancho |
| 21/10/23 | Comprar los componentes que vamos a necesitar. | Carlos Caravaca y Erick Sancho |
| 28/10/23 | Comenzar con el diseño del circuito a implementar. Realizar pruebas simuladas y físicas con el arduino. | Carlos Caravaca Erick Sancho |
| 04/11/23 | Iniciar la programación del proyecto. Iniciar diseño del circuito. | Erick Sancho Carlos Caravaca |
| 11/11/23 | Continuar la programación y enseñar avances. | Erick Sancho y Carlos Caravaca |
| 18/11/23 | Iniciar el informe técnico final. Continuar con la programación. | Erick Sancho Carlos Caravaca |
| 25/11/23 | Afinar detalles del proyecto y terminar la programación. Realizar presentación final. Enseñarle avance al profesor. | Erick Sancho Carlos Caravaca |
| Diciembre | Terminar pendientes del proyecto. (Si los hay) Presentación final del proyecto. | Carlos Caravaca Erick Sancho |

Tabla 1: Tabla con el cronograma.

9. Conclusiones, observaciones y recomendaciones.

Se logra desarrollar eficazmente el funcionamiento de una alarma láser que garantizando la seguridad y buen uso de cada uno de los componentes. El sistema de alarma es una forma eficaz de proteger a las personas, los bienes y la información de posibles intrusiones o amenazas, innovando mediante el uso de un láser de baja potencia para crear líneas de detección que se pueden interrumpir por un movimiento no deseado, cuando se interrumpe una línea de detección, se activa una alarma sonora y visual. El sistema de alarma es eficiente energéticamente y tiene una larga vida útil gracias al láser de baja potencia que utiliza una pequeña cantidad de energía, lo que permite al sistema funcionar durante un período prolongado con una sola batería, por último el sistema también está diseñado para ser fácil de mantener, lo que reduce los costos de operación.

Para mejorar el proyecto, se podrían realizar las siguientes recomendaciones:

- Agregar más líneas de detección lo cual aumentaría la cobertura del sistema de alarma y lo haría más difícil de evadir.
- Utilizar un láser de mayor potencia ya que brindaría un sistemas más sensible que detecte intrusiones más pequeñas.
- Agregar un sistema de videovigilancia ya que permitiría a los usuarios ver quién o qué ha activado la alarma.

Referencias

- [1] microjpm.com. Tienda de componentes electronicos. <https://www.microjpm.com/>.
- [2] Arduino. Libraries. <https://www.arduino.cc/reference/en/libraries/>.

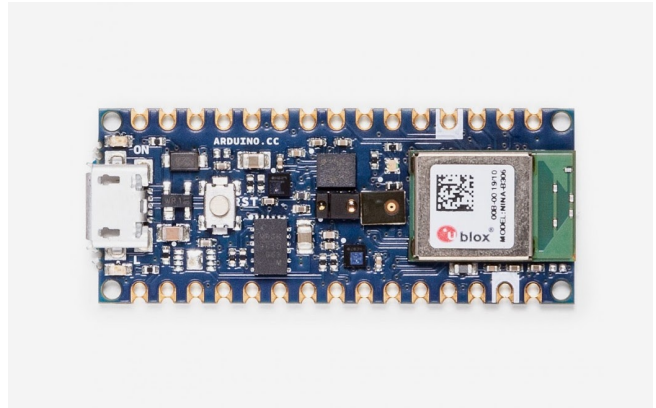
10. Anexos.

```
1 # Codigo realizado para el arduino
2
3 /*
4  * Proyecto
5  * Curso: Laboratorio de Microcontroladores
6  * Estudiantes:
7  * Carlos A. Caravaca Mora - B61512
8  * Erick Sancho Alvarado - B87388
9  * Ciclo: II
10 * Ano: 2023
11 * Descripcion: Alarma con modulo laser
12 */
13
14 // variables para los pines de cada componente
15 int buzzer = 6;    //D6
16 int laser = 2;     //D2
17 int fotoresistencia = A6;    //A6
18 int estado_fotore = 2;
19 int estado;
20
21 void setup(){
22     // se inicializan como entrada o salida
23     pinMode(buzzer, OUTPUT);
24     pinMode(laser, OUTPUT);
25     pinMode(fotoresistencia, INPUT);
26     Serial.begin(9600); // Inicia la comunicaci n serial
27 }
28 void loop(){
29     // lectura de la fotoresistencia
30     estado_fotore = analogRead(fotoresistencia);
31     Serial.println("Lectura de fotoresistencia: " + String(estado_fotore
32 ));
33     digitalWrite(laser,HIGH);
34
35     if (estado_fotore > 400){
36         // interrupcion de laser
37         estado = 1;
38     }else{
39         // no interrupcion de laser
40         estado = 2;
41     }
42     switch (estado) {
43         case 1:{
44             unsigned char i;
45             int x = 1;
46             while(x < 10)
47             {
48                 //salida de frecuencia uno
49                 for(i=0;i<80;i++)
50                 {
51                     digitalWrite(buzzer,HIGH);
```

```

51         delay(1); //wait for 1ms
52         digitalWrite(buzzer, LOW);
53         delay(1); //wait for 1ms
54     }
55     //salida de frecuencia dos
56     for(i=0; i<100; i++)
57     {
58         digitalWrite(buzzer, HIGH);
59         delay(2); //wait for 2ms
60         digitalWrite(buzzer, LOW);
61         delay(2); //wait for 2ms
62     }
63     x++;
64 }
65 }
66     break;
67 case 2:
68     // no hace nada, se sale y continua el bucle
69     break;
70 default:
71     break;
72 }
73 }

```



Description

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

Target areas:

Maker, enhancements, IoT application



Features

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



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



Contents

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



1 The Board

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

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

1.1 Ratings

1.1.1 Recommended Operating Conditions

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

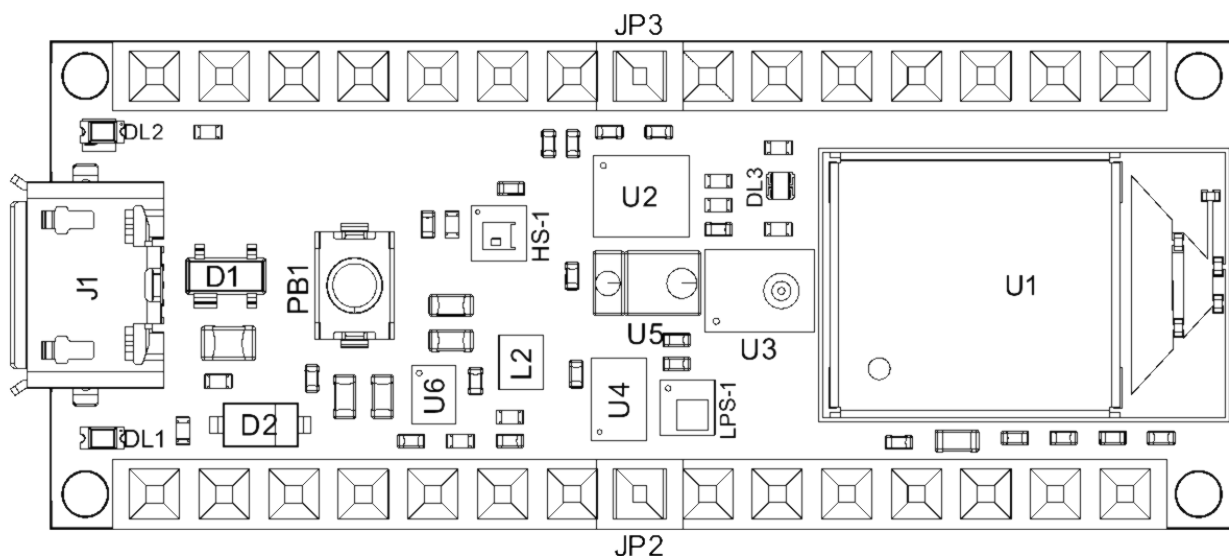
1.2 Power Consumption

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

2 Functional Overview

2.1 Board Topology

Top:



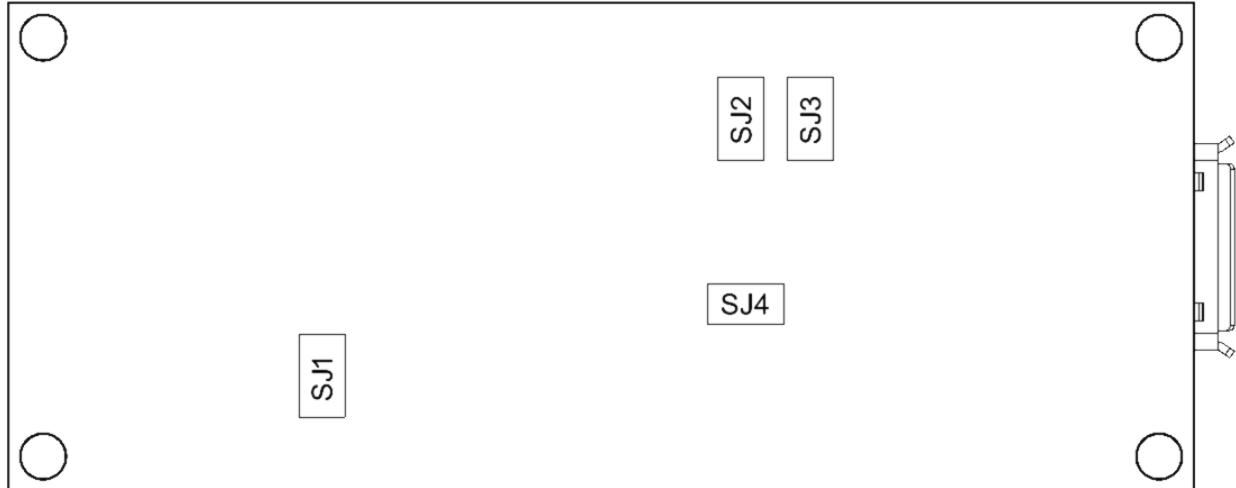
Board topology top

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



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

Bottom:



Board topology bot

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

2.2 Processor

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

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

2.3 Crypto

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

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



2.4 IMU

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

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

2.5 Barometer and Temperature Sensor

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

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

2.6 Relative Humidity and Temperature Sensor

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

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

2.7 Digital Proximity, Ambient Light, RGB and Gesture Sensor

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

2.7.1 Gesture Detection

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

2.7.2 Proximity Detection

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



2.7.3 Color and ALS Detection

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

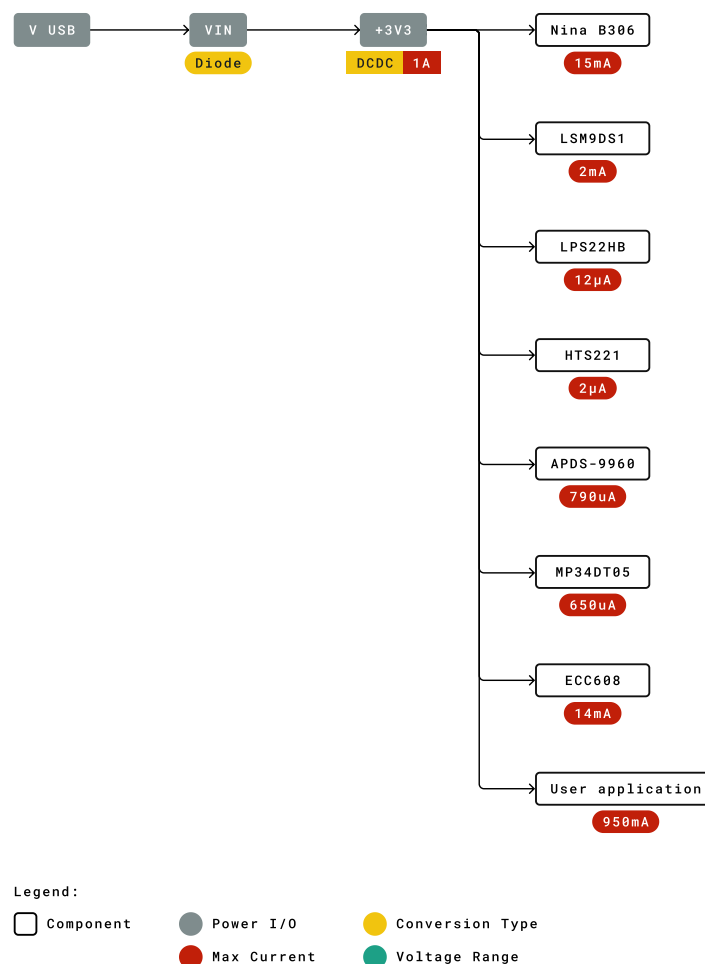
2.8 Digital Microphone

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

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

2.9 Power Tree

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



Power tree

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



3 Board Operation

3.1 Getting Started - IDE

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

3.2 Getting Started - Arduino Web Editor

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

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

3.3 Getting Started - Arduino IoT Cloud

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

3.4 Sample Sketches

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

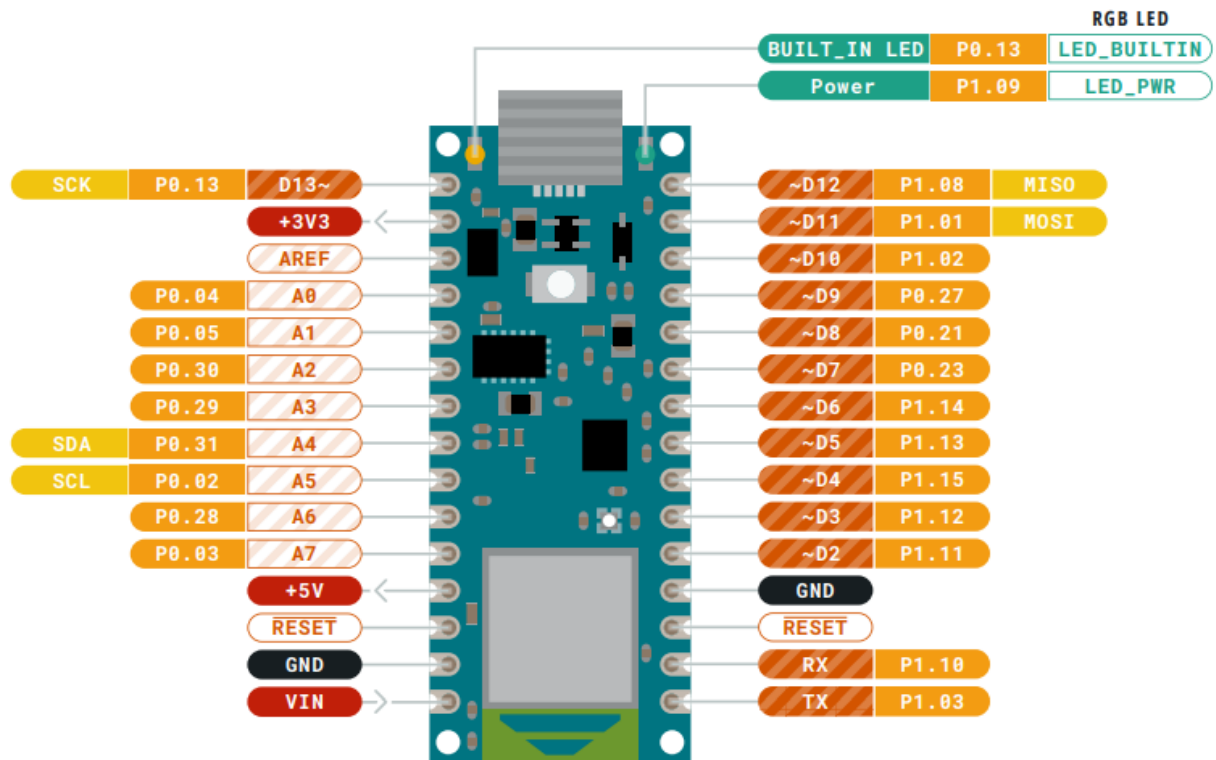
3.5 Online Resources

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

3.6 Board Recovery

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

4 Connector Pinouts



Pinout

4.1 USB

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

4.2 Headers

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

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



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

4.3 Debug

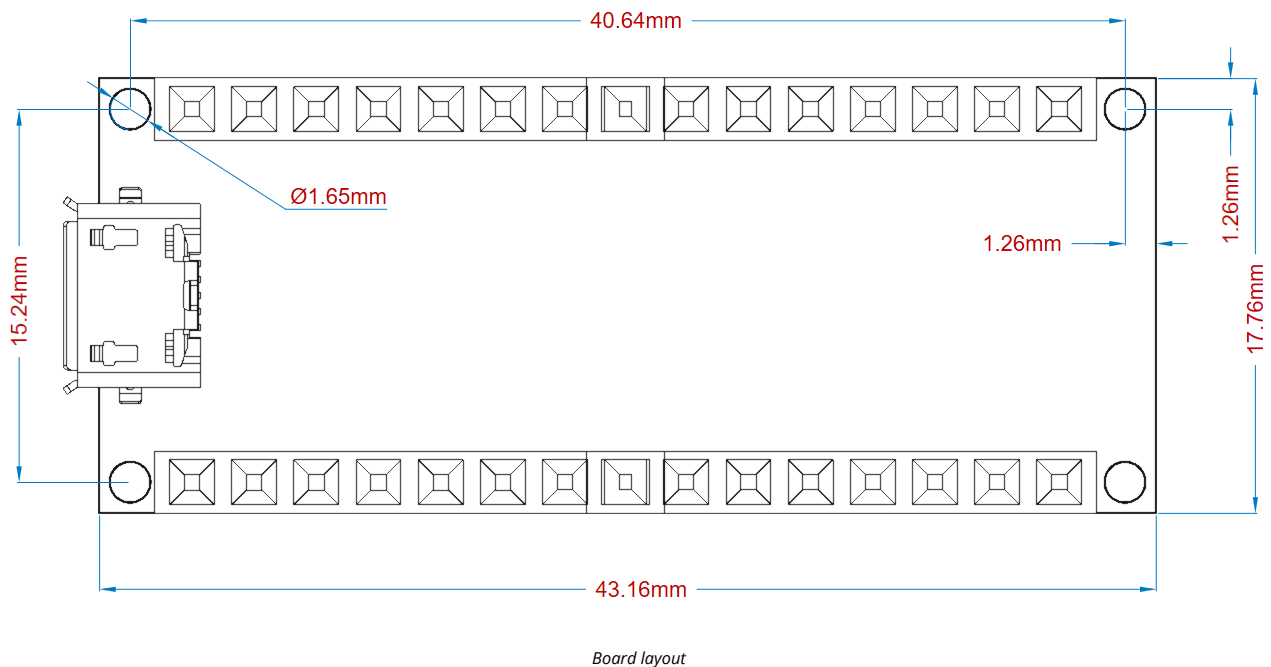
On the bottom side of the board, under the communication module, debug signals are arranged as 3x2 test pads with 100 mil pitch with pin 4 removed. Pin 1 is depicted in Figure 3 – Connector Positions

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

5 Mechanical Information

5.1 Board Outline and Mounting Holes

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



6 Certifications

6.1 Declaration of Conformity CE DoC (EU)

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

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

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

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

Exemptions : No exemptions are claimed.

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



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

6.3 Conflict Minerals Declaration

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

7 FCC Caution

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

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

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

FCC RF Radiation Exposure Statement:

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

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

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

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

- (1) l'appareil n'a pas 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 |