

Home (/) / Microcontroller Based Projects (/microcontroller-projects)  
/ Arduino Nano 33 BLE Sense Review - What's New and How to Get Started?

## Arduino Nano 33 BLE Sense Review - What's New and How to Get Started?

Published November 8, 2019

4



Aswinth Raj (/users/aswinth-raj)

Author



Arduino Nano 33 BLE Sense Board

Arduino has been the go-to development platform for **fast prototyping** and idea validation. Many of us would have started with the Arduino UNO development board (<https://circuitdigest.com/microcontroller-projects/getting-started-with-arduino-uno-blinking-led>), but today as we progress towards the Internet of things (<http://circuitdigest.com/internet-of-things-iot-projects>), Computer Vision (<https://circuitdigest.com/tags/opencv>), Artificial Intelligence (<https://circuitdigest.com/tags/ai>), Machine learning, and other futuristic technologies, the humble Arduino UNO could no longer cope up with its 8-bit Microcontroller. This called for the launch of new boards with more powerful processors that have built-in Wi-Fi, Bluetooth, GSM and other wireless capabilities, like the popular **MKR1000** or **MKR GSM 1400**. In this context, Arduino has recently launched a new version of its Nano called the **Arduino Nano 33**.

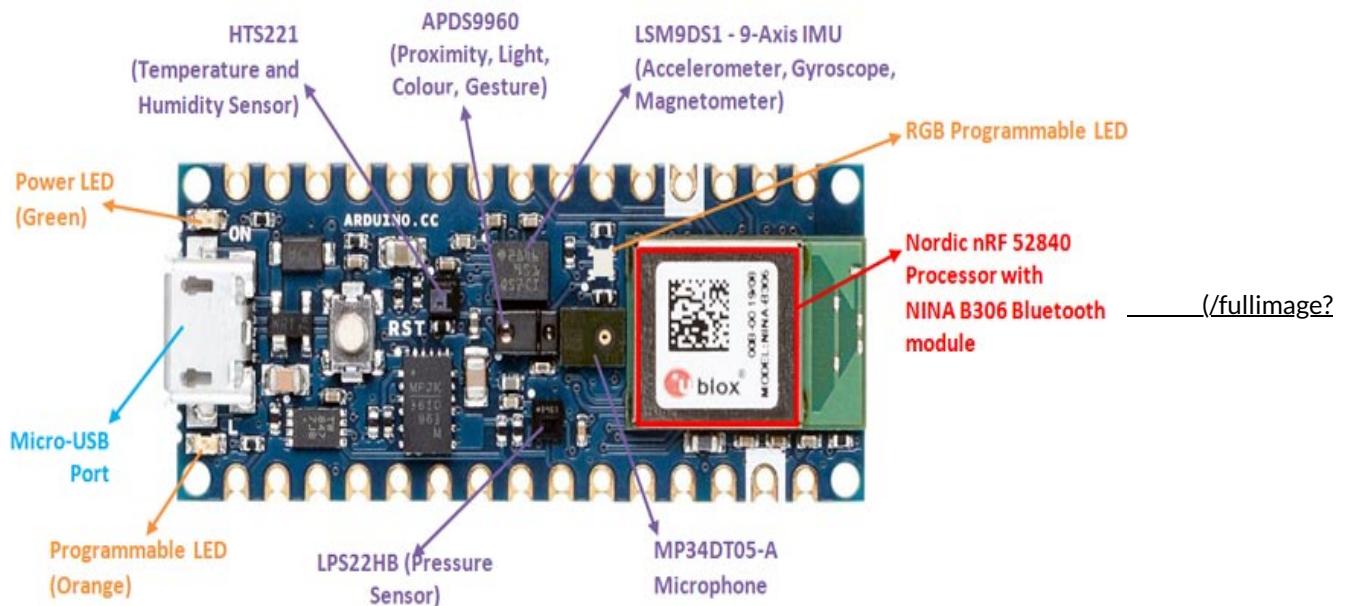
There are totally two types of Arduino Nano 33 boards, namely Arduino Nano 33 IoT and Arduino Nano 33 BLE sense. The main difference between both the modules is that the **Arduino Nano 33 BLE sense** module has some built-in sensors (will get into details later) while the Arduino Nano 33 IoT does not have them. In this article we will review the Arduino Nano 33 BLE sense board, introduce you to its features and functionalities and finally write a sample code to read the sensor values and display on serial monitor. So let's get learning....!

## Arduino Nano 33 BLE Sense

The name “Arduino Nano 33 BLE Sense” is mouthful, but the name itself puts out some important information. It is called “Nano” because the dimensions, pinout and form factor is very similar to the classic [Arduino Nano](#) (<https://circuitdigest.com/arduino-nano-projects>), it is actually planned to be used as replacement for Arduino Nano in your existing projects, but the catch is that this new module **operates on 3.3V** while the classic Nano operates on 5V. So I think this is where the name “33” comes in, to indicate that the board operates on 3.3V. Then the name “[BLE](#) (<https://circuitdigest.com/tags/ble>)” indicates that the module supports **Bluetooth Low Energy (BLE5.0)** and the name “sense” indicates that it has on-board sensors like accelerometer, gyroscope, magnetometer, temperature and humidity sensor, Pressure sensor, Proximity sensor, Colour sensor, Gesture sensor, and even a built-in microphone. We will get into the details of BLE and other sensors later but for now this is how an Arduino Nano 33 BLE sense board looks straight out of un-boxing.

## Arduino Nano 33 BLE Sense Hardware Overview

At the first look of the board you can find a lot of components crowded on the top, most of which are sensors that I told earlier. But the main brain is hidden behind the metal casing on the right side. This casing holds the **Nordic nRF52840 processor** that contains a powerful Cortex M4F and the **NINA B306** module for BLE and Bluetooth 5 communication. This allows the board to operate on very low power and communicate using Bluetooth 5 making it ideal for **low power mesh network applications** in [home automation](#) (<https://circuitdigest.com/home-automation-projects>), and other connected projects. Also since the nRF processor supports the **ARM Mbed OS** it also provides some software improvements which we will discuss later. The sensors, LEDs, pushbuttons and other important stuff that you should know on your board are marked in the below image.



<https://circuitdigest.com/inlineimages/u2/Arduino-Nano-33-BLE-Sense-Hardware-Overview.jpg>

As you can see from the above image, the board is power-packed with sensors that can help you in building the right of the box without even having to connect the board to any external sensors. The board is aimed to be used in **wearable devices** and other smart portable devices like Fitness bands, Glucose monitoring, Pedometers, smartwatch, weather station, Home security etc where you will be using most of these sensors. And like always all these sensors have pre-built

libraries for Arduino which you can use readily. At the end of this article we will read values from all these sensors and display it on the serial monitor. The sensor details on Arduino Nano 33 BLE sense board along with its required libraries are tabulated below

Sensor Name	Parameters	Links
LSM9DS1 - ST Microelectronics	Accelerometer, Gyroscope, Magnetometer	<a href="https://www.st.com/resource/en/datasheet/DM00103319.pdf">LSM9DS1 Datasheet (<a href="https://www.st.com/resource/en/datasheet/DM00103319.pdf">https://www.st.com/resource/en/datasheet/DM00103319.pdf</a>)</a> <a href="https://github.com/arduino-libraries/Arduino_LSM9DS1">Arduino LSM9DS1 Library (<a href="https://github.com/arduino-libraries/Arduino_LSM9DS1">https://github.com/arduino-libraries/Arduino_LSM9DS1</a>)</a>
LPS22HB - ST Microelectronics	Pressure	<a href="https://www.st.com/resource/en/datasheet/lps22hb.pdf">LPS22HB Datasheet (<a href="https://www.st.com/resource/en/datasheet/lps22hb.pdf">https://www.st.com/resource/en/datasheet/lps22hb.pdf</a>)</a> <a href="https://github.com/arduino-libraries/Arduino_LPS22HB">Arduino LPS22HB Library (<a href="https://github.com/arduino-libraries/Arduino_LPS22HB">file:///D:/30%20oct/New%20folder/upload-%20title%20image/LPS22HB%20Datasheet</a>)</a>
HTS221 - ST Microelectronics	Temperature and Humidity	<a href="https://www.st.com/resource/en/datasheet/hts221.pdf">LPS22HB Datasheet (<a href="https://www.st.com/resource/en/datasheet/hts221.pdf">https://www.st.com/resource/en/datasheet/hts221.pdf</a>)</a> <a href="https://github.com/arduino-libraries/Arduino_HTS221">Arduino HTS221 Library (<a href="https://github.com/arduino-libraries/Arduino_HTS221">https://github.com/arduino-libraries/Arduino_HTS221</a>)</a>
APDS9960 - Avago Tech.	Proximity, Light, Colour, Gesture	<a href="https://components101.com/sites/default/files/component_datasheet/APDS9960.pdf">LPS22HB Datasheet (<a href="https://components101.com/sites/default/files/component_datasheet/APDS9960.pdf">https://components101.com/sites/default/files/component_datasheet/APDS9960.pdf</a>)</a> <a href="https://github.com/arduino-libraries/Arduino_APDS9960">Arduino APDS9960 Library (<a href="https://github.com/arduino-libraries/Arduino_APDS9960">https://github.com/arduino-libraries/Arduino_APDS9960</a>)</a>
MP34DT05 - ST Microelectronics	Microphone	<a href="https://www.st.com/resource/en/datasheet/mp34dt05-a.pdf">MP34DT05 Datasheet (<a href="https://www.st.com/resource/en/datasheet/mp34dt05-a.pdf">https://www.st.com/resource/en/datasheet/mp34dt05-a.pdf</a>)</a> <a href="https://github.com/arduino/ArduinoCore-nRF528x-mbedos/tree/master/libraries/PDM">Inbuilt-PDM Library (<a href="https://github.com/arduino/ArduinoCore-nRF528x-mbedos/tree/master/libraries/PDM">https://github.com/arduino/ArduinoCore-nRF528x-mbedos/tree/master/libraries/PDM</a>)</a>

Most of these sensors are from **ST Microelectronics** and they support low power operation making it ideal for battery operated designs. Few people might already be familiar with the APDS9960 sensor since it is already available as a spate module and we have also used the APDS9960 sensor with Arduino previously (<https://circuitdigest.com/microcontroller-projects/how-to-use-adps9960-rgb-and-gesture-sensor-with-arduino>). For more information on these sensors, you can visit the respective datasheet and also make sure you have added the entire provided library to your Arduino IDE to begin using them with your Arduino Nano 33 BLE sense board. To add a library you can use the given link to get to the respective GitHub page and download the ZIP file, then use Sketch -> Include Library -> Add.ZIP Library or you can also use the library manager on Arduino IDE and add these libraries.

### Arduino Nano 33 BLE sense Board Technical Specifications:

Powered by the [Nordic nRF52840 Processor \(\[https://infocenter.nordicsemi.com/pdf/nRF52840\\\_PS\\\_v1.1.pdf\]\(https://infocenter.nordicsemi.com/pdf/nRF52840\_PS\_v1.1.pdf\)\)](https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf), the Arduino Nano 44 BLE board has the following technical specifications

- Operating Voltage: 3.3V
- USB-Input Voltage: 5V

- Input-pin Voltage: 4.5V to 21V
- Chip: NINA-B3 – RF52840
- Clock: 64MHz
- Flash : 1MB
- SRAM: 256 KB
- Wireless Connectivity: Bluetooth 5.0 / BLE
- Interfaces: USB, I2C, SPI, I2S, UART
- Digital I/O Pins: 14
- PWM Pins: 6 (8-bit resolution)
- Analog Pins: 8 (10-bit or 12-bit configurable)

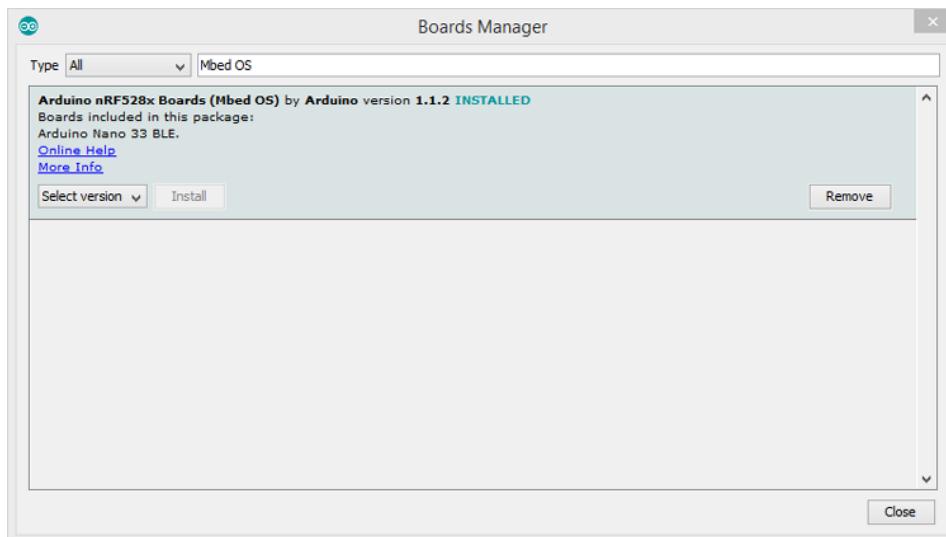
## Software improvements with Arduino Nano 33 BLE sense

Just like all Arduino boards out there, the Arduino Nano 33 BLE sense can be programmed with the Arduino IDE. But, you have to use the board manager and add the board details to your IDE before you can begin. As we know the nRF 52840 can be programmed using ARM Mbed OS, this means that our Arduino Nano 33 board supports Real-Time Operating System (RTOS). (<https://circuitdigest.com/tags/rtos>) With Mbed OS programming we can run multiple threads at the same time in the program to perform multi-tasking. Also, the power consumption of the board will be greatly reduced, every time we call the delay function the board will enter into **tickles mode** during the delay time to save power and would jump back into operation once the delay is over. It is reported that this operation will consume 4.5uA less than a normal Arduino delay operation.

That being said, the Mbed OS integration with Arduino IDE is relatively new and it is going to take some time before we can fully utilize the full power of Mbed OS with Arduino IDE. So for a quick start-up, we will write a program to read all the sensor's values and display it on the serial monitors.

## Preparing your Arduino IDE for Arduino Nano 33 BLE sense

Launch your Arduino IDE and go to Tools -> Boards -> Board Manager to launch your Arduino Board manager. Now search for “Mbed OS” and install the package. It should take some time for the installation to complete.



Once the installation is done, close the dialog box and connect your Arduino 33 board using a micro USB cable with your laptop. As soon as you connect the board windows will automatically start installing the required drivers for the board. Then open your Arduino IDE and select Tools -> Board -> Arduino Nano 33. Then also select the right COM port by checking Tools -> Port, mine is connected to port COM3 but yours might vary. After the port is selected your IDE bottom right corner should look like this

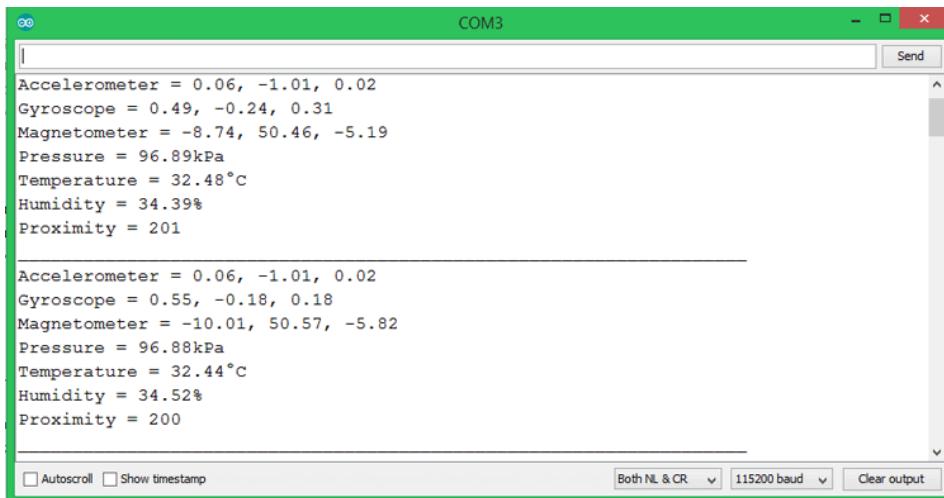


Now to quickly check if everything is working we can use an example program, let's try the one provided at File -> Examples -> PDM -> PDMSerialPlotter. This program will use the on-board microphone to listen for audio and plot it on a serial plotter. You can upload the program and check if the board and the IDE are working.

Now if you experience a ridiculously slow compiling then you are not alone, many people including me face this issue and at the time of writing this article, there seems to be no solution. It takes me around 2-3 minutes to compile and upload simple programs and when I tried some BLE programs or tried to work with Mbed OS the compilation time increased to more than 10 minutes which did not encourage me to try anything further. This is because of the **Mbed OS integration with Arduino IDE**, let's hope someone from the wonderful Arduino Community comes up with a solution for this.

## Program to read sensor data and display on Serial Monitor

If we do not use the BLE or core Mbed OS functionalities of the board the compile time was reasonable. So I wrote a simple sketch to read all the sensor values and display it on the serial monitor like show below



The complete code to do the same is given at the bottom of this page, but make sure you have installed all the libraries mentioned above. The explanation of the code is as follows.

Start the program by including all the required header files. Here we will be using all the four sensors except for the microphone

```
#include <Arduino_LSM9DS1.h> //Include the library for 9-axis IMU
#include <Arduino_LPS22HB.h> //Include library to read Pressure
#include <Arduino_HTS221.h> //Include library to read Temperature and Humidity
#include <Arduino_APDS9960.h> //Include library for colour, proximity and gesture recognition
```

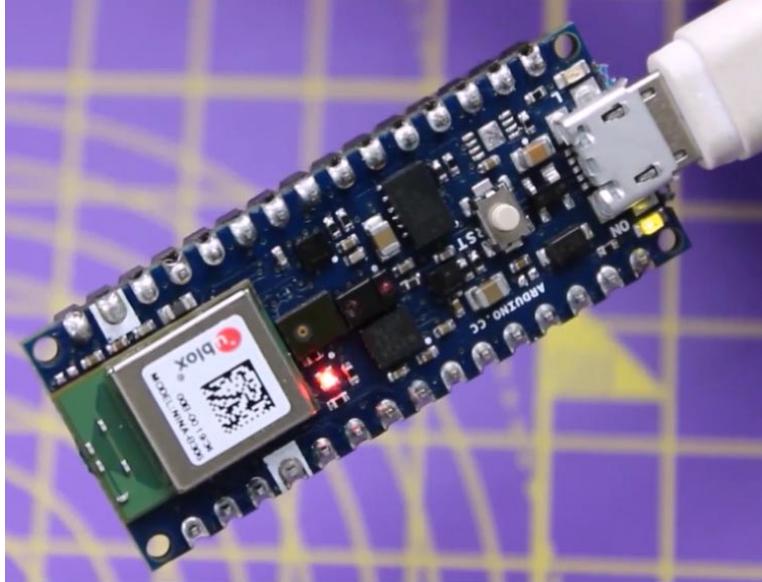
Inside the setup function we initialize the serial monitor at 9600 baud rate to display all the sensor values and also initialize all the required libraries. The code inside setup is shown below

```
void setup(){
    Serial.begin(9600); //Serial monitor to display all sensor values
    if (!IMU.begin()) //Initialize IMU sensor
    { Serial.println("Failed to initialize IMU!"); while (1);}
    if (!BARO.begin()) //Initialize Pressure sensor
    { Serial.println("Failed to initialize Pressure Sensor!"); while (1);}
    if (!HTS.begin()) //Initialize Temperature and Humidity sensor
    { Serial.println("Failed to initialize Temperature and Humidity Sensor!"); while (1);}
    if (!APDS.begin()) //Initialize Colour, Proximity and Gesture sensor
    { Serial.println("Failed to initialize Colour, Proximity and Gesture Sensor!"); while (1);}
}
```

Inside the loop function, we read the required sensor values from the library and then print them on the serial monitor. The syntax can be referred from the example program of each library, we have read the accelerometer, gyroscope, magnetometer, pressure, temperature, humidity and proximity sensor values and displayed them on the serial monitor. The code to measure the accelerometer value is shown below, likewise, we can measure for all sensors.

```
//Accelerometer values
if (IMU.accelerationAvailable()) {
    IMU.readAcceleration(accel_x, accel_y, accel_z);
    Serial.print("Accelerometer = ");Serial.print(accel_x); Serial.print(", ");Serial.print(accel_y);Serial.pr
}
delay (200);
```

## Arduino Nano 33 BLE- Uploading the code



Uploading the code to Nano 33 is similar to any other boards, but do note that the board has two COM ports. When you click on the upload button the Arduino IDE compiles the code and then resets the board automatically through software command, this will put the board in boot loader mode and upload your code. Because of this, once the upload is done you

might notice that the Arduino IDE has automatically changed its COM port to a different number and you might want to change it back before you open your serial monitor.

So this is pretty much my experience with the Arduino Nano 33 board so far, I will try building something with its sensors and BLE features sometime later in the future. How was your experience with the board? What would you want me to build with it? Leave the answers in the comment section and we will discuss more.

## Code

```
#include <Arduino_LSM9DS1.h> //Include the library for 9-axis IMU
#include <Arduino_LPS22HB.h> //Include library to read Pressure
#include <Arduino_HTS221.h> //Include library to read Temperature and Humidity
#include <Arduino_APDS9960.h> //Include library for colour, proximity and gesture recognition
void setup(){
    Serial.begin(9600); //Serial monitor to display all sensor values
    if (!IMU.begin()) //Initialize IMU sensor
    {
        Serial.println("Failed to initialize IMU!"); while (1);}
    if (!BARO.begin()) //Initialize Pressure sensor
    {
        Serial.println("Failed to initialize Pressure Sensor!"); while (1);}
    if (!HTS.begin()) //Initialize Temperature and Humidity sensor
    {
        Serial.println("Failed to initialize Temperature and Humidity Sensor!"); while (1);}
    if (!APDS.begin()) //Initialize Colour, Proximity and Gesture sensor
    {
        Serial.println("Failed to initialize Colour, Proximity and Gesture Sensor!"); while (1);}
    }
float accel_x, accel_y, accel_z;
float gyro_x, gyro_y, gyro_z;
float mag_x, mag_y, mag_z;
float Pressure;
float Temperature, Humidity;
int Proximity;
void loop()
{
    //Accelerometer values
    if (IMU.accelerationAvailable())
    {
        IMU.readAcceleration(accel_x, accel_y, accel_z);
    }
    //Proximity sensor values
    if (APDS.proximityAvailable())
    {
        APDS.readProximity(Proximity);
    }
    //Temperature and Humidity sensor values
    if (HTS.humidityAvailable())
    {
        HTS.readHumidity(Humidity);
    }
    //Pressure sensor values
    if (BARO.pressureAvailable())
    {
        BARO.readPressure(Pressure);
    }
    //Gesture sensor values
    if (APDS.gestureAvailable())
    {
        APDS.readGesture(Gesture);
    }
    //Print all sensor values to serial monitor
    Serial.print("Accel X: ");
    Serial.print(accel_x);
    Serial.print(" Accel Y: ");
    Serial.print(accel_y);
    Serial.print(" Accel Z: ");
    Serial.print(accel_z);
    Serial.print(" Gyro X: ");
    Serial.print(gyro_x);
    Serial.print(" Gyro Y: ");
    Serial.print(gyro_y);
    Serial.print(" Gyro Z: ");
    Serial.print(gyro_z);
    Serial.print(" Mag X: ");
    Serial.print(mag_x);
    Serial.print(" Mag Y: ");
    Serial.print(mag_y);
    Serial.print(" Mag Z: ");
    Serial.print(mag_z);
    Serial.print(" Pressure: ");
    Serial.print(Pressure);
    Serial.print(" Temperature: ");
    Serial.print(Temperature);
    Serial.print(" Humidity: ");
    Serial.print(Humidity);
    Serial.print(" Proximity: ");
    Serial.print(Proximity);
    Serial.println();
    delay(1000);
}
```

## Video

### Arduino Nano 33 BLE Sense Board Review



## Tags

[Arduino \(/Tags/Arduino\)](#)    [Arduino Nano 33 BLE Sense \(/Tags/Arduino-Nano-33-Ble-Sense\)](#)[Arduino Nano \(/Tags/Arduino-Nano\)](#)    [Review \(/Tags/Review\)](#)    [Getting Started \(/Tags/Getting-Started\)](#)

## Comments

Submitted by MJ (/users/mj) on Tue, 11/12/2019 - 06:42

[Permalink \(/comment/30765#comment-30765\)](#)

### Trouble connecting Nano BLE ([/comment/30765#comment-30765](#))

Thank you for this article. I would love to see more about the BLE features. I recently purchased one of these boards as my first entry into Arduino, and have been able to read sensor data as expected but can't seem to discover my Arduino bt peripheral from any of my devices (running Mac OSX, Windows, iOS). I'm using the ArduinoBLE library and everything seems to be initializing correctly from my Nano, but no dice. There's probably something very obvious I'm missing here.

Submitted by Ruud Kapteijn (/users/ruud-kapteijn) on Thu, 12/31/2020 - 04:20

[Permalink \(/comment/33331#comment-33331\)](#)

### Thank you very much for this ([/comment/33331#comment-33331](#))

Thank you very much for this comprehensive article. Very well written and really practical. It gives a clear overview of the possibilities of the board. I expect mine tomorrow and will run your code to get to know it.

The idea is the use the board to build trust vectoring. The 9-axis IMU should make this possible. In stage 2 it will be augmented by the u-blox SAM-M8Q GPS which also runs on 3.3V.

For this project the board will need to direct standard 6V servos with JR wiring. Can this be done directly from the output pins or is an amplification required due to the 3.3V? If so, what would be the best solutions?

Thanks for your advice on beforehand and with Best Regards...

Submitted by Marcos Ezequie... (/users/marcos-ezequiel-cervantes-castro) on Wed, 01/27/2021 - 11:35

[Permalink \(/comment/33461#comment-33461\)](#)

### hey, i loved this, i just ([/comment/33461#comment-33461](#))

hey, i loved this, i just bought an arduino nano 33 ble, y just want to know how to use the bluetooth :( how to access to, in which pin?? how;?? thanks if someone answer me :(

Submitted by NIYOYITA Amani (/users/niyoyita-amani) on Wed, 02/16/2022 - 17:34

[Permalink \(/comment/34930#comment-34930\)](#)

### a very good article here, ([/comment/34930#comment-34930](#))

a very good article here, this helped me to develop my custom PCB

Log in (/user/login?destination=/microcontroller-projects/arduino-nano-33-ble-sense-board-review-and-getting-started-guide%23comment-form) or register (/user/register?destination=/microcontroller-projects/arduino-nano-33-ble-sense-board-review-and-getting-started-guide%23comment-form) to post comments



## FEATURED PRODUCTS FROM DIGIKEY



PowerPlane Busbar Power Connectors (<https://bit.ly/46znd18>)

High-current drop-in replacement connectors support a wide range of power distribution applications.

(<https://bit.ly/46znd18>)



Black Plated RF Connectors and Adapters (<https://bit.ly/3N59DvF>)

Industrial plating option with enhanced durability for a variety of harsh environments

(<https://bit.ly/3N59DvF>)



AD8410A/AD8411A Current-Sense Amplifiers (<https://bit.ly/46yC503>)

AD8410A/AD8411A Current-Sense Amplifier with PWM Rejection, High Voltage, High Bandwidth Amplifiers

(<https://bit.ly/46yC503>)



NCV84120 Single-Channel High-Side Driver (<https://bit.ly/46vZZZS>)

onsemi's NCV84120 is a self-protected, very low  $I_Q$  high-side driver with analog current sense

(<https://bit.ly/46vZZZS>)



T6793 CO2 Sensor Module (<https://bit.ly/47PkeTI>)

T6793 CO2 sensor module is designed for applications requiring high-performance in a compact package

(<https://bit.ly/47PkeTI>)



(<https://bit.ly/3tRUkzW>)

#### WR-TBL Terminal Blocks (<https://bit.ly/3tRUkzW>)

WR-TBL terminal blocks can be customized and developed for a customer's specific application



(<https://bit.ly/49bWVVi>)

#### SPM Power Inductors (<https://bit.ly/49bWVVi>)

SPM inductors offer larger current, lower Rdc, smaller size, and superior DC bias characteristics.



(<https://bit.ly/3s8luAO>)

#### EPC2252 Automotive Qualified 80 V GaN FET (<https://bit.ly/3s8luAO>)

EPC's enhancement mode power transistor is ideal for designing higher-resolution LiDAR systems



## Join 100K+ Subscribers

Your email is safe with us, we don't spam.

Type your email address

Subscribe

Be a part of our ever growing community.



Semicon Media is a unique collection of online media, focused purely on the Electronics Community across the globe. With a perfectly blended team of Engineers and Journalists, we demystify electronics and its related technologies by providing high value content to our readers.



## COMPANY

[Privacy Policy \(/privacy-policy\)](#)    [Cookie Policy \(/cookie-policy\)](#)    [Terms of Use \(/terms-of-use\)](#)  
[Contact Us \(/contact\)](#)    [Advertise \(/advertise\)](#)

---

## PROJECT

[555 Timer Circuits \(/555-timer-circuits\)](#)    [Op-amp Circuits \(/op-amp-circuits\)](#)    [Audio Circuits \(/audio-circuits\)](#)  
[Power Supply Circuits \(/smps-power-supply-circuits\)](#)    [Arduino Projects \(/arduino-projects\)](#)    [Raspberry Pi Projects \(/simple-raspberry-pi-projects-for-beginners\)](#)  
[MSP430 Projects \(/msp430-projects\)](#)    [STM32 Projects \(/stm32-projects-and-tutorials\)](#)    [ESP8266 Projects \(/esp8266-projects\)](#)  
[PIC Projects \(/pic-microcontroller-projects\)](#)    [AVR Projects \(/avr-microcontroller-projects\)](#)    [8051 Projects \(/8051-microcontroller-projects\)](#)  
[ESP32 Projects \(/esp32-projects\)](#)    [IoT Projects \(/internet-of-things-iot-projects\)](#)    [PCB Projects \(/diy-pcb-projects\)](#)  
[Arduino ESP8266 Projects \(/arduino-esp8266-projects\)](#)    [All Microcontroller Projects \(/microcontroller-projects\)](#)

---

## OUR NETWORK



(<https://circuitdigest.com>)



(<https://components101.com>)



(<https://iotdesignpro.com>)

Copyright © 2023 Circuit Digest (/). All rights reserved.