

## Práctica 1 Arquitectura Arduino Nano 33 BLE

### 1.1 Objetivos de la práctica:

- Familiarización con el sistema de desarrollo Arduino Nano 33 BLE.
- Manejo de los periféricos típicos de cualquier microcontrolador: ADC, PWM, timers, GPIOs...

Trabajaremos con la placa de desarrollo [Arduino Nano 33 BLE](#), un módulo que utiliza el potente procesador nRF52840 de Nordic Semiconductors, que tiene las siguientes características:

- ARM Cortex-M4F 64MHz
- 32 bit
- Unidad de coma flotante (FPU)
- 1 MB Flash
- 256 KB RAM



Este procesador es parte del módulo [NINA-B306](#) incluido en el Arduino Nano 33 BLE y que también incluye comunicaciones Bluetooth 5:

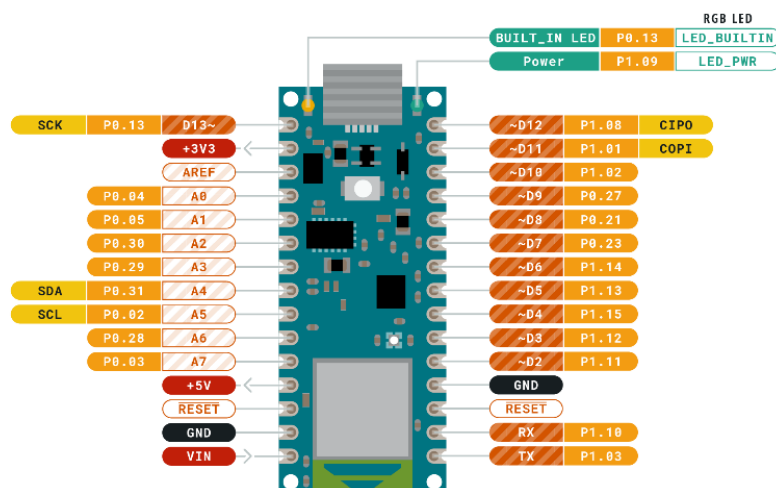
Item	NINA-B3x1	NINA-B3x2	NINA-B3x6
Bluetooth version	5.0	5.0	5.0
Band support	2.4 GHz, 40 channels	2.4 GHz, 40 channels	2.4 GHz, 40 channels
Typical conducted output power	+7.5 dBm	+8 dBm	+8 dBm
Radiated output power (EIRP)	+10.5 dBm (with approved antennas)	+10 dBm	+10 dBm
RX sensitivity (conducted)	-94 dBm	-94 dBm	-94 dBm
RX sensitivity, long range mode (conducted)	-100 dBm	-100 dBm	-100 dBm
Supported 2.4 GHz radio modes	Bluetooth Low Energy IEEE 802.15.4 <sup>1</sup> Proprietary 2.4 GHz modes <sup>1</sup>	Bluetooth Low Energy IEEE 802.15.4 <sup>1</sup> Proprietary 2.4 GHz modes <sup>1</sup>	Bluetooth Low Energy IEEE 802.15.4 <sup>1</sup> Proprietary 2.4 GHz modes <sup>1</sup>
Supported Bluetooth Low Energy data rates	1 Mbps 2 Mbps 500 kbps 125 kbps	1 Mbps 2 Mbps 500 kbps 125 kbps	1 Mbps 2 Mbps 500 kbps 125 kbps
Module size	10.0 x 11.6 mm	10.0 x 15.0 mm	10.0 x 15.0 mm

Para la captura de datos cuenta con:

- IMU de 9 grados de libertad (LSM9DS1) y 16 bits de resolución.
- Micrófono digital omnidireccional (MP34DT05).
- Sensor de luz ambiente y color (APDS9960) que permite la detección de gestos, proximidad, etc.
- Sensor de presión (LPS22HB).
- Sensor capacitivo de temperatura y humedad (HTS221).



## ARDUINO NANO 33 BLE



## 1.2 Tareas a realizar:

1. Con una periodicidad de un segundo, lee el valor del ADC y muéstralo por consola. Utiliza la salida de 3V3 de la placa con un potenciómetro para generar la tensión analógica. **Antes de conectarla al módulo comprueba que no superas 3V3 con un multímetro.**  
Material de apoyo: <https://www.programmingelectronics.com/sprintf-arduino/>
2. Utilizando los timers hardware genera una interrupción cada 10 segundos que lea el ADC y lo muestre por pantalla.  
Material de apoyo: <https://forum.arduino.cc/t/five-hardware-timers-example/905798>
3. Genera una salida PWM a 5kHz proporcional a la lectura del ADC. Comprueba en el osciloscopio tanto la medida analógica como la salida PWM.
4. Utilizando el tipo string gestiona por la UART los periféricos mediante un protocolo de tal modo que si le mandas:
  - a. ADC: Envíe la lectura del ADC actual
  - b. ADC(x): envíe la lectura del ADC cada x segundos. Si x=0, deja de mandar datos. Utiliza interrupción de Timer para la temporalización.
  - c. PWM(x): comanda el duty cycle del módulo PWM con números del 0 al 9.Material de apoyo:  
<https://www.arduino.cc/reference/en/language/variables/data-types/string/>  
<https://cpp4arduino.com/2018/10/23/what-is-string-interning.html>
5. Conecta dos placas que se comuniquen entre sí utilizando I2C. Haz que una de ellas encienda o apague un led externo según los comandos que reciba de la otra. Utiliza un pin de salida digital para el encendido y apagado del led.  
Material de apoyo:  
<https://docs.arduino.cc/tutorials/nano-33-ble-sense/i2c/>
6. Muestrea los datos de los sensores acelerómetro, giróscopo y magnetómetro incluidos en la placa y cada 100 ms y mándalos cada segundo por UART.  
Material de apoyo:  
<https://docs.arduino.cc/tutorials/nano-33-ble-sense/imu-accelerometer/>  
<https://docs.arduino.cc/tutorials/nano-33-ble-sense/imu-gyroscope/>  
<https://docs.arduino.cc/tutorials/nano-33-ble-sense/imu-magnetometer/>
7. Combinando los dos apartados anteriores, haz que dos placas se comuniquen por I2C.  
PLACA 1: captura datos de los tres sensores muestreados cada 200 ms durante 1 segundo a demanda del usuario. Tras ese segundo los manda por I2C a la otra placa.  
PLACA 2: cuando recibe datos por I2C los muestra por pantalla y enciende un LED durante 1 segundo al recibirlo.

## 1.3 Hitos evaluables de la práctica:

1. Enseñar salida PWM y señal analógica en el osciloscopio. (tarea 3)
2. Enseñar terminal UART mostrando las funciones de control (tarea 4)
3. Enseñar y explicar comunicación I2C utilizando el trigger del osciloscopio. (tarea 7)

## Anexo – Recursos interesantes

Heap: <https://cpp4arduino.com/2018/11/06/what-is-heap-fragmentation.html>

Strings:

<https://cpp4arduino.com/2018/11/21/eight-tips-to-use-the-string-class-efficiently.html>

<https://cpp4arduino.com/2018/10/23/what-is-string-interning.html>

Progmem: <https://www.arduino.cc/reference/en/language/variables/utilities/proGMEM/>

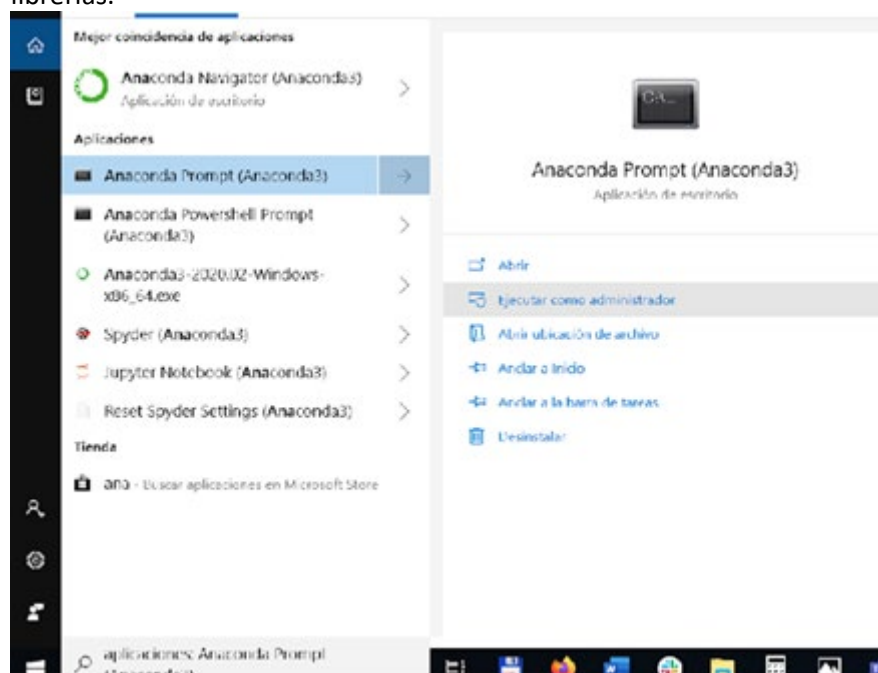
Socket wifi: <https://techtutorialsx.com/2018/05/17/esp32-arduino-sending-data-with-socket-client/>

Termite: [https://www.compuphase.com/software\\_termite.htm](https://www.compuphase.com/software_termite.htm)

Arduino IDE: <https://www.arduino.cc/en/software>

Instalación del entorno ESP32: <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

Instalar librerías:



- Pyserial: pip install pyserial
- Matplotlib: pip install matplotlib