Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE0624 – Laboratorio 5
III ciclo 2023

STM32/Arduino: GPIO, Giroscopio, comunicaciones, TinyML.

Estudiantes:
Kevin Campos Castro
Josué Salmerón Córdoba
Grupo 1
Profesor: Marco Villalta

12 de febrero de 2024

Índice

1. Resumen **1**

2. Nota teórica **2**

3. Desarrollo/Análisis **7**

4. Conclusiones y recomendaciones **11**

5. Anexos **13**

Índice de figuras

1. Diagrama de bloques del Nano BLE 33 Sense . Tomado de [1]. 3
2. Diagrama de pines del Nano BLE 33 Sense . Tomado de [2]. 4
3. Características eléctricas de nRF52480. Tomado de [1]. 4
4. Características eléctricas de la placa. Tomado de [2]. 4
5. Diagrama de pines del LSM9DS1, tomado de [3]. 5
6. Diagrama del circuito. 6
7. Registro del giroscopio 7
8. Graficación del movimiento 7
9. Datos de aceleración importados. 8
10. Datos del giroscopio importado. 8
11. Gráfica del entrenamiento y su pérdida. 8
12. Gráfica del entrenamiento y su pérdida iniciando en 100. 9
13. Gráfica del error absoluto para observar rendimiento. 9
14. Movimiento brazo arriba. 10
15. Movimiento puño hacia la PC. 10
16. Movimiento circular. 10
17. Movimiento brazo arriba. 10
18. Resumen de los movimientos. 10

Índice de tablas

1. Lista de equipos 5

1. Resumen

En este trabajo se presenta una de muchas aplicaciones en TinyML que posee el microcontrolador Arduino Nano 33 Ble Sense, en este caso se trata de la detección de movimientos realizados con el brazo. Con ayuda del ejemplo de la librería de TensorFlowLite, en especial el `IMU_Capture` se logró hacer uso del giroscopio y la acelaración de los movimientos para luego hacer un entrenamiento de los datos con el notebook de Colab disponible en [4] y con el ejemplo `IMU_Classifier` la placa logra identificar mostrando la probabilidad más alta de alguno de los 3 movimientos realizados. Por tanto, este trabajo cumple satisfactoriamente con lo solicitado porque el arduino es capaz de identificar los movimientos una vez realizados.

2. Nota teórica

En esta sección se describen los componentes principales que se utilizaron para el desarrollo del HAR-Human Activity Recognition.

Arduino Nano 33 BLE

El arduino Nano 33 BLE Sense es un módulo miniatura que contiene un módulo NINA B306, basado en Nordic nRF52480 y contiene un M4F Cortex, un cripto chip el cual puede almacenar certficados de forma segura y pre-compartir llaves y un IMU de 9 ejes. El módulo puede ser montado como un componente DIP o como componente SMT, directamente soldado por la via de los pads.

Características generales

Las características más importantes de este mcu se mencionan a continuación [1]

- CPU: ARM Cortex-M4 a 64MHz con FPU, 32-bit, 1MB Flash, 256kB SRAM.
- Bluetooth 5, IEEE 802.15.4-2006, 2,4 GHz.
- ARM TrusZone Cryptocell 310 security subsystem, secure boot.
- USB 2.0, QSPI, SPI.
- 48 GPIOs.
- 12-bit, ADC con 8 canales.
- 64 comparadores de nivel, 15 del tipo low-power.
- Sensor de temperatura.
- 4×4-canales PWM.
- Periféricos de audio: I2S, PDM
- 5×32-bit timers.
- 4× SPI maestros3× SPI esclavos.
- 2×I2C.
- 2× UART.
- decodificador de cuadratura (QDEC).
- 3× RTC.

Diagrama de bloques

La figura 1 representa el diagrama de bloques de la placa.

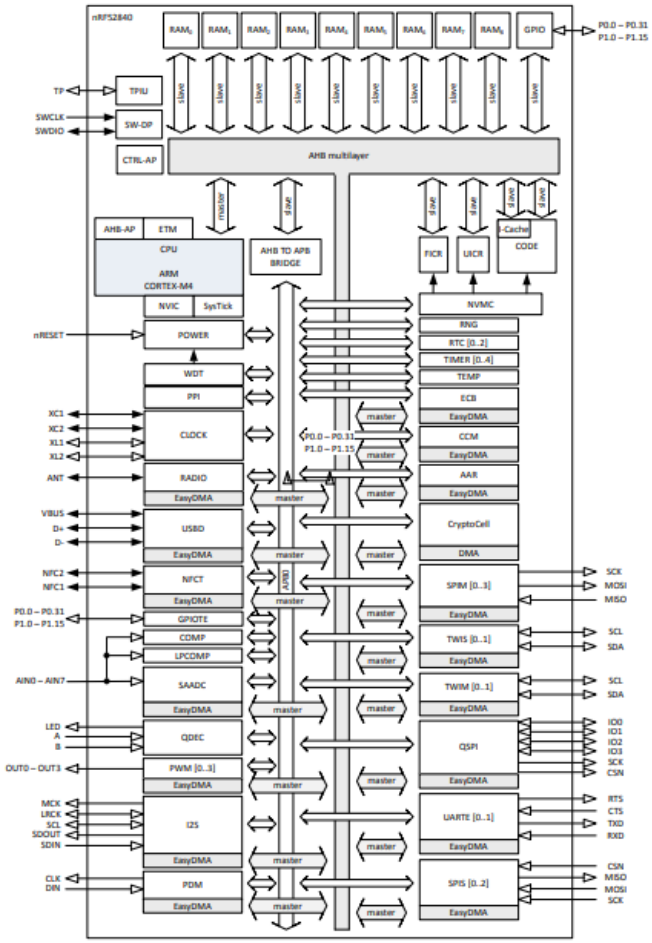


Figura 1: Diagrama de bloques del Nano BLE 33 Sense . Tomado de [1].

Diagrama de pines

El diagrama de la fugura 2 brinda de manera más detallada la distribución de los pines.

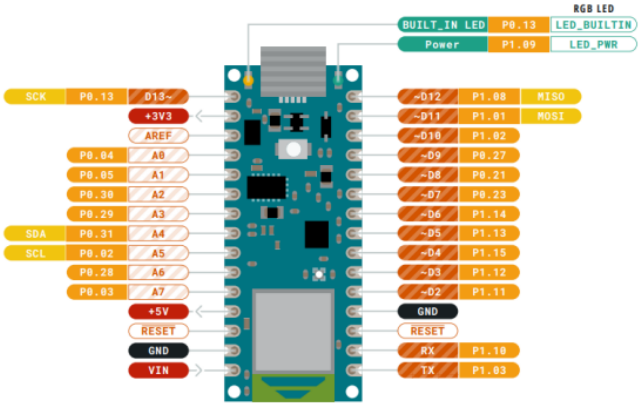


Figura 2: Diagrama de pines del Nano BLE 33 Sense . Tomado de [2].

Características eléctricas

Aquí se tomaron dos referencias para tener más claro este detalle, primero se muestran los valores máximos del mcu nRF52480 y los de la placa respectivamente.

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

Figura 3: Características eléctricas de nRF52480. Tomado de [1].

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

Figura 4: Características eléctricas de la placa. Tomado de [2].

Periféricos utilizados

En este laboratorio no se utilizaron periféricos externos, además, el único sensor que se utilizó es el giroscopio de 9 ejes que viene incorporado en el Arduino Nano 33 BLE. A continuación se presentan los puntos más importantes de este sensor (LSM9DS1) para realizar este laboratorio [3]:

- 3 canales de aceleración, 3 canales de velocidad angular, 3 canales de campo magnético.
- ±2/±4/±8/±16 g aceleración lineal escala completa.
- ±245/±500/±2000 dps velocidad angular escala completa.
- Salida de datos de 16 bits.
- Interfaces serie SPI / I2C.
- Tensión de alimentación analógica 1,9 V a 3,6 V.

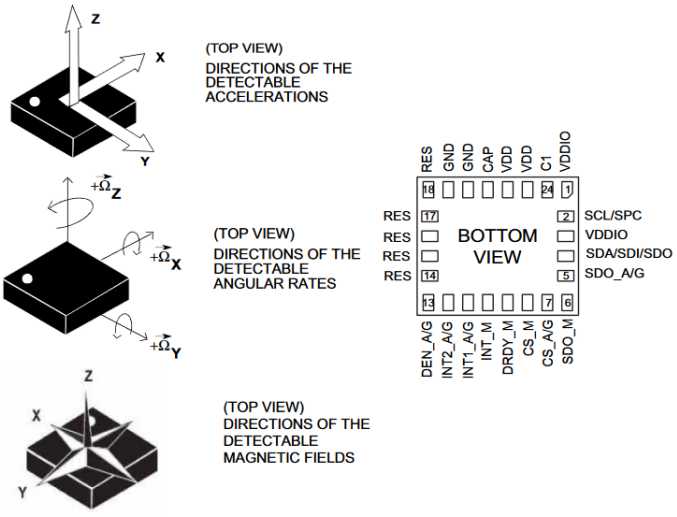


Figura 5: Diagrama de pines del LSM9DS1, tomado de [3].

Lista de componentes

Tabla 1: Lista de equipos

Componente	Cantidad	Precio
Arduino Nano 33 BLE	1	60$
Total		60$

Diseño del circuito

El diseño del circuito es muy simple ya que basta con conectar el arduino nano 33 ble sense a la computadora por medio de un cable USB, tal como se muestra en la figura 6.

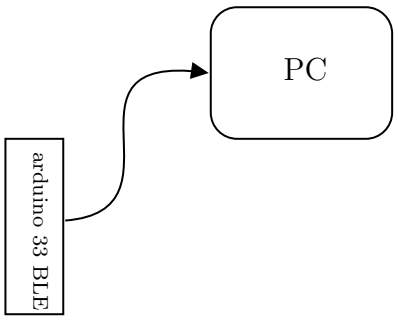


Figura 6: Diagrama del circuito.

A partir de lo mostrado anteriormente se consultaron las referencias dadas por el profesor para realizar el laboratorio.

3. Desarrollo/Análisis

Gracias a lo consultado en [5], fue posible realizar todas las configuraciones necesarias para empezar con el uso del programa `IMU_ Capture.ino` disponible en [6], el cual se encarga de leer la aceleración y el giroscopio de la placa e imprime por un segundo en el monitor serial (por consola del IDE) cuando una velocidad significativa es detectada. Además, es posible activar el `Serial Plotter` para graficar los datos. Lo anterior se muestra a continuación.

```
aX,aY,aZ,gX,gY,gZ
-1.171,-0.242,1.188,22.461,-57.861,23.987
-1.115,-0.205,1.227,0.061,-74.158,51.208
-1.023,-0.071,1.269,-17.578,-83.252,76.294
-0.943,0.076,1.271,-27.161,-79.285,85.510
-0.841,0.138,1.245,-23.560,-61.584,74.951
-0.774,0.148,1.192,-10.803,-36.926,53.040
-0.767,0.129,1.127,5.798,-12.817,28.503
-0.777,0.050,1.060,22.522,5.798,8.911
-0.761,-0.052,0.982,32.715,15.320,2.014
-0.753,-0.093,0.913,29.724,15.869,8.057
-0.741,-0.098,0.863,14.648,12.207,18.738
-0.712,-0.082,0.833,0.732,5.188,24.780
-0.674,-0.063,0.811,-2.991,-0.854,21.729
-0.619,-0.048,0.786,2.563,-4.822,9.094
-0.545,-0.050,0.795,17.395,-5.188,-10.010
-0.462,-0.120,0.830,35.950,-2.930,-30.396
-0.369,-0.186,0.854,49.500,-0.610,-43.518
-0.321,-0.201,0.887,53.650,2.625,-47.180
-0.278,-0.256,0.918,54.321,4.578,-46.204
-0.235,-0.315,0.933,54.565,5.920,-40.039
-0.207,-0.328,0.975,49.377,7.080,-28.564
-0.185,-0.358,1.001,43.274,9.705,-18.494
-0.218,-0.341,1.001,35.461,12.695,-11.841
-0.268,-0.290,1.018,28.503,14.282,-8.728
-0.324,-0.250,1.027,24.658,15.442,-9.399
-0.375,-0.233,1.028,20.691,15.076,-11.719
-0.418,-0.224,1.035,14.099,12.207,-14.160
-0.447,-0.212,1.037,5.798,7.263,-17.090
-0.481,-0.203,1.042,-2.869,2.747,-20.569
-0.521,-0.201,1.049,-9.277,-1.038,-24.780
-0.550,-0.196,1.043,-12.939,-2.808,-29.968
-0.567,-0.217,1.047,-15.564,-6.775,-35.706
-0.610,-0.262,1.042,-18.311,-11.292,-40.649
```

Figura 7: Registro del giroscopio

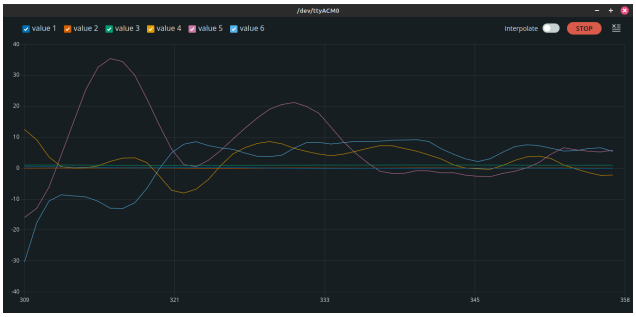


Figura 8: Graficación del movimiento

De la figura 7, se tiene las coordenadas que registran el movimiento realizado, que de hecho es un golpe hacia la pantalla de la PC. Luego, en la figura 8 muestra las ondas del golpe. Hecho esto, se procede a realizar un pequeño script de Python para grabar todos estos datos en un archivo .csv. Por lo que una vez cargado el código `IMU_ Capture.ino` se cierra el IDE de Arduino y se ejecuta el script. De donde se realizaron 3 movimientos.

- Golpe hacia la pantalla (en una sola dirección).
- Alzar el brazo con diferentes direcciones.
- Movimiento circular en contra de las manecillas del reloj.

Se tomaron 1750 muestras para los 3 movimientos, un aproximado de 25 s realizando la misma tarea. Ahora, con base a estos datos se realizó el entrenamiento con ayuda de [4].

Para el entrenamiento se tomaron los resultados obtenidos y se procedió a usar como base el ejemplo de reconocimiento de gestos dado en la presentación de la clase [4]. El primer paso fue introducir los datos, el resultado de esto se observa a continuación:

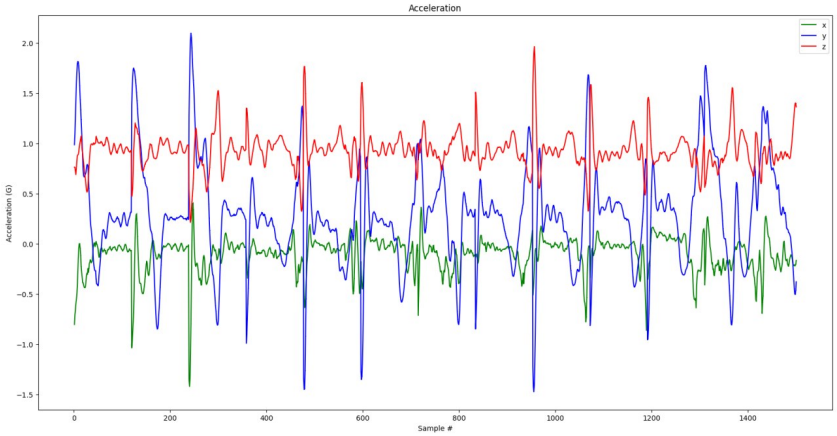


Figura 9: Datos de aceleración importados.

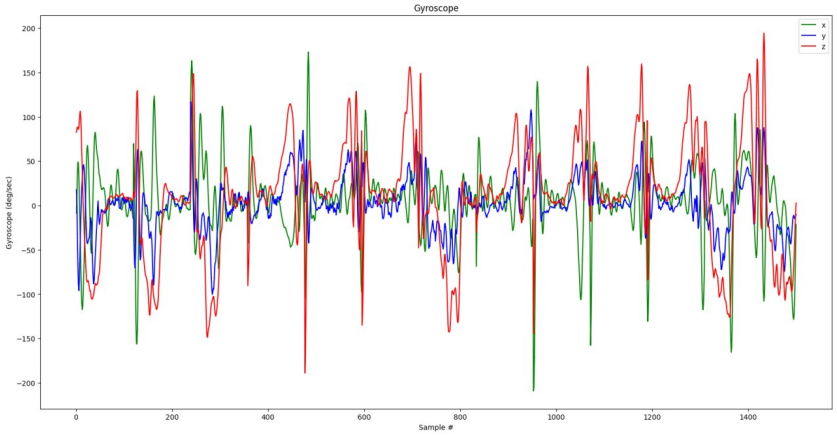


Figura 10: Datos del giroscopio importado.

Una vez realizado este paso se empieza a entrenar el modelo, para ello se preparan los datos para que su formato coincida, una vez realizado lo anterior se entrena la red siguiendo las indicaciones del enunciado y tomando el 60 % de los datos para el entrenamiento, un 20 % para la validación y otro 20 % para realizar pruebas. Los resultados del entrenamiento se muestran en las siguientes figuras:

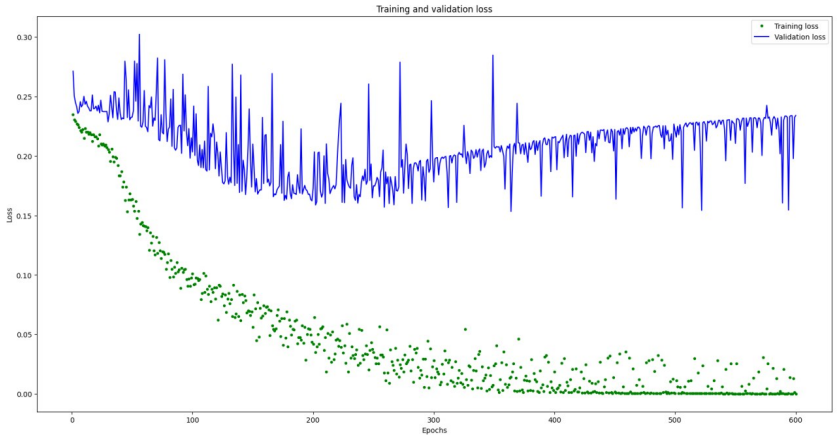


Figura 11: Gráfica del entrenamiento y su pérdida.

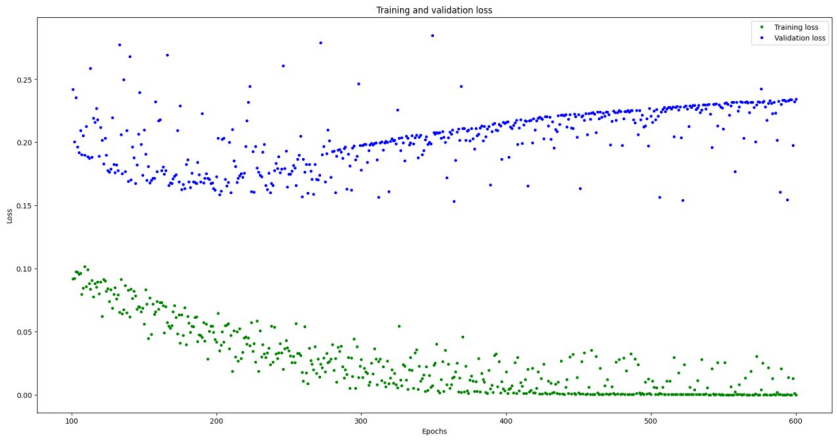


Figura 12: Gráfica del entrenamiento y su pérdida iniciando en 100.

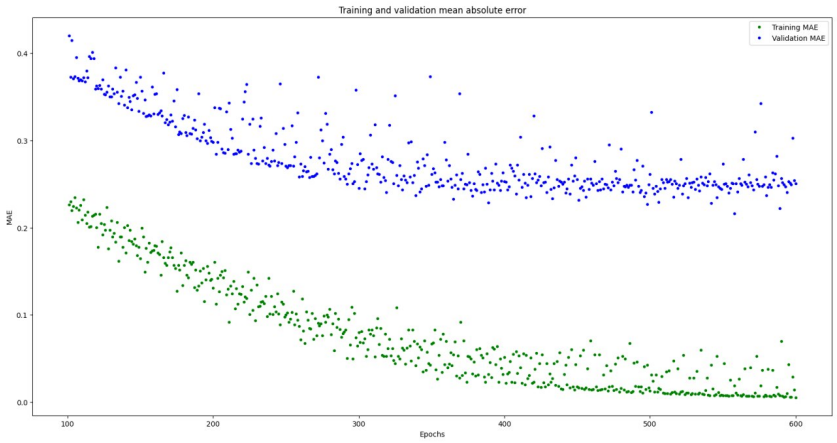


Figura 13: Gráfica del error absoluto para observar rendimiento.

Con lo anterior se obtiene un archivo de encabezado "model.h" que es usado para poder determinar los movimientos escogidos para este laboratorio.

Ahora, gracias al entrenamiento realizado y siguiendo con la ayuda de los ejemplos vistos en clase, se toma la referencia de [6] a la cual se le hacen unas pequeñas modificaciones, primero se le añade el archivo `model.h` para ver los resultados del entramiento pero ahora en la placa. Lo otro es la eliminación de algunos encabezados que no serán necesarios ya que el profesor indicó que son más orientados a los desarrolladores de tensorflow tales como: `tensorflow/lite/micro/micro_error_ reporter.h` y `tensorflow/lite/version.h`. También se eliminó la variable global `tflite::MicroErrorReporter tflErrorReporter` y el acceso a la memoria de esta misma variable. Por otro, lado se tuvo que añadir los mapas de los gestos:

Listing 1: Nombre de los archivos csv

```
// array to map gesture index to a name
const char* GESTURES[] = {
  "punch",
  "arm_up",
  "circle"
};
```

Hecho esto, se verificó y cargó el código a la placa, lo cual tomó cierto tiempo en comparación a otros ejemplos. Una vez terminado esto, se realizaron los movimientos y por medio del serial monitor se mostraron las probabilidades de los movimientos ejecutados, donde el que posee mayor probabilidad indica el movimiento o acción hecha con la placa.

Figura 14: Movimiento brazo arriba.

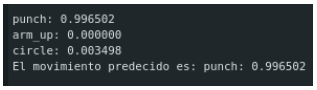


Figura 15: Movimiento puño hacia la PC.

Figura 16: Movimiento circular.

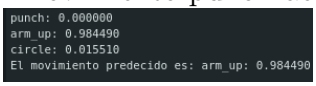


Figura 17: Movimiento brazo arriba.

Primero se levantó el brazo y es sencillo notar que en la figura 14 el movimiento es detectado correctamente ya que las demás probabilidades o valores con base al entrenamiento fueron totalmente despreciables. El siguiente movimiento fue un puñetazo hacia la PC y note que en la figura 15 la placa logra detectarlo sin problema alguno, con una similitud muy ínfima con el movimiento circular. Por último, se movió la placa en forma circular y el resultado de la figura 16 detecta este movimiento con algunas similitudes con el levantamiento del brazo hacia y el puñetazo, no obstante el valor más alto fue el movimiento circular. En resumen, la figura 18 es la consola o el monitor serial de Arduino que va mostrando estas probabilidades dependiendo del movimiento que se ejecute con la placa.

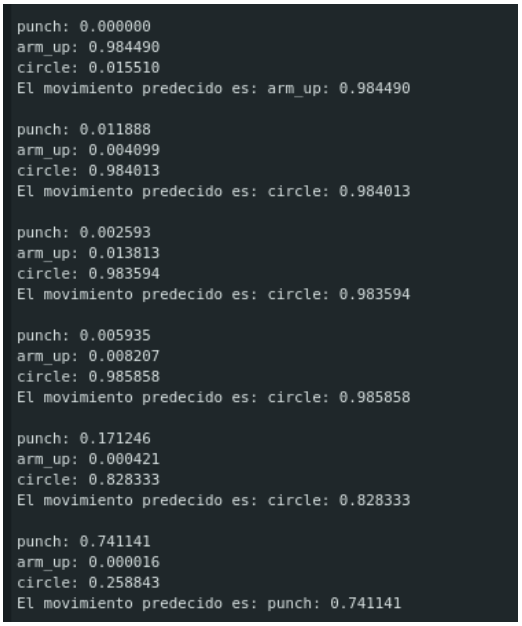


Figura 18: Resumen de los movimientos.

4. Conclusiones y recomendaciones

Las conclusiones para este trabajo son:

- Los ejemplos brindados fueron de gran ayuda para realizar las pruebas con el giroscopio.
- El notebook de Colab está muy bien hecho porque es rápido, lo cual tiene sentido al tratarse de Python, y realiza un muy buen entrenamiento de los datos porque a la hora de llevar el archivo `model.h` al código `IMU_ Classifier`, resulta que que la placa es consistente cuando asigna una probabilidad al movimiento realizado previamente.
- Tensorflow lite es una herramienta muy poderosa porque permite hacer uso de técnicas de machine learning en dispositivos que no tienen un rendimiento tan alto.
- Por tanto, el programa cumple satisfactoriamente porque los movimientos realizados fueron hechos con la misma cantidad de muestras; 1750. Para el caso del puñetazo se hizo en una misma dirección hacia la pantalla de la PC. En cambio, los otros dos: movimiento circular se ejecutó en sentido horario y anti-horario, y el brazo hacia arriba se hizo en diferentes ejes. Además, el buen entrenamiento de los datos mencionado previamente ayudó mucho para identificar los movimientos realizados.

A modo de recomendación, prestar mucha atención en clases con **todo** lo que menciona el profesor, por supuesto estudiar muy bien los ejemplos recomendados por él así como la literatura. Lo otro muy importante es hacer los movimientos con diferentes ejes y así no tener un mismo patrón en un muestreo de 1750 muestras ya que no habrá diversidad para identificar el movimiento realizado.

Referencias

[1] Nordic Infocenter. nrf52840 datasheet. Nordic Infocenter, https://infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf, Mayo 2019. Accedido en febrero de 2024.

[2] Arduino Docs. Arduino nano 33 ble. Arduino Docs, https://docs.arduino.cc/resources/datasheets/ABX00030-datasheet.pdf, Mayo 2019. Accedido en febrero de 2024.

[3] ST life.augmented. 9-axis inemo inertial module (imu): 3d magnetometer, 3d accelerometer, 3d gyroscope with i2c and spi. https://www.st.com/en/mems-and-sensors/lsm9ds1.html#overview, setiembre 2015. Accedido en febrero de 2024.

[4] Arduino Docs. Tiny ml on arduino. Arduino, https://colab.research.google.com/github/arduino/ArduinoTensorFlowLiteTutorials/blob/master/GestureToEmoji/arduino_tinyml_workshop.ipynb#scrollTo=9J33uwpNtAku, febrero 2024. Accedido en febrero de 2024.

[5] Arduino Docs. Get started with machine learning on arduino. Arduino Docs, https://docs.arduino.cc/tutorials/nano-33-ble-sense/get-started-with-machine-learning/, febrero 2024. Accedido en febrero de 2024.

[6] Arduino. Imu capture. Arduino, https://github.com/arduino/ArduinoTensorFlowLiteTutorials/tree/master/GestureToEmoji/ArduinoSketches/IMU_Capture, Agosto 2020. Primer parte, captura de gestos.

5. Anexos

A continuación, se muestran las hojas del fabricante de los componentes usados para este laboratorio. Y el link del repositorio de este trabajo: `https://github.com/JosueC07183/Labo5`.

Feature list

Features:

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

Applications:

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

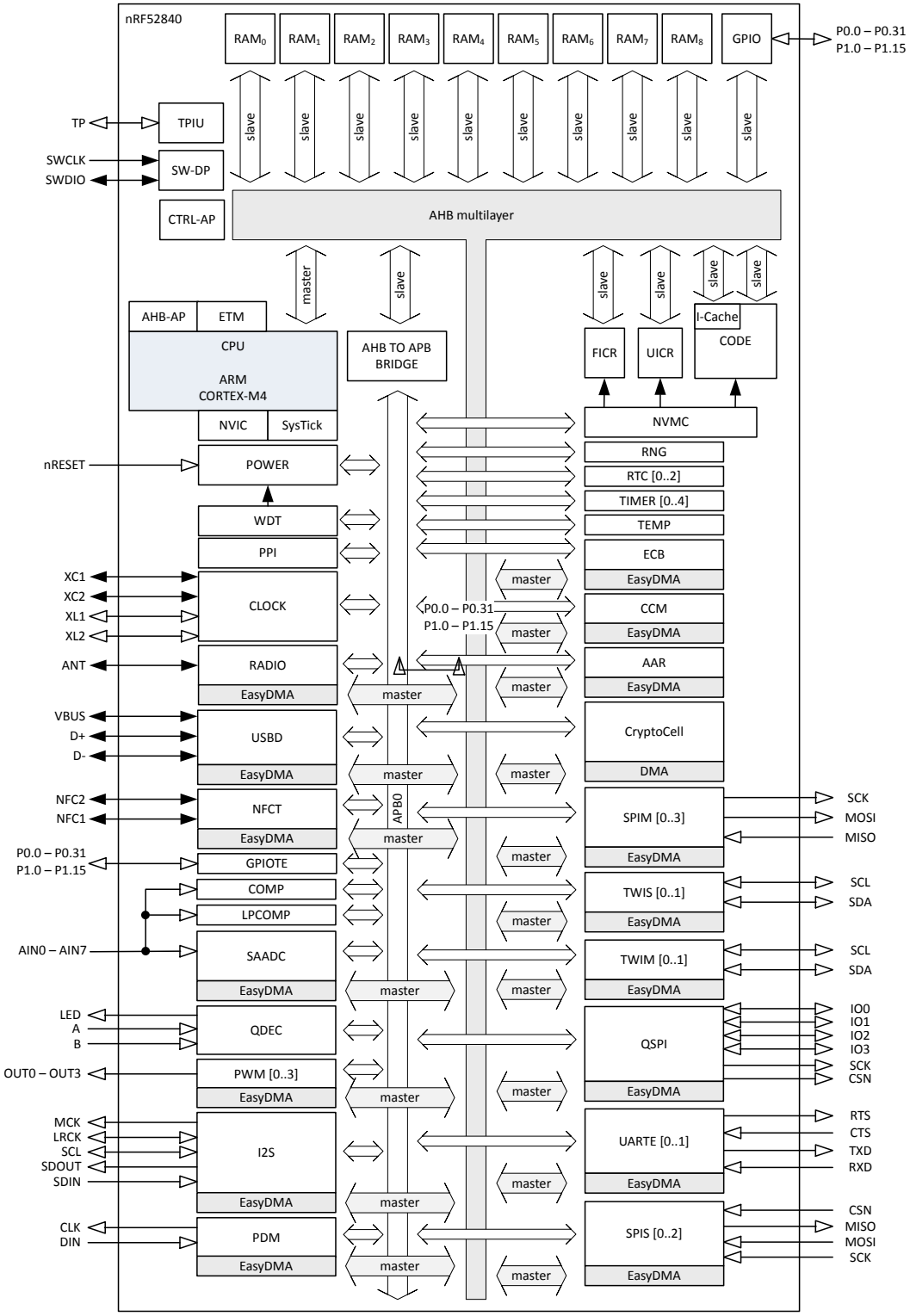
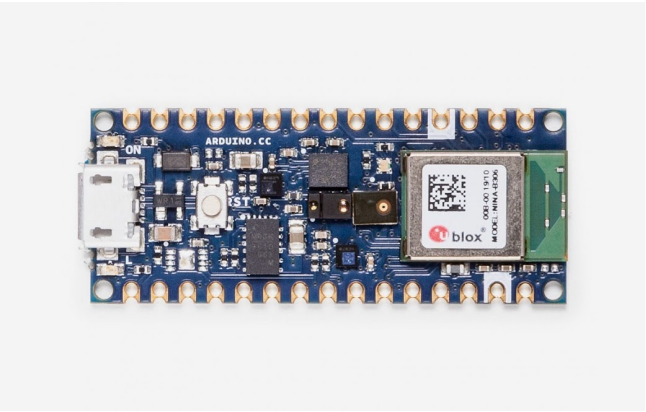


Figure 1: Block diagram

Product Reference Manual
SKU: ABX00031



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

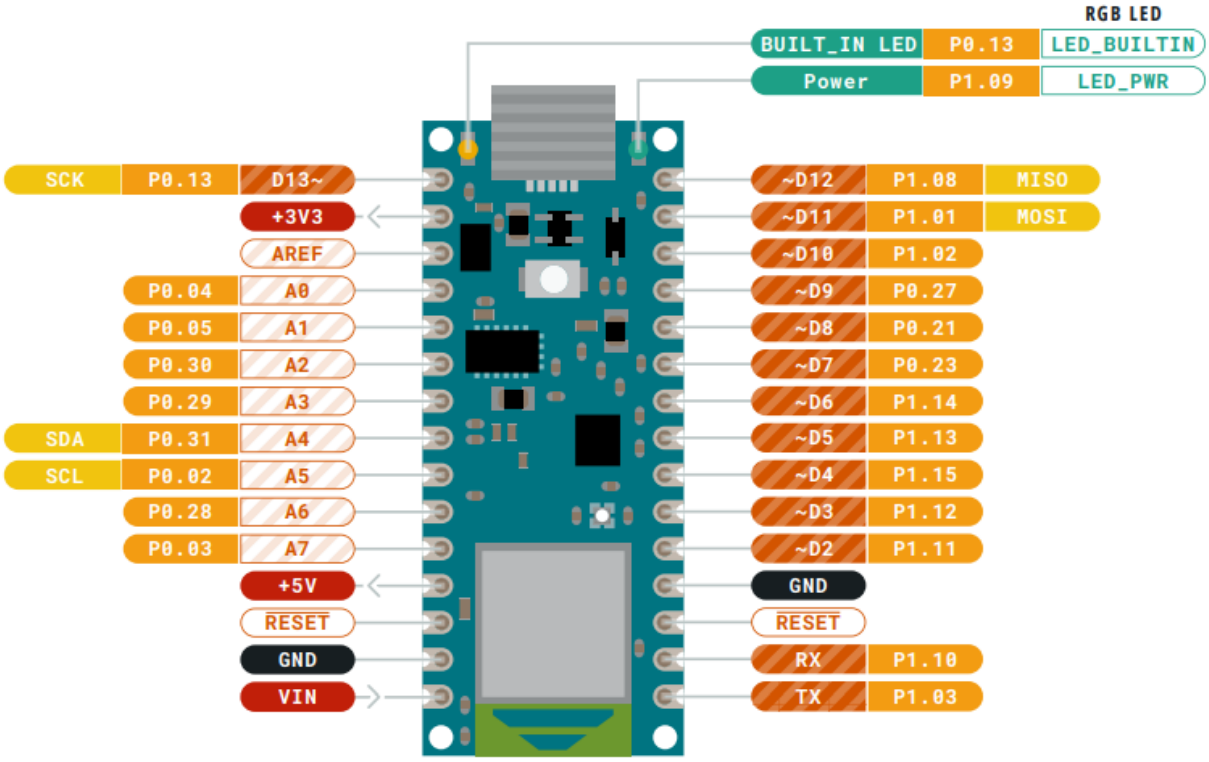
Target areas:

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
  - ±2/±4/±8/±16 g linear acceleration full scale
  - ±4/±8/±12/±16 gauss magnetic full scale
  - ±245/±500/±2000 dps angular rate full scale
  - 16-bit data output
- **LPS22HB** (Barometer and temperature sensor)
  - 260 to 1260 hPa absolute pressure range with 24 bit precision
  - High overpressure capability: 20x full-scale
  - Embedded temperature compensation
  - 16-bit temperature data output
  - 1 Hz to 75 Hz output data rateInterrupt functions: Data Ready, FIFO flags, pressure thresholds
- **HTS221** (relative humidity sensor)
  - 0-100% relative humidity range
  - High rH sensitivity: 0.004% rH/LSB
  - Humidity accuracy: ± 3.5% rH, 20 to +80% rH
  - Temperature accuracy: ± 0.5 °C,15 to +40 °C
  - 16-bit humidity and temperature output data

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



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