

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0624: Laboratorio de Microcontroladores

1 ciclo 2023

Laboratorio 5

Arduino Nano BLE y ML

Jimmy Manley Peña

Roberto Acevedo Mora

Profesor: Marco Villalta Fallas

2 de julio del 2023

Resumen

En esta práctica se desarrolla un dispositivo de reconocimiento de actividad humana HAR (Human Activity Recognition), mediante el uso de la placa Arduni Nano Ble y una red neuronal entrenada utilizando la librería TensorFlow.

Se requiere detectar 3 tipos de movimientos distintos ante una entrada de datos que corresponden a cambios en movimiento captados por el giroscopio. Estos movimientos corresponden al movimiento que se realiza un puño, un movimiento circular y también se entrena para detectar un movimiento de flexión. Se realiza una captura de muestras con cada uno de estos movimientos utilizando un script de Python que lee los datos enviados por la placa y los guarda en un archivo csv. A partir de estos datos se modela una red neuronal con diversas capas y neuronas que por medio de aprendizaje automático podrá dar una predicción a un estímulo recibido luego de ser entrenada con los datos proporcionados. Para que el modelo pueda ser usado con TinyML se necesita convertir el modelo de TensorFlow a TensorFlow Lite y exportarlo como un archivo de encabezado .h, el cual se incluye en otro programa que se cargará en la placa y podrá realizar una predicción según las lecturas del giroscopio.

Link del repositorio: <https://github.com/robertoacevedomora/Laboratorio-5>

1. Nota teórica

1.1 Microcontrolador: El Arduino Nano 33 BLE utiliza el microcontrolador Nordic nRF52840, el cual es un chip ARM Cortex-M4 de 32 bits con una velocidad de reloj de 64 MHz. Este microcontrolador ofrece un alto rendimiento y características avanzadas.

Bluetooth Low Energy (BLE): La placa está diseñada para admitir la comunicación inalámbrica a través de Bluetooth Low Energy. Esto permite conectar la placa a otros dispositivos compatibles con BLE, como teléfonos inteligentes, tabletas y computadoras.

E/S digitales y analógicas: El Arduino Nano 33 BLE tiene 14 pines de E/S digitales, de los cuales 6 se pueden utilizar como pines PWM (Modulación por

Ancho de Pulso) para generar señales analógicas. Además, hay 8 pines analógicos para leer señales analógicas.

Memoria: El microcontrolador nRF52840 tiene 1 MB de memoria flash para cargar el código y 256 KB de RAM para la ejecución del programa. Esta memoria se puede utilizar para crear aplicaciones complejas y almacenar datos.

Alimentación: El Arduino Nano 33 BLE se puede alimentar a través del conector USB o mediante una fuente de alimentación externa de 7 a 21 V. Cuenta con un regulador de voltaje integrado que suministra 3,3 V a los componentes y pines de E/S.

Soporte de software: El Arduino Nano 33 BLE es compatible con el entorno de desarrollo integrado (IDE) de Arduino, lo que permite programar la placa utilizando el lenguaje de programación Arduino. P

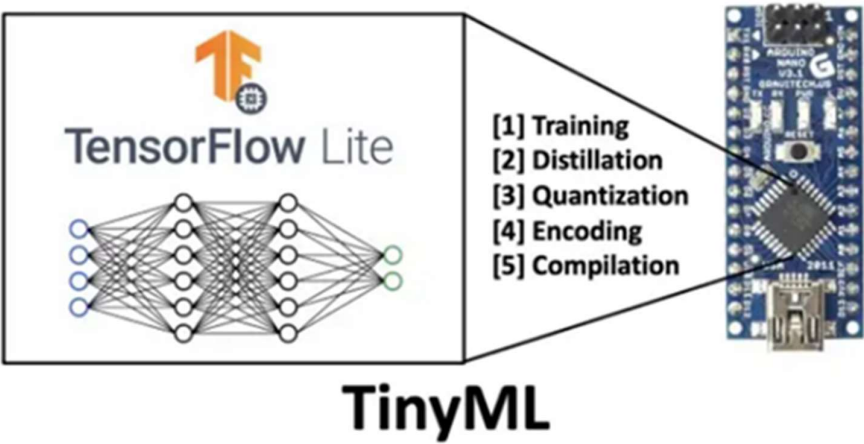
Sensores integrados: La placa también incluye algunos sensores integrados, como un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Estos sensores permiten detectar la orientación y los movimientos de la placa.

Tamaño compacto: El Arduino Nano 33 BLE tiene un tamaño reducido, similar al del modelo Arduino Nano tradicional. Esto la hace adecuada para proyectos en los que el espacio es limitado.

1.2 TensorFlow Lite TensorFlow es una biblioteca de software de código abierto para inteligencia artificial y aprendizaje automático con redes neuronales profundas. TensorFlow Lite es un marco de aprendizaje profundo de código abierto diseñado para la inferencia en el dispositivo. Proporciona un conjunto de herramientas que permite el aprendizaje automático en el dispositivo al dar la posibilidad de que los desarrolladores ejecuten sus modelos entrenados en dispositivos y computadoras móviles, integrados y de IoT. Es compatible con plataformas como Linux, sistemas operativos de dispositivos móviles y MCU. TensorFlow Lite está especialmente optimizado para el aprendizaje automático en el dispositivo (Edge ML). Como modelo Edge ML, es adecuado para la implementación en dispositivos de borde con recursos limitados.

1.3 Tiny Machine Learning (TinyML) se define como un campo de crecimiento exponencial en tecnologías y aplicaciones de aprendizaje automático que incluye hardware (circuitos integrados y dedicados), algoritmos y software capaces de realizar análisis de datos de sensores en dispositivos en las áreas de visión, biomédicos, audio, IMU, entre otros [5]. Estos se caracterizan por consumir potencia

extremadamente baja, típicamente en el rango de mW y hasta menor. Gracias a los anteriormente mencionados permite una variedad de casos de uso continuamente encendidos y enfocado a dispositivos que funcionan con baterías.



1.2. Periféricos usados

1.3. Otros componentes electrónicos utilizados:

BOM:

Componente	Tipo	Precio	Cantidad
Arduino Nano 33 BLE Kit		\$30.00	1
Total	-	\$30.31	4

Tabla 1: Lista de componentes utilizados

2. Librerías utilizadas

Para llevar a cabo el laboratorio 5 fue necesario instalar la librería Arduino_LSM9DS1 (giroscopio.ino), y Arduino_TensorFlowLite (tinym1.ino). Para esta última se incluyeron los archivos encabezados TensorFlowLite.h, all_ops_resolver.h, micro_error_reporter.h, micro_interpreter.h, schema_generated.h en el sketch tinym1.ino.

3. Desarrollo/Análisis de resultados

El laboratorio 5 consiste en desarrollar un reconocedor de actividad humana. Para llevar a cabo dicha tarea se utiliza como microcontrolador el Arduino nano ble 33. Se efectuó un script en el entorno virtual Arduino IDE para leer los datos del giroscopio, así como el acelerómetro. Posteriormente se procedió a guardar los datos recopilados de los ejes del giroscopio y el acelerómetro para cada uno de los gestos que se realizan en archivos csv para poder utilizarlos en Google Colab para el modelo de aprendizaje automático. Para enviar los datos del microcontrolador a un archivo csv utilizamos un script de Python que se comunica con el puerto COM6, por lo tanto, una vez que se carga el script giroscopio.ino al microcontrolador y se encuentre conectado mediante un cable usb a la computadora, los datos del giroscopio y acelerómetro son guardados en un archivo csv.

Se escogieron 3 gestos que se realizan con la mano para realizar el laboratorio, el gesto “puño” que es un movimiento con el cerrado puno hacia adelante; “círculo”, corresponde a un movimiento circular con la mano y “flexión”, que con el brazo estirado se realiza un movimiento de flexión. Para cada gesto se procedió a tomar datos durante 15 segundos, posteriormente estos datos se guardan en un archivo csv, un archivo por cada gesto.

Posteriormente, tomamos como referencia el archivo `arduino_tinyml_workshop.ipynb` y lo abrimos en Google Colab. Configuramos el Jupyter Notebook, después se procedió a subir los archivos `puno.csv`, `circulo.csv` y `flexión.csv`, así mismo, graficamos los datos del giroscopio y el acelerómetro. Una vez analizados y preparados los datos, se construye y se entrena el modelo. El modelo entrenado se convierte a un formato TensorFlow Lite. Para finalizar esta etapa, codificamos el modelo en un archivo de encabezado (`model.h`) C++ y descargamos dicho archivo para utilizarlo en el script donde analizamos los datos obtenidos en TensorFlow Lite.

Proseguimos el laboratorio creando un nuevo script en Arduino IDE para analizar los datos obtenidos por TensorFlow Lite, en dicho código incluimos las librerías `LSM9DS1.h`, `TensorFlowLite.h`, `all_ops_resolver.h`, `micro_error_reporter.h`,

micro_interpreter.h, schema_generated.h. También incluimos el archivo encabezado model.h, procedente del modelo entrenado.

En dicho código inicializamos las variables globales de Tensor Flow lite. Creamos una memoria estática y un array para enviar los datos del mapa generado por los gestos a cada dato que corresponde a cada uno de los tres gestos. Inicializamos el IMU, así mismo creamos el intérprete con 4 parámetros para inicializarlo. Normalizamos los datos IMU, de esta forma el monitor del program Arduino IDE va a mostrarnos el resultado de la relación del gesto que realizamos con respecto a los 3 gestos que estudiamos, por ejemplo, si llevamos a cabo una flexión el resultado del monitor específicamente para este gesto debería de ser cercano al 0.9 %. De esta forma, imprimimos el gesto y su probabilidad.

Análisis de resultados

Se subió el archivo .HEX generado por el script tinym1.ino al Arduino Nano Ble 33. Reiniciamos el microcontrolador y para ver los resultados se lleva a cabo los 3 gestos, en esta parte es importante tomar en cuenta que el movimiento debe ser relativamente rápido , de esta forma el microcontrolador detecta de mejor forma los movimientos que realizamos.

El primer ejercicio que se lleva a cabo es una flexión, tal y como se aprecia en la figura 3 cuando realizamos la flexión el nivel de probabilidad para dicho gesto es cercano al 99%, los otros dos gestos muestran un nivel de probabilidad mínimo.

El segundo ejercicio consiste en realizar un círculo, tal y como se aprecia en la figura 1 cuando realizamos el círculo el nivel de probabilidad para dicho gesto es cercano al 99%, los otros dos gestos muestran un nivel de probabilidad mínimo.

Por último, se efectúa un movimiento con el puño adelante, tal y como se aprecia en la figura 2 cuando realizamos el gesto puño el nivel de probabilidad para dicho gesto es cercano al 99%, los otros dos gestos muestran un nivel de probabilidad mínimo.

```
puno: 0.000000  
circulo: 0.999945  
flexion: 0.000055
```

```
puno: 0.000000  
circulo: 0.999997  
flexion: 0.000002
```

```
puno: 0.000000  
circulo: 0.999974  
flexion: 0.000026
```

Figura 1. Monitor serial gesto círculo

```
puno: 0.998265  
circulo: 0.000010  
flexion: 0.001725
```

```
puno: 0.999861  
circulo: 0.000000  
flexion: 0.000139
```

```
puno: 0.998410  
circulo: 0.000007  
flexion: 0.001583
```

Figura 2. Monitor serial gesto puño

```
puno: 0.014116  
circulo: 0.051266  
flexion: 0.934618
```

```
puno: 0.008784  
circulo: 0.011790  
flexion: 0.979426
```

```
puno: 0.010730  
circulo: 0.005864  
flexion: 0.983405
```

Figura 3. Monitor serial gesto flexión

Es importante tomar en cuenta la posición en la que debe estar colocada la tarjeta para poder llevar a cabo los gestos de la manera correcta, para el caso del “puño” la tarjeta debe estar invertida y sostenida por la mano derecha, en caso contrario no se puede detectar el movimiento. En las figura 4 se muestra como llevar a cabo dicho movimiento.



Figura 4. Movimiento “puño”.

Al igual que para el “puño”, para realizar el gesto “círculo” y que sea detectado de forma correcta el controlador debe ser sostenido con la mano como se muestra en la figura 5. Posteriormente se procede a realizar un círculo en el aire.

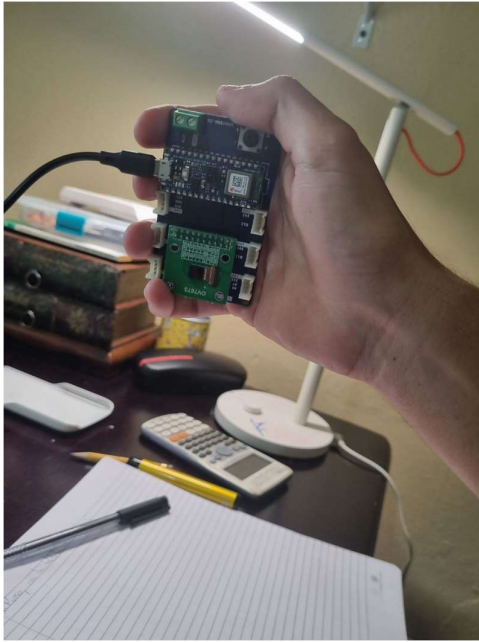


Figura 5. Movimiento círculo.

Finalmente, para el gesto flexión la tarjeta debe ser sostenida como se muestra en la figura 6. Luego se realiza un movimiento de flexión subiendo el brazo derecho.

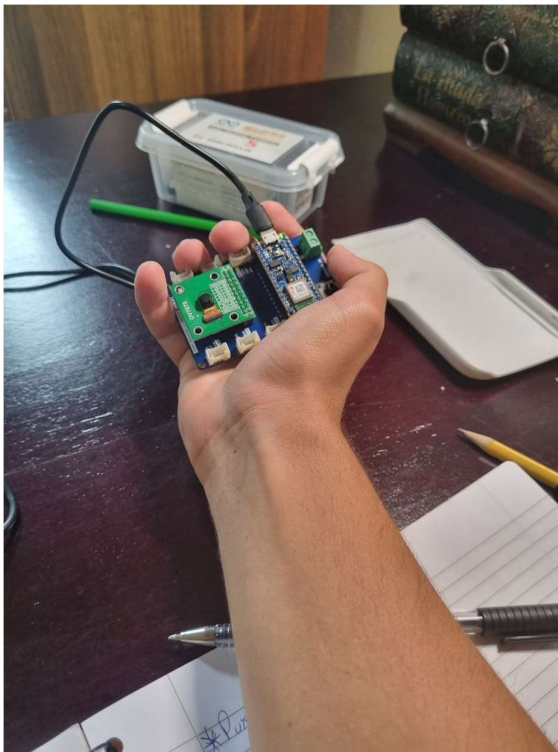


Figura 6. Gesto flexión

5. Conclusiones y recomendaciones

Con el laboratorio 5 fue posible reconocer los sensores del giroscopio y el acelerómetro del microcontrolador arduino nano ble 33, en el laboratorio 4 se utilizó de igual forma el giroscopio y acelerómetro del microcontrolador discovery kit stm32. Comparando ambos microcontroladores, es concluyente que la programación del script del arduino es más simple y flexible que con el stm32. Así mismo, el arduino nano es desde el punto de vista físico más pequeño lo cual permite utilizarlo en aplicaciones donde el espacio físico sea más reducido. De igual forma el uso y acceso de las bibliotecas es más sencillo en arduino.

La sección más relevante de este laboratorio consistió en la aplicación y el uso de machine learning para desarrollar una red neuronal de 2 capas con tres salidas, que cada salida corresponde a los gestos que fueron previamente entrenados. Es importante y de gran aprendizaje tener conocimiento básico sobre el área de “machine learning” que es una rama de la inteligencia artificial, esto debido a la relevancia que ha comenzado a tomar en los últimos años especialmente en temas de ingeniería eléctrica para llevar a cabo aplicaciones en varios campos de estudio. Otro aspecto interesante es la recopilación de datos, la gestión de datos es un campo de estudio que ha ganado importancia porque permite interpretar los datos numéricos que disponemos, con el objetivo de tomar las decisiones más eficientes.

Referencias

Arduino Nano BLE 33 [Nano 33 BLE | Arduino Documentation](#)

Tiny Machine Learning <https://docs.arduino.cc/hardware/nano-33-ble>

Anexos:

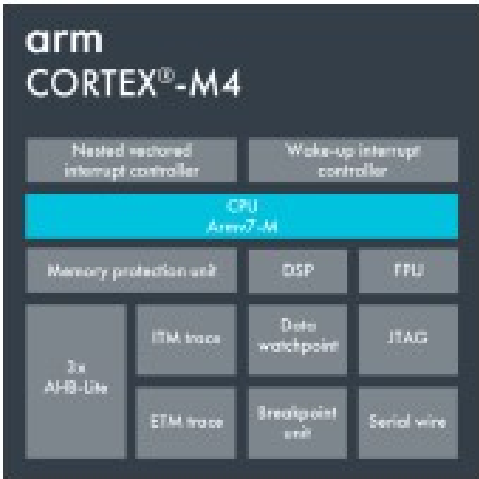


Figure 1: Block diagram of the Cortex-M4 processor

Overview

The Cortex-M4 processor is developed to address an efficient, easy-to-use blend of control and signal processing functionality. It provides the use benefits of the Cortex-M family of processors and includes motor control, automotive, power management, and automation markets.

Features

Feature	Description
Architecture	Armv7E-M
Bus Interface	3x AMBA AHB-Lite AMBA ATB interface
ISA Support	Thumb/Thumb-2
Pipeline	3-stage + branch sp
DSP Extension	Single-cycle 16/32- Single-cycle dual 16 8/16-bit SIMD arith Hardware Divide (2-
Floating-Point Unit	Optional single prec IEEE 754 compliant



Description

Nano 33 BLE is a miniature sized module containing a NINA B306 module, based on Nordic nRF52840, a 9 axis IMU. The module can either be mounted as a DIP component (when mounting pin headers) or soldered it via the castellated pads.

Target areas:

Maker, enhancements, basic IoT application scenarios