

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE-0624 Laboratorio de Microcontroladores

Laboratorio 5

STM32/Arduino: GPIO, Giroscopio, comunicaciones, TinyML

Carlos Alberto Caravaca Mora - B61512.

Erick Sancho Alvarado - B87388.

Profesor: MSc. Marco Villalta Fallas.

31 de octubre, 2023

Índice

1. Introducción.	1
2. Nota teórica.	2
2.1. Arduino Nano 33 BLE	2
2.2. Arduino IDE (2.2.1)	3
2.3. Biblioteca Arduino TensorFlowLite.	3
2.4. Edge Impulse	3
2.5. Diseño del circuito	3
3. Análisis de resultados.	4
3.1. Adquisición de información:	4
3.2. Procesamiento y Etiquetado:	4
3.3. Entrenamiento del Modelo:	4
3.4. Aplicación del Modelo:	4
3.5. Detección de Comandos y Comunicación:	5
3.6. Plataforma IoT:	5
4. Conclusiones y recomendaciones.	6

Índice de figuras

1. Topología de la placa. 2

1. Introducción.

En el presente informe se documenta la creación de un controlador por voz utilizando el kit Arduino Nano 33 BLE mediante Tiny Machine Learning. Se realizaron preparativos, incluyendo la obtención del kit, instalación de software necesario, y creación de una cuenta en Edge Impulse como trabajo previo, el proceso de desarrollo abarca la captura de comandos de voz, la implementación y entrenamiento de un modelo de red neuronal, y la ejecución de acciones asociadas a los comandos detectados.

Durante el laboratorio, se registraron tres comandos de voz, cada uno con una duración mínima de dos minutos y variaciones en entonaciones. Es destacable la importancia de ajustar la ventana de muestras para el entrenamiento de la red neuronal donde la implementación final permite la comunicación entre el microcontrolador y la PC, así como la conexión a una plataforma IoT para activar acciones correspondientes a los comandos detectados.

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

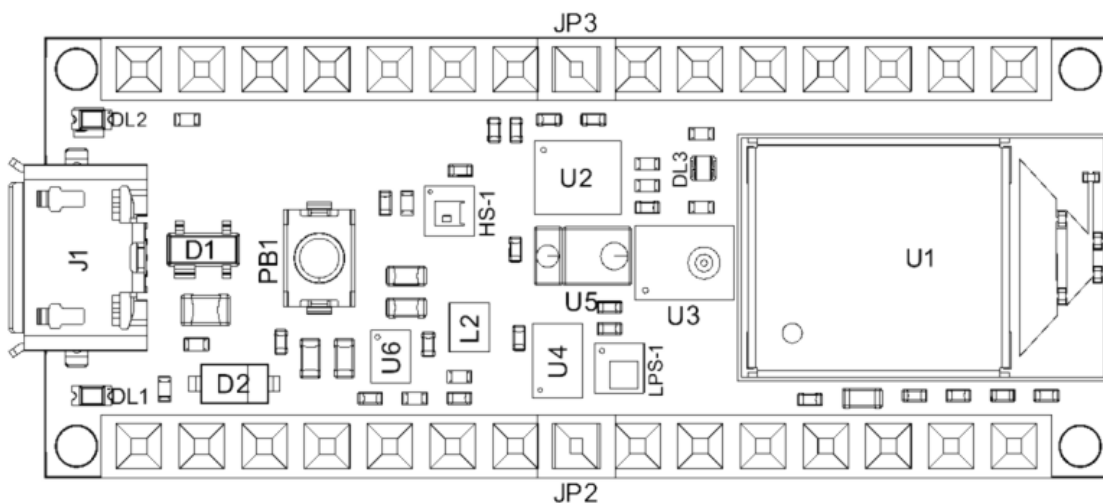
2. Nota teórica.

2.1. Arduino Nano 33 BLE

El Arduino Nano 33 BLE es una placa de desarrollo compacta y potente diseñada para aplicaciones de Internet de las cosas (IoT) y proyectos de electrónica embebida, el dispositivo se basa en el microcontrolador Nordic nRF52840, que cuenta con un procesador ARM Cortex-M4 de 32 bits y conectividad Bluetooth de baja energía (BLE). Presenta las siguientes características básicas:

- Microcontrolador: utiliza el nRF52840 de Nordic Semiconductor, que ofrece un rendimiento robusto con su arquitectura ARM Cortex-M4 y una velocidad de reloj de hasta 64 MHz.
- Conectividad Bluetooth: incorpora Bluetooth Low Energy (BLE), lo que permite la comunicación inalámbrica eficiente y de bajo consumo de energía, siendo ideal para proyectos IoT.
- Puertos y Periféricos: a pesar de su tamaño compacto, el Nano 33 BLE ofrece una variedad de pines de entrada/salida digital y analógica, así como interfaces para sensores y módulos adicionales.
- USB: posee un puerto USB que facilita la conexión y programación, permitiendo cargar programas directamente en la placa.
- Sensores Integrados: algunos modelos incluyen sensores integrados como giroscopio y acelerómetro, ampliando las capacidades para proyectos que requieren detección de movimiento.
- Compatibilidad con IDE Arduino: es compatible con el entorno de desarrollo integrado (IDE) de Arduino, lo que facilita la programación para usuarios familiarizados con la plataforma.

El Arduino Nano 33 BLE es una opción versátil para proyectos de IoT y desarrollo de dispositivos conectados, ofreciendo un equilibrio entre tamaño compacto, rendimiento y capacidades de conectividad inalámbrica y la topología de la placa se ve de la siguiente manera:



2.2. Arduino IDE (2.2.1)

El Arduino IDE 2.x se caracteriza por ser una reescritura importante y no comparte código con el IDE 1.x, es un entorno de desarrollo diseñado para facilitar la programación y carga de código en placas Arduino.. Está basado en el marco Theia IDE y construido con Electron. Las operaciones de backend, como la compilación y la carga, se descargan a una instancia arduino-cli que se ejecuta en modo demonio. Este nuevo IDE se desarrolló con el objetivo de preservar la misma interfaz y experiencia de usuario de la versión principal anterior para proporcionar una actualización sin fricciones. [1]

2.3. Biblioteca Arduino TensorFlowLite.

TensorFlowLite es una versión ligera de TensorFlow, un marco de aprendizaje automático desarrollado por Google. La versión Lite está diseñada específicamente para dispositivos con recursos limitados, como microcontroladores, y la biblioteca Arduino TensorFlowLite se enfoca en hacer que esta tecnología sea accesible en placas Arduino. La biblioteca es compatible con diversas placas Arduino, incluyendo el Arduino Nano 33 BLE utilizado en el laboratorio mencionado. Permite aprovechar capacidades de aprendizaje automático en proyectos de IoT y dispositivos embebidos, se puede ejecutar en dispositivos pequeños como microcontroladores. [2] Está diseñado para ejecutar modelos de aprendizaje automático en dispositivos con memoria limitada, las características de uso son las siguientes:

- Realizar predicciones en modelos entrenados.
- Clasificar audio.
- Identificar acciones humanas en vídeo.
- Entrenar a un agente de juego.
- Extraer texto de imágenes.

2.4. Edge Impulse

Edge Impulse es una plataforma integral de desarrollo de aprendizaje automático (ML) centrada en la implementación eficiente de modelos de machine learning en dispositivos edge, como microcontroladores y sistemas integrados. La plataforma simplifica el proceso de desarrollo y ofrece herramientas para la captura y etiquetado de datos, entrenamiento de modelos además de desarrollo sin codificación. Cuenta con una interfaz gráfica amigable que permite a los usuarios, incluso sin experiencia en programación de machine learning, crear modelos personalizados y diseñar pipelines de procesamiento de señales. Edge Impulse también facilita la exportación de modelos a diversas plataformas de hardware, incluyendo placas Arduino, generando código optimizado para su implementación.[3]

2.5. Diseño del circuito

Se utilizan los siguientes componentes para realizar el laboratorio:

Componente	Tipo	Cantidad	Precio ¢
Arduino Nano 33 BLE [?]	Kit de Tiny Machine	1	13244.61 c/u
Protoboard [?]	ensamble a presión, 1 bloque y 2 tiras	1	5400 c/u
Batería [4]	Pila alcalina "9V" (cuadrada)	1	2 300 c/u
Broche [5]	Porta pila "9V"	1	260 c/u
Cables [6]	Juego de 80 cables de 15 cm tipo Dupont	1	2 990 c/u

Tabla 1: Lista de componentes utilizados.

3. Análisis de resultados.

En la sección de resultados, podemos mencionar que no logramos implementar el laboratorio, por ende, se mencionaran aspectos teóricos de lo que pudo haber sido este laboratorio en caso de si implementarse de manera correcta.

3.1. Adquisición de información:

En esta fase, la meta era programar el Arduino Nano 33 BLE para capturar información del micrófono y enviarla a la computadora a través del puerto USB. La implementación típicamente involucraría la configuración de objetos de audio, como el objeto `AudioInputI2S`, que se conecta al micrófono, y el objeto `AudioOutputUSB`, que permite la transmisión de datos al puerto USB. La expectativa era que esta configuración permitiera al Arduino funcionar como un dispositivo de adquisición de audio, capturando señales del micrófono y facilitando su transmisión a la computadora para su posterior procesamiento.

3.2. Procesamiento y Etiquetado:

En la siguiente etapa, se requería un script de Python para procesar los datos del micrófono y etiquetarlos según el tipo de comando de voz. La expectativa era la creación de un archivo que contuviera datos de audio etiquetados con los comandos específicos, como "luces", "televisor", "puerta", etc., junto con períodos de ruido de fondo. Esto implicaría la utilización de bibliotecas como `sounddevice` para la captura de audio y `numpy` para el manejo eficiente de los datos. El script debería haber sido capaz de diferenciar entre las distintas categorías de comandos y etiquetar los datos de manera apropiada.

3.3. Entrenamiento del Modelo:

En el paso de entrenamiento del modelo, la expectativa era desarrollar un modelo de TensorFlow Lite o utilizar Edge Impulse para cargar, configurar y entrenar una red neuronal con los datos etiquetados previamente. Esto implica la construcción de una arquitectura de red neuronal, la compilación del modelo con funciones de pérdida y optimización adecuadas, y finalmente, el entrenamiento del modelo con los datos de audio etiquetados. La expectativa era que el modelo aprendiera patrones distintivos asociados con cada comando de voz y pudiera generalizar correctamente a nuevas muestras.

3.4. Aplicación del Modelo:

Una vez que el modelo estaba entrenado, se esperaba exportarlo correctamente al Arduino Nano 33 BLE. Esto implicaría el uso de herramientas proporcionadas por TensorFlow Lite o

Edge Impulse para convertir el modelo entrenado en un formato compatible con el microcontrolador. La exportación exitosa permitiría que el modelo fuera incorporado en el código del Arduino y utilizado para la detección de comandos de voz.

3.5. Detección de Comandos y Comunicación:

La fase de detección de comandos implicaba la implementación de un programa en el microcontrolador que utilizara el modelo previamente entrenado para reconocer el tipo de comando realizado. Se esperaba que el microcontrolador pudiera capturar el audio del entorno, aplicar el modelo y luego comunicar los resultados a la PC, indicando qué comando de voz había sido detectado.

3.6. Plataforma IoT:

Finalmente, la integración con la plataforma IoT requería el envío de comandos reconocidos a una plataforma como Thingsboard y activar un widget que representara la acción asociada. La comunicación entre el microcontrolador y la plataforma IoT sería esencial para lograr esta integración, y se esperaba que la plataforma reflejara de manera efectiva los comandos recibidos.

4. Conclusiones y recomendaciones.

- El laboratorio estuvo muy interesante, se utilizó la misma placa del laboratorio, lo cual estuvo interesante para finalizar el curso de esta manera. Esto facilitó un poco la realización del mismo.
- La planificación teórica destaca la necesidad de un enfoque completo desde la adquisición de datos hasta la integración con IoT, subrayando la importancia de una estrategia detallada para el éxito del proyecto.
- El análisis teórico señala factores críticos como la adquisición de datos, el entrenamiento del modelo y la integración con IoT, subrayando la importancia de abordar estos aspectos desde el principio.
- Aunque la implementación no se llevó a cabo, el análisis teórico proporciona una perspectiva de como pudo haber sido la implementación real.

Referencias

- [1] Arduino IDE. Arduino IDE 2.2.1. <https://www.arduino.cc/en/software>.
- [2] TensorFlow. TensorFlow Lite para microcontroladores. <https://www.tensorflow.org/lite/microcontrollers?hl=es-419>, 2023.
- [3] edgeimpulse. Edge Impulse. <https://edgeimpulse.com/>, 2023.
- [4] steren.cr. Pila alcalina "9V"(cuadrada). <https://www.steren.cr/pila-alcalina-9v-cuadrada.html>.
- [5] steren.cr. Broche porta pila "9V". <https://www.steren.cr/broche-porta-pila-9v.html>.
- [6] steren.cr. Juego de 80 cables de 15 cm tipo Dupont. <https://www.steren.cr/juego-de-80-cables-de-15-cm-tipo-dupont.html>.