



Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0624 Laboratorio de Microcontroladores

EIE
Escuela de
Ingeniería Eléctrica

MSc. Marco Villalta Fallas - II Ciclo 2022

Laboratorio # 5

STM32/Arduino: GPIO, Giroscopio, comunicaciones, TinyML

Instrucciones Generales:

Este laboratorio se puede realizar de manera individual o en pareja. Se debe utilizar git para registrar el avance y aporte de cada estudiante(aquel estudiante que no registre aportes/contribuciones tendrá una nota de cero), en caso de hacer el laboratorio en pareja se debe utilizar un mismo repositorio.

El laboratorio debe entregarse antes del 27 de noviembre a las 23:59.

Utilice capturas de pantalla para demostrar la funcionalidad, estas capturas de pantalla deben mostrar sólo la información pertinente al paso correspondiente.

Entregue un archivo comprimido que incluya un directorio llamado **informe** con los archivos necesarios para generar el PDF del informe (.tex, imágenes, código, entre otros) y un directorio llamado **src** con los archivos de código fuente que lleven a la solución. Cualquier otro formato o entrega tardía no se revisará y el laboratorio tendrá una nota de cero.

Trabajo previo

Antes de empezar con el desarrollo de este laboratorio deberá realizar previamente los siguientes puntos:

- En caso de utilizar una tarjeta SMT32F429 Discovery:

- Retirar de la bodega una tarjeta de desarrollo STM32F49 Discovery, un cable micro USB - USB para la comunicación con la computadora y un cable mini USB - USB para la alimentación de la tarjeta.
- Instalar la biblioteca LibOpenCM3, los ejemplos de la biblioteca LibOpenCM3, revisar los ejemplos, instalar para Python3 varios módulos(Pandas, TensorFlow, matplotlib, numpy), un compilador gcc-arm y el programador ST-Link.
- Revisar la guía en <https://github.com/MarkoSagadin/MicroML>, compilar los ejemplos provistos por el profesor para la tarjeta SMT32F49.
- Crear una cuenta en <https://edgeimpulse.com/>.

- En caso de utilizar un kit Arduino Nano 33 BLE:

- Retirar de la bodega un kit de Tiny Machine Learning - Arduino Nano 33 BLE
- Instalar la última versión del Arduino IDE (2.0.1) y el soporte Arduino Mbed OS Nano boards(v3.4.1)
- Crear una cuenta en <https://edgeimpulse.com/>.

HAR - Human Activity Recognition

Se deberá desarrollar un reconocedor de actividad humana. Por lo tanto debe cumplir con los siguientes requisitos:

1. Utilizar una placa Discovery STM32F49 o un kit Arduino Nano 33 BLE.
2. Realizar un programa para el microcontrolador que capture la información del giroscopio y se envíe a la computadora por el puerto USB.

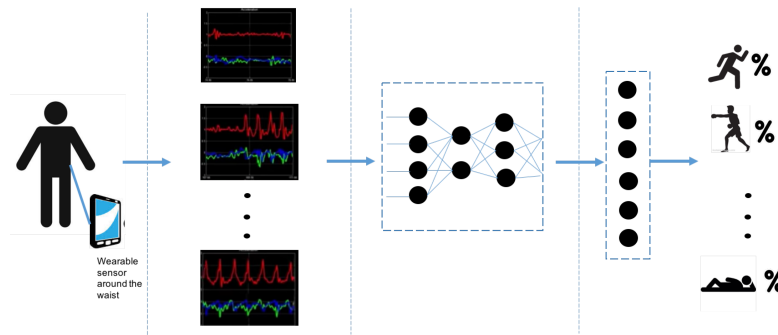


Figura 1: HAR

3. Realizar un script de Python que guarde la información recibida del giroscopio y que se encuentre etiquetada con el tipo de movimiento que se efectúa, el script debe guardar esta información en un archivo CSV.
4. Debe registrar 3 movimientos (por ejemplo: mover brazo hacia arriba o hacia abajo, mover brazo en círculos, golpe de brazo, estacionario, etc) por un periodo de tiempo (p.e. 5-15 segundos). Eso mientras se ejecuta el programa de registro de giroscopio en conjunto con el script de comunicaciones a la PC.
5. Realizar un script de Python para crear un modelo de TensorFlow lite con lo siguiente:
 - a) Que cargue la información de los movimientos guardados en el archivo CSV del paso anterior
 - b) Configure una red neuronal (p.e. una red 2 capas (una de entrada y una oculta/intermedia) con varias neuronas cada una y 3 salidas (una por cada tipo de movimiento realizado), usando ReLU como función de activación (excepto para la de salida que se recomienda softmax), algoritmo de optimización rmsprop y métrica de pérdida *mae*).
 - c) Entrene la red con el 60 % de los datos, se utilizará el 20 % de los datos para validar y el resto 20 % para pruebas.
 - d) Exporte el modelo obtenido con TensorFlow Lite.
6. Realizar un programa que utilice el modelo construido en el paso anterior para detectar el tipo de movimiento realizado, los movimientos detectados deben de quedar registrados en la PC por lo que se debe realizar la comunicación entre el microcontrolador y la PC.

Algunas observaciones adicionales:

- Escoger suficiente memoria para los tensors (varios megas, en caso contrario habría problemas con la reserva de memoria)
- La cantidad de muestras por movimiento deberían ser aproximadamente las mismas.
- Se debe ajustar la ventana de muestras para entrenar la red puesto que puede alterar la eficiencia y resultados
- El orden de las bibliotecas es importante para la compilación (STM32)
- Utilice los ejemplos vistos en clase.