

# Inteligencia artificial on the edge

Jorge Cuartero Galindo, Belén Calvo López, Nicolas Medrano Marqués

Grupo de Electrónica de Potencia y Microelectrónica (I3A)  
Universidad de Zaragoza, Pedro Cerbuna 12, 50009, Zaragoza, Spain.  
Tel. +34-876553427, e-mail: [819569@unizar.es](mailto:819569@unizar.es)

## Resumen

El presente trabajo busca desarrollar un prototipo capaz de detectar movimientos y posiciones de la mano para su uso como sistema de apuntado en un ordenador. Para ello, la adquisición y procesado de los datos se llevará a cabo con un microprocesador que hará uso una red neuronal.

## Desarrollo del prototipo

El desarrollo de este prototipo se dividirá en 4 fases: interfaz hardware de adquisición de datos, procesado, envío y ejecución de las órdenes correspondientes en el receptor.

### Interfaz hardware

El dispositivo (Fig. 1) irá montado sobre un guante debido a su aplicación. Está constituido por unas bandas piezorresistivas encargadas de determinar la flexión de los dedos; un bloque central encargado de sujetar todas las piezas y alojar el microcontrolador (Arduino RP2040 Connect [1]); y una batería encargada de alimentar el sistema.

Los datos a adquirir serán los correspondientes a la flexión de los dedos y la dirección donde señala la mano. Para lo primero se conformará un divisor de tensión con los piezorresistivos (Flex Sensor de Spectra Symbol) y una resistencia fija (Fig. 2), de modo que una flexión de los dedos se traduce en un valor determinado de voltaje que mediremos con los ADC del microcontrolador (Fig. 3).

Para la orientación de la mano, haremos uso del módulo LSM6DSOXTR del microcontrolador, una unidad de medición inercial (IMU) de seis ejes que incluye un acelerómetro y un giróscopo.

### Procesado de los datos

Para los datos referentes a la orientación, primero tendremos que calibrar el sistema, corrigiendo errores de offset, ruido, cuadratura, así como

reescalar los datos. A continuación, definimos un sistema de coordenadas para convertir los datos obtenidos (aceleraciones y velocidades angulares), en los ángulos correspondientes. Debido a que se busca poder señalar diferentes posiciones en una pantalla, definimos el ángulo respecto al eje horizontal (eje de la gravedad), y el ángulo en el plano vertical, que se corresponderán con los ejes X e Y de la pantalla.

Para los datos referentes a la flexión de los dedos, desarrollamos una red neuronal que a partir de la información proporcionada por cada dedo sea capaz de identificar 10 posiciones definidas previamente. Debido a la naturaleza del problema, entrenamos a la red mediante el método de *one-hot encoding* [2].

Para la obtención del *dataset* de entrenamiento, se ha desarrollado un programa que, a través de la pantalla de un PC solicita al usuario las diferentes posiciones. Tras unos segundos, las medidas de los dedos son establecidas y almacenadas (Fig. 3) junto a una etiqueta de la posición correspondiente. Este proceso se ejecutó en varios días y posiciones distintas del brazo para maximizar la adaptabilidad de la red. Finalmente se dispuso de 1953 ejemplos, destinados el 80% al entrenamiento, un 10% a validación y el 10% restante al test.

La red fue diseñada y entrenada en Python, empleando la librería Tensor Flow. Entre los objetivos propuestos, se priorizó mantener una alta exactitud en la identificación de los patrones ocupando la menor cantidad de memoria posible, de forma que pudiese integrarse en un microcontrolador. Para la implementación en el microcontrolador [3], usamos TensorFlow lite. Esta librería simplifica la definición de una red y reduce la resolución de los parámetros del modelo (los pesos), obteniendo un fichero de definición compatible con nuestro dispositivo a través de la librería Arduino adecuada [4].

Tras diferentes pruebas se concluye que una arquitectura 4:40:20:10 (menos de 4 kb) obtiene los resultados deseados.

### Envío de los datos por RF

El siguiente paso es hacer llegar los datos ya procesados (2 ángulos y la posición de la mano) al ordenador. Para ello decide emplear el protocolo WiFi, usando un dispositivo intermedio como *hotspot* (el teléfono móvil por simplicidad) al cual se conectarán el ordenador y el microcontrolador.

Además, debido a que nos interesa que los datos lleguen en tiempo real, aunque perdamos algunos en el proceso, optemos por el protocolo UDP, que no requiere de verificación de la red.

### Interfaz de control remoto

Una vez recibidos los datos en el ordenador, son interpretados por un programa en Python. La librería *win32api* se utiliza para convertir la información de los ángulos en órdenes para el ratón, indicando con la

posición de los dedos instrucciones como la presión en los botones del ratón o el giro de la rueda.

Asimismo, una interfaz gráfica desarrollada con PyQt5 accesible con distintas posiciones de la mano servirá para acceder a opciones adicionales del dispositivo (Fig. 4).

## Conclusiones

Este trabajo integra múltiples ámbitos como la inteligencia artificial, computación on the edge o las comunicaciones para lograr la implementación de un sistema capaz de ejecutar múltiples órdenes en un ordenador a partir de movimientos de la mano.

## REFERENCIAS

- [1]. <https://docs.arduino.cc/hardware/nano-rp2040-connect>, Último acceso 03/04/2024
- [2]. <https://medium.com/thedeephub/one-hot-encoding-with-tensorflow-and-keras-569cfa91b162>, Último acceso 25/03/2024
- [3]. Gian Marco Iodice, "TinyML Cookbook". Packt publishing Ltd., 2022, Birmingham
- [4]. Simone Salerno, <https://github.com/cloquentarduino>. Último acceso 22/03/2024

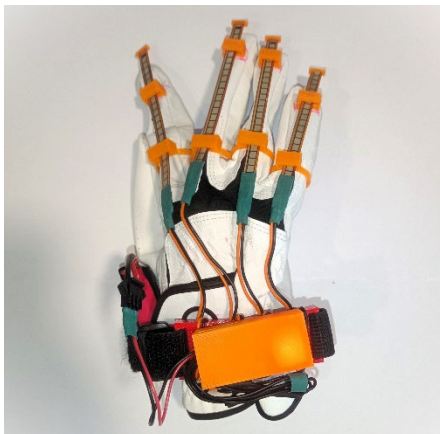


Figura 1. Imagen del montaje final del guante.

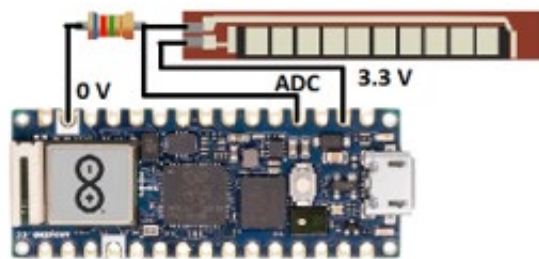


Figura 2. Esquema del divisor de tensión de los piezorresistivos

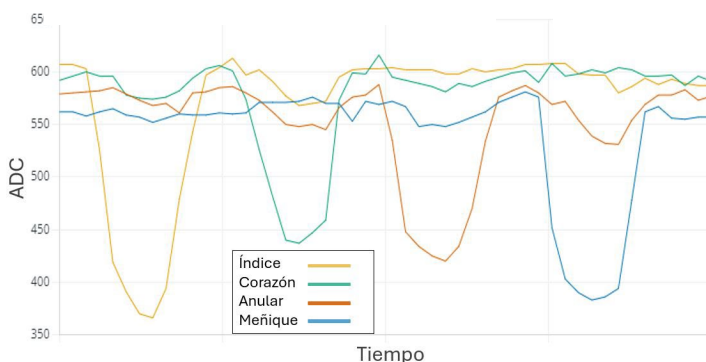


Figura 3. Valores obtenidos en el ADC al flexionar sucesivamente los dedos

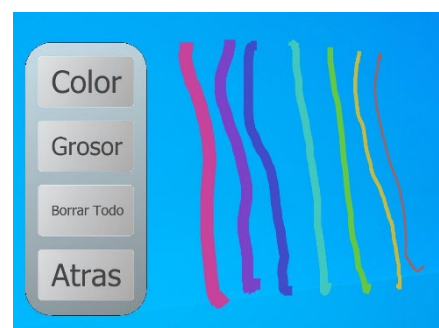


Figura 4. Captura de la función "Pintar" accesible desde el menú.