

Jornadas de Automática

Optimización del registro en tiempo real de sensor de fuerza

Sempere, J.M.^a, Gracia, D.I.^{b,c}, Azorín, J.M.^{b,c,d}, Úbeda, A.^a, Iáñez, E.^{b,c,*}

^aHuman Robotics Group, Universidad de Alicante, Crta. de San Vicente del Raspeig S/N, 03690, San Vicente del Raspeig, Alicante, España.

^bBrain-Machine Interface Systems Lab, Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, 03202 Elche, España.

^cInstituto de Investigación en Ingeniería de Elche-I3E, Universidad Miguel Hernández de Elche, Avda. de la Universidad s/n, 03202 Elche, España.

^dValencian Graduate School and Research Network of Artificial Intelligence – valgrAI, Camino de Vera, s/n, Edificio 3Q, 46022 Valencia, España.

To cite this article: Sempere, J.M., Gracia, D.I., Azorín, J.M., Úbeda, A., Iáñez, E. 2024. Optimization of real-time force sensor recording. *Jornadas de Automática*, 45. <https://doi.org/10.17979/ja-cea.2024.45.10934>

Resumen

El estudio aborda la necesidad de cuantificar la presión superficial con baja latencia en la transmisión de datos. Utilizando el sensor de fuerza resistivo FSR 406 y el microcontrolador ESP32 se ha desarrollado una arquitectura para optimizar el envío y registro de datos. La arquitectura incluye la activación de triggers basados en umbrales de voltaje o botones pulsadores, permitiendo la sincronización precisa entre dispositivos. El sistema envía datos uno a uno o bloques de cadenas, asegurando la integridad y continuidad de la información. Además, se muestran ejemplos donde se logra una sincronización efectiva, facilitando la comparación de la presión superficial con otras señales fisiológicas sin generar desfases. Este estudio optimiza el registro y transmisión de datos, mejorando la precisión y eficiencia en aplicaciones fisiológicas y biomecánicas y, demostrando una arquitectura robusta para la sincronización y análisis de múltiples dispositivos.

Palabras clave: Redes de sensores, Diseño lógico, diseño físico e implementación de sistemas informáticos embebidos, Algoritmos en tiempo real, planificación y programación, Consolidación de señales de sensores, Desarrollos en medición y procesamiento de señales

Optimization of real-time force sensor recording

Abstract

The study addresses the need to quantify surface pressure with low latency in data transmission. Using the FSR 406 resistive force sensor and the ESP32 microcontroller, an architecture has been developed to optimize data sending and recording. The architecture includes triggers activation based on voltage thresholds or push buttons, allowing precise synchronization between devices. The system sends data one by one or in blocks of strings, ensuring the integrity and continuity of the information. In addition, examples are shown where effective synchronization is achieved, facilitating the comparison of surface pressure with other physiological signals without generating time lags. This study optimizes data recording and transmission, improving accuracy and efficiency in physiological and biomechanical applications and demonstrating a robust architecture for synchronization and analysis of multiple devices.

Keywords: Sensor networks, Logical design, physical design, and implementation of embedded computer systems, Real-time algorithms, scheduling, and programming, Sensor signal consolidation, Developments in measurement, signal processing

*Autor para correspondencia: eianez@gmail.com

1. Introducción

La precisión y eficiencia en la recogida y procesamiento de datos biomédicos son fundamentales, especialmente en aplicaciones que requieren mediciones exactas y en tiempo real (Agarwal et al., 2022). Por ello, este trabajo se enfoca en la optimización de la cuantificación de la presión superficial mediante el uso del sensor de fuerza resistivo FSR 406 y el microcontrolador ESP32, desarrollando una arquitectura que optimiza tanto el empaquetamiento como el registro de datos.

En la búsqueda de métodos eficaces y económicos para medir la fuerza y el contacto, se han seleccionado los sensores de fuerza resistiva (FSR) debido a su bajo costo y su alta capacidad de transmisión de datos. En el estudio realizado por (Sadun et al., 2016) se demuestra la efectividad de los sensores de fuerza resistiva como sensores de bajo coste. Además, en el análisis de (Giovanelli and Farella, 2016), esta tecnología sobresale por su eficiencia y coste económico, ofreciendo una alternativa viable frente a tecnologías más costosas como las celdas de carga o los sensores piezoeléctricos.

Por otro lado, se ha seleccionado el microcontrolador ESP32 por su destacada eficiencia en el manejo de múltiples flujos de datos en tiempo real tal y como muestran (Foltýnek et al., 2019). Esta característica lo convierte en una herramienta idónea para diversas aplicaciones en el ámbito del Internet de las Cosas (IoT), desde la automatización del hogar hasta el monitoreo continuo de señales, como respaldan (Babiuch et al., 2019) en su artículo.

Con una arquitectura que optimiza tanto el empaquetamiento como el registro de la presión superficial, el presente trabajo pretende establecer un sistema estándar que sea directamente aplicable en escenarios clínicos o de investigación, proporcionando datos precisos y sin introducir desfases en otras señales fisiológicas.

2. Material y métodos

En este apartado se describe el hardware utilizado, principalmente el sensor de fuerza cuadrado FSR 406 y el microcontrolador ESP32 así como su calibración y registro en tiempo real desde Matlab.

2.1. FSR 406 y selección de resistencia de 330Ω

El sensor FSR 406 es un sensor de fuerza resistivo cuadrado de 43.69x43.69mm ((Electronics, 2024)). Pertenece a una familia de sensores que muestran un decrecimiento de la resistencia de forma no lineal con el incremento de la presión ejercida (ver Figura 1).

Su tiempo de activación es menor a 3μs y, para realizar la conversión fuerza-voltaje, el dispositivo FSR se conecta a una resistencia en una configuración de divisor de tensión siguiendo la siguiente ecuación:

$$V_{OUT} = \frac{R_M \cdot V_+}{R_M + R_{FSR}} \quad (1)$$

donde:

- V_{OUT} es el voltaje de salida obtenido del sensor de fuerza resistivo.

- V_+ es el voltaje de entrada al circuito.
- R_M es la resistencia de medida que permanece fija.
- R_{FSR} es la resistencia variable del sensor de fuerza resistivo, que cambia su resistencia en función de la fuerza aplicada.

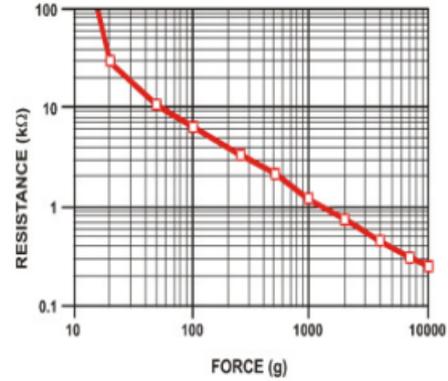


Figura 1: Curva fuerza del sensor FSR 402 (Electronics, 2024). Muestra la correlación entre la resistencia generada por el sensor frente a la fuerza aplicada en su superficie.

En esta configuración, a medida que la fuerza aplicada al FSR aumenta, R_{FSR} disminuye, lo que provoca que el voltaje V_{out} aumente si se mantiene constante el voltaje de entrada V_+ y la resistencia R_M . De esta manera, incrementar R_M aumenta la sensibilidad del sensor, permitiendo detectar cambios más pequeños en la fuerza con mayor precisión (ver Figura 2).

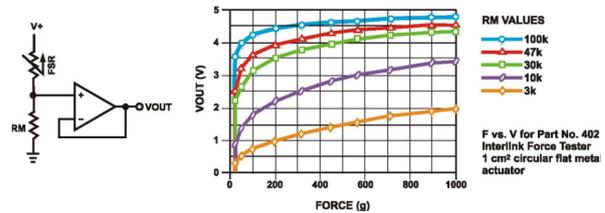
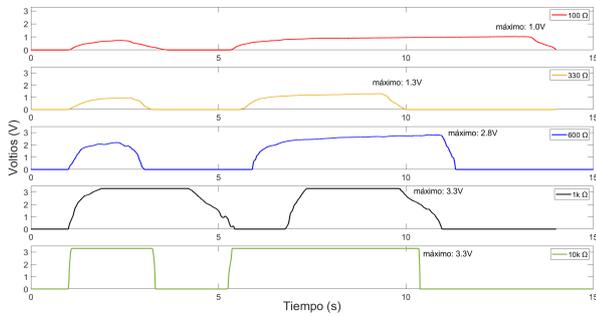


Figura 2: Curvas de voltaje de salida en función de las resistencias utilizadas. *Gráfica orientativa del modelo FSR 402 (Electronics, 2024)

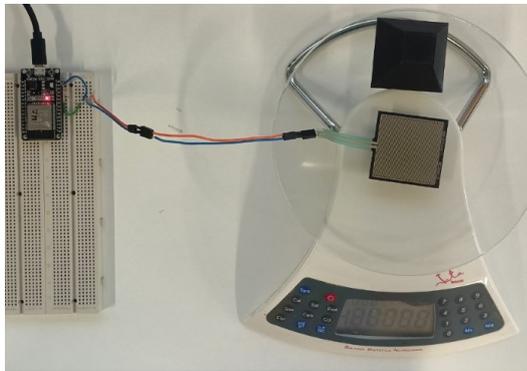
El valor buscado teniendo en cuenta la Figura 2 para dicha resistencia es de 10kΩ ya que, de esta forma, se logra una alta sensibilidad del sensor frente a fuerzas mínimas y, además, para fuerzas más moderadas, los valores obtenidos tienen un mayor rango de valores y mayor linealidad antes de terminar saturando.

Sin embargo, en la práctica, al utilizar una resistencia de 10kΩ se obtuvieron registros que saturaban demasiado rápido el sensor (ver Figura 3(a)) por lo que se valoraron resistencias menores a costa de perder sensibilidad. Para esta valoración, se aplicó sobre el sensor FSR 406 una fuerza de 1 kg para determinar la sensibilidad frente a presiones leves y, posteriormente, se volvió a aplicar una fuerza de hasta 3 kg para comprobar la saturación del sensor. Para llevar a cabo la aplicación de las fuerzas se utilizó una báscula que soportaba hasta 3 kg (ver Figura 3(b)). Tras comprobar la sensibilidad para

resistencias de 100Ω , 330Ω , 660Ω , $1k\Omega$ y $10k\Omega$, se decidió utilizar una resistencia de 330Ω ya que demostró una buena sensibilidad y la saturación del sensor se alcanzaría en fuerzas muy elevadas (ver Figura 3(a)). Además, ante una presión rápida sobre el sensor puede obtenerse una curva más definida y lineal frente a una saturación prácticamente instantánea del sensor cuando se utilizan resistencias de mayor valor.



(a) Sensibilidad obtenida para $100(\Omega)$, $330(\Omega)$, $600(\Omega)$, $1k(\Omega)$ y $10k(\Omega)$.



(b) Setup comprobación sensibilidad sensor FSR 406.

Figura 3: Método y resultados obtenidos de la sensibilidad del sensor FSR 406 en función de la resistencia para la configuración de divisor de tensión.

Por otro lado, es probable que la diferencia tan grande entre los datos teóricos y los obtenidos en la práctica se deban a la diferencia entre las superficies de los sensores FSR 402 y FSR 406, ya que el sensor FSR 402 presenta una superficie efectiva de 169.17 mm^2 y el sensor FSR 406 presenta una superficie efectiva de 1568.16 mm^2 .

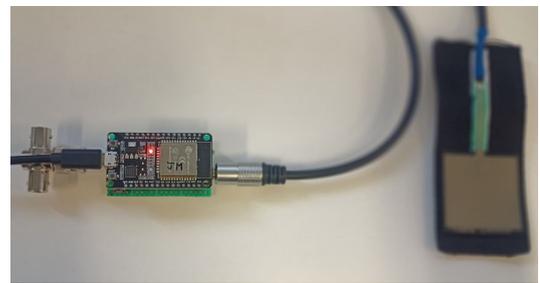
2.2. Microcontrolador ESP32

El microcontrolador ESP32 se eligió por su distinción frente a las placas Arduino convencionales. Estas últimas suelen tener velocidades de reloj de 16 MHz, con un solo núcleo de procesamiento. En contraste, el microcontrolador ESP32 cuenta con dos núcleos de procesamiento de 240 MHz cada uno (Kareem and Dunaev, 2021). Esta característica permite una gestión más eficiente de múltiples tareas al ejecutar procesos en paralelo. Además, la conexión entre el microprocesador y el ordenador se establece de forma directa a través de un puerto COM para reducir cualquier tipo de latencia en el envío de los datos. Aunque el módulo ESP32 dispone de conexión wifi y bluetooth, se ha optado por el envío de datos mediante el puerto COM para asegurar un envío sin retardos y sin pérdidas de los datos y asegurar una mejor sincronización.

La Figura 4 ilustra el sistema final, que integra el microcontrolador ESP32 y el sensor de fuerza FSR 406. En la Figura 4(a), se muestra el microcontrolador ESP32 montado en el circuito impreso, equipado con un botón para activar los triggers y conectores coaxial y jack para medir los voltajes del sensor. La Figura 4(b) destaca al sensor FSR 406 y un adaptador coaxial que permite el envío de triggers a varios dispositivos externos.



(a) Microcontrolador ESP32



(b) Microcontrolador ESP32 y sensor FSR 406

Figura 4: Versión final sistema de registro y envío de datos a través del sensor FSR 406 y el microcontrolador ESP32.

El sensor de fuerza, junto con la resistencia de medida, están conectados en forma de divisor de tensión a un pin de entrada de conversión analógico digital del microcontrolador ESP32. Este dispone de 12 bits de resolución, es decir 4096 unidades sobre los 3.3V máximos de registro. No obstante, al utilizar una resistencia de medida 330Ω para conseguir que la medida registrada del sensor de fuerza sea más lineal, reducimos el máximo voltaje a sólo 1.3V, lo que implica una reducción de la precisión en los datos. Por tanto, en este caso, los datos del sensor de fuerza tendrán únicamente 1614 valores posibles para representar los valores registrados. Con un rango de fuerza máximo del sensor de 100 Newtons tenemos una resolución de 0.06 Newtons por bit. Para las aplicaciones que está destinado, bien para conocer el instante de inicio de una fuerza o conocer de forma aproximada la fuerza ejercida, esta resolución es suficiente. En caso de requerir una mayor precisión y calidad la solución radica en utilizar sensores de fuerza con costes muy superiores a la solución económica planteada.

3. Implementación

Para poder entender el correcto registro y envío de los datos analógicos detectados por el sensor FSR 406, primero es crucial entender cuál es el flujo de la comunicación entre el ordenador y el microcontrolador ESP32.

En primer lugar, se debe establecer una conexión a través del puerto serial o puerto COM desde el software (en este caso

Matlab).

Una vez realizada la conexión, el microcontrolador ESP32 se encuentra a la espera de recibir como parámetros de entrada la frecuencia de muestreo a la que se desea trabajar (por defecto 2 kHz) y el intervalo de transmisión, es decir, el período de tiempo donde debe almacenar los datos y enviarlos al receptor (por defecto, 0.5 segundos).

Una vez que se han enviado ambos parámetros de entrada, el microcontrolador ESP32 estará a la espera de recibir el comando para comenzar a registrar y enviar los datos al sistema cuando se cumpla el intervalo de tiempo que se le ha especificado. Además, a partir de ese momento tendrá la capacidad de enviar triggers tanto al sistema como a los dispositivos que tenga conectados por conectores coaxiales siempre y cuando se cumpla el condicional especificado en el microcontrolador. Por tanto, desde el momento que recibe los parámetros de configuración, el microcontrolador ESP32 ya está registrando y comprobando la señal del sensor, pero sin almacenar los datos hasta recibir la señal de inicio.

3.1. Envío de triggers

Actualmente, se han tenido en cuenta 3 opciones distintas para activar el envío del triggers, y se puede variar entre ellas en función de la versión cargada en el microcontrolador:

- **Activación por botón:** Este condicional se activa al pulsar un botón que ha sido añadido en el circuito de la placa PCB y que está conectado a uno de los pines convertidores de analógico a digital (ADC) del microcontrolador ESP32.
- **Superación de umbral:** Este condicional se activa cuando el sensor envía voltajes superiores a un umbral especificado (por defecto 0.24V, que corresponde aproximadamente a un 18 % del máximo de fuerza medible).
- **Superación de umbral y activación por botón:** Este condicional combina las dos técnicas ya comentadas anteriormente, por lo que el envío del trigger únicamente se produce cuando se sobrepasa el umbral de fuerza y se pulsa simultáneamente el botón del microcontrolador.

3.2. Envío de datos

Una vez recibidas la frecuencia de muestreo y el intervalo de transmisión, el microcontrolador está a la espera de recibir la señal de inicio para comenzar el envío de los datos del sensor FSR 406.

Una vez enviada la señal, el microcontrolador va a seguir el flujo de ejecución que se puede observar en la Figura (5), donde constantemente el microcontrolador ejecutará el bloque llamado "LOOP" donde estará a la espera de recibir alguna lectura desde el puerto serial. En este punto se está comprobando si se ha recibido el comando para iniciar o detener el envío de datos. Si no se da el caso, comprobará si debe enviar los datos, el trigger o ambos. Los datos se enviarán cuando se haya registrado el tiempo indicado en la configuración y el registro esté activo. Mientras que el trigger se activará siempre que el registro esté activo y se supere el umbral indicado.

Por otro lado, el bloque llamado "INTERRUPCIÓN" es una función que se ejecuta en paralelo al bloque principal y su

intervalo de activación va en función de la frecuencia de muestreo especificada en el sensor. Esta función es la encargada de registrar los valores del sensor y comprobar si la condición del trigger y el tiempo del intervalo de transmisión se han cumplido. Es decir, si se hubiera especificado una frecuencia de muestreo de 2000 Hz y un intervalo de transmisión de 0,5 s, la función de interrupción se va a activar cada $\frac{1000000}{2000} = 0,5$ ms y cuando se hayan almacenado $2000 \times 0,5 = 1000$ datos, se realizará el envío desde el "LOOP".

Además, en función de la forma en la que se realice el envío y recepción de los datos se puede distinguir entre 2 tipos distintos:

- **Envío de datos uno a uno:** En esta variante, una vez se han registrado los datos indicados en el intervalo de transmisión a la frecuencia de muestreo indicada se procederá al envío de todos los datos al cliente (Matlab). Por ejemplo a 2 kHz y con un intervalo de 0.5 segundos será necesario enviar 1000 valores del sensor. En este caso se envían de forma consecutiva y por tanto el cliente debe procesar 1000 recepciones en cada intervalo de 0.5 segundos.
- **Envío de datos por bloques:** En esta segunda variante, la carga computacional en el envío de los datos se distribuye entre el microprocesador ESP32 y el sistema de recepción de los datos haciendo que el microcontrolador transforme los valores en cadenas y los concatene separándolos con un carácter específico para, posteriormente almacenar estas cadenas en un conjunto de cadenas divididas por bloques para segmentar el envío de datos. Cuando todas las posiciones de este conjunto de cadenas se completan, el microcontrolador comienza con el envío de cada una de las cadenas mientras que el registro simultáneo de los nuevos valores recibidos del sensor comienzan a guardarse en otro conjunto de cadenas auxiliar para no tener pérdidas en los datos. Con este método, si la frecuencia de muestreo del sensor y el intervalo de transmisión siguen siendo de 2 kHz y 0,5 s respectivamente, cada 0,5 s el sistema debe procesar 50 recepciones de bloques de cadenas de 20 datos cada una (por defecto son 20). Luego, los datos recibidos se separan con el carácter especial introducido en las cadenas y se reconvierten a su valor numérico reposicionando los valores en un vector general del sistema receptor. Este método descarga al cliente de un gran número de lecturas del buffer haciéndolo más óptimo.

En ambos casos, se han evaluado diferentes frecuencias de muestreo de 100 a 2.000 Hz para asegurar que los datos pueden ser recibidos en tiempo real en Matlab, sin ningún retraso en la comunicación. En el caso del envío de datos uno a uno, cuando se envían 0.5 segundos de datos correspondientes a 1000 datos (a una frecuencia de muestreo de 2.000 Hz), el tiempo de procesado de Matlab asciende a los 0.29 segundos. Aunque es suficiente, ya que es inferior al intervalo de tiempo, en caso de que se requiera parte del intervalo de 0.5 segundos para procesar otras señales en tiempo real es preferible poder reducir dicho tiempo. Por ello, al implementar la solución por bloques, consigue reducirse este tiempo y la carga de procesamiento al tener que recibir una cantidad de recepciones 20

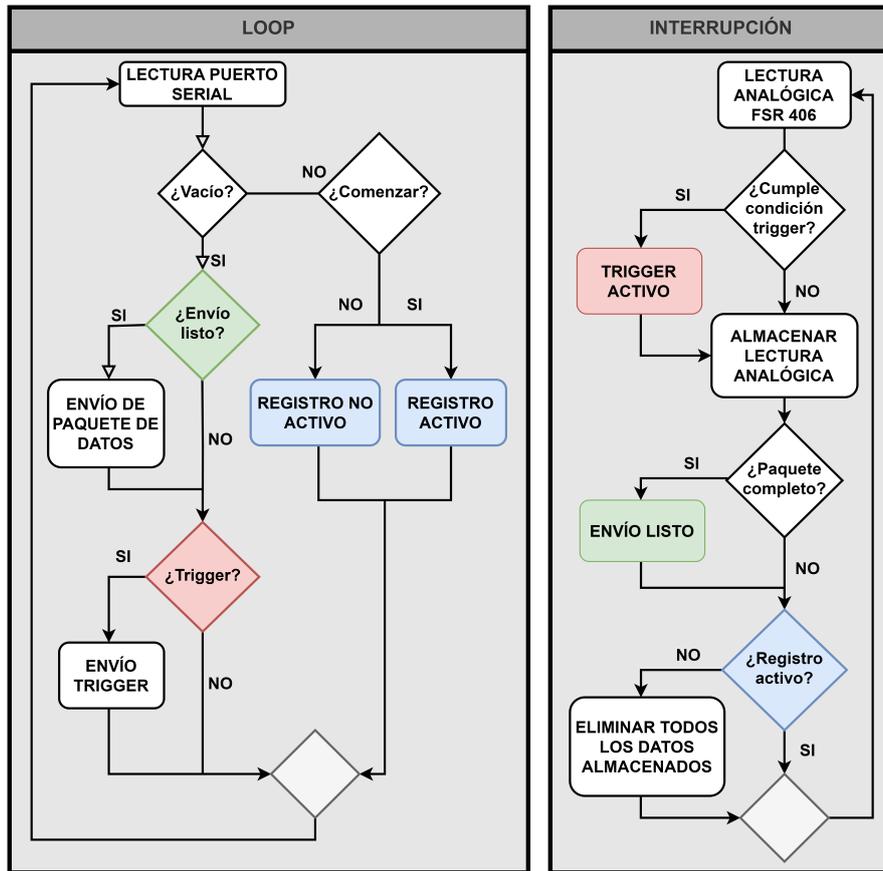


Figura 5: Diagrama de flujo de la lógica de ejecución del microcontrolador ESP32.

veces inferior. Se ha comprobado además, que utilizando frecuencias de muestreo superiores, comienzan a producirse desfases en la recepción de datos teniendo por tanto como límite los 2000 Hz implementados.

4. Resultados

Para demostrar la funcionalidad y efectividad del sistema desarrollado en este trabajo, en esta sección se van a mostrar los registros obtenidos utilizando el sensor de fuerza resistivo FSR 406 y el microcontrolador ESP32 para enviar datos y triggers a Matlab y a los dispositivos conectados mediante cables coaxiales. Para verificar la sincronización, se ha utilizado un equipo de electroencefalografía que dispone de una entrada de trigger mediante cable coaxial. Durante las pruebas, la arquitectura de Matlab ha registrado simultáneamente los valores recibidos del electroencefalógrafo junto con el trigger y del sistema FSR406-ESP32.

A continuación, se detallan los tres métodos diferentes de envío de triggers empleados en el sistema:

4.1. Activación por botón

Como se puede observar en la Figura 6, este método ha sido utilizado para corroborar el correcto envío de los triggers de manera sencilla y directa ya que no es posible utilizarlo para corroborar la sincronización entre dispositivos.

4.2. Superación de umbral

Este método es fundamental para la sincronización de las lecturas entre varios dispositivos, especialmente desde el inicio de una actividad hasta su finalización (ver Figura 7). Este método es crucial para aplicaciones donde es necesario registrar eventos específicos que ocurren cuando se excede una fuerza determinada, garantizando así una alta precisión en la sincronización de datos. La sincronización entre las señales puede comprobarse verificando la muestra donde se activa el trigger con la muestra del sensor de fuerza cuyo valor ha sido indicado como umbral (línea negra horizontal de la Figura 7).

4.3. Superación de umbral y activación por botón

La doble condición presente en este método es especialmente útil cuando se requiere una alta precisión para sincronizar las lecturas desde el comienzo hasta el final de una actividad, y más concretamente en escenarios donde la fuerza aplicada oscila constantemente alrededor del umbral determinado. La Figura 8 ilustra cómo se implementa este método. Esta combinación asegura que solo se registren eventos verdaderamente significativos, facilitando la sincronización de los datos recogidos. De esta forma, al igual que en el caso anterior, se puede comprobar la sincronización del registro verificando la muestra de activación del trigger frente al valor del sensor de fuerza cuando se supera el umbral. Como se ha comentado, en este caso se sincroniza la prueba al completo marcando el trigger mediante la habilitación con el botón en el inicio y el fin del experimento, evitando que si la fuerza generada fluctúa

mucho no se generen triggers indeseados que dificulten la sincronización de la señal posterior con otros equipos.

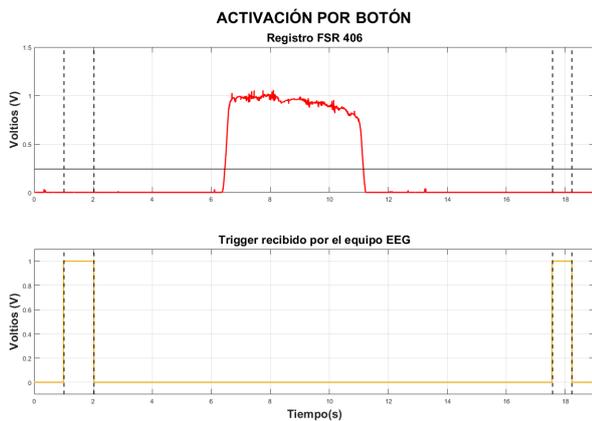


Figura 6: Registros de fuerza sincronizados a través del envío de triggers por botón.

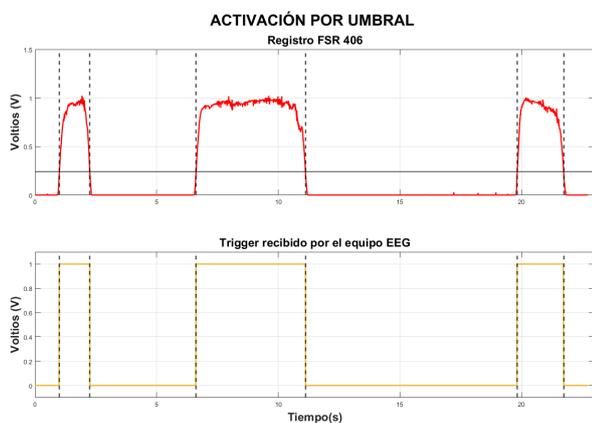


Figura 7: Registros fuerza sincronizados a través del envío de triggers por superación de umbral.

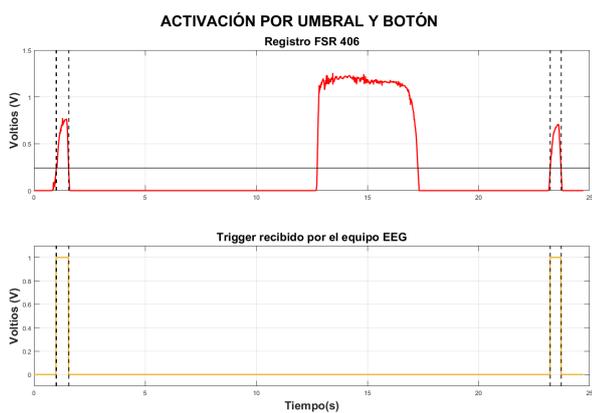


Figura 8: Registros fuerza sincronizados a través del envío de triggers por superación de umbral y activación por botón.

5. Conclusiones

Este estudio ha demostrado con éxito la viabilidad y eficacia de una arquitectura optimizada para la cuantificación y transmisión de datos de presión superficial. A lo largo del

desarrollo e implementación, se han abordado y resuelto varios desafíos técnicos que permitieron mejorar la eficiencia del sistema llegando a realizar lecturas a 2000 Hz sin introducir ningún tipo de desfase en el sistema general que controla el registro de todos los dispositivos de adquisición de datos, pudiendo asegurar una sincronización precisa entre distintos dispositivos y, habiendo sido ya utilizado por (Gracia et al., 2023) para determinar en qué momento iniciaba y terminaba la ventana de interés del estudio.

Como trabajos futuros, se pretende incrementar el número de sensores de fuerza que puedan estar conectados simultáneamente al sistema, lo que permitiría un monitoreo más completo y detallado de las superficies de interés. Además, se está desarrollando un algoritmo para la corrección de los valores obtenidos con el objetivo de conseguir linealizar la curva del sensor de fuerza, mejorando así la precisión y utilidad de los datos registrados en aplicaciones prácticas.

Agradecimientos

Este trabajo ha sido realizado en el marco del proyecto de I+D+i ReGAIT (PID2021-124111OB-C31), financiado por MICIU/AEI/10.13039/501100011033 y por FEDER, UE, en el marco del proyecto MYOREHAB (PCI2023-143405), financiado por el Ministerio de Ciencia, Innovación y Universidades y la Agencia Estatal de Investigación MCI/AEI/10.13039/501100011033, y por la Consellería de Educación, Universidades y Empleo de la Generalitat Valenciana y por el Fondo Social Europeo a través de la ayuda CIACI-F/2022/108.

Referencias

- Agarwal, R., Kumar, A., Yadav, S., 2022. Precision Measurements in Healthcare Systems and Devices. Springer Nature Singapore, Singapore, pp. 1–10.
DOI: 10.1007/978-981-19-1550-5_6 – 1
- Babiuch, M., Foltýnek, P., Smutny, P., 5 2019. Using the esp32 microcontroller for data processing. Proceedings of the 2019 20th International Carpathian Control Conference, ICCCC 2019.
DOI: <https://doi.org/10.1109/CarpathianCC.2019.8765944>
- Electronics, I., 2024. Fsr 400 series data sheet. <https://cdn.sparkfun.com/assets/c/4/6/8/b/2010-10-26-DataSheet-FSR406-Layout2.pdf>.
- Foltýnek, P., Babiuch, M., Šuránek, P., 2019. Measurement and data processing from internet of things modules by dual-core application using esp32 board. Measurement and Control 52 (7-8), 970–984.
DOI: 10.1177/0020294019857748
- Giovanelli, D., Farella, E., 2016. Force sensing resistor and evaluation of technology for wearable body pressure sensing. Journal of Sensors 2016.
DOI: 10.1155/2016/9391850
- Gracia, D. I., Ortiz, M., Candela, T., Iáñez, E., Sánchez, R. M., Díaz, C., Azorín, J. M., 6 2023. Design and evaluation of a potential non-invasive neurostimulation strategy for treating persistent anosmia in post-covid-19 patients. Sensors 2023, Vol. 23, Page 5880 23, 5880.
DOI: 10.3390/S23135880
- Kareem, H., Dunaev, D., 5 2021. The working principles of esp32 and analytical comparison of using low-cost microcontroller modules in embedded systems design. 2021 4th International Conference on Circuits, Systems and Simulation, ICCSS 2021, 130–135.
DOI: 10.1109/ICCS51193.2021.9464217
- Sadun, A. S., Jalani, J., Sukor, J. A., 2016. Force Sensing Resistor (FSR): a brief overview and the low-cost sensor for active compliance control. In: Jiang, X., Chen, G., Capi, G., Ishll, C. (Eds.), First International Workshop on Pattern Recognition. Vol. 10011. International Society for Optics and Photonics, SPIE, p. 1001112.
DOI: 10.1117/12.2242950