

DETECCIÓN Y CLASIFICACION DE GOLPES DE BOXEO

Jorge Almingol Estrada

Master Universitario en Ingeniería Electrónica. Universidad de Zaragoza
María de Luna, 1. 50018 Zaragoza (Spain)

Resumen

El objetivo de este proyecto es el de poder detectar y clasificar diferentes golpes típicos de boxeo en inferencia. Para ello se ha hecho uso de los siguientes componentes hardware: Una placa de desarrollo basada en ESP32 y un sensor de aceleración, giro y campo magnético. El sistema se ha implementado mediante Arduino mediante un sistema operático de tiempo real.

1. CARACTERIZACIÓN DEL SISTEMA Y CAPTURA DE DATOS.

1.1. Repaso de los diferentes golpes clásicos de boxeo

Puesto que vamos a detectar diferentes golpes de boxeo lo primero es hacer una revisión de estos y una breve descripción de cada movimiento:

Directo: El brazo de ataque coloca los nudillos en posición horizontal para impactar y luego se retrae rápidamente a la posición original de defensa.

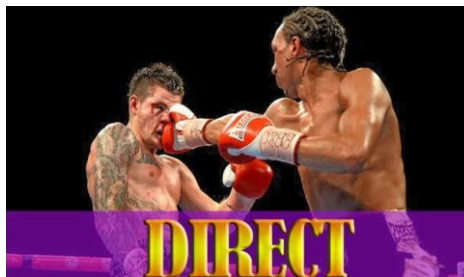


Figura 1 Golpe Directo de boxeo.

Crochet o cruzado: Uno de los más potentes del boxeo se realiza sin abrir el brazo y colocándolo paralelo al suelo, de manera lateral y siendo acompañado con un sincronizado movimiento de pie, cadera, torso y hombro.

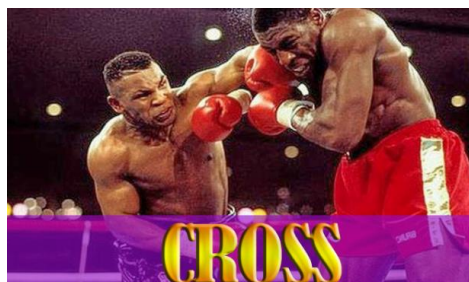


Figura 2 Crochet de boxeo.

Gancho o Uppercut: Se aplica de modo vertical, acompañando el movimiento del brazo con una leve flexión en las rodillas para dar más impulso hacia arriba.



Figura 3 Gancho de boxeo.

Hook: es un golpe similar al Uppercut, pero tiene como objetivo golpear al rival en el torso



Figura 4 Hook de boxeo

Estos 4 golpes por tanto serían la base de un deporte que en la puesta en práctica resulta de mayor complejidad y una gran estrategia de movimiento.

1.2. Protocolo de adquisición de los datos

Para realizar una caracterización de estos movimientos vamos a realizar una captura de diferentes magnitudes durante los distintos golpes **al sacco**. Para ello haremos uso del sensor de aceleración y giro capacitivo GY-91. Este sensor es capaz de medir las aceleraciones y los giros en las 3 dimensiones por tanto es perfecto para la tarea demandada. Un microcontrolador ESP32 se encarga de adquirir, adaptar y procesar los datos procedentes del sensor así como realizar las diferentes tareas necesarias para el tratamiento y posterior clasificación de los datos.

Tras diversas pruebas de adquisición con diferentes periodos entre medidas se determinó que 10ms resultaba un buen periodo de adquisición puesto que caracteriza bien la forma de onda correspondiente al movimiento. Por tanto el sensor tomara muestras con una frecuencia de 100 Hz. Un ejemplo de una muestra de

Para poder realizar la caracterización también es necesario determinar la duración del golpeo. Esto se determinó analizando las formas de onda de los diferentes golpes en detalle. Tras este análisis se determinó que la duración aproximada de estos golpes era algo menor a 30 ciclos, es decir 300ms. Un ejemplo de un golpeo único se muestra en la Figura 6.

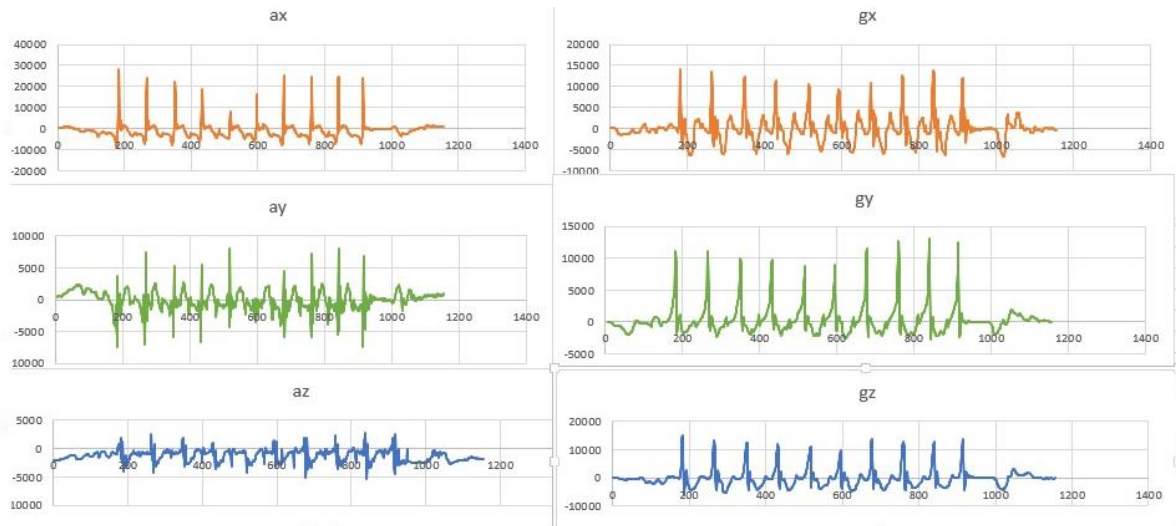


Figura 5 Adquisición de diferentes muestras de un golpe directo al aire

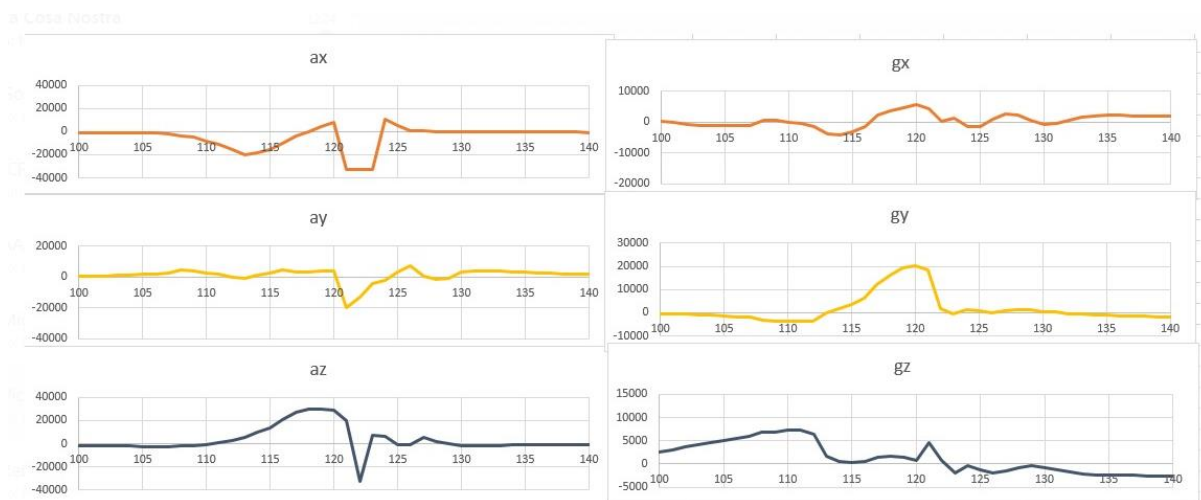


Figura 6 Croquet, medida individual del golpe al saco.

Por tanto a la hora de caracterizar un golpe utilizaremos los datos de adquisición de una ventana temporal de 300ms con 180 muestras (30 de cada una de las 6 señales). Para evitar el caso de que un golpe quede dividido en dos ventanas temporales se utiliza la técnica de solapamiento de estas ventanas. Este método consiste en que cada ventana comienza en el punto medio de la anterior.

Una vez recogidos estos datos por el microcontrolador emparejado al sensor (ESP32) es este el encargado de su procesado.

Por último una vez procesados estos datos se procede a evaluar si es golpe y en tal caso informar de cual es mediante el envío de un socket mediante WiFi al dispositivo oportuno.

1.3. Consumo energético y alimentación

La demanda energética del dispositivo es un tema clave ya que debemos asegurar que reciba energía suficiente y que, además, vaya a tener un tiempo de suministro energético suficiente para cumplir con la tarea a realizar. En primer lugar hemos recogido los distintos consumos que se dan para esta aplicación (Tabla 1).

Tabla 1 Consumos

Dispositivo	Consumo (mA)
ESP32	50
ESP32 sleep mode	15
WiFi	130
Sensor	3.8

Teniendo los consumos se ha determinado el tiempo de uso de cada uno de estos dispositivos. En el caso del sensor de giro, está siempre conectado por tanto tendrá un consumo constante. En el caso del ESP 32 hay que ver cuánto tiempo está en modo normal y cuanto en modo sleep. Para ello tenemos en cuenta las diferentes tareas que realiza y los tiempos en cada una de ellas (Tabla 2).

Tabla 2 Tiempos de trabajo de las diferentes tareas

	tiempo sleep (ms)	tiempo on (ms)	periodo (ms)
Tarea 1	10	2.3	12.3
Tarea 2	100	4.2	104.2
Tarea 3	10	3.2	13.2

En cuanto al módulo WiFi solamente está activo cuando lo está el micro controlador, por tanto la proporción de tiempo on, off será la misma que la del ESP32. Cabe destacar que para determinar el tiempo que realmente está activo el micro controlador se ha determinado un tiempo promedio de tarea y se han sumado los tiempos on de las 3 tareas durante ese tiempo común. Con todo esto nos queda la siguiente tabla de consumos (Tabla 3)

Tabla 3 Consumos individuales y total

	T Sleep (ms)	T on (ms)	periodo (ms)	Consumo (mA)
ESP 32	7.276101193	4.697232141	11.9733333	28.73077324
Sensor	0	∞	∞	3.8
WiFi	7.276101193	4.697232141	11.9733333	51.00001489
			total	83.53078813

Por tanto el modulo tendrá un consumo medio de 83.5mA. Para la alimentación durante las pruebas se ha utilizado un Smart Bank de 2000mAh y una corriente máxima de 1^a. Esto por tanto nos asegura que pueda recibir la corriente media y la corriente de pico de 185mA (suma de todos los procesos en el punto máximo de trabajo) y que la batería tendrá una duración suficiente ya que aguantaría un día de uso.

$$\frac{2000 \text{ mAh}}{83.5 \text{ mA}} = 23.95 \sim 24h$$

En una aplicación futura el uso de una Smart Bank no es el modo de alimentación óptimo ya que estas baterías tienen muchas pérdidas energéticas. En caso de buscar un método de alimentación eficiente lo más óptimo sería la utilización en serie de 2 baterías de litio las cuales proporcionarán 3V bastante constantes que, una vez próxima a la descarga llegará a descender cerca de los 2.7V habiéndose consumido un 90% de la batería. Estos 2,7V superan la tensión mínima del micro de 2,55V.

Desde el punto de vista de la comodidad otra opción sería incluir una batería recargable de litio o de polímero de litio pero esta opción conlleva una pérdida de eficiencia. Esta pérdida se debe a que la tensión de salida de estas baterías ronda entre los 3,7 y los 4,2V. Esto implica la necesidad de reducir la tensión a los 3.3V de consumo del micro, y por tanto a la pérdida energética correspondiente a esa conversión.

2. Análisis de los datos

Una vez determinado el protocolo de adquisición otro tema de interés es el de tratamiento de los datos. Para este modelo, el procesado a realizar va a consistir en cálculos estadísticos. Puesto que la señal del golpe es nítida y no se aprecian efectos de posible ruido. Las magnitudes son muy significativas, llegando a saturar el sensor en 16g haciendo que cualquier pequeña fluctuación sea despreciable. Por lo tanto no vamos a implementar ningún filtrado de datos.

Los cálculos que se realizan consisten por tanto en el cálculo de la **media** y la **varianza** de los datos en cada ventana temporal. De esta forma podemos analizar las características de los datos en conjunto y conseguimos ver que atributos destacan respecto a otros que, por tanto, resultaran ser los más excitados por cada movimiento.

Para realizar este análisis se recogen los datos de las medias y las varianzas directamente del sensor y se muestran en gráficas mediante el uso de Excel. A modo de ejemplo se realizó una serie de 8 movimientos, 2 de cada clase, en el siguiente orden: directo, crochet, hook y gancho (Figura 7).

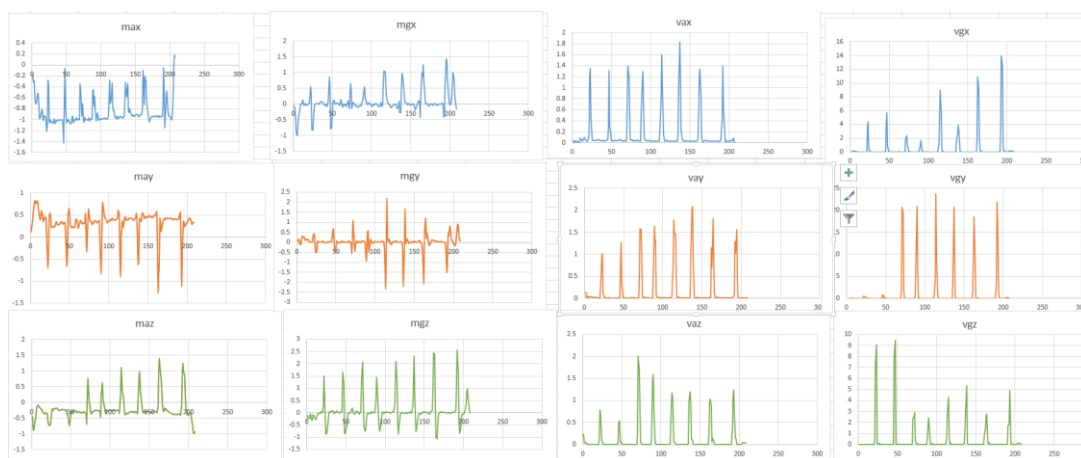


Figura 7 Medias y varianzas durante los diferentes golpes

Para definir las características de los golpes se realizó un número mucho mayor de medidas y se agruparon para realizar las comparaciones. En primer lugar era necesario definir cuándo se ha dado un golpe y por el contrario descartar los datos que no sean relevantes. La detección de un golpe se caracteriza por existir una aceleración negativa en el eje Y de movimiento y a su vez existir una varianza en los tres ejes de giro superior a uno.

Teniendo ya definido cuando se ha dado un golpe queda determinar que golpe es, o por el contrario descartarlo. Se agruparon los datos correspondientes a 60 golpes y se analizan sus características.

En primer lugar el **Directo**. Este golpe se caracteriza por tener una varianza de giro en y un orden de magnitud menor a la del resto de golpes. Además es el único que tiene una media de aceleración en el eje z negativa.

El siguiente golpe es el **Crochet**, Este golpe se define por tener una significativa varianza en el eje de giro y superior a 10 (también ocurre en hook y gancho), además de, a diferencia de los siguientes golpes, tiene la varianza del giro en z superior a 4, en el eje z la media de los datos es siempre menor a 0,6 y la varianza siempre superior a 2.

El **Hook** es muy similar al **Gancho**, ambos golpes coinciden en tener mucha varianza en el giro en y, como ocurriría con el Crochet. Las características del Crochet lo diferencian de estos por lo tanto evaluara si es un crochet y una vez descartado analizara si se trata de alguno de estos dos. Para el caso del **Hook**, la media de los datos de aceleración en el eje z están entre 0,8 y 1 siendo siempre menores a 1, y además los valores de varianza de los datos de giro en x son siempre menores a 5.

Con el **Gancho** ocurre lo contrario. La media de los datos de aceleración en el eje z están entre son siempre mayores que 1 y los valores de varianza de los datos de giro en x son siempre mayores que 5.

Las tareas comentadas de detectar si es un movimiento y caracterizarlo se realizaran mediante una **máquina de estados**.

En la Figura 8 se muestran los diferentes datos de todos los puntos. Aunque es un poco compacto se pueden apreciar ciertos detalles aquí comentados. Además también se incluye la tabla de datos. Los valores de crochet son reducidos pero se comprobaron paralelamente un número mayor de datos.

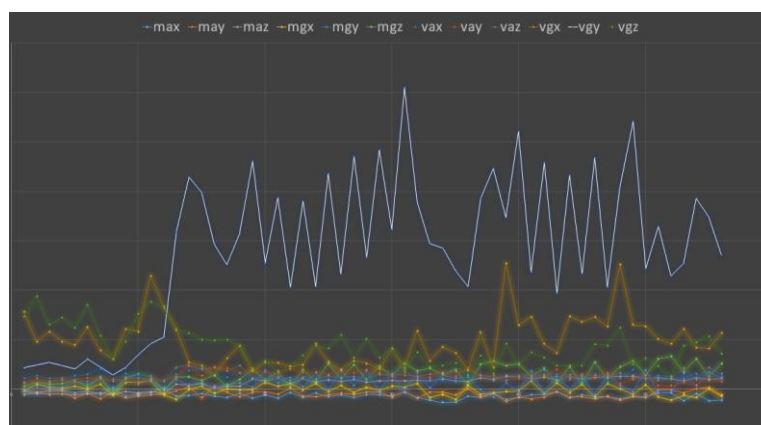


Figura 8 Gráfica que muestra los datos recogidos de diferentes golpes realizados

		mah	may	maz	mgx	mgv	mgz	vah	vay	vaz	vgh	vgy	vgz
Directo	1	-0.21	-0.13	-0.56	-0.22	-0.14	0.25	1.08	0.67	0.63	7.81	2.12	7.33
	2	-0.55	-0.29	-0.43	0.45	0.13	0.66	1.3	1.04	0.64	4.76	2.42	9.36
	3	-0.05	-0.25	-0.56	0.18	0.1	0.47	1.09	0.79	0.7	5.76	2.68	6.5
	4	-0.11	-0.23	-0.53	0.06	0.14	0.51	1.1	1.04	0.89	4.81	2.37	7.19
	5	-0.43	-0.97	-0.62	0.29	0.74	0.86	1.37	0.86	0.71	4.44	2.04	6.19
	6	-0.18	-0.41	-0.5	-0.02	0.5	0.19	1.47	1.11	0.74	6.28	3.05	8.51
	7	-0.58	-1.02	-0.37	0.47	1.04	1.2	2.02	1.03	1.09	3.89	2.23	5.38
	8	0.14	-0.35	-0.61	-0.62	-0.18	-0.48	1	0.96	0.76	3.01	1.41	3.01
	9	-0.72	-0.91	-0.26	0.64	1.08	1.15	1.54	0.93	0.99	6.05	2.12	4.8
	10	-0.51	-0.72	-0.39	0.67	1.13	1.48	1.51	0.68	0.46	5.84	3.43	7.59
	11	-0.39	-0.58	-0.3	0.78	0.97	1.26	1.29	1.11	0.9	11.47	4.58	8.85
	12	0.16	-0.55	-0.49	-0.58	-0.02	-0.18	0.83	0.86	0.73	8.37	5.25	8.12
croche	13	-0.72	-0.22	0.47	-1.07	0.95	1.22	1.49	1.19	2.19	6.02	15.32	5.9
	14	-0.66	0.21	0.42	-0.07	0.38	1.21	1.87	2.36	2.37	2.71	21.43	5.66
	15	-0.46	-0.92	0.53	0.26	0.65	0.85	1.96	1.31	2.04	2.35	19.87	4.97
	16	-0.72	0.26	0.14	-0.44	0.05	1.42	1.33	2.22	2.17	1.39	14.67	4.93
	17	-0.9	-0.33	0.29	0.02	0.48	0.29	1.18	2.08	1.84	3.08	12.58	4.98
	18	-0.47	-0.81	0.27	-0.13	0.62	0.95	1.62	1.61	2.34	4.35	15.69	4.4
hook	19	-0.94	-0.35	0.99	-0.09	0.18	1.95	1.09	1.9	1.12	1.81	23.07	2.12
	20	-0.58	-0.29	0.83	0.66	1.93	0.95	1.01	1.68	0.92	2.69	12.76	2.85
	21	-0.97	-0.55	1.09	0.25	0.63	1.7	1.05	1.64	1.2	2.63	19.38	1.81
	22	-0.35	0.08	0.51	0.52	1.18	0.18	0.7	1.99	0.92	2.29	10.28	2.03
	23	-0.9	-0.71	1.04	-0.08	0.21	1.95	1.13	1.67	1.19	2.46	19.02	3.4
	24	-0.37	-0.79	0.68	0.52	1.7	0.73	0.94	1.69	1.02	4.53	10.35	4.38
	25	-0.79	-0.58	0.92	-0.25	-0.08	2.53	1.27	1.47	1.34	2.74	21.77	4.13
	26	-0.6	-0.44	0.75	0.37	1.89	1.05	1.02	1.84	1.19	1.96	11.63	5.44
	27	-0.9	-0.56	0.94	-0.35	0.36	2.41	1.14	1.61	1.25	2.86	23.5	3.19
	28	-0.56	-0.32	0.69	0.17	1.84	0.97	0.91	1.97	1	2.61	13.34	5.1
	29	-0.6	-0.31	0.78	-0.11	-0.2	2.27	1.42	1.58	1.48	1.99	24.18	3.16
	30	-0.73	-0.82	0.83	0.26	1.54	1.36	1.26	1.26	1.43	4.04	16.16	4.12
	31	-0.35	-0.28	0.82	0.18	-0.32	2.59	1.88	1.62	1.38	2.14	30.49	1.87
	32	-0.85	-0.97	0.87	0.55	1.66	1.79	1.02	1.18	1.26	5.89	18.96	3.7
	33	-1.14	-0.32	0.84	-0.9	0.09	2.09	0.89	1.18	1.48	2.86	14.74	2.22
	34	-1.39	-0.31	0.99	-0.56	1.43	1.47	0.84	1.71	1.59	4.24	14.25	2.39
	35	-1.33	-0.62	0.78	-1.07	1.07	1.98	0.76	1.26	1.76	3.66	11.99	2.62
	36	-0.75	0.03	0.76	0.33	1.2	0.42	1.33	2.11	1.91	1.93	10.36	1.41
gancho	37	-0.85	-0.81	1.14	-0.56	-0.05	2.47	1.38	1.31	1.42	5.7	19.28	3.41
	38	-0.49	-0.47	0.91	-0.39	-0.86	2.85	1.37	1.35	1.12	2.17	22.32	1.55
	39	-1.07	-1.27	1.31	-0.27	0.6	2.34	0.82	0.9	1.24	12.75	17.33	4.62
	40	-0.86	-0.84	1.26	-0.17	-0.26	2.57	1.27	1.11	1.25	6.44	26.06	2.42
	41	-0.61	-1.03	1	0.78	1.81	1.25	0.91	1.38	1.12	7.32	11.86	3.71
	42	-0.51	-0.83	1.19	-0.47	-0.47	2.1	1.9	1.25	1.45	4.53	22.93	3.15
	43	-0.27	-0.3	0.88	0.63	1.47	0.59	1.03	1.97	1.22	3.67	9.67	2.37
	44	-0.91	-0.84	1.18	-0.35	-0.08	2.21	1.13	0.98	1.45	7.35	21.62	2.37
	45	-0.62	-0.81	1	0.49	1.43	0.47	0.76	1.47	1.28	6.76	11.62	2.4
	46	-0.82	-1.01	1.16	-0.8	-0.62	2.7	1.53	1.19	1.36	7.29	23.41	4.48
	47	-0.81	-0.76	1.2	0.51	1.44	1.06	0.72	1.66	1.21	6.33	10.32	4.41
	48	-0.98	-1.13	1.25	0	0.34	2.21	1.08	0.53	1.62	12.59	20.47	6.2
	49	-0.75	-0.79	1.31	-0.52	-0.24	2.82	1.94	0.96	1.15	6.48	27.09	2.48
	50	-0.83	-0.89	1.08	0.41	1.89	1.4	0.87	1.32	1.08	6.36	12.19	3.06
	51	-0.39	-0.12	1.13	-0.79	-0.38	3.04	1.34	0.33	0.83	5.05	16.45	2.25
	52	-0.4	-0.08	1	-1.13	-0.33	3.28	1.16	0.35	1.06	4.57	11.43	1.28
	53	-1.22	-0.4	0.84	-0.65	1.58	1.75	1.1	0.69	2.11	6.08	12.74	4.4
	54	-0.48	-0.01	1.14	-0.87	0.33	3.04	1.53	0.42	0.98	4.17	19.3	4.7
	55	-1.19	0.12	0.91	0	1.68	1.27	1.18	1.08	2.15	4.13	17.39	5.29
	56	-1.12	-0.62	1.09	-0.75	1.12	2.54	1.24	0.74	1.51	5.66	13.59	3.54

Figura 9 Tabla de datos de los movimientos capturados de cada golpeo

3. Implementación de las tareas

Para la realización de esta aplicación hemos hecho usos del sistema operativo (OS) en tiempo real FreeRTOS. El OS nos permite crear tareas con diferentes prioridades y se encarga de gestionar el tiempo de uso del procesador por cada tarea.

En el programa realizado para este detector de golpes hemos implementado 3 tareas. Una primera tarea es la encargada de la adquisición de datos de cada ventana temporal. La segunda se encarga del procesamiento de los datos y por ultimo una tercera tarea consistente en una máquina de estados que en función de los datos recibidos determina si es un golpe y en caso de serlo que golpe es. No se ha asignado prioridad a las tareas puesto que no se trabaja con tiempos ajustados pero en caso de hacerlo la prioridad máxima la tendría la adquisición de datos ya que

no podemos permitirnos perder información. La tarea de segundo nivel sería el tratamiento de los datos y por ultimo ya la tarea menos prioritaria sería la tarea encargada de la clasificación y las comunicaciones.

Otra de las ventajas de utilizar FreeRTOS es que nos permite mantener el microprocesador en modo Sleep en los periodos de tiempo en los que no se está ejecutando ninguna tarea. Aunque esta aplicación sea dedicada a una tarea puntual y no resulta muy significativo el ahorrar energía siempre es importante que el sistema consuma la menor energía posible haciéndolo así más eficiente energéticamente.

Vamos a realizar una descripción más exhaustiva de cada tarea

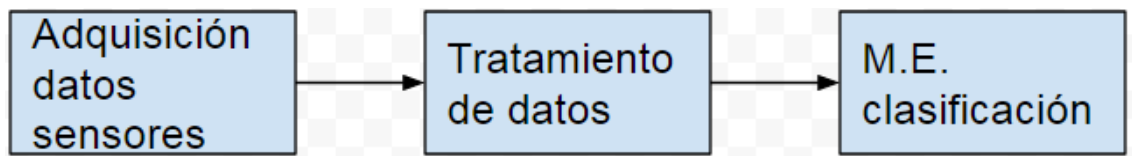


Figura 10 Tareas del Sistema Operativo

3.1. Adquisición de datos.

Esta tarea, activada cada 10ms, se encarga en primer lugar de adquirir los datos del sensor y adaptarlos a valores representativos. Estos valores se transforman en unidades de aceleración medidos en fuerzas G. Utilizamos fuerzas G ya que se adapta bien a los rangos de las magnitudes del problema facilitando así los cálculos posteriores.

A continuación el sistema almacena los datos en las ventanas temporales alternas. Esto se ha realizado mediante la introducción de los diferentes datos en dos matrices con 6 filas (puesto que es el número de señales que medimos) y tantas columnas como muestras tomamos en cada ventana que, en este caso, son 30. Ambas matrices tienen un índice de columna desfasado a la mitad del tamaño de ventana de una respecto a la otra obteniendo de esta forma las mediciones solapadas comentadas anteriormente.

Cuando una de las matrices se completa, esta es copiada en una tercera matriz la cual será procesada en la tarea de procesamiento del OS.

Además esta tarea añade los datos del sensor a un documento de texto contenido en una memoria SD adjunta al microcontrolador para poder mostrar los datos adquiridos permitiéndonos ver las formas de onda de los diferentes golpes y trabajar con ellas (Figuras 6 y 7).

3.2. Tratamiento de los datos

Esta tarea, como su nombre indica se encarga de calcular tanto la media como la varianza de cada una de las señales. Esto, puesto que recibe una matriz de datos, consistirá en calcular la media y la varianza de cada una de sus filas.

Una vez calculadas las introduce en dos nuevos vectores de datos (medias y varianzas), disponibles para ser evaluados en la máquina de estados.

Esta tarea se ejecuta cada 100ms, evalúa si tiene una matriz que analizar y en caso de no ser así vuelve a estar en reposo. Puesto que las ventanas son de 300ms la tarea recibirá datos cada 150ms pero a modo de evitar solapamientos se decide que revise la disponibilidad de información con una frecuencia mayor.

3.3. Clasificación y comunicación

Esta tercera tarea será la encargada de determinar si los datos son un golpe, que golpe es y enviarlo al receptor mediante un socket WiFi. Para ello hace uso de una máquina de estados (Figura 11) con 5 diferentes estados. Reposo, Detector de movimiento, Clasificador, Comunicaciones, Estado previo a Reposo.

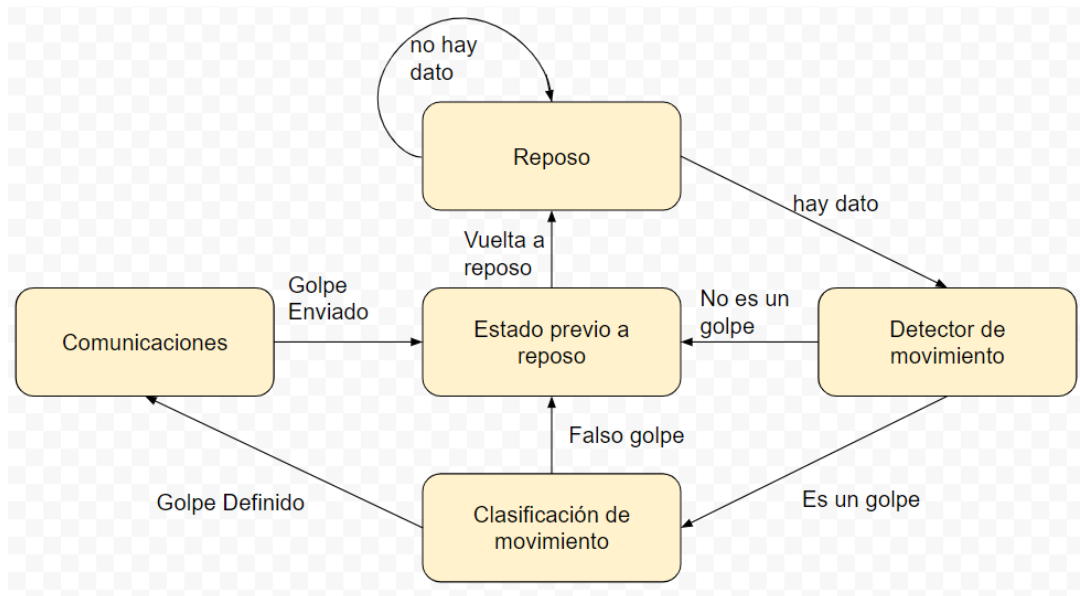


Figura 11 Esquema de la máquina de estados

Reposo: Este estado se encarga de comprobar si llegan datos de la Tarea 2, en caso de recibirlos pasa al estado de Detector de movimiento, sino, sigue igual.

Detector de movimiento: comprueba los datos recibidos y decide si se trata de un golpe o son datos sin valor. En caso de no tener valor pasa al estado previo a reposo. En caso contrario el estado siguiente será el clasificador.

Clasificador de movimiento: Este estado analiza los datos y decide cuál de los 4 golpes es mediante una lógica de descarte. En primer lugar comprueba si se trata de un directo, en caso de no ser un directo comprueba si se trata de un croquet. Mismo proceso con el hook y por ultimo con el gancho. En caso de no coincidir con las características de ninguno se descarta el movimiento como falso golpe y la maquina pasa al estado previo a reposo. Si es un golpe correcto pasa al estado de comunicaciones.

Comunicaciones: Esta tarea tiene la función de enviar un socket con el resultado de la evaluación del golpeo mediante WiFi a un servidor. Además, también envía los valores de las medias y varianzas para facilitar el análisis durante los ensayos. La salida de este estado es siempre pasar al estado previo a reposo.

Paso previo a Reposo: Este estado almacena el diferente resultado obtenido en cada uno de los posibles pasos para evitar solapamientos de datos.

Por último se incluye un diagrama de flujo que muestra todo el programa en conjunto.

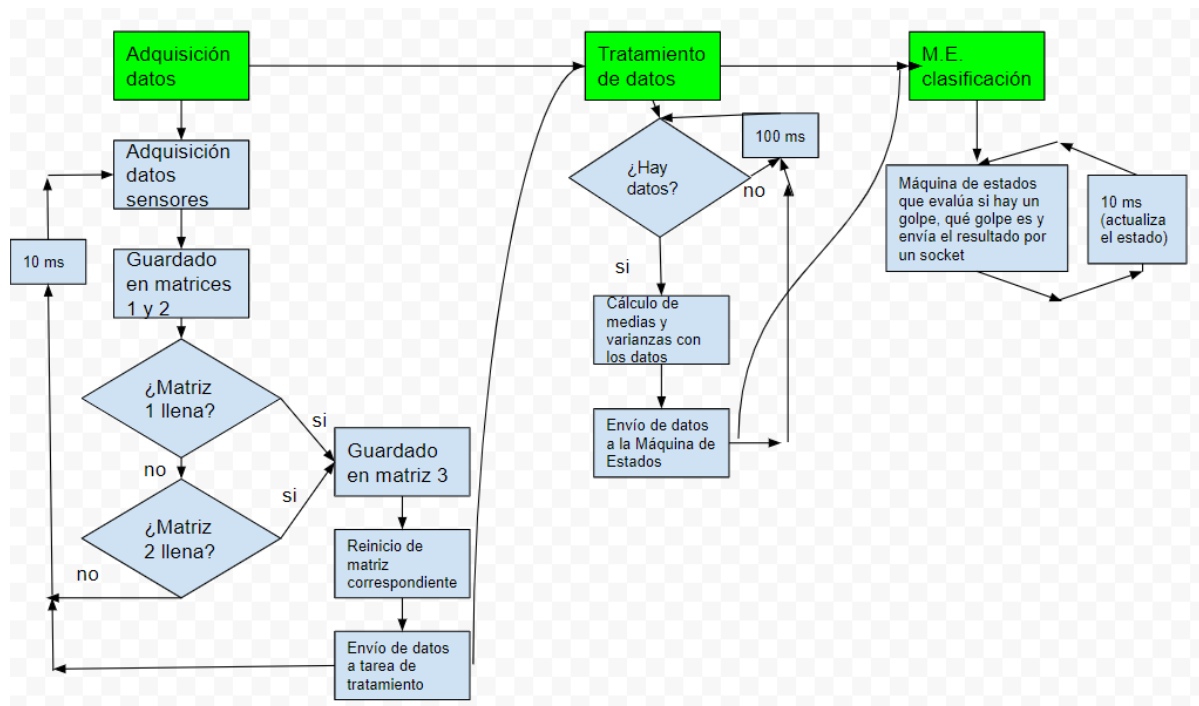


Figura 12 Diagrama de flujo del programa del sensor

4. Preparación de los ensayos y demostración

En este apartado vamos a hablar de cómo se preparó el entorno de medidas. En primer lugar se colocó el sensor en un guante de boxeo como puede verse en la Figura 14. Es importante realizar correctamente este posicionamiento para una correcta captura de datos y tener cierta repetitividad en las medidas.

El sensor se sujeta con cinta de carroceros debido a su comodidad ya que hay que ponerlo y quitarlo muchas veces. Quedaría como trabajo futuro implementar un sistema de sujeción más cómodo y con mejor apariencia.

Respecto a la alimentación del sensor se realiza mediante una power bank por el conector USB de la placa. Esto ofrece mucha versatilidad en su uso. Dentro de un trabajo futuro también estaría desarrollar un sistema de alimentación optimizado

En las diferentes pruebas se capturaron los datos con esta disposición y siempre mediante golpes a un saco de boxeo. En las etapas de caracterización los datos se almacenaban en la tarjeta micro SD de la placa. Una vez ya desarrollada la aplicación el sensor sigue almacenando los datos, pero no es realmente útil ya que la comunicación es mediante WiFi. Para ello creamos un servidor con SocketTest y esperamos a recibir datos del sensor (Figura 15).



Figura 13 Placa con la tarjeta y el sensor y a su derecha la power bank utilizada.



Figura 14 Montaje del sensor en un guante de boxeo y saco de pruebas

Tras realizar varias pruebas de funcionamiento y depurar diversos errores se consiguió un funcionamiento bastante aceptable que diferenciaba perfectamente el directo de los otros 3 golpes más curvos y asimismo también diferenciaba estos tres. Los 3 movimientos curvos varían en la altura, crochet en la parte superior del saco, hook en la parte intermedia y gancho en la parte baja. Aunque diferencia correctamente los golpes sí que en ocasiones se confunde con los más cercanos, es decir confunde croché con hook parte media y alta, hook con crochet o gancho ya que es una parte intermedia, y gancho con hook cuando se golpea la parte baja. Si los movimientos están bien definidos en el momento del golpeo no suele confundirlos. En la siguiente imagen (Figura 15) se puede observar como muestra una serie de golpes y los datos de cada golpe, permitiendo de esta forma compararlos en vivo y poder ajustar el sensor.

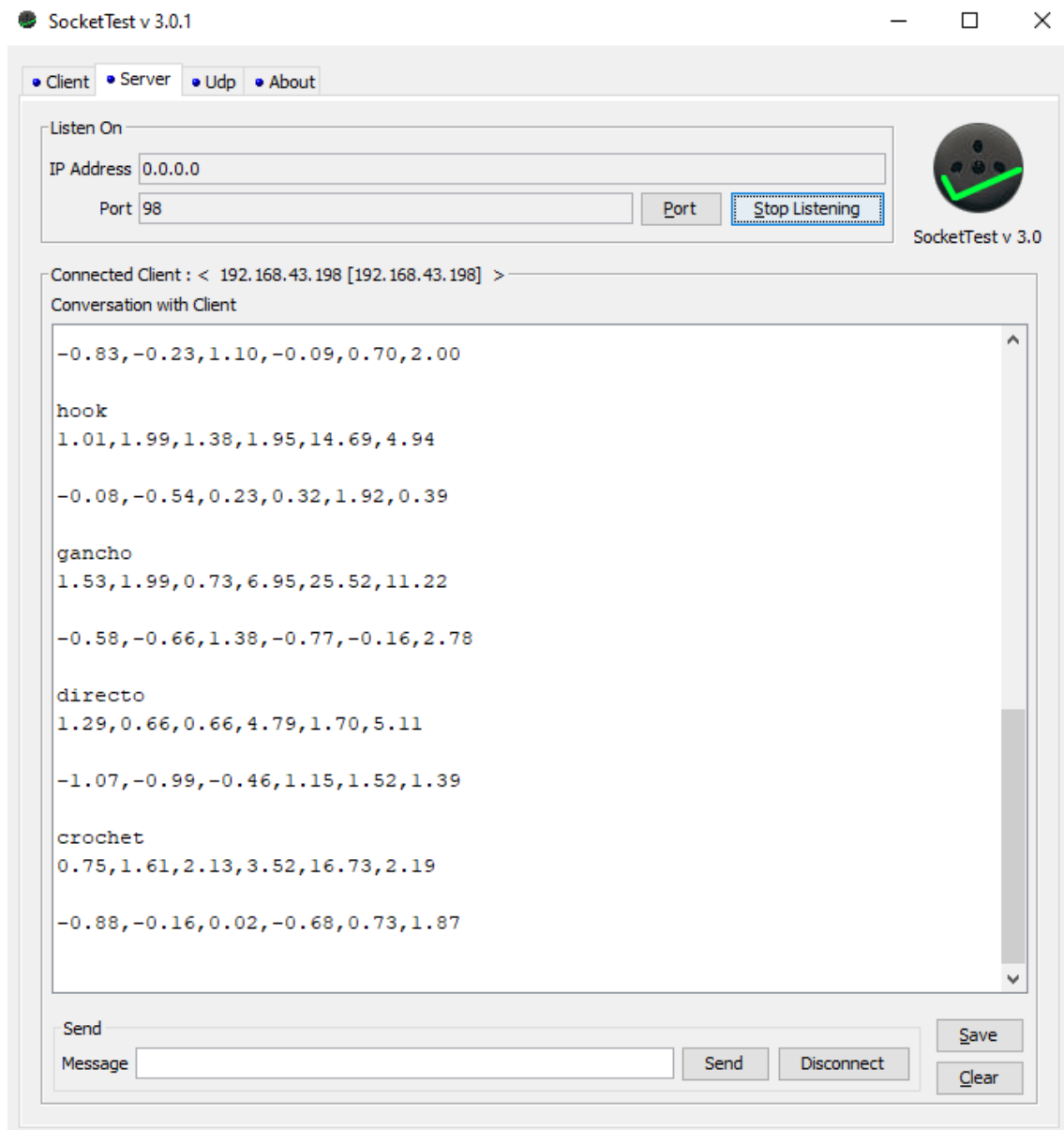


Figura 15 Recepción de datos en SocketTest

5. Trabajo Futuro

Dentro de este amplio proyecto quedan pendientes varias tareas a implementar como también mejorar el diseño.

En primer lugar una interesante implementación en el modelo sería la de incluir un sistema de auto apagado. Esto consistiría en poner el procesador en modo deep-sleep tras un largo periodo de tiempo en el que no se ha realizado ningún golpeo. Posteriormente mediante la pulsación de un botón incluido en la tarjeta se reactivaría el sistema.

Una segunda mejora sería continuar ajustando los datos para que el clasificador, aun ya funcionando bastante bien, funcione mejor.

Otra mejora del modelo sería fijarle una batería dedicada evitando el uso de baterías externas y asimismo diseñar un sistema de sujeción con correas que además cubra el sensor y la batería y permita sostenerlo en la muñeca del golpeador sin riesgo de caídas o de sufrir daños. Este sistema resultaría mucho más cómodo a la hora de insertarlo en un guante.

Además de estas hay otras muchas posibles mejoras del sistema que tras resolverlas se llegaría a obtener un aparato completamente desarrollado y dedicado. Esto conllevaría una gran cantidad de tiempo y preparación; superiores al que se le puede dedicar a un trabajo con fines lectivos.

6. Conclusiones

Se ha comprobado la amplia gama de posibilidades que ofrece un microcontrolador esp32 trabajando con sensores. En este caso con la simple adhesión de un sensor que mide aceleración y giro hemos podido desarrollar un sistema de detección y caracterización de golpes de boxeo.

También cabe destacar la facilidad de desarrollo de un sistema multitarea que ofrece un sistema operativo como FreeRTOS, utilizado en este proyecto, permitiendo separar las tareas de cálculo de las de análisis y de las de procesamiento de datos.

Por último destacar que, aunque es un sistema que queda abierto a muchas posibles mejoras, en definitiva se ha conseguido realizar un detector con unos elementos y un programa realmente sencillos.