

PickMouse:  
Dispositivo apuntador de reconocimiento  
gestual.

**Autor: Eduardo Rafael Sepúlveda de Ugarte**  
Ingeniería de Informática (2.º ciclo)  
PFC - Sistemas empotrados

**Consultor: Antoni Morell Pérez**

Fecha de entrega: 12 / Junio / 2016



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	PickMouse. Dispositivo apuntador de reconocimiento gestual.
<b>Nombre del autor:</b>	Eduardo Rafael Sepúlveda de Ugarte
<b>Nombre del consultor:</b>	Antoni Morell Pérez
<b>Fecha de entrega:</b>	06/2016
<b>Área del Trabajo Final:</b>	Sistemas empotrados.
<b>Titulación:</b>	<i>Plan de Estudios del Estudiante</i>
<b>Resumen del Trabajo (máximo 250 palabras):</b>	
<p>El proyecto consiste en desarrollar un dispositivo periférico autónomo e inalámbrico, que pueda ser utilizado como dispositivo apuntador (como el ratón), que esté incorporado en un guante, de forma que, con el movimiento de un dedo, se pueda mover el puntero de un dispositivo remoto, de forma inalámbrica. Y a través de unos botones, se pueda ejecutar los eventos de los botones del ratón.</p> <p>Para la captación de los movimientos del dedo se utilizará un acelerómetro.</p> <p>Para activar el movimiento del ratón, se pulsará un botón, de forma que se pueda desactivar el movimiento del ratón para realizar otras acciones (ej: escribir con el teclado...) sin mover el puntero.</p> <p>Con dos botones más, realizaremos las acciones de clic derecho e izquierdo del ratón.</p> <p>El último botón permitirá ejecutar el evento de "Mouse Down", es decir, hacer clic sin soltar el botón izquierdo del ratón.</p> <p>Con un módulo Wifly, nos conectaremos de forma inalámbrica al dispositivo a controlar.</p> <p>El microcontrolador LPC1769 gestionará todos los procesos.</p>	

**Abstract (in English, 250 words or less):**

This project aims to be an approach to a new concept of pointing device. This is intended to create an autonomous and wireless wearable device that can perform pointer movements of a user interface and allow execute the actions of mouse buttons.

To do so is to install on a glove an accelerometer device for motion capture a WiFly device that allows wireless communication with the remote device to be controlled and a set of buttons to activate and deactivate the device and generating events of the mouse buttons. In addition, to detect the movements, a study of patterns of signals received from the accelerometer and how to detect them be necessary.

The goal is to move your mouse to a Windows system wirelessly and able to work with it smoothly.

The objectives have been met and the results far from perfect suggest that with more development time may reach certain levels of quality and precision.

The project has been very interesting because it looks like it may open a new way of research about interaction with user interfaces.

**Palabras clave (entre 4 y 8):**

Wearable, Wireless, mouse, pointer, handsfree, embedded.

# Índice

<b>1. Introducción</b> .....	1
1.1 Contexto y justificación del Trabajo .....	1
1.2 Objetivos del Trabajo .....	2
1.3 Enfoque y método seguido .....	2
1.4 Planificación del Trabajo.....	3
1.4.1 Tareas a realizar: .....	3
1.4.2 Planificación / PEC:.....	3
1.4.3 Diagrama de Gantt: .....	4
1.5 Recursos utilizados:.....	4
<b>2. Diseño de hardware:</b> .....	6
2.1 Descripción del Hardware:.....	6
2.2 Esquema de conexiones hardware:.....	11
2.3 Test de Hardware .....	11
2.3.1 Test Modulo Wifly:.....	11
2.3.2 Test Acelerómetro: .....	13
<b>3. Diseño de software:</b> .....	18
3.1 Software para el sistema embebido sobre FreeRTOS:.....	18
3.1.1 Diseño de módulos:.....	18
3.1.2 Diagrama principal del sistema: .....	19
3.1.3 Tarea de control de botones:.....	20
3.1.4 Tarea de captación de movimientos:.....	21
3.1.5 Filtrado de Ruido .....	28
3.2 Software para sistema remoto (Windows): .....	29
<b>4. Valoración económica del proyecto:</b> .....	31
<b>5. Conclusiones</b> .....	32
<b>6. Glosario</b> .....	34
<b>7. Bibliografía</b> .....	35

## Lista de figuras

Ilustración 1. Concepto del proyecto. ....	1
Ilustración 2. Planificación. ....	3
Ilustración 3. Diagrama de Gantt. ....	4
Ilustración 4. Montaje de prototipo. ....	5
Ilustración 5. LPCXpresso LPC1769. ....	6
Ilustración 6. Acelerómetro MMA7361. ....	7
Ilustración 7. Módulo de comunicaciones WiFly RN-XV. ....	7
Ilustración 8. Conversor USB-serie CP2102. ....	8
Ilustración 9. Interruptores. ....	8
Ilustración 10. Placas de pruebas. ....	9
Ilustración 11. Brazo robótico con mando S300355. ....	9
Ilustración 12. Analizador Lógico SALEAE Logic 8. ....	10
Ilustración 13. Características Analizador Lógico SALEAE Logic 8. ....	10
Ilustración 14. Esquema de conexiones hardware. ....	11
Ilustración 15. Prueba conexión red Wifi. ....	11
Ilustración 16. Prueba conectividad por UDP. ....	12
Ilustración 17. Prueba cadena de texto. ....	13
Ilustración 18. Esquema de conexión para la prueba. ....	14
Ilustración 19. Entorno de pruebas. ....	14
Ilustración 20. Valores ADC del LPCEXpresso1769. ....	15
Ilustración 21. Valores del analizador lógico. ....	16
Ilustración 22. Valores puntuales obtenidos con el analizador lógico. ....	16
Ilustración 23. Valores especificaciones del acelerómetro MMA7361. ....	16
Ilustración 24. Test con 2000 medidas. ....	17
Ilustración 25. Diseño de módulos. ....	18
Ilustración 26. Diagrama principal del sistema. ....	19
Ilustración 27. Diagrama de Flujo. Captación del estado de botones. ....	20
Ilustración 28. Montaje brazo robótico. ....	21
Ilustración 29. Ejes acelerómetro. ....	22
Ilustración 30. Movimiento hacia arriba. ....	22
Ilustración 31. Patrón Eje X. ....	23
Ilustración 32. Patrón Eje Y. ....	23
Ilustración 33. Patrón Ejes mayor detalle. ....	24
Ilustración 34. Solución detección movimiento lateral. ....	24
Ilustración 35. Algoritmo para la captación del movimiento. ....	26
Ilustración 36. Valores calculados. ....	27
Ilustración 37. Algoritmo nuevo tras rotación de valores almacenados. ....	28
Ilustración 38. Valores pico de ruido. ....	28
Ilustración 39. Pico de ruido. ....	28
Ilustración 40. Detección pico de ruido. ....	29
Ilustración 41. Funciones. ....	31
Ilustración 42. Estudio económico. ....	31

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

Si clasificamos el conjunto de dispositivos apuntadores (capaces de mover el puntero de una interface de usuario) por su conectividad (inalámbricos y por cable), nos encontramos que los dispositivos por cable suelen ser utilizados en entornos profesionales (mouse, trackball, touchpad, tableta digitalizadora...) y sin embargo los apuntadores inalámbricos se utilizan para un entorno lúdico (mandos de videoconsolas). Además, estos dispositivos inalámbricos suelen ser dispositivos pesados, los cuales no nos permiten realizar otra acción mientras tenemos el dispositivo en la mano.

La propuesta de este trabajo es realizar un dispositivo apuntador wearable que no obligue al usuario a coger un dispositivo nuevo con la mano ya que lo lleva siempre encima, ni tampoco a desplazar su mano hasta un lugar específico, permitiéndole libertad espacial total inalámbrica, de forma que no solo pueda ser utilizado con fines lúdicos sino también en un entorno profesional.

Para esto propongo estudiar y plantear una solución a través de un sistema embebido, que se instale en un guante cómodo y que sea capaz de reconocer los gestos de un dedo para mover el puntero de una interfaz de usuario.

La idea final, es un único dispositivo capaz de ser utilizado en distintos entornos, por ejemplo: en un PC para mover el ratón sin necesidad de levantar la mano del teclado, en una presentación sin necesidad de estar apuntando al proyector o pantalla, en un sistema de reproducción de películas en el salón de una casa o en un videojuego sin tener que cambiar de dispositivo y sin inutilizar una mano.



Ilustración 1. Concepto del proyecto.



## 1.2 Objetivos del Trabajo

Acoplar un sistema embebido a un guante para desarrollar un dispositivo wearable (Picmouse).

PickMouse debe poderse conectar de forma inalámbrica a un sistema (ej: Windows) y enviarle la información necesaria.

PickMouse debe poder reconocer los movimientos de un dedo y generar la información necesaria para mover el puntero en el sistema remoto (Windows)

PickMouse debe tener unos botones con los que se puedan simular los botones izquierdo y derecho del mouse y poder activar y desactivar el dispositivo.

PickMouse debe tener un identificador visual (LED) para informar de cuando el dispositivo está listo para usarse y si está activa la captura de movimientos.

## 1.3 Enfoque y método seguido

Debido a que se trata de un proyecto de investigación, la estrategia se ha basado en intentar implementar este proyecto con elementos nuevos, el objetivo no es mejorar una solución ya existente, sino crear un nuevo concepto de dispositivo apuntador.

Además, hubiera hecho falta encontrar una plataforma de hardware y software libre sobre la que basarse y que se pudiera adaptar al concepto de dispositivo wearable (ligero, ergonómico etc...) y tras una breve fase de búsqueda, no encontré nada parecido y esto habría requerido mucho más tiempo sin presentar en principio grandes ventajas.

El proyecto, por tanto, está enfocado en desarrollar el dispositivo desde cero y la metodología la he basado en priorizar la lectura de la documentación de los componentes y diseño de conexiones hardware para poder desarrollar lo antes posible pequeños programas de pruebas para poder asegurar lo antes posible si los componentes de captación de movimientos eran suficientes y poder cambiarlos en caso de no encontrar unos resultados aceptables. Una vez realizada la aceptación del hardware se ha procedido a desarrollar el resto de partes del proyecto.

## 1.4 Planificación del Trabajo

### 1.4.1 Tareas a realizar:

El conjunto de tareas a realizar está dividido en cinco partes:

1. Diseño de hardware:
  - Selección de componentes.
  - Realización de esquemas de conexiones.
  - Comprobación del hardware.
2. Diseño de software:
  - Selección del sistema receptor de los movimientos captados por el dispositivo "PickMouse".
  - Selección del protocolo de comunicaciones.
  - Diseño de módulos de software necesarios.
  - Diseño de ciclo de vida y ejecución.
3. Desarrollo software:
  - Desarrollo de controladores para LPC1769.
  - Desarrollo del software de gestión de los movimientos gestuales y calibración del acelerómetro.
  - Desarrollo del software de comunicación con sistemas remotos a través de redes inalámbricas (usando Wifi) para el microcontrolador LPC1769.
  - Desarrollo de receptor de comunicaciones y generación de movimientos del puntero y eventos "clic".
4. Test y pruebas:
  - Integración de los módulos y gestión de multitarea.
  - Calibraciones de la sensibilidad de captación de movimientos
  - Test de usabilidad y rendimiento.
5. Documentación:
  - Desarrollo de la memoria del proyecto:
  - Desarrollo de la presentación del proyecto:

### 1.4.2 Planificación / PEC:

Nombre	Fecha de inicio	Fecha de fin	Entregas PEC	
			PEC1:	Propuesta de trabajo y Panificación.
☐ ● FASE DE DISEÑO / REVISION	10/03/16	13/03/16		
● Diseño de hardware	10/03/16	11/03/16		
● Diseño de software	12/03/16	13/03/16		
☐ ● FASE DE DESARROLLO	14/03/16	17/05/16		
● Desarrollo de drivers	14/03/16	31/03/16	PEC2	Primera entrega del trabajo.
● Desarrollo de SW de control gestual	1/04/16	20/04/16		
● Desarrollo de SW de comunicaciones	21/04/16	17/05/16		
☐ ● FASE DE TEST	18/05/16	25/05/16		
● Integracion de todos los modulos	18/05/16	20/05/16	PEC3	Segunda entrega del trabajo.
● Test y calibraciones finales	21/05/16	25/05/16		
☐ ● FASE DE DOCUMENTACION	26/05/16	19/06/16		
● Creación de la memoria	26/05/16	12/06/16		Entra de la memoria.
● Creación de la presentación	13/06/16	19/06/16		Entrega de presentacion y codigo.

**Ilustración 2. Planificación.**

### 1.4.3 Diagrama de Gantt:

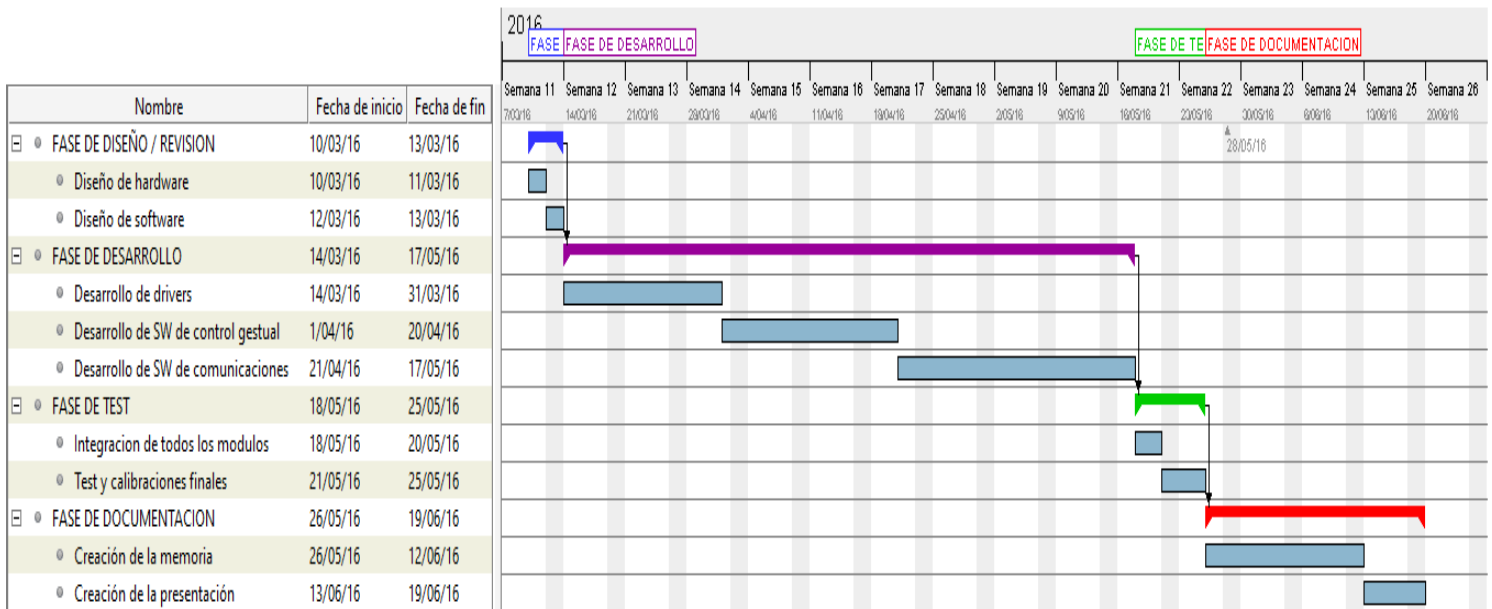


Ilustración 3. Diagrama de Gantt.

### 1.5 Recursos utilizados:

#### Hardware:

- 1 Microcontrolador LPCXpresso 1769:
- 1 Modulo Wifly RN-XV.
- 1 Convertidor USB 2.0 to UART VP2102.
- 1 Acelerómetro MMA7361 / Adaptador XBEE to DIP.
- 4 Interruptores de 4 pines.
- 1 Batería externa con conexión USB.
- 1 Guante tela/plástico.
- Juegos de cables para placas de pruebas.

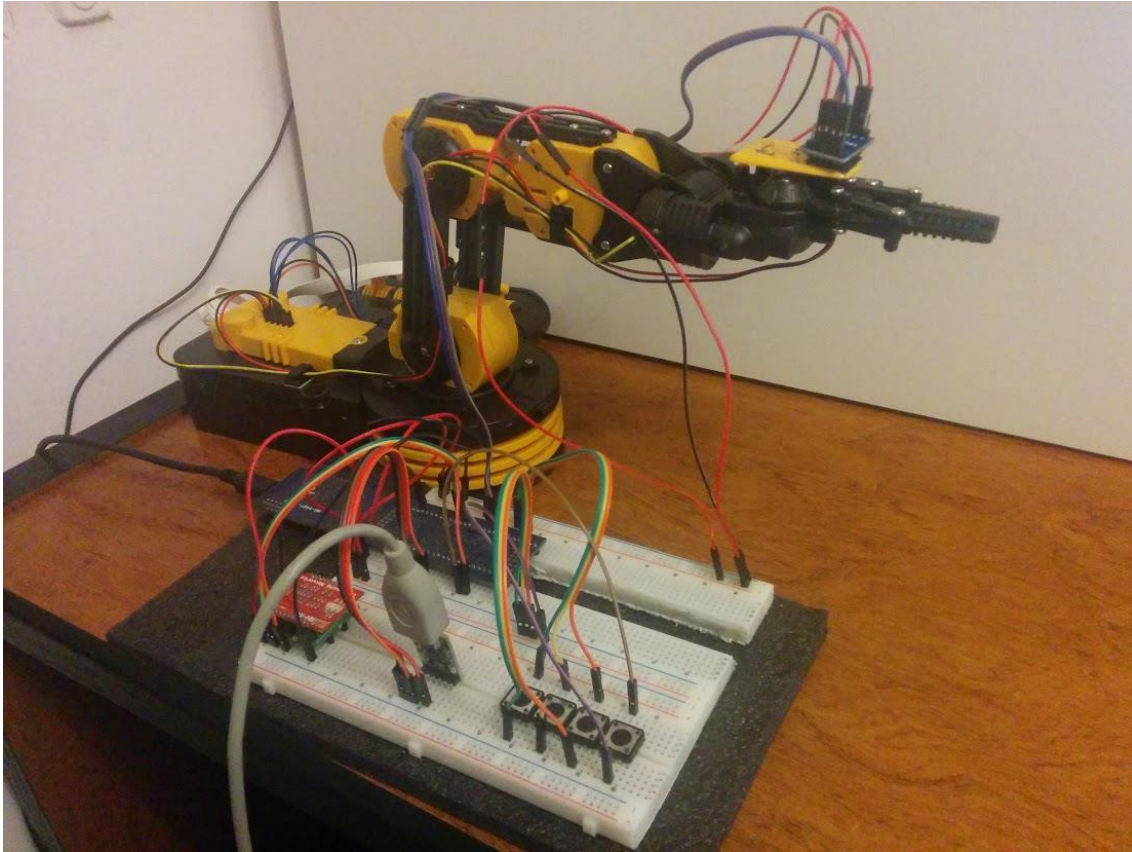
#### Herramientas para el desarrollo:

- 1 Placas de pruebas.
- 1 Brazo Robótico.
- 1 Analizador Lógico Saleae 8.

#### Software:

- IDE LPCXpresso
- Sistema Operativo FreeRTOS (FreeRTOS\_Library)
- Librería para LPCExpresso1769 (CMSISv2p00\_LPC17xx ).
- IDE MS Visual Studio. Para el desarrollo del programa receptor en el entorno Windows.
- LibreOffice Draw, MS Word, MS PowerPoint, GanttProject 2.7.2 Ostrava
- Aplicación para Android UDP Sender / Receiver.

## 1.6 Montaje de prototipo.



**Ilustración 4. Montaje de prototipo.**

## 1.7 Indicaciones de Uso

El dispositivo PickMouse necesita de una etapa de Inicialización o calibración una sola vez en el momento que se enciende.

Durante esta fase el dedo índice del usuario debe reposar sobre una mesa horizontal y mantenerlo quieto durante unos pocos segundos.

En esta fase, el dispositivo calibrará la captación de movimientos y se conectará a la red Wifi local en la que se halle el dispositivo a controlar.

En el momento en que la inicialización sea correcta se encenderá un LED rojo el cual indicará que el dispositivo está listo para usarse.

Este led no indicará ningún tipo de conexión con un dispositivo remoto.

Mediante el botón número 4 se podrá desactivar la captación de movimientos del ratón y activarla de nuevo sin necesidad de pasar por una nueva etapa de calibración.

Mediante el botón número 3 se permite enviar el evento “LEFT\_DOWN” (pulsar sin soltar) y “LEFT\_UP” (levantar) del botón izquierdo del ratón, para permitir coger objetos del escritorio y moverlos.

El botón 2 permite el clic del botón derecho del ratón.

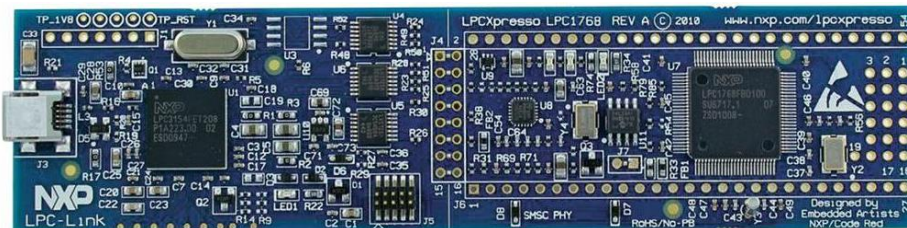
El botón 1 permite el clic del botón izquierdo del ratón.

## 2. Diseño de hardware:

El proyecto se basa en el kit proporcionado por la universidad.

### 2.1 Descripción del Hardware:

- LPCXpresso LPC1769



**Ilustración 5. LPCXpresso LPC1769.**

Se trata de una tarjeta con un microcontrolador Cortex-M3 con conector JTAG para depurar y con capacidad de establecer modos de consumo reducido.

Las especificaciones del microcontrolador son las siguientes.

- Procesador: NXP's Cortex-M3 LPC1769 microcontrolador en LQFP100.
- Memoria Flash: 512kB
- Memoria de datos: 64kB SRAM
- Reloj: 12MHz para la CPU.
- Dimensiones: 35 x 140mm
- Alimentación: 3.15V-3.3V alimentación externa a través de USB vía JTAG (LPC-LINK)
- Conectores: LPC1769 dispone de pines de expansión (2x27 filas de pines).
- Interfaces de comunicaciones:
  - 1 controlador USB 2.0.
  - 4 UARTs.
  - 2 CAN
  - 1 Ethernet
  - 8 entradas analógicas (ADC) de 12 bits.
  - Reloj de tiempo real.
  - PWM.

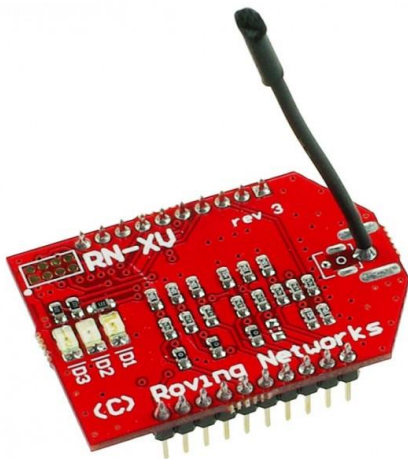
- Acelerómetro MMA7361



**Ilustración 6. Acelerómetro MMA7361.**

Se trata de un acelerómetro analógico de bajo consumo. Las principales características son:

- Alimentación: 2.2v - 3.6v
  - Alta sensibilidad con dos modos (1,5g y 6g).
  - Tres salidas analógicas, una para cada eje.
  - Tiempo de respuesta 0,5ms.
  - Sensor y salida digital dedicada de caída libre (0g).
- Módulo de comunicaciones WiFi RN-XV:



**Ilustración 7. Módulo de comunicaciones WiFi RN-XV**

Este es un dispositivo que nos permite conectarnos a una red wifi 802.11 b/g. Es posible comunicarnos con él a través de comandos por su puerto serie.

Las principales características son:

- Basado en 802.15.4
  - Muy bajo consumo: 4uA (sleep), 38mA (activo)
  - El stack TCP/IP interno soporta: DHCP, UDP, DNS, ARP, ICMP, cliente HTTP, cliente FTP y cliente TCP
  - Potencia configurable por software: 0dBm a 12dBm
  - Interfaz: TTL UART
  - Velocidad de datos: Hasta 464Kbps port UART
  - Soporta redes Adhoc.
  - 8 pines I/O
  - 3 salidas analógicas para sensores
  - RTC para time-stamping, auto-sleep y auto-wakeup
  - Alimentación: 3.3VDC
- 
- Conversor USB-serie CP2102:

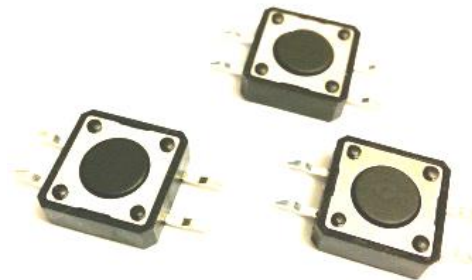


**Ilustración 8. Conversor USB-serie CP2102.**

Convertidor de USB a Serial(TTL). Esta placa se conecta directamente al módulo USB y soporta auto-reset.

Especificaciones:

- Especificación USB 2.0 con 12 Mbps de velocidad completa .
  - USB estándar tipo A conector de 6 pines macho y TTL . 6pins de 3.3V , RST , TXD , RXD GND y + 5V .
  - Carácter Evento transmisión descanso apoyo Line.
  - Suspend estados USB compatibles mediante SUSPENDEN pines.
  - Rango de temperatura : -40 a 85 .
  - Soporta Windows 98SE , 2000 , XP , Vista, Window7 , Linux 2.40 .
- 
- Interruptores:



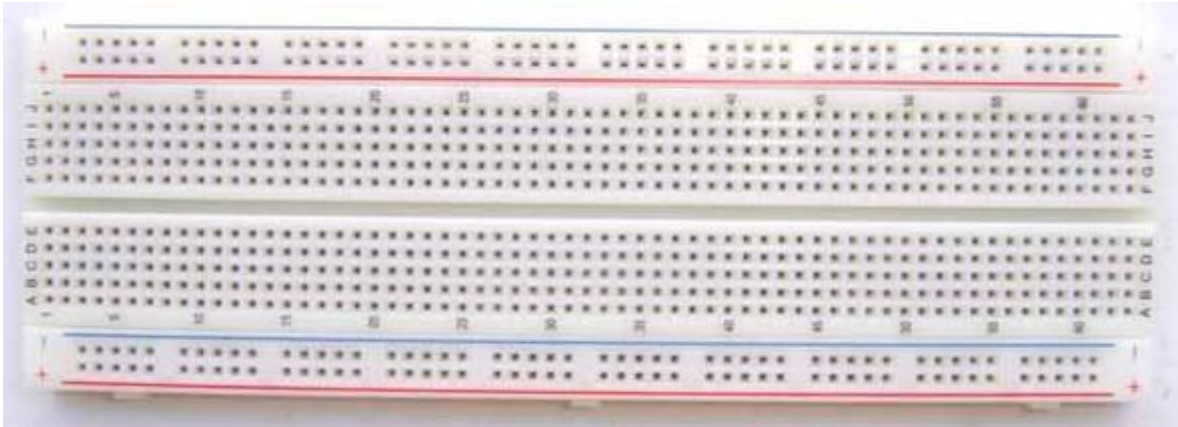
**Ilustración 9. Interruptores.**

Se trata de cuatro interruptores sencillos de cuatro polos.

Herramientas de apoyo utilizadas:

- Placas de pruebas:

Se han utilizado dos placas para realizar el esquema eléctrico de conexiones inicial.



**Ilustración 10. Placas de pruebas.**

- Brazo robótico con mando S300355:



**Ilustración 11. Brazo robótico con mando S300355.**

Brazo robot con mando cableado tiene cinco grados de libertad. Equipado con 5 motores independientes, las palancas de la unidad de control controlan la rotación de la base, el movimiento del codo y de la muñeca y la apertura y cierre de la pinza. He utilizado este brazo robótico que permite simular los movimientos de un dedo de forma constante sin vibraciones, debido a que en los primeros estudios de movimiento se hacía muy difícil realizar las pruebas con una mano real, ya que el pulso introducir medidas difíciles de analizar. Con ayuda de este brazo se pudieron realizar movimientos “limpios” con los que estudiar el comportamiento del acelerómetro en situaciones ideales.



- Analizador Lógico SALEAE Logic 8:



Ilustración 12. Analizador Lógico SALEAE Logic 8.

Se trata de un analizador lógico de 8 canales que permite la grabación de canales analógicos y digitales.

Sus principales características son:

		Logic 8
<b>Inputs</b> <a href="#">More</a>		
Multi-Use (A/D/Both) <i>z</i>		8
Digital-only		-
<b>Digital</b> <a href="#">More</a>		
Sample Rate (max) <i>z</i>		100 MS/s
Fastest Digital Signal		25 MHz
Sample Depth (typical) <i>z</i>		10+ Billion
Supported Logic Levels		1.8V - 5.5V
Works with RS-232, 422/3, 485		Yes, directly
<b>Analog</b> <a href="#">More</a>		
Sample Rate (max) <i>z</i>		10 MS/s
Bandwidth (-3dB) <i>z</i>		1 MHz
Number of bits <i>z</i>		10 bits
Input Voltage Range <i>z</i>		0 V to 5 V

Ilustración 13. Características Analizador Lógico SALEAE Logic 8.

## 2.2 Esquema de conexiones hardware:

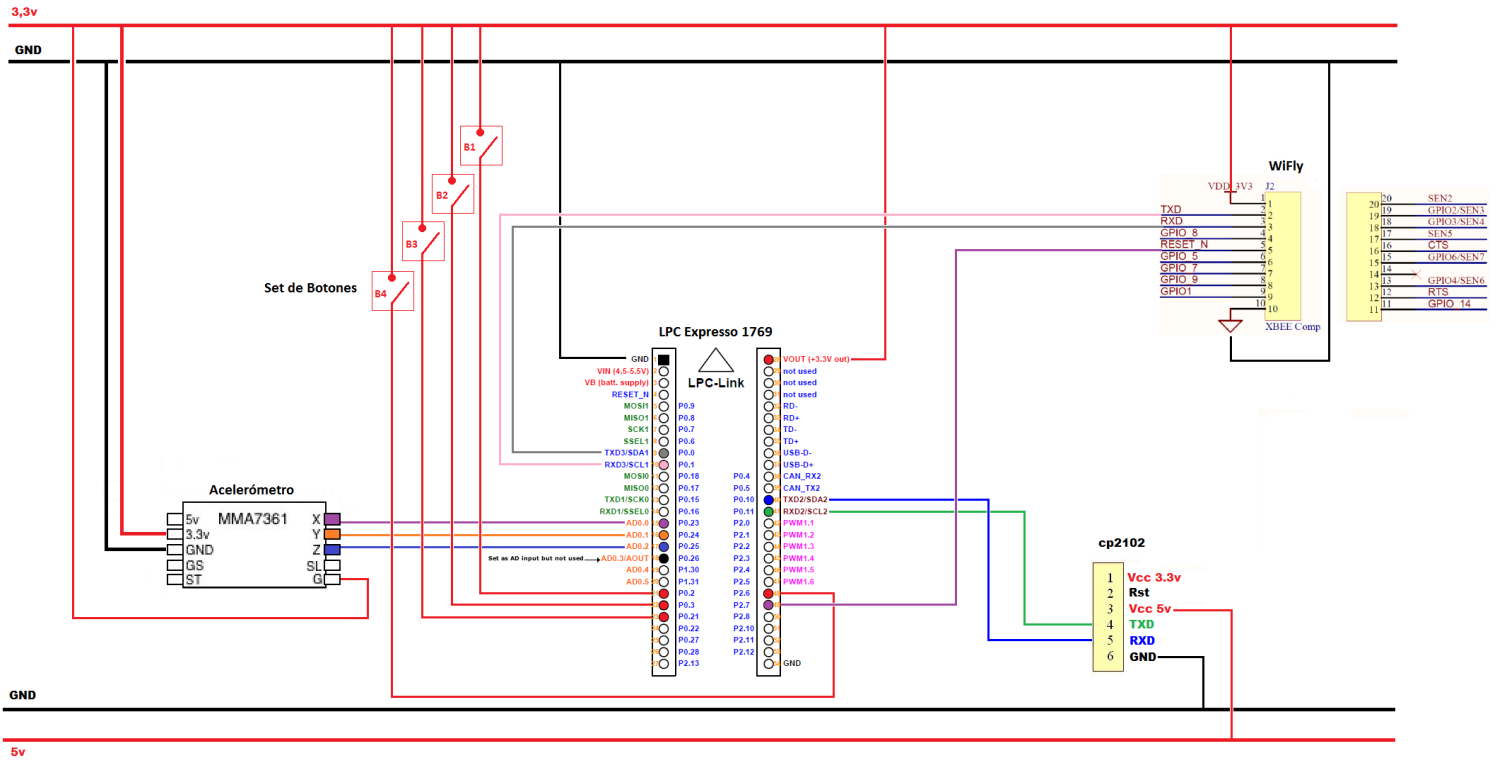


Ilustración 14. Esquema de conexiones hardware.

## 2.3 Test de Hardware

### 2.3.1 Test Modulo Wifly:

La primera prueba consistió en asegurar la conexión a una red Wifi, en las siguientes trazas se puede apreciar como la conexión es correcta:

```

Conectando a WLAN...

Check_Read READED: CMD

Check_Read READED: set w a 4
AOK
<4.00>

Check_Read READED: set w s MYWIFINETWORK
AOK
<4.00>

Check_Read READED: set w p PASSWDMEGASECURE
AOK
<4.00>

Check_Read READED: join MYWIFINETWORK

Auto-Assoc MYWIFINETWORK chan=5 mode=MIXED SCAN OK

<4.00>
Joining MYWIFINETWORK now..

Associated!

DHCP: Start

WLAN connected.
    
```

Ilustración 15. Prueba conexión red Wifi.

La segunda prueba consistió en comprobar la conectividad por UDP.

En las siguientes trazas se puede ver como la configuración se realiza correctamente:

```
Configurando modulo WiFly...
SENDING: set ip proto 1
Wifly_Set READED: set ip proto 1
AOK
<4.00>

SENDING: set broadcast interval 0
Wifly_Set READED: set broadcast interval 0
AOK
<4.00>

SENDING: set ip host 255.255.255.255
Wifly_Set READED: set ip host 255.255.255.255
AOK
<4.00>

SENDING: set ip flags 0x00
Wifly_Set READED: set ip flags 0x00
AOK
<4.00>

SENDING: set ip remote 8888
Wifly_Set READED: set ip remote 8888
AOK
<4.00>

SENDING: set ip local 8888
Wifly_Set READED: set ip local 8888
AOK
<4.00>

SENDING: set wlan join 1
Wifly_Set READED: set wlan join 1
AOK
<4.00>
```

**Ilustración 16. Prueba conectividad por UDP.**

La tercera prueba consistió en enviar una cadena de texto "TEST OK!" a una aplicación de Android receptora de paquetes UDP y el resultado fue satisfactorio:

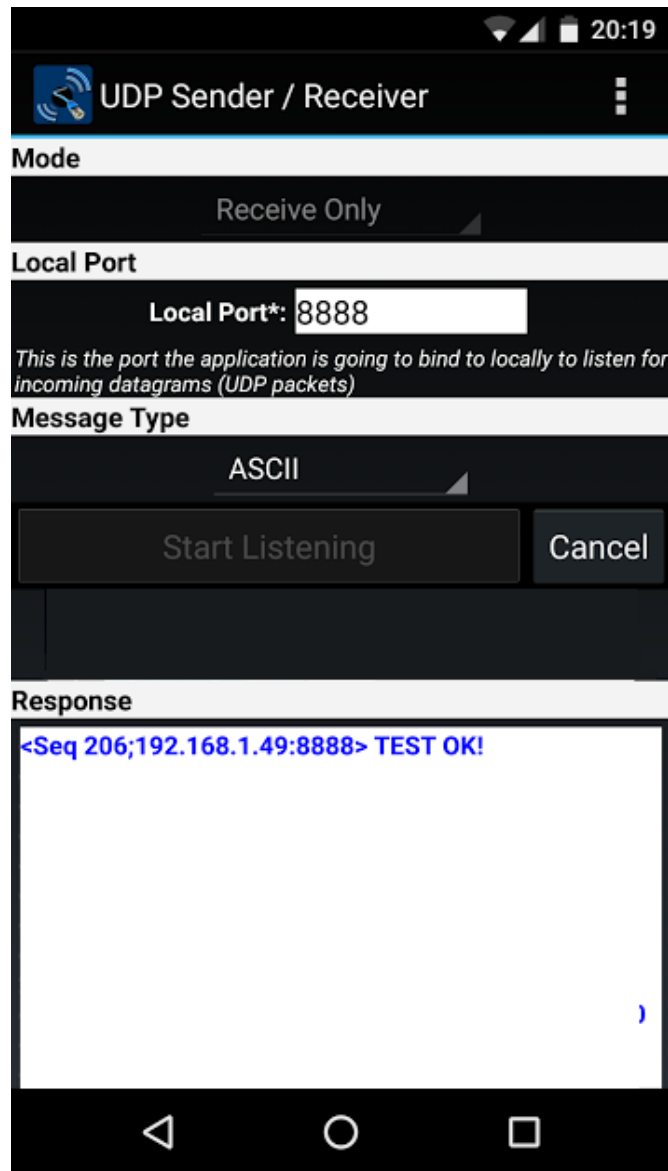


Ilustración 17. Prueba cadena de texto.

### 2.3.2 Test Acelerómetro:

Para estas pruebas he utilizado un analizador lógico que conecté a la salida de los canales del acelerómetro para poder analizar los datos emitidos por el acelerómetro y compararlos con los proporcionados por el ADC del microcontrolador.

Este es el esquema de conexión para la prueba:

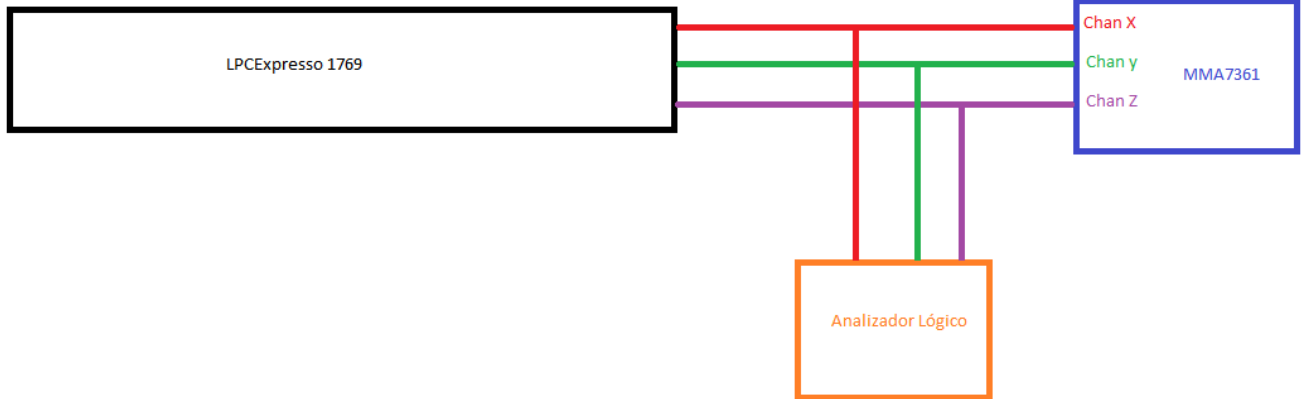


Ilustración 18. Esquema de conexión para la prueba.

Fotografía del entorno de pruebas:

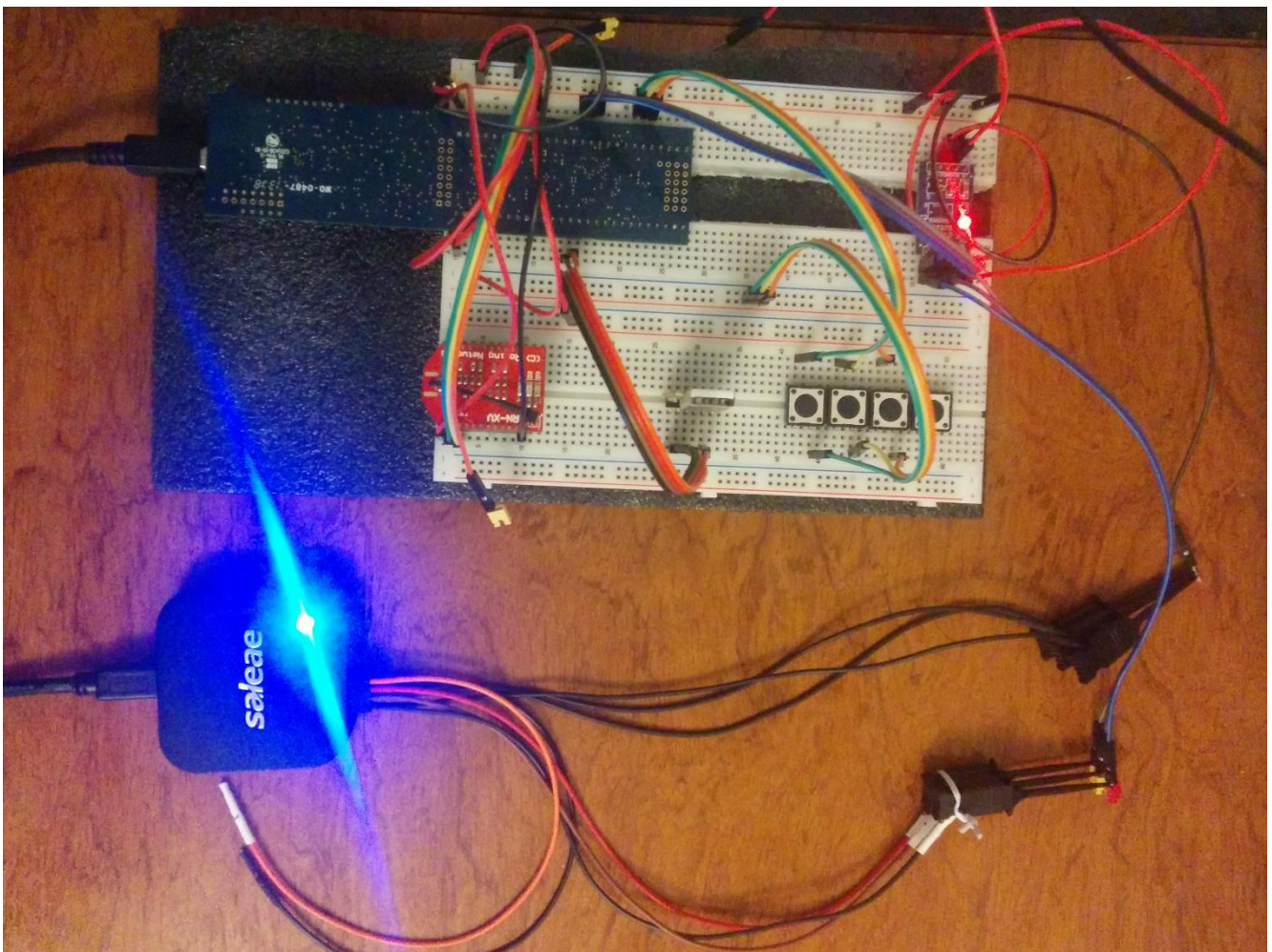


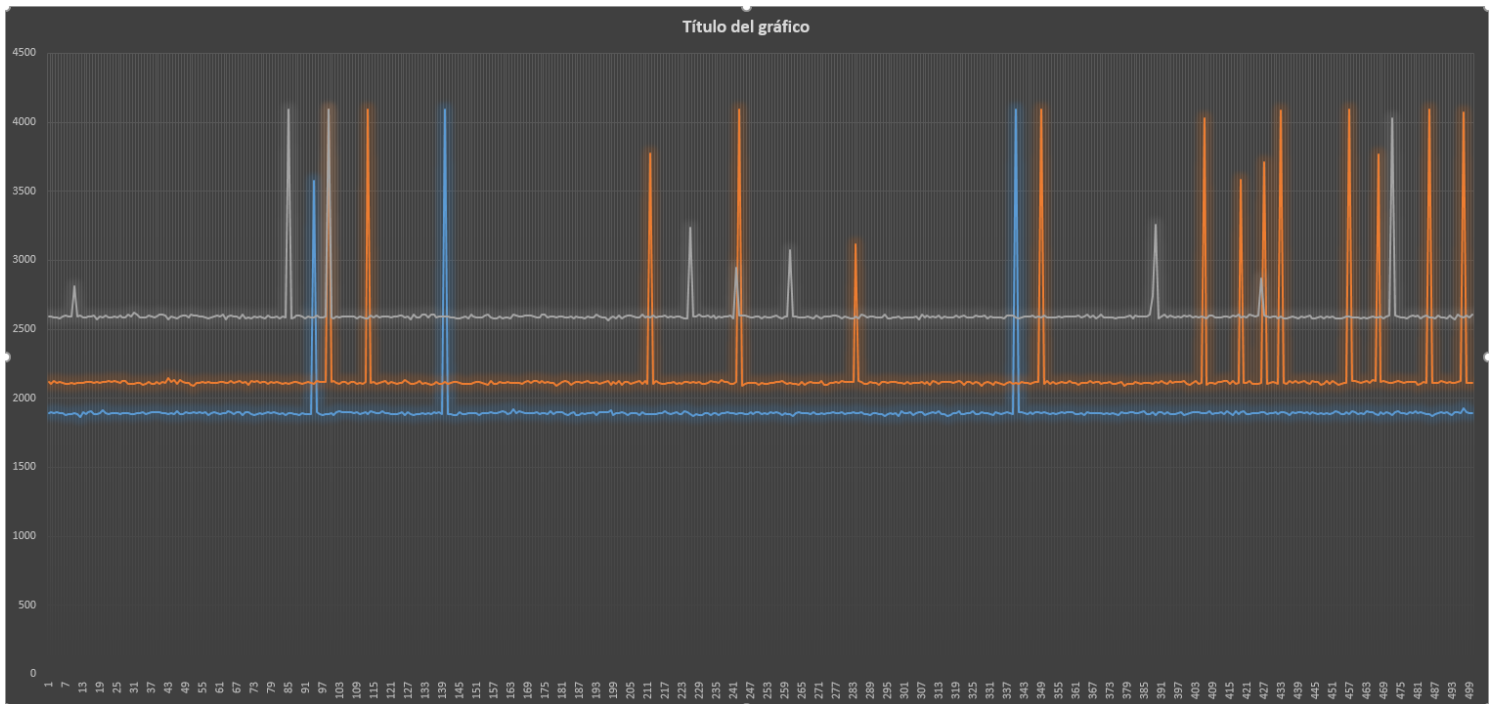
Ilustración 19. Entorno de pruebas.

He usado un test sencillo en el que se configura el ADC y se toman 500 muestras teniendo el acelerómetro en completo reposo.

Estas muestras las he utilizado como fuente para hacer una gráfica en Excel. Como indica el diagrama y fotografía he conectado a los 3 canales de salida del acelerómetro un analizador lógico marca SALEAE.

Durante las pruebas he arrancado el analizador lógico, el cual ha grabado el voltaje enviado por el acelerómetro al microcontrolador.

Estos son los valores recogidos por el ADC del LPCExpreso1769.



**Ilustración 20. Valores ADC del LPCExpreso1769.**

Se puede apreciar que la mayor parte del tiempo los valores son constantes, pero en bastantes ocasiones tenemos “picos” da la sensación de que hay algún tipo de ruido.

Estos son los valores recogidos por el analizador lógico en la misma prueba:

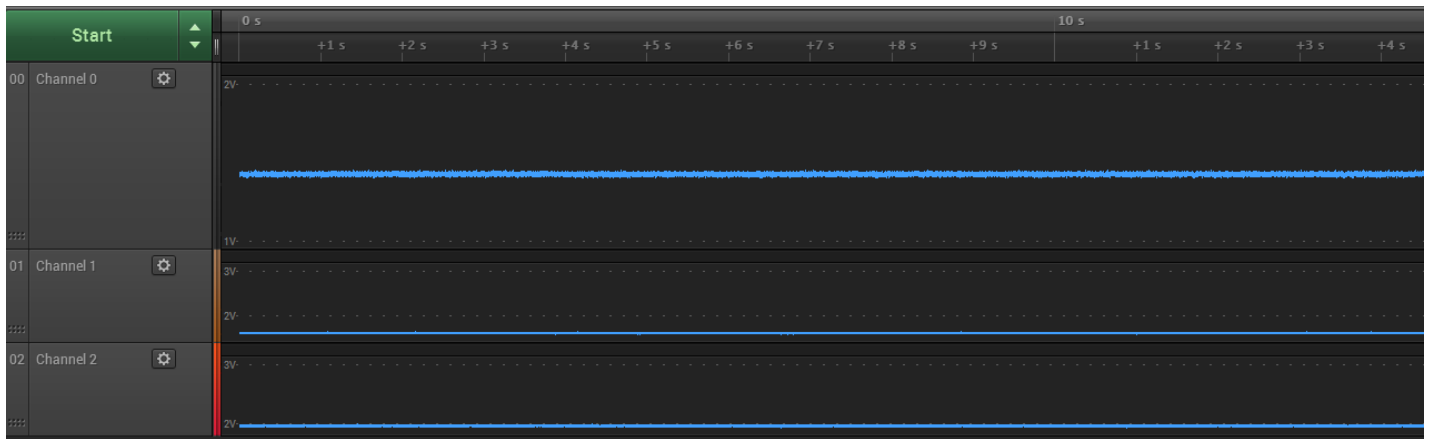


Ilustración 21. Valores del analizador lógico.

En la imagen se aprecia que a pesar de lo que dice el ADC la entrada analógica es constante, y esos picos no son producidos por el acelerómetro.

A continuación, una muestra de unos valores puntuales obtenidos con el analizador lógico:

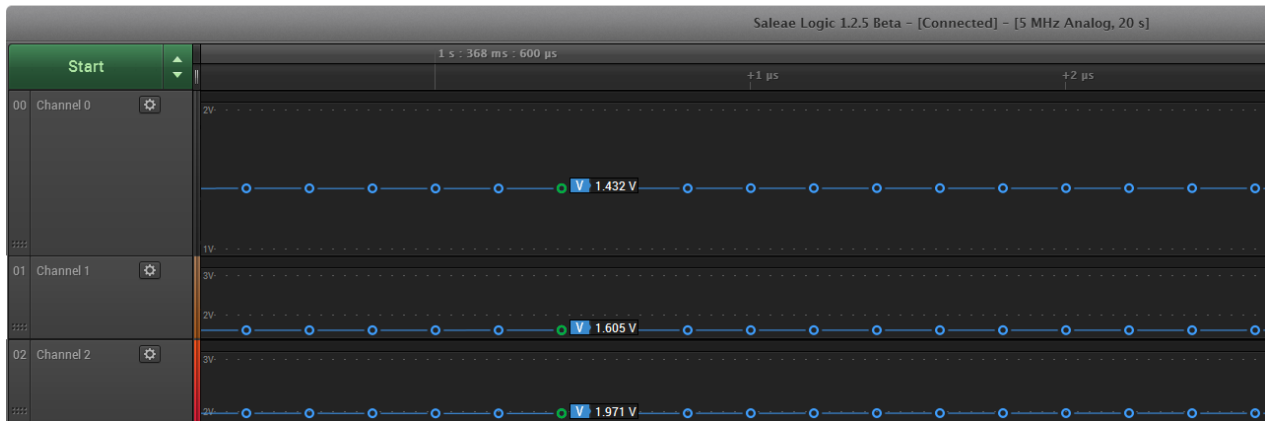


Ilustración 22. Valores puntuales obtenidos con el analizador lógico.

Los valores son muy parecidos a los proporcionados en las especificaciones del acelerómetro MMA7361.

Output Signal					
Zero-g ( $T_A = 25^\circ\text{C}$ , $V_{DD} = 3.3\text{ V}$ ) <sup>(5), (6)</sup>	$V_{OFF}$	1.485	1.65	1.815	V

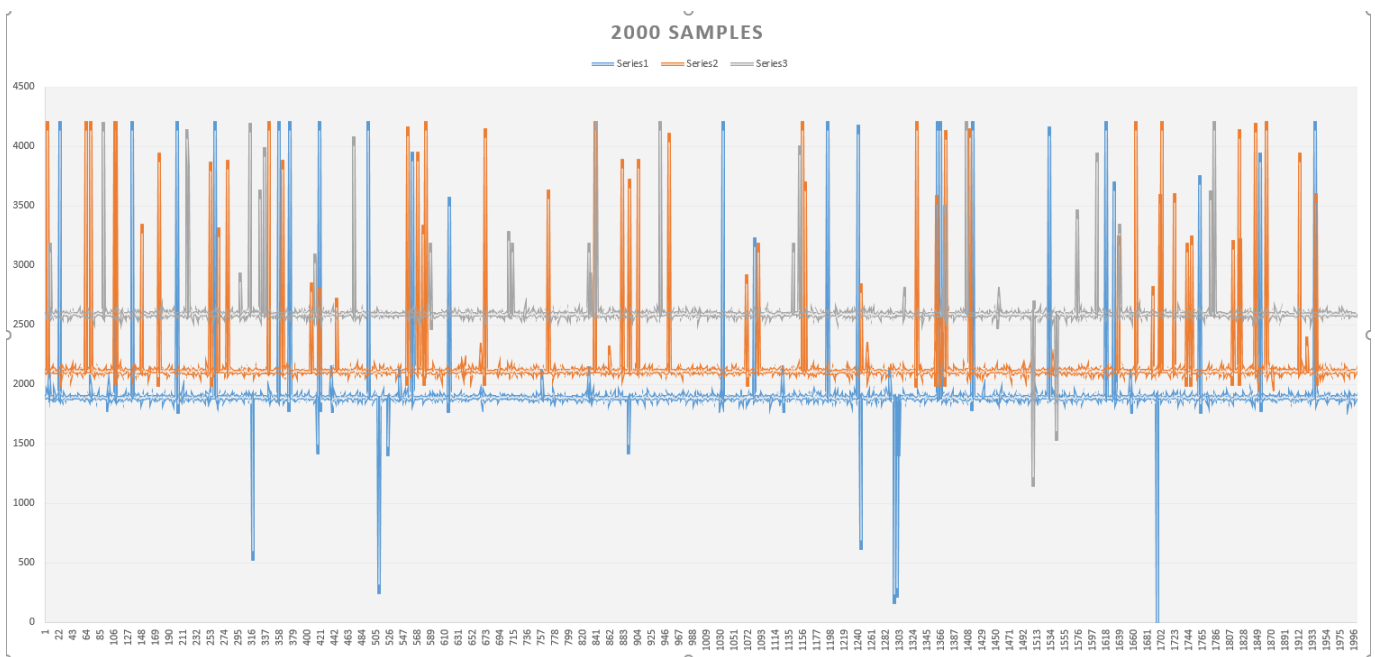
Ilustración 23. Valores especificaciones del acelerómetro MMA7361.

Por lo tanto, parece que el acelerómetro tiene unos valores en reposo “normales” y que no es la fuente de los picos.

En este foro comentan el mismo problema y parece es algo ya conocido y que no hay solución aun por parte del fabricante...

<https://www.lpcware.com/content/forum/lpcxpresso-lpc1769-adc-work-not-correctly>

A continuación, el mismo test con 2000 medidas:



**Ilustración 24. Test con 2000 medidas.**

Las conclusiones del test fueron, que debido a que el uso de este microcontrolador es un requisito del proyecto, haría falta implementar un filtro de software para eliminar el ruido introducido por el ADC. (Su estudio y solución se exponen en el apartado de diseño de software).



### 3. Diseño de software:

#### 3.1 Software para el sistema embebido sobre FreeRTOS:

##### 3.1.1 Diseño de módulos:

El modulo central es la función main, que utiliza el resto de módulos como se muestra en la siguiente imagen, en la que se define la funcionalidad general de cada módulo.

Módulos de software del Sistema.

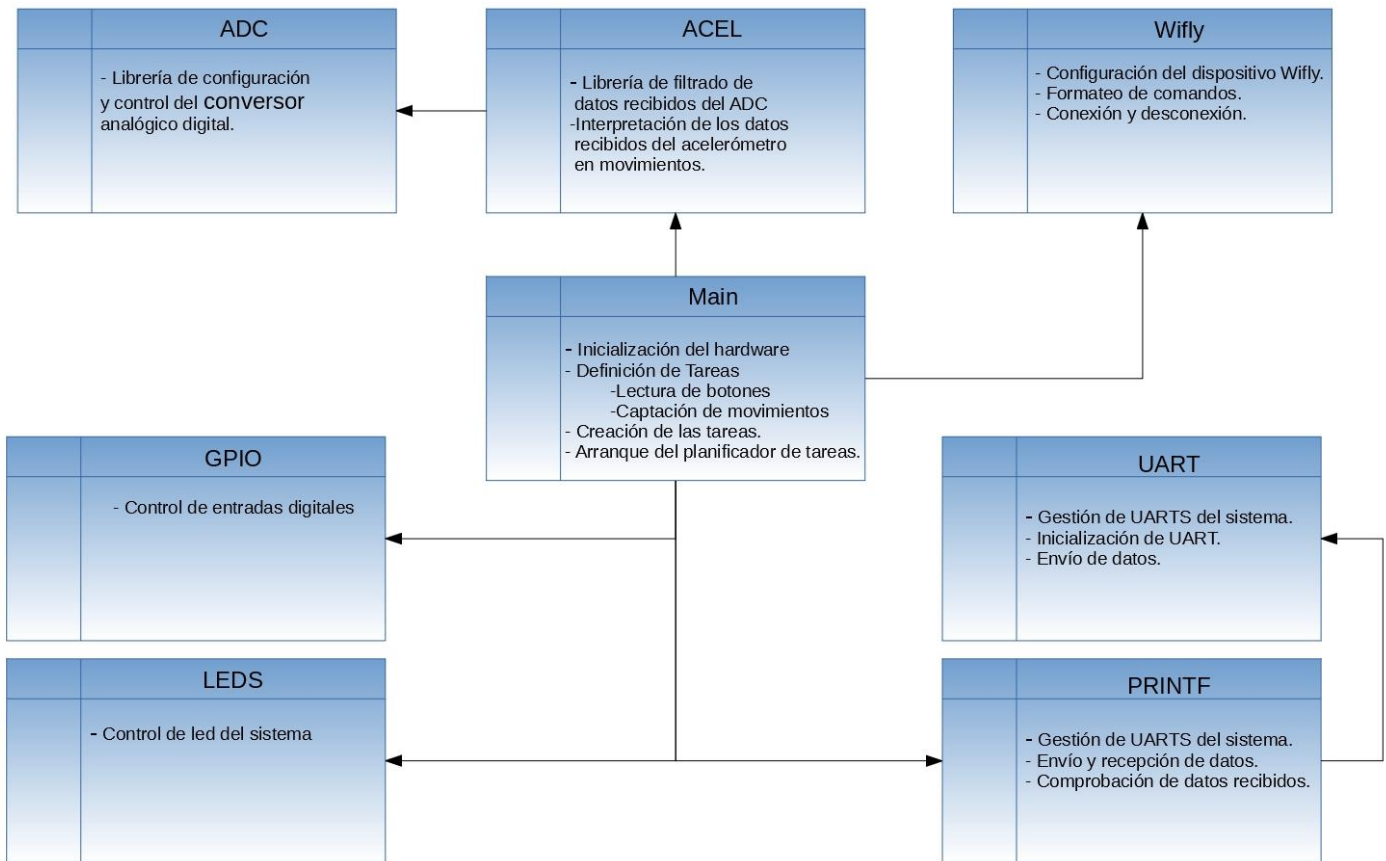
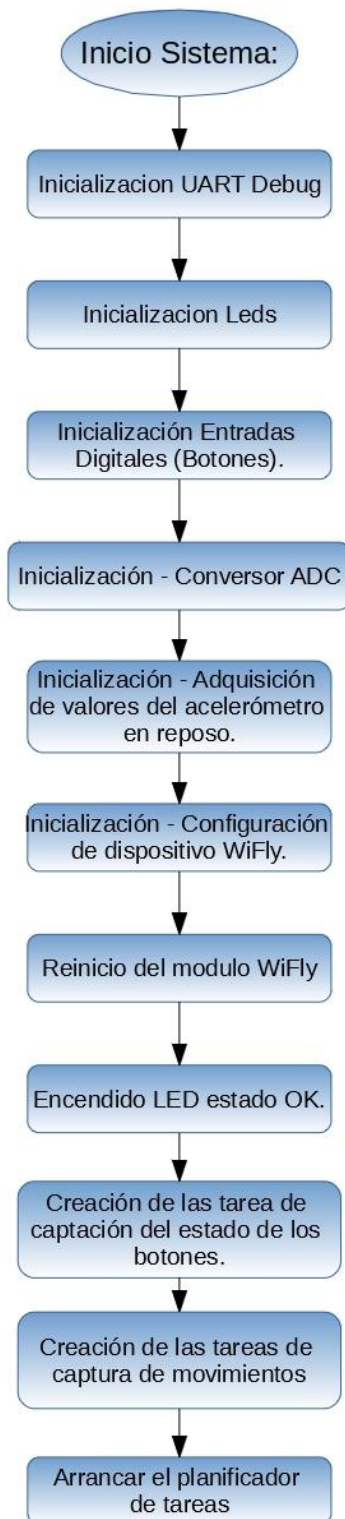


Ilustración 25. Diseño de módulos.

### 3.1.2 Diagrama principal del sistema:



**Ilustración 26. Diagrama principal del sistema.**

### 3.1.3 Tarea de control de botones:

A continuación, el diagrama de Flujo de la tarea de captación del estado de los botones.

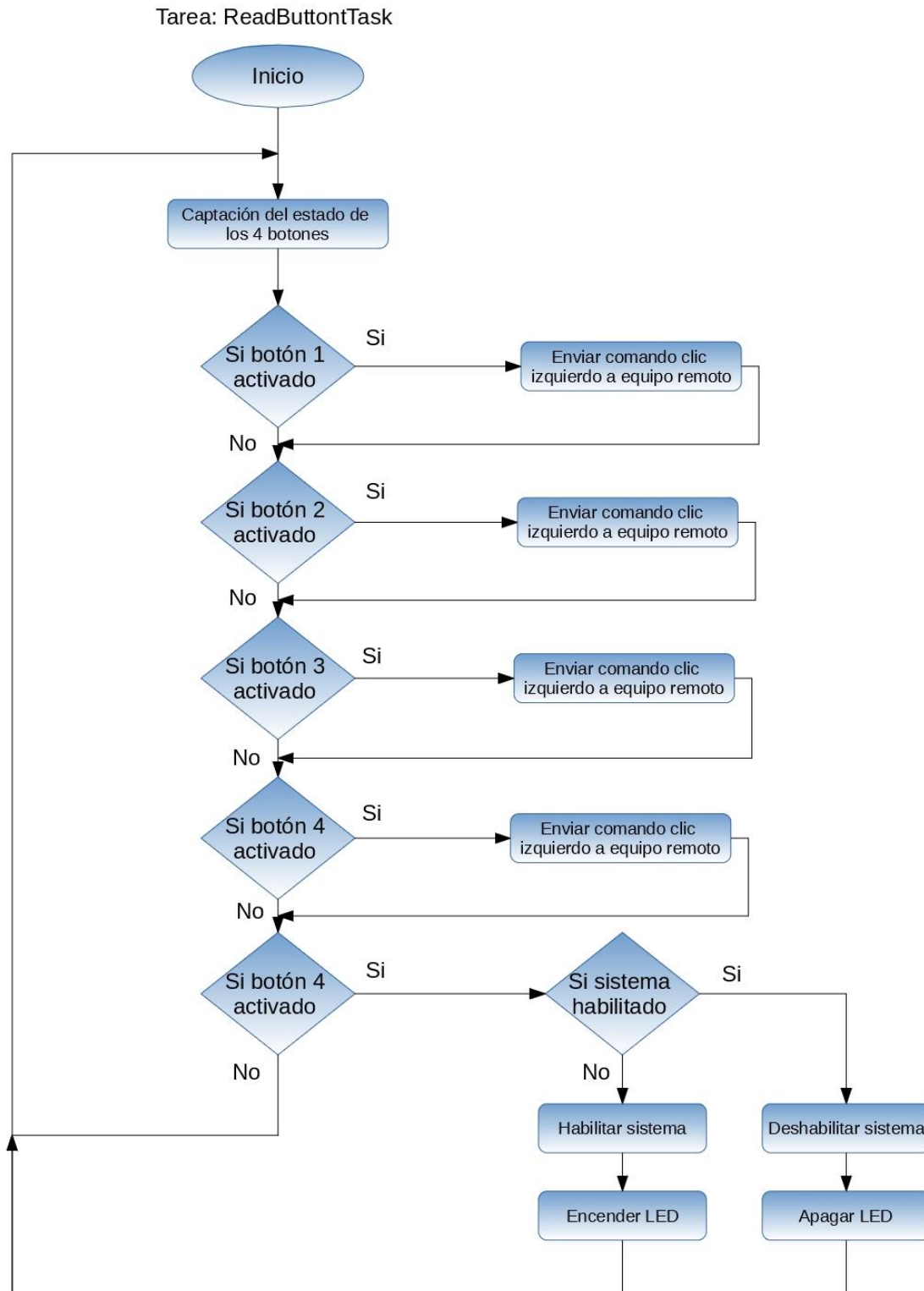


Ilustración 27. Diagrama de Flujo. Captación del estado de botones.

### 3.1.4 Tarea de captación de movimientos:

Captación de valores reales y reconocimiento de los gestos:

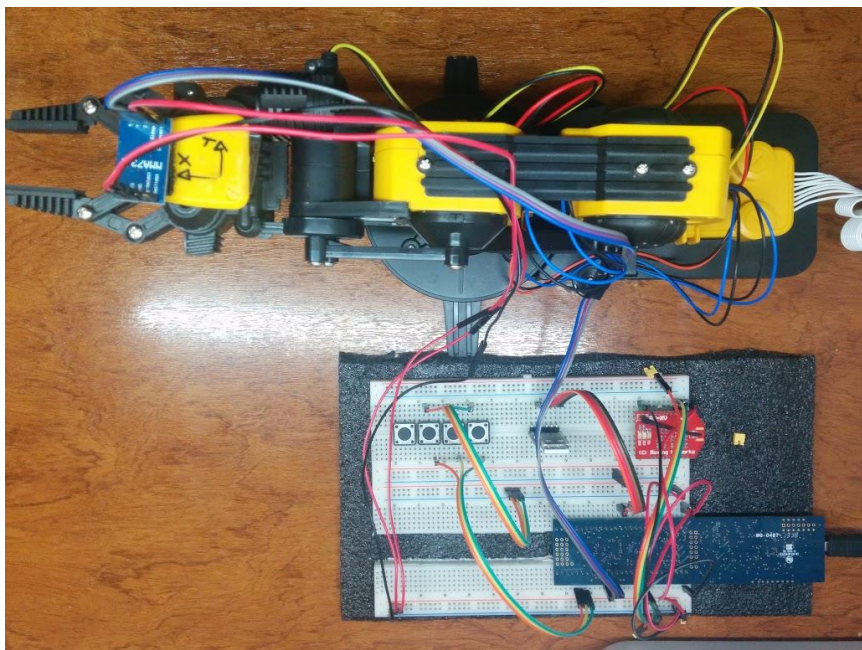
Esta parte del desarrollo se ha basado en cientos de pruebas analizando los movimientos que podemos realizar con los dedos teniendo el acelerómetro colocado en uno de ellos y cómo podemos transformar estos en la información que necesitamos para mover un puntero de un ratón.

Tras muchas pruebas, he notado que era muy necesario cambiar la metodología de las pruebas ya que para cada prueba había que colocarse el acelerómetro en la mano, evitar el “mal pulso” generar un movimiento y analizar los resultados obtenidos.

El objetivo era reconocer patrones de valores que nos indicaran un movimiento, pero esto es un problema si lo hacemos de primeras con un caso real (una mano).

Para conseguir acelerar el proceso de pruebas y poder repetir aproximadamente los mismos movimientos y poder estudiarlos he utilizado un brazo robótico que simula los movimientos de un dedo.

Gracia a él, he podido generar movimientos “limpios” que estudiar y llegar a interesantes conclusiones. Sobre el he montado el acelerómetro quedando el montaje final:

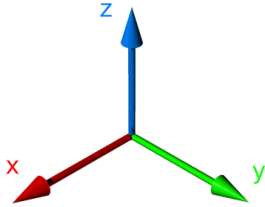


**Ilustración 28. Montaje brazo robótico.**

Analizando las necesidades del proyecto, necesito saber cuándo iniciar un movimiento en 4 direcciones posibles de un puntero de un ratón:

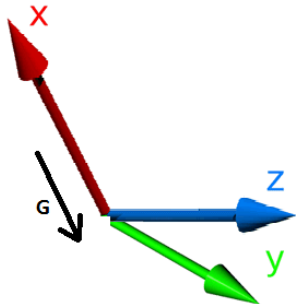
- Arriba
- Abajo
- Izquierda
- Derecha.

Sin embargo, el acelerómetro dispone de 3 ejes



**Ilustración 29. Ejes acelerómetro.**

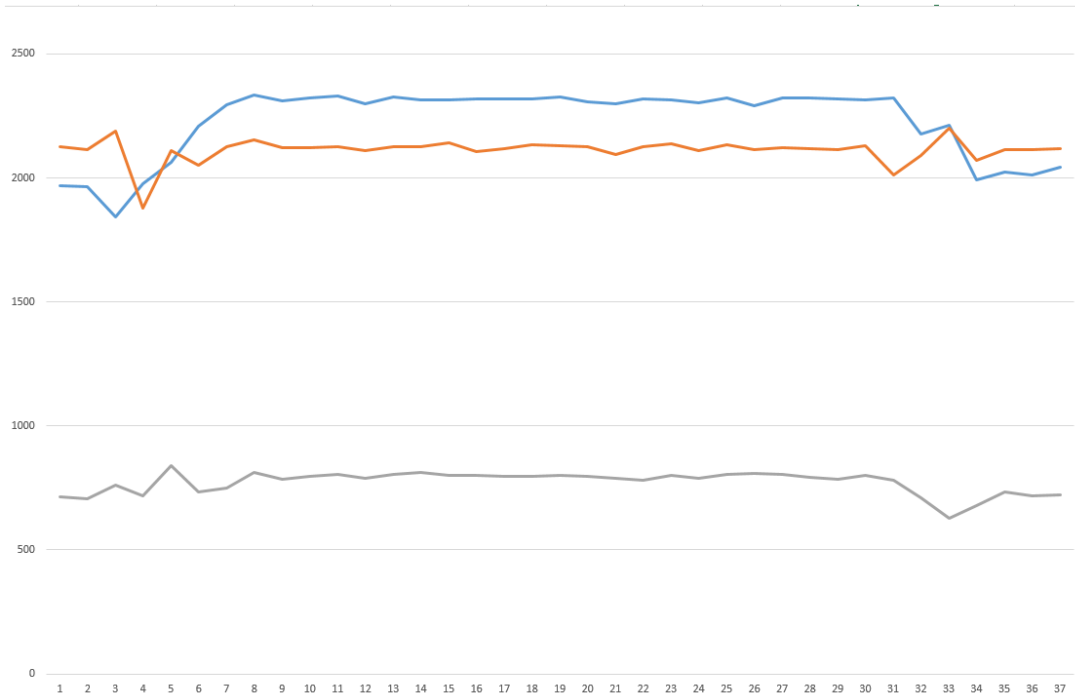
Al elevar el acelerómetro el eje X sufre una aceleración constante debida a la fuerza de la gravedad ya que al elevarlo, el eje X se aproxima a lo que sería el eje Z estando en reposo.



**Ilustración 30. Movimiento hacia arriba.**

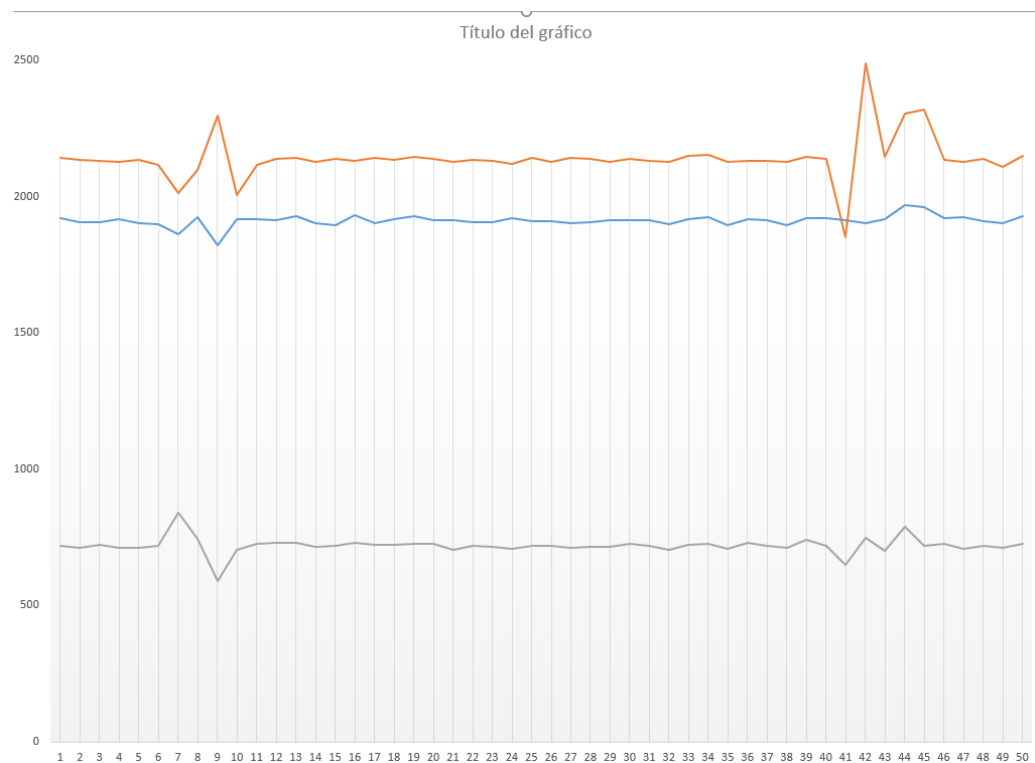
Por tanto, la detección del movimiento consiste en captar el patrón de comienzo del movimientos en el que se inicia un aumento en la aceleración del eje X que posteriormente se mantiene constante mientras se mantenga la inclinación y que se detendrá si se vuelve a la posición inicial.

En la siguiente imagen se muestra el patrón detectado de un movimiento hacia arriba durante un tiempo y luego la vuelta al estado de reposo (línea azul, Eje X).



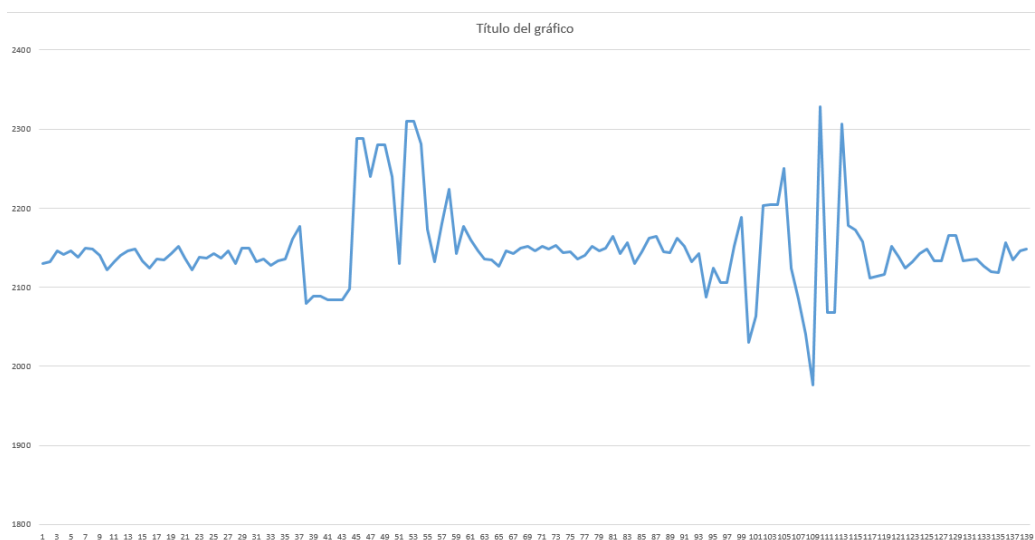
**Ilustración 31. Patrón Eje X.**

Sin embargo, los movimientos al desplazar el acelerómetro de izquierda a derecha provocan un resultado muy diferente ya que no se produce ninguna aceleración constante en el eje Y sino una variación esporádica como la de la siguiente gráfica (línea naranja Eje Y):



**Ilustración 32. Patrón Eje Y.**

Otro ejemplo más en detalle:



**Ilustración 33. Patrón EjeY mayor detalle.**

Como se puede apreciar, la azul en este último diagrama corresponde a el eje Y, y este ha recibido 2 movimientos en sentidos opuestos casi idénticos. Sin embargo, la forma de la curva de los datos cambia constantemente, en ocasiones y el número de casos diferentes en los que se origina el movimiento es enorme. He probado distintos algoritmos y no he conseguido acertar en más de un 30% de los casos y en la mayoría he tenido falsos positivos.

A este problema hay que añadir el problema de conversos A/D que si produce ruido durante el movimiento en algunas ocasiones (0,01%) no conseguimos filtrarla si se produce a la vez que un movimiento, esto complica tremendamente la detección del ruido y mucho más la detección del patrón de movimiento.

### **SOLUCION al problema de la detección de movimiento latera izquierda y derecha (eje Y):**

Tras muchos quebraderos de cabeza y horas mejorando los algoritmos y la creación de patrones de movimiento he pensado que hay una solución mucho más eficaz, desde el punto de vista del desarrollo.

Si cambiamos el movimiento que produce el desplazamiento del puntero (IZQUIERDA DERECHA) en vez de moviendo de izquierda a derecha, usaríamos un movimientos de **ROTAION** hacia la izquierda o la derecha de la mano.



**Ilustración 34. Solución detección movimiento lateral.**

Con este movimiento se consigue una aceleración constante en el eje Y según se rote hacia un lado o el otro el acelerómetro, lo que significa girar la mano (como un volante) en vez de desplazarla de izquierda a derecha.

Utilizando este movimiento el patrón de reconocimiento es muy parecido al del eje X y por tanto los esfuerzos en afinación de la precisión se retroalimentan en ambos ejes.

Por último, el eje Z se ha descartado en principio para el reconocimiento de los patrones de movimiento. Es cierto que podría utilizarse la información de este eje para complementar la de los otros dos y aumentar la precisión, pero por ahora parece que es suficiente con la información de dos ejes.

En el siguiente diagrama (siguiente página) expongo el algoritmo implementado para la captación del movimiento.

Este algoritmo se basa en:

- La captura de los valores del acelerómetro.
- Aplicación de un algoritmo de filtrado de ruido
- Cálculo de variaciones de los 3 últimos valores obtenidos por el acelerómetro.
- Algoritmo de cálculo de inicio de un movimiento y su dirección.
- En función del movimiento encontrado enviar el comando apropiado al dispositivo remoto.



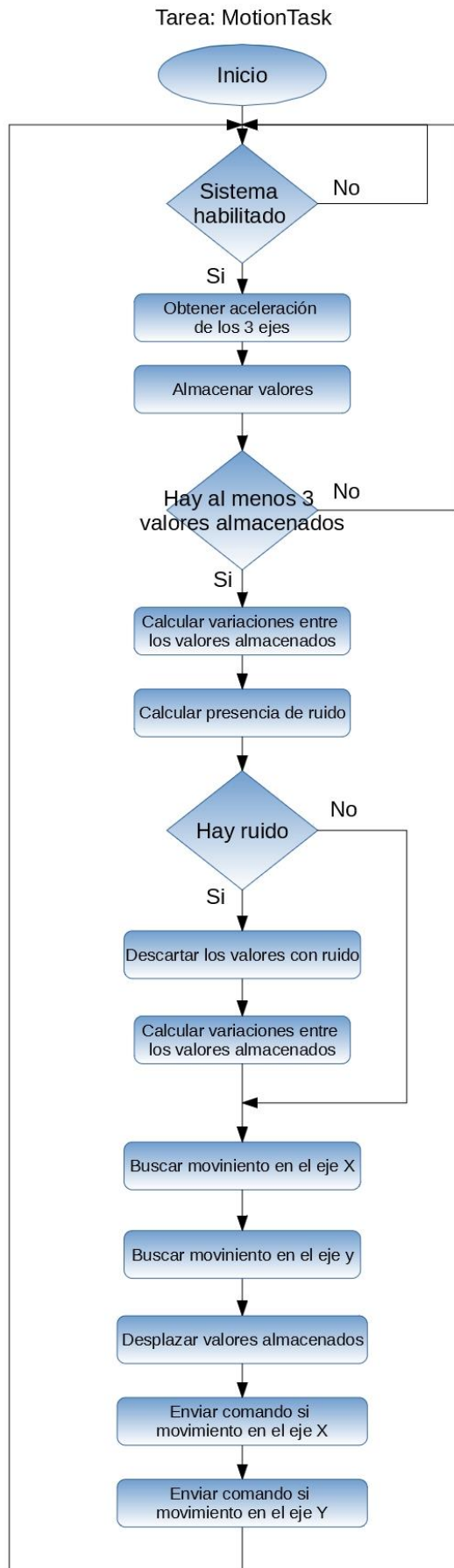
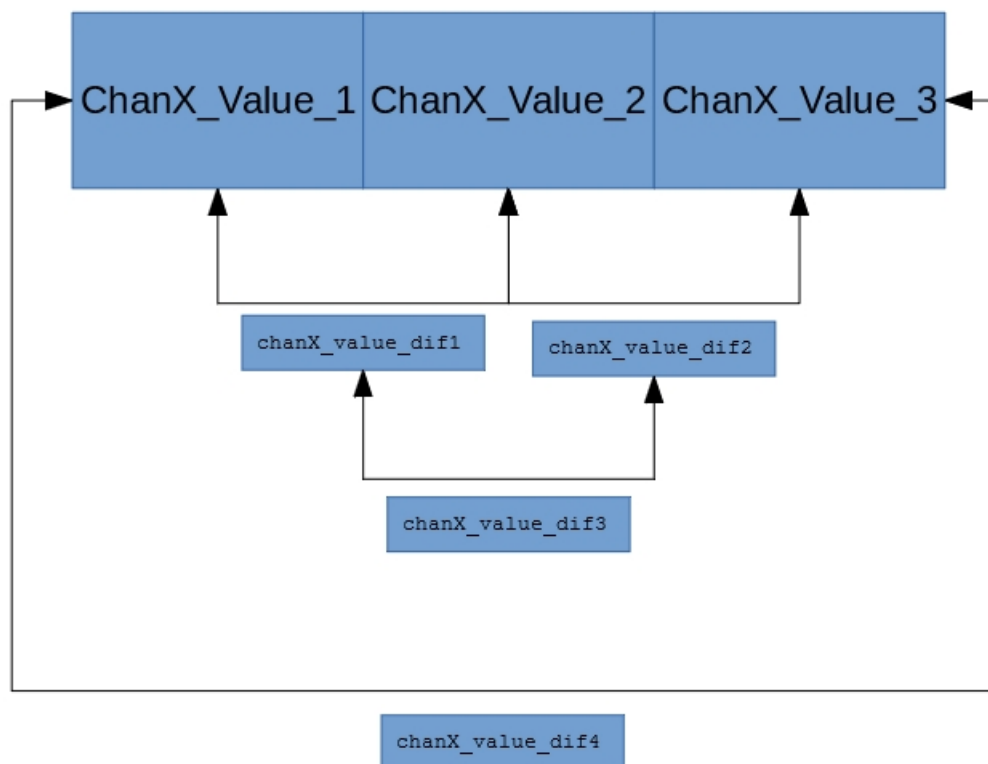


Ilustración 35. Algoritmo para la captación del movimiento.

Para el filtrado del ruido (que se verá en el siguiente apartado) y el cálculo del movimiento (punto actual), se guardan los 3 últimos valores recibidos del acelerómetro (chanX\_Value\_X) y se calculan las diferencias entre los valores obtenidos (chanX\_value\_difX).

En este esquema se indican todos los valores calculados con las diferencias entre los valores almacenados:



**Ilustración 36. Valores calculados.**

Para averiguar si hay un movimiento, en el inicio del sistema (como ya se ha explicado con anterioridad) hay una tarea para establecer cuáles son los valores en reposo de cada uno de los tres ejes.

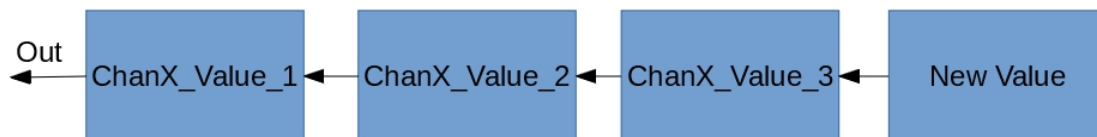
Se determina que hay movimiento en una dirección si el valor recibido por el acelerómetro no ha sido descartado por el filtro y excede, en cuatro posibles niveles, los valores en reposo.

Estos niveles son los siguientes:

- Movimiento ligero: el valor excede en 200 unidades el valor en reposo.
- Movimiento moderado: el valor excede en 400 unidades el valor en reposo.
- Movimiento grande: el valor excede en 600 unidades el valor en reposo.
- Movimiento muy grande: el valor excede en 800 unidades el valor en reposo.

En función del nivel del movimiento y de la dirección (estos valores pueden ser negativos), se enviará un comando diferente al equipo remoto para que ejecute un movimiento en esa dirección más fuerte o más suave. Los comandos se listarán en el apartado con la descripción del módulo de comunicaciones.

Finalmente, después de cada cálculo, los valores almacenados rotan para dar cabida a uno nuevo e iniciar el algoritmo de nuevo.



**Ilustración 37. Algoritmo nuevo tras rotación de valores almacenados.**

### 3.1.5 Filtrado de Ruido

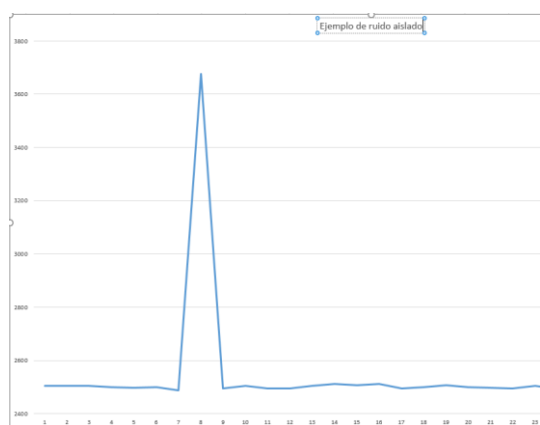
Para solucionar los ruidos generados por el ADC, se ha estudiado los valores del ruido llegando a la conclusión que en la gran mayoría de los casos tienen una forma en la gráfica muy parecida. En el ejemplo de valores siguiente se ve como el "pico" de ruido aparece entre dos valores **normales**

X	Y	Z
1890	2107	2596
1905	2103	2588
<b>1893</b>	2111	2601
<b>4095</b>	2127	2584
<b>1890</b>	2101	2596
1896	2111	2593
1885	2110	2576

**Ilustración 38. Valores pico de ruido.**

Para la eliminación de este punto, se analizan 3 puntos consecutivos de las medidas captadas de forma que, si el punto intermedio varía en gran medida con los contiguos y los contiguos son muy cercanos, eliminamos este punto aislado.

Por suerte, este patrón se repite en el 99.9% de los casos.



**Ilustración 39. Pico de ruido.**

Por tanto, la manera más eficiente y sencilla para detectar que un valor es “ruido” es aplicar el siguiente algoritmo y si se cumple entonces hay detección de ruido:

```
chanX_value_dif1 > 50  &&  chanX_value_dif2 > 50  &&  chanX_value_dif3 < 50
```

#### Ilustración 40. Detección pico de ruido.

Como se puede ver esta regla define la figura del “ruido” de la Figura anterior XXX.

Con este sencillo algoritmo aplicado a cada canal del acelerómetro nos libramos de la gran mayoría de valores erróneos debidos al ruido.

### 3.2 Software para sistema remoto (Windows):

Las comunicaciones inalámbricas como ya se mostró en el apartado de test, por parte del dispositivo empotrado LPEXpresso + módulo Wifly, las comunicaciones funcionan perfectamente. Únicamente tras configurarlo apropiadamente solo hemos de escribir sobre el puerto de comunicaciones del módulo Wifly e indicarle lo que queremos enviar a nuestro dispositivo remoto.

La escritura en el puerto es segura y las dos tareas “captación de clic de botones” y “captación de movimientos” envían comandos distintos haciendo uso de las comunicaciones con el módulo WIFly sin problemas llamando a la función de escritura en la UART del módulo Wifly que gestiona el acceso a la sección crítica con un semáforo.

Por parte del dispositivo receptor de las comunicaciones, se ha desarrollado un pequeño programa de consola en C para Windows que abre un socket UDP y escucha los comandos enviados por PickMouse.

Estos comandos se componen de un solo carácter ASCII e indican en qué dirección y cuando ha de desplazar el puntero del ratón. Además, hay comandos para indicar los clics de los botones del mouse.

Estos son los comandos enviados:

- A: Subir 5 puntos el puntero del ratón.
- B: Subir 10 puntos el puntero del ratón.
- C: Subir 20 puntos el puntero del ratón.
- D: Subir 40 puntos el puntero del ratón.

- F: Bajar 5 puntos el puntero del ratón.
- G: Bajar 10 puntos el puntero del ratón.
- H: Bajar 20 puntos el puntero del ratón.
- I: Bajar 40 puntos el puntero del ratón.
  
- K: Mover 5 puntos el puntero del ratón a la izquierda.
- L: Mover 10 puntos el puntero del ratón a la izquierda.
- M: Mover 20 puntos el puntero del ratón a la izquierda.
- N: Mover 40 puntos el puntero del ratón a la izquierda.
  
- P: Mover 5 puntos el puntero del ratón a la derecha.
- Q: Mover 10 puntos el puntero del ratón a la derecha.
- R: Mover 20 puntos el puntero del ratón a la derecha.
- S: Mover 40 puntos el puntero del ratón a la derecha.
  
- X: Ejecutar un clic de botón izquierdo del ratón.
- Y: Ejecutar un clic de botón derecho del ratón.
- Z: Ejecutar una presión o soltar el botón izquierdo del ratón. (se utiliza para hacer clic sin soltar y luego poder soltar).

Existen 4 comandos distintos para cada dirección para permitir cierta suavidad en el movimiento del puntero, de forma que, por ejemplo, con forme más elevemos el dedo, con mayor rapidez se moverá el puntero hacia arriba, y lo mismo para cada dirección.

En cuanto al código del programa para Windows se compone de las siguientes funciones:

- `_tmain`: En el que se inicializan las comunicaciones por UDP y mediante un bucle `while` se queda escuchando un socket abierto, y en función del comando recibido ejecuta la acción recibida.
- `LeftClick`: Que ejecuta un evento de clic del botón izquierdo del ratón.
- `LeftHalfClic`: Que ejecutara una presión (sin soltar), o soltar el botón izquierdo del ratón, en función de si se ha presionado antes el botón izquierdo del ratón o no.
- `RightClick`: Que ejecuta un evento de clic del botón derecho del ratón.

```

C:\Users\Magneto\Desktop\PICKMOUSE.exe
Initialising Winsock...
Initialised.
Socket created.
Bind done

Waiting for data...
Data: U
UP!
Data: D
DOWN!
Data: R
Right!
Data: L
Left!

```

Ilustración 41. Funciones.

## 4. Valoración económica del proyecto:

En la siguiente valoración económica se ha tenido en cuenta que el guante utilizado para las pruebas solo ha costado 1€, sin embargo, el guante final debería ser de mayor calidad y además con algunos refuerzos para contener la electrónica, por esto se ha presupuestado un precio unitario por guante de 10€ teniendo en cuenta que se harían pedidos grandes.

No se ha incluido en esta valoración el convertidor UART-USB debido a que se trata de un dispositivo inalámbrico y aunque se ha utilizado para depurar el código no estará en el dispositivo final.

Se ha añadido el presupuesto de una pequeña batería para dotarle de una autonomía aceptable 1300mA, sin embargo, no se ha hecho un estudio energético del dispositivo y queda pendiente una mejor elección de la batería a instalar.

El precio de un solo dispositivo sería de:

Material	Precio Unitario	Cantidad	Precio total
LPC1769	20 €	1	20
Wifly RN-XV	32	1	32
Adaptador UART-USB	4	1	4
Acelerómetro MMA 7361	9,5	1	9,5
Interruptor	1	4	4
Guante	10	1	10
Horas de trabajo	12	300	3600
Bateria	10	1	10
			3689,5

Ilustración 42. Estudio económico.

Sin embargo, si repartimos el coste de desarrollo entre el total de volumen de ventas el precio del dispositivo se verá reducido.

Estimando unas ventas de 500.000 unidades debido al gran furor inicial que va a suponer en el mercado en los primeros meses el precio del dispositivo quedaría de la siguiente manera:

Precio del dispositivo sin contar las horas de ingeniería: 89,5 €

El precio de todos los dispositivos sin el coste de desarrollo.  
 $89,5 * 500.000 = 44.750.000$  €

Sumándoles el coste de desarrollo.  
 $44750000 + 3600 = 44753600$  €

Precio final por unidad.  
 $44753600 / 500.000 = 89,5072$  €

Parece por tanto un proyecto viable a partir de la venta de 500.000 unidades, al cual se le puede añadir un margen interesante de beneficios.

## 5. Conclusiones

Conclusiones derivadas del estudio de la captura de movimientos con la mano:

- Para este tipo de captura de movimientos con acelerómetro sería necesario utilizar un giroscopio para el eje Y de forma que no hiciese falta girar la mano para mover el puntero de izquierda a derecha.
- La captura de movimientos es muy complicada ya que intervienen factores como el pulso o movimientos involuntarios.
- Además, he descubierto el efecto de que el usuario crea tener el dedo apuntando a un lado, pero si no se fija en su dedo en ocasiones el dedo no apunta donde el usuario cree que está apuntando.

Lecciones aprendidas:

- La gestión de dispositivos hardware a bajo nivel implica un intenso estudio de la documentación de cada dispositivo y de sus manuales.
- Existe una gran comunidad en internet con foros en los que se puede sacar información fundamental para los problemas encontrados con el desarrollo de sistemas embebidos.

- La utilización de sistemas operativos en tiempo real es muy útil y de fácil utilización si se realiza un estudio correcto de las necesidades del proyecto.
- Lidar con problemas de hardware como me ha sucedido con el AD puede llegar a tener un gran impacto en el proyecto y en las fechas de entrega, ya que son problemas difíciles de detectar y los últimos en los que se busca el problema.

#### Cumplimiento de objetivos:

- Creo que he alcanzado todos los objetivos planteados, aunque algunos de ellos han cambiado de forma, como por ejemplo los movimientos de izquierda y derecha no se ejecutan moviendo el dedo de izquierda a derecha sino rotando la mano hacia la izquierda o derecha.
- La sensibilidad de los movimientos del ratón es muy mejorable pero posible si se continúa evolucionando esa parte del desarrollo.
- Las comunicaciones inalámbricas son muy sencillas y podrían reforzarse incluyendo la identificación del dispositivo y seguridad en las comunicaciones.
- La planificación del proyecto ha sido adecuada, sin embargo, los problemas encontrados con el conversor analógico digital provocaron algunas demoras en las entregas.

#### Futuras mejoras para el proyecto:

- Quedan algunas posibilidades de mejora como ya he comentado:
  - o Mejora en la seguridad de las comunicaciones.
  - o Identificación de equipos en las comunicaciones.
  - o Configuración de la red Wifi.
  - o Mejorar la suavidad y velocidad de los movimientos del puntero del ratón.



## 6. Glosario

**Acelerómetro:** cualquier instrumento destinado a medir aceleraciones en distintos ejes y que en caída gravitacional libre hacia el centro de la Tierra medirá un valor de cero, ya que, a pesar de que su velocidad es cada vez mayor, está en un marco de referencia en el que no tiene peso.

**ARM** Arquitectura RISC (Reduced Instruction Set Computer) de 32 bits desarrollado por ARM Holdings.

**GPIO** General Purpose Input/Output, es un pin genérico de entrada/salida de propósito general.

**FreeRTOS** Free Real Time Operating System, es un sistema operativo en tiempo real para dispositivos embebidos.

**LPC1769** Micro-controlador Cortex-M3 que puede ser usado para aplicaciones en dispositivos empotrados con un alto nivel de integración y un consumo de energía bajo.

**Periférico** Dispositivo externo al procesador, que produce entrada y salida de información.

**Sistema operativo en tiempo real** Sistema operativo diseñado especialmente para sistemas en tiempo real. Se utilizan en los sistemas que hay que cumplir unas especificaciones temporales determinadas. Sigla **RTOS**

**Scheduller o planificador** Elemento del núcleo del sistema operativo responsable de decidir que tarea ha de utilizar el procesador en un momento determinado.

**UART** Universal Asynchronous Receiver/Transmitter, es un chip que se encarga de manejar las interrupciones de los dispositivos conectados al puerto serie, convertidos a datos en formato paralelo y transmitirlos al bus del sistema.

**Tarea** Cualquier calculo individual, conjunto de cálculos, la lógica de la toma de decisiones, intercambio de información o combinación de ellas que ha de llevar a término en tiempo de ejecución el planificador. Es la abstracción de los RTOS. Las tareas comparten un espacio de memoria común.

**Wi-Fi** Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

**WiFiRN171** Componente que capacita a sistemas empotrados a conectarse a redes Wireless compatibles con 802.11b/g.

**Wireless** Transferencia de datos donde no se utilizan cables para conectar diferentes ordenadores.

## 7. Bibliografía

**NXP.** LPCWARE (Software & Support for NXP MCUs). Submitted by lpcxpresso-support on Mon, 2013-10-21. Disponible en: <http://www.lpcware.com/lpcxpresso>

**LPC1769 Configurator:** Configurador automatico de registros para la placa LPCEXPRESO1769. Disponible en: [http://nanohome.be/nxp/LPC1769\\_Pins.php](http://nanohome.be/nxp/LPC1769_Pins.php)

**Sistemas empotrados:** Documentación de la asignatura. [http://materials.cv.uoc.edu/continguts/PID\\_00158470/index.html](http://materials.cv.uoc.edu/continguts/PID_00158470/index.html)

**UOC.** IniciCortexM3. Administrator, XWiki.org, 2013-09-07. Disponible en: <http://cv.uoc.edu/webapps/xwiki/wiki/matembeddedsystems/home/view/Material/IniciCortexM3>

**Fundación Wikipedia.** Wikipedia, la enciclopedia libre. GNU Free Documentation License, 2013. Disponible en: <http://es.wikipedia.org/>

**Embedded Artists.** The art of Embedded Systems Development. Embedded Artists AB, 2013. Disponible en: [http://www.embeddedartists.com/products/lpcxpresso/lpc1769\\_xpr.php](http://www.embeddedartists.com/products/lpcxpresso/lpc1769_xpr.php)

**NXP Semiconductors.** LPC1769/68/67/66/65/64/63. NXP B.V. 2013. Disponible en: [http://www.nxp.com/documents/data\\_sheet/LPC1769\\_68\\_67\\_66\\_65\\_64\\_63.pdf](http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf)

**Freescale semiconductor.** Micromachined Accelerometer (MMA7361L). 2008. Disponible en: [http://www.freescale.com/files/sensors/doc/data\\_sheet/MMA7361L.pdf](http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf)

**WiFi Command Reference,** Advanced Features & Applications User's Guide: Disponible en: <http://cdn.sparkfun.com/datasheets/Wireless/WiFi/rn-wiflycr-ug-v1.2r.pdf>

**FreeRTOS documentación:** Disponible en: <http://www.freertos.org/index.html>