

# Proyecto: Guante-Mouse

Fosco, Camilo  
Sampayo, Sebastián  
Schiffmacher, Christian

3° de Diciembre de 2013

Laboratorio de Microcomputadoras

---

## Índice

<b>1. Objetivos del Proyecto</b>	<b>3</b>
<b>2. Descripción del Proyecto</b>	<b>3</b>
2.1. Detección del movimiento . . . . .	3
2.2. Conversión de movimiento a datos . . . . .	4
2.3. Procesamiento de los datos . . . . .	5
2.4. Comunicación con la PC . . . . .	6
<b>3. Diagrama en Bloques (hardware)</b>	<b>11</b>
<b>4. Diagrama de Flujo (firmware)</b>	<b>11</b>
<b>5. Circuito Esquemático</b>	<b>19</b>
<b>6. Listado de Componentes y Costos</b>	<b>20</b>
<b>7. Resultados</b>	<b>20</b>
<b>8. Conclusiones</b>	<b>21</b>
<b>9. Apéndices</b>	<b>22</b>
9.1. Filtro pasa-bajos . . . . .	22
9.2. Selección de Velocidad . . . . .	23

## 1. Objetivos del Proyecto

El proyecto busca ofrecer una mayor comodidad a la hora de manejar la computadora. Con el Guante Mouse, el usuario puede controlar el cursor de su PC de forma intuitiva y agradable, con simples movimientos de la mano. Además, representa una forma nueva y más interactiva de usar la computadora, y permite evitar problemas médicos generados por el uso de un mouse común como el síndrome del túnel carpiano.

## 2. Descripción del Proyecto

El Guante Mouse está compuesto por un circuito montado sobre un guante que detecta la inclinación de la mano del usuario y procesa la información para mover el cursor de la PC de forma análoga, en tiempo real. Los movimientos del dorso de la mano son captados por un acelerómetro y procesados por un 8052, que transmite los datos a la computadora de forma cableada, a través de un circuito preexistente de mouse con *trackball*.

El guante presenta 2 pulsadores que cumplen la función de los 2 botones de un mouse común (izquierdo, derecho), y permite activarse y desactivarse juntando o separando respectivamente el dedo indice y mayor.

Se describe a continuación el funcionamiento del proyecto parte por parte.

1. Detección del movimiento
2. Conversión de movimiento físico a dato digital
3. Procesamiento de los datos
4. Comunicación con la PC

### 2.1. Detección del movimiento

Con el objetivo de medir el movimiento de la mano se investigaron los diferentes módulos acelerómetros de 3 ejes que ofrece el mercado en la actualidad y que se encuentran en un rango de precios razonables para el proyecto. Básicamente se encontraron 2 tipos:

1. Con salida digital
2. Con salida analógica

Para el primer caso hubiera sido necesario estudiar el protocolo de comunicación utilizado por el fabricante. Esto traería como consecuencia adaptar el software para poder hacer las pruebas de verificación para ver si lo que se está interpretando en el receptor es efectivamente lo que se quiso trasmisir desde el módulo acelerómetro.

Por otro lado, en el segundo caso se tiene una salida analógica para cada eje, por lo tanto utilizando un conversor analógico/digital básico se puede obtener con seguridad y relativa precisión y velocidad, el valor de la aceleración medida.

Por estas razones se optó por el módulo “*MMA7361L 3-Axis Accelerometer ±1.5/6g with Voltage Regulator*” el cual posee el integrado *MMA7361L*, un regulador de tensión de 3.3V para alimentarlo, y una serie de capacitores y resistencias necesarios para el correcto funcionamiento de este chip según la hoja de datos del fabricante. Las características que posee este acelerómetro son:

- Dimensiones: 0.5" x 0.9" x 0.09" (1.2×2.3×0.23 cm) (sin contar los pines)
- Tensión de alimentación: 2.2-16 V
- Corriente de alimentación: 0.5 mA
- Formato de salida: 3 tensiones analógicas (una por cada eje) centradas a la mitad de la tensión del pin de 3.3V
- Rango de sensibilidad (seleccionable utilizando el pin g-Select):

- MMA7361L:  $\pm 1.5\text{g}$  (default) o  $\pm 6\text{g}$
- MMA7341L:  $\pm 3\text{g}$  (default) o  $\pm 11\text{g}$
- Peso sin los pines: 0.025 oz (0.7 g)
- *Sleep Mode*

Como se puede ver en las especificaciones existen dos modelos: el 7361 y el 7341. Utilizando una aplicación móvil de teléfono se realizaron unas primeras mediciones para tener una aproximación de en que rango se estaría trabajando. El resultado fue que nunca se sobrepasaría los 1.5g en ningún eje, por lo tanto se decidió comprar el de mayor sensibilidad ( $\pm 1.5\text{g}$ ).

## 2.2. Conversión de movimiento a datos

El sensor de movimiento (acelerómetro) entrega una tensión en su salida proporcional a la aceleración medida. Sin embargo, el microcontrolador solo puede interpretar valores digitales en sus entradas, i.e. 0V o 5V. Por lo tanto es necesario conectar en el medio algún tipo de conversor analógico/digital.

Para esta tarea se eligió el ADC0808, el cual posee un multiplexor a su entrada permitiendo que se utilice para varias entradas analógicas distintas(hasta 8) controladas por 3 pines, con una única salida digital de 8 bits. Además, la conexión entre este y el 8051 se puede hacer de forma directa, dado que el integrado está pensado para dichos fines (posee salidas TTL). Las especificaciones del conversor son las siguientes:

- Multiplexor de 8 entradas
- Alimentación de 5V
- Error de  $\pm 1/2 \text{ LSB}$
- Baja potencia: 15 mW
- Tiempo de conversión:  $100\mu\text{s}$
- Frecuencia de reloj: 10 - 1280 Hz
- Posibilidad de ajustar el rango de tensión de entrada para aprovechar los 256 valores posibles.

Si bien la frecuencia de reloj se indica hasta 1.3 Mhz, por comodidad se utilizó la salida *ALE* del 8051, la cual se encuentra a 2 Mhz con un cristal de 12Mhz, ya que funcionaba correctamente.

Por otro lado para el conexionado se “puentearon” los pines *START* y *ALE* como se indica en las aplicaciones típicas de la hoja de datos. Un pulso alto en esta entrada da comienzo a la conversión. Este nodo se conectó a la salida de un pin del microcontrolador para controlar el momento en que se requiera un dato.

Otra salida es el pin *EOC*, la cual indica que ha finalizado la conversión correctamente y ya se puede tomar la salida. Sin embargo, dado que la velocidad de conversión no era algo determinante en el proyecto (dado que el máximo de conversión es del orden de los microsegundos), se optó por esperar siempre un tiempo mayor al tiempo de conversión máximo, de aproximadamente  $130\mu\text{s}$ .

El algoritmo para tomar la medición desde el 8051 es básico:

```

Seleccionar entrada (setear controles del mux)
Enviar pulso en START
Esperar un tiempo de conversión
Almacenar medición en una variable

```

### 2.3. Procesamiento de los datos

Una vez almacenado un valor de aceleración entre 0 y 255 en una variable global se da comienzo al procesamiento de la señal.

Para esta tarea lo primero que se realizó fue una serie de mediciones con el acelerómetro imprimiendo los valores por pantalla (a través del puerto serie). Estos datos se levantaron con MATLAB y se graficaron para tener una idea visual de la señal obtenida. El resultado fue una señal ruidosa que no aprovechaba el rango máximo (0-255), los valores máximos de estas mediciones se encontraban entre 80 y 170 aproximadamente. Por esto se decidió ajustar mejor la referencia de tensión del conversor de modo de aprovechar el rango 0-255 y mejorar la precisión. Además se diseñó un filtro de tipo pasa-bajos para suavizar la señal, lo cual tendría repercusión directa en la *performance* del producto final (movimientos más suaves del puntero en la pantalla):

$$y(n) = y(n-1) \cdot (1 - \alpha) + x(n) \cdot \alpha$$

con  $\frac{1}{256} \leq \alpha \leq 1$  es proporcional a la frecuencia de corte del filtro (si  $\alpha = 1$  es un pasa-todo, ver apéndice).

A la hora de implementar el filtro en *Assembler* se consideró  $\alpha = N$  (ver apéndice) y se utilizó el siguiente algoritmo para optimizar el error de truncamiento:

$$y(n) = \frac{y(n-1) \cdot (N-1) + x(n)}{N}$$

De esta forma resultó imprescindible utilizar una rutina de división en 16 bits, dado que el numerador será un número mayor a 255 con alta probabilidad. Luego, como aritméticamente resulta imposible que el resultado exceda 255 (dados que  $y(n-1)$  y  $x(n)$  son de 8 bits), el resultado de la división se puede truncar al *byte* menos significativo con completa seguridad.

Los resultados del filtro implementado se pueden ver a continuación donde se graficó en MATLAB la medición del acelerómetro en azul, la medición filtrada con el 8051 en verde, y la medición filtrada con la función *filter* de MATLAB (mucho más precisa/teórica):

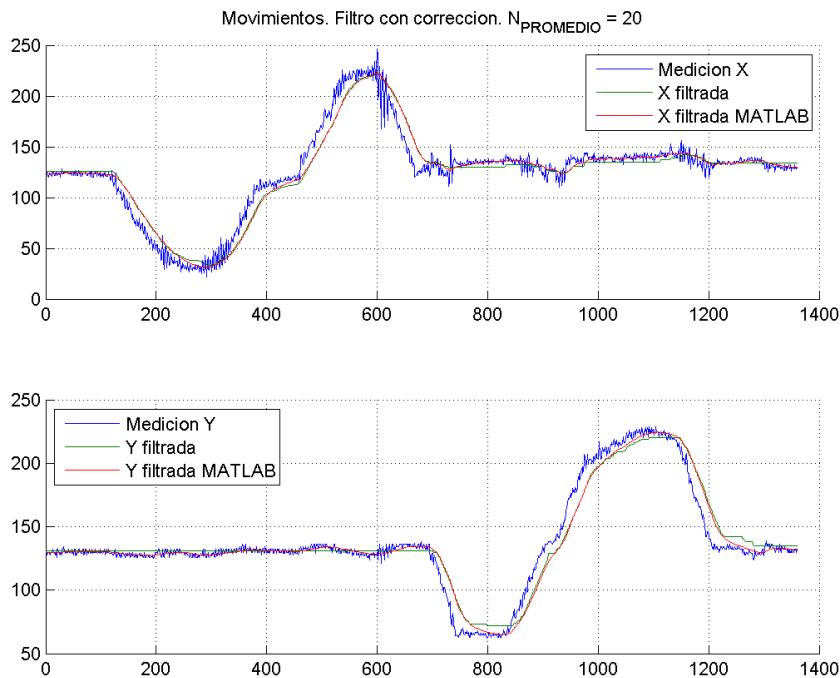


Figura 1: Resultados del filtro implementado

## 2.4. Comunicación con la PC

A la hora de definir la comunicación entre el 8052 y la PC, se barajaron diferentes opciones. Entre ellas, se terminó optando por la comunicación a través de un circuito existente de mouse, tanto por la mayor facilidad de código, como por el desafío que representaba estudiar de forma crítica un circuito existente y adecuarlo a las necesidades del proyecto.

Se abrió entonces el mouse *Logitech CPAM9948*, y se analizó su circuito interno, que se muestra en la figura siguiente:

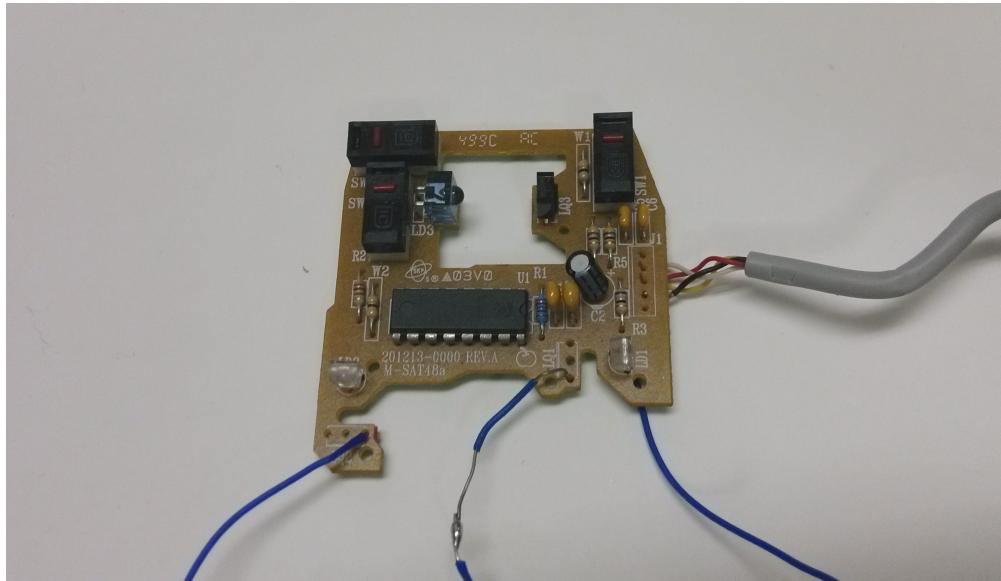


Figura 2: Circuito del Mouse

El mouse funciona gracias a dos pares receptor-emisor infrarrojos, uno para cada eje. Entre el emisor y el receptor se coloca una rueda con ranuras equiespaciadas. Se descubrió que al mover el mouse, se mueven dichas ruedas y provocan interrupciones en el haz del emisor que generan una señal activa en el receptor. Esta señal es captada por el microprocesador del mouse y, según su forma, hace que el cursor se mueva a la derecha o a la izquierda. Para descubrir la forma de dichas señales, se midió el punto de conexión entre el microprocesador del mouse y el fotoreceptor. De esa forma, se buscó determinar qué señal producía el receptor para cada tipo de movimiento, con el objetivo de reproducirla. Considerando que cada eje se comporta de la misma forma, se conectó el osciloscopio al punto de conexión entre el receptor y el microprocesador del mouse, y se movió el mouse en el eje x, midiendo la señal generada por el receptor con el osciloscopio. Se vió rápidamente que la señal difería según si el movimiento era hacia la izquierda o a la derecha. Se obtuvo lo siguiente:

Señal izquierda:

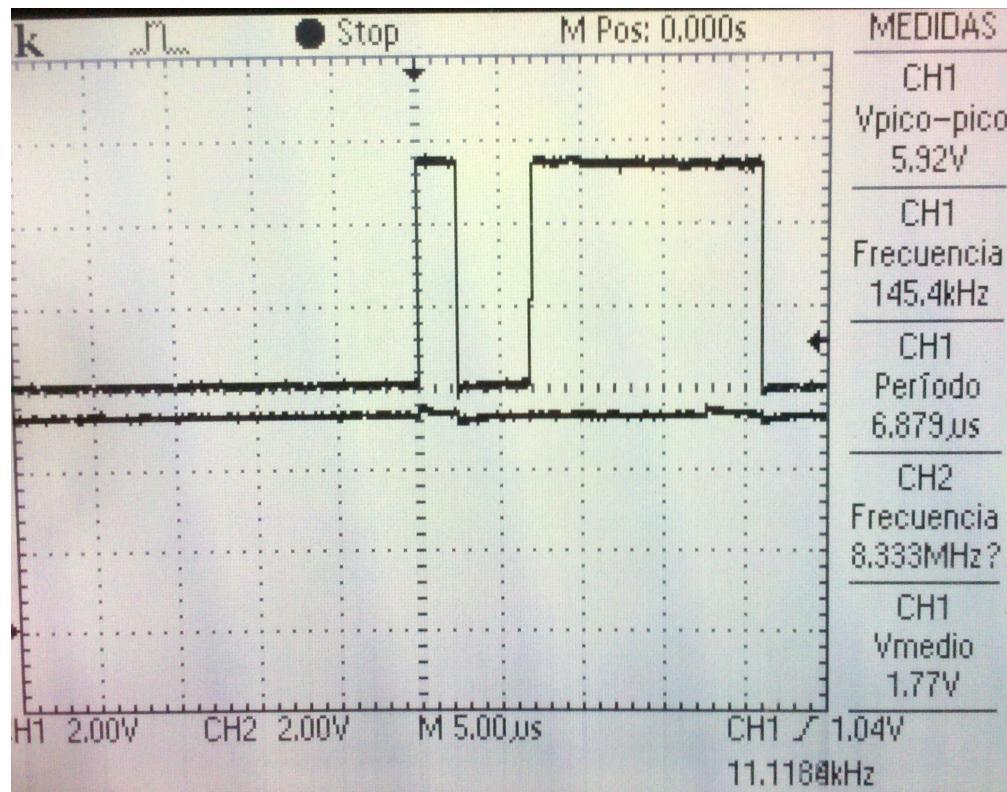
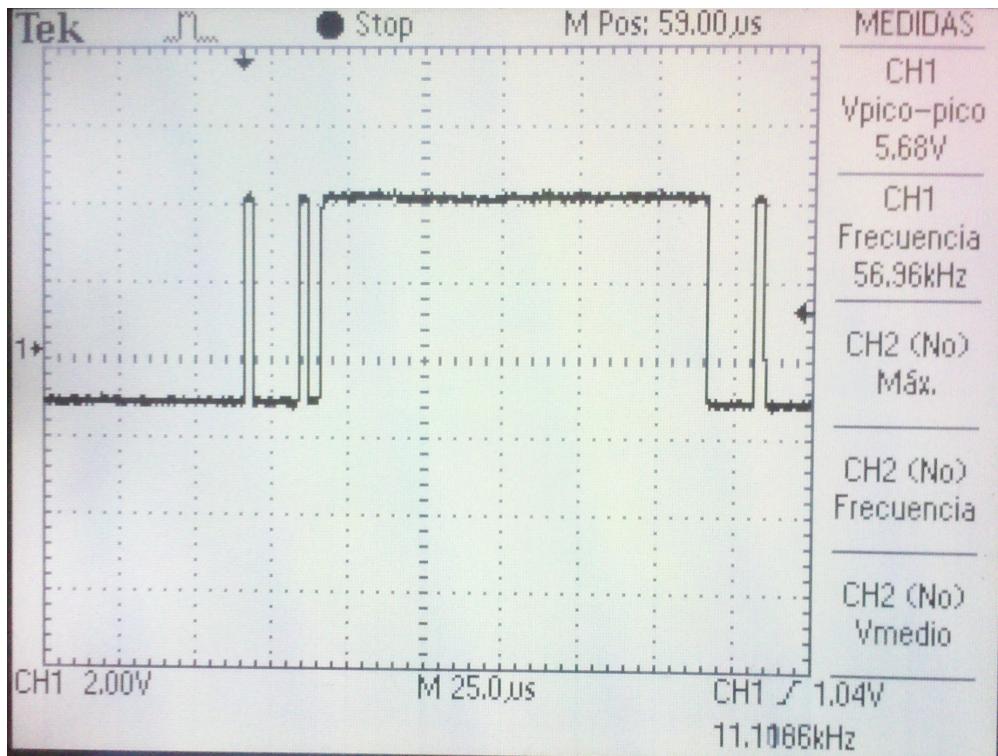


Figura 3: Circuito del Mouse

Señal derecha:



Se observa que son señales periódicas, y que las señales izquierda y derecha son modificaciones simples de la señal con mouse quieto (se agrega un pulso en cierto punto), manteniendo aproximadamente el período. Se buscó entonces reproducir estas señales con el 8051, y enviarlas directamente al microprocesador. Despues de varios intentos, se descubrió que la señal medida era una combinación de una señal patrón generada por el microprocesador del mouse con una señal que llamaremos activa, generada por el receptor al captar la interrupción del haz de luz. Se definió entonces lo siguiente:

Señal patrón = señal con mouse quieto:

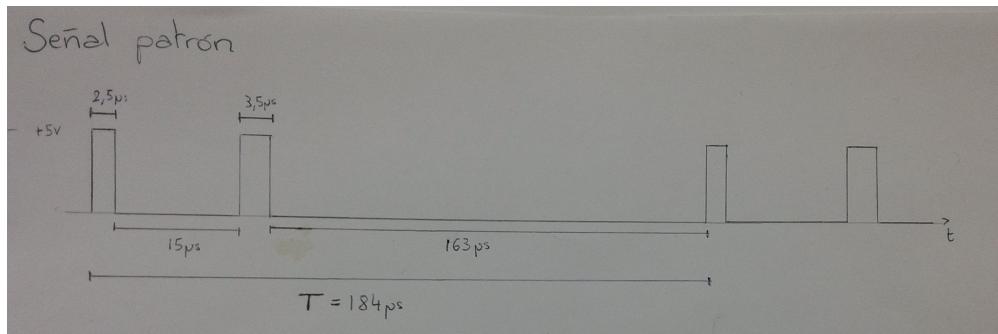


Figura 5: Circuito del Mouse

Señal activa izquierda:

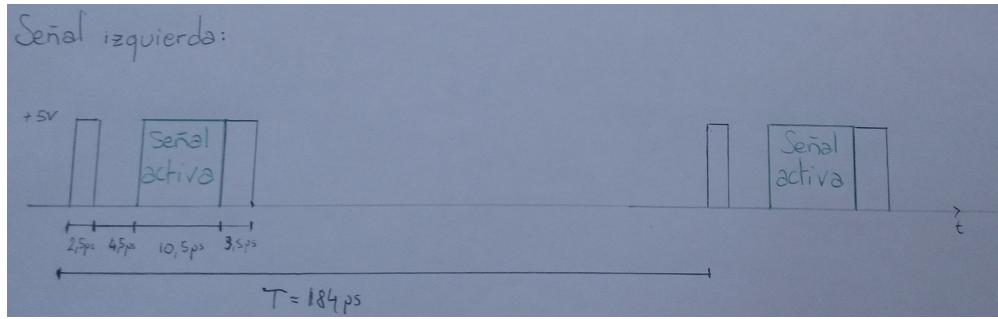


Figura 6: Circuito del Mouse

Señal activa derecha:

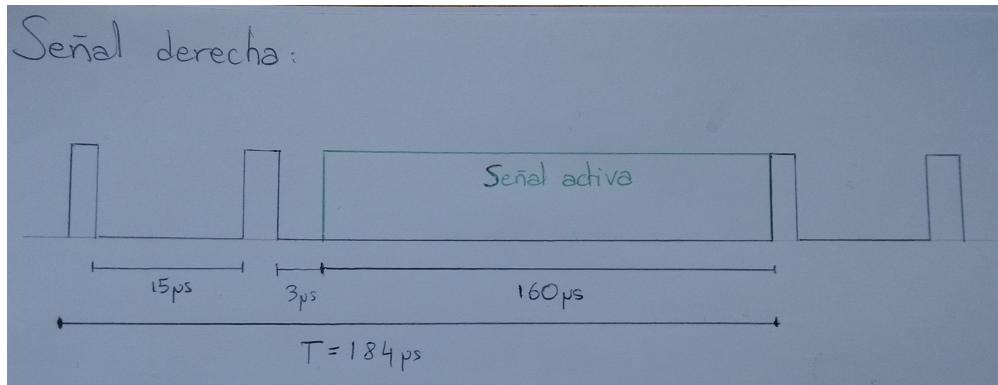


Figura 7: Circuito del Mouse

Donde la posición relativa de cada pulso de señal activa con respecto a la señal patrón es crucial para el correcto movimiento del cursor. Cabe destacar que cada señal activa aparece una vez por período de señal patrón, ya que debe ubicarse en un punto preciso del período. El microprocesador del mouse capta entonces la perturbación de su señal patrón, e interpreta esa perturbación detectando en qué posición, relativa a su señal, se generó esta nueva señal que se viene a superponer. A partir de eso, mueve el cursor hacia la izquierda o hacia la derecha. Cada señal activa detectada mueve el cursor una cierta cantidad finita de pixels (pequeña). Por lo tanto, mientras más señales activas detecte en un cierto intervalo de tiempo, más rápido se verá el movimiento del cursor.

Una vez entendido el funcionamiento del circuito del mouse, se procedió a crear el programa que emularía a los receptores generando las señales activas. Se programaron las señales activas vistas anteriormente, y simplemente colocándolas en la posición correcta con respecto a la señal patrón, se logró desplazar el cursor hacia ambos lados. Para lograr que la posición de las señales activas sea la correcta con respecto a la señal patrón, se diseñó un sistema de sincronización utilizando un pin más: se quiso conectar un nuevo pin del puerto 3 al punto de salida de la señal patrón, para leer esta señal y poder sincronizar el programa, pero la tecnología open collector del 8052 requiere que se imponga un 1 lógico (+5V) en un pin de lectura. Al tener esa tensión sobre el pin de señal patrón, el microprocesador del mouse detecta la continua de +5V como una perturbación constante de la señal patrón, y genera un movimiento aleatorio e incontrolable. Para evitar este problema, se colocó un buffer digital de tensión entre el pin del microprocesador del mouse (conectado a la entrada del bufer) y el pin de lectura del 8051 (conectado a la salida del bufer). De esa forma, los 5V del pin de lectura no se transmiten al pin del microprocesador, pero la señal patrón llega al pin de lectura. Cabe destacar que, al ser la señal patrón una señal con componentes de alta frecuencia (pulsos de  $2\mu s$ ), el buffer no transmite la señal de forma perfecta, sino que los pulsos cuadrados se deforman y quedan con forma casi triangular. Aun así, los pulsos deformados son captados como 1 lógico por el 8052, lo que permite que el algoritmo de sincronización funcione perfectamente.

Finalmente, la comunicación con la PC resulta ser de la siguiente forma: Se conecta un pin del puerto 3 (pin de sincronización) con el pin de señal patrón del microprocesador del mouse, mediante un buffer (para que no se superpongan los 5V de lectura sobre el pin de señal patrón). Se hace esto para los dos ejes. Se conecta otro pin del puerto 3 (pin de salida) directamente sobre el pin de señal patrón. Se mantiene este pin en cero, hasta que se quiera generar movimiento. También se hace para ambos ejes. El programa recorre un lazo infinito, que se puede ver de la siguiente forma:

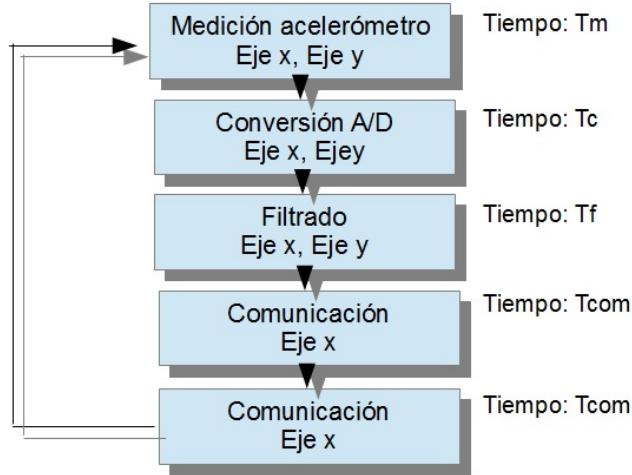


Figura 8:

Como se puede ver, durante un tiempo Tcom parametrizable, se generan señales activas que son transmitidas al mouse por el pin de salida. Ya que el cursor se mueve más rápido mientras más señales activas detecte en un intervalo fijo de tiempo, se optó por enviar una cantidad de señales activas aproximadamente proporcional al valor de aceleración medido. La relación exacta entre cantidad de señales activas enviadas y valor de aceleración está explicada en el Apéndice 2. Una vez determinada la cantidad de señales activas a enviar, se mandan las señales activas una atrás de la otra, y luego se espera hasta que se complete el tiempo Tcom. De esa forma, se logra que la velocidad del cursor sea aproximadamente proporcional (ver Apéndice 2) al valor de aceleración medido, es decir a la inclinación de la mano.

Se tiene que tomar en cuenta que el tiempo Tcom debe ser elegido con criterio: si es grande, permite enviar más cantidad de señales activas, por lo tanto se tienen más velocidades distintas de cursor, lo que otorga una mayor precisión. Sin embargo, al ser grande, se prolonga por más tiempo el movimiento en cada eje, lo que provoca que la sensación de fluidez se pierda cuando se desplaza el cursor en diagonal: se observa un movimiento escalonado. Se optó por un Tcom relativamente bajo, ya que se notó que la fluidez aumentaba drásticamente comparada con la de un valor de Tcom más alto, y la precisión seguía siendo aceptable. En el programa, el tiempo Tcom está determinado por la variable MAX\_ACCELERACION, ya que en cada etapa de comunicación con el circuito del mouse, se generan como máximo esta cantidad de períodos de señal activa, y siempre se tiene esta cantidad de períodos totales (señal activa + señal patrón para completar). El tiempo Tcom es entonces MAX\_ACCELERACION multiplicado por el periodo de las señales (tanto la señal activa como la señal patrón tienen el mismo periodo, ya que la primera se ubica siempre en el mismo punto del periodo de la segunda).

### 3. Diagrama en Bloques (hardware)

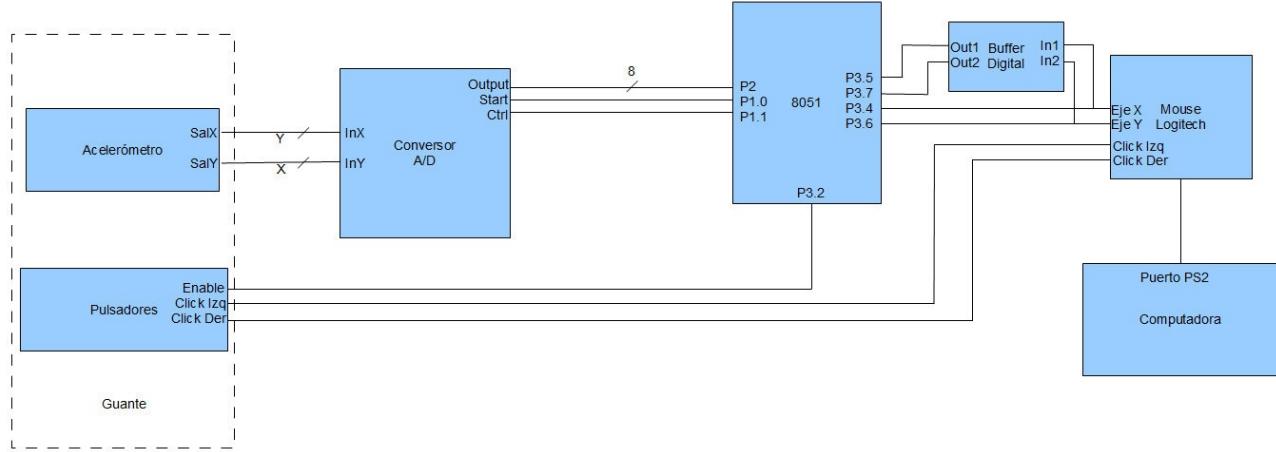


Figura 9: Diagrama en bloques

El proyecto, a nivel hardware, se puede dividir en 2 partes. Una parte móvil fijada al guante y una parte fija.

La parte móvil esta conformada por el acelerómetro que mide los movimientos de la mano y los pulsadores correspondientes al *click* derecho, al *click* izquierdo y al *enable* del 8051.

El acelerómetro mide los datos de los ejes X e Y y envía una señal analógica por cada eje al conversor A/D para ser convertida en una señal digital de 1byte cada una.

Esta señal es enviada al 8051 que procesa y analiza la señal siempre y cuando el pulsador de *enable* este oprimido.

Una vez hecho esto, el 8051 envía una señal al mouse para que este se comunique con el puerto PS2 de la computadora, lo que hace que se mueva el puntero.

Para enviar la señal correcta, se utilizó un buffer de tensión que sirve para sincronizar la señal estática del mouse con la señal enviada por el microprocesador.

Los pulsadores correspondientes a los *clicks* derecho e izquierdo están conectados directamente al mouse.

### 4. Diagrama de Flujo (firmware)

A continuación se muestran los diagramas de flujo del programa.

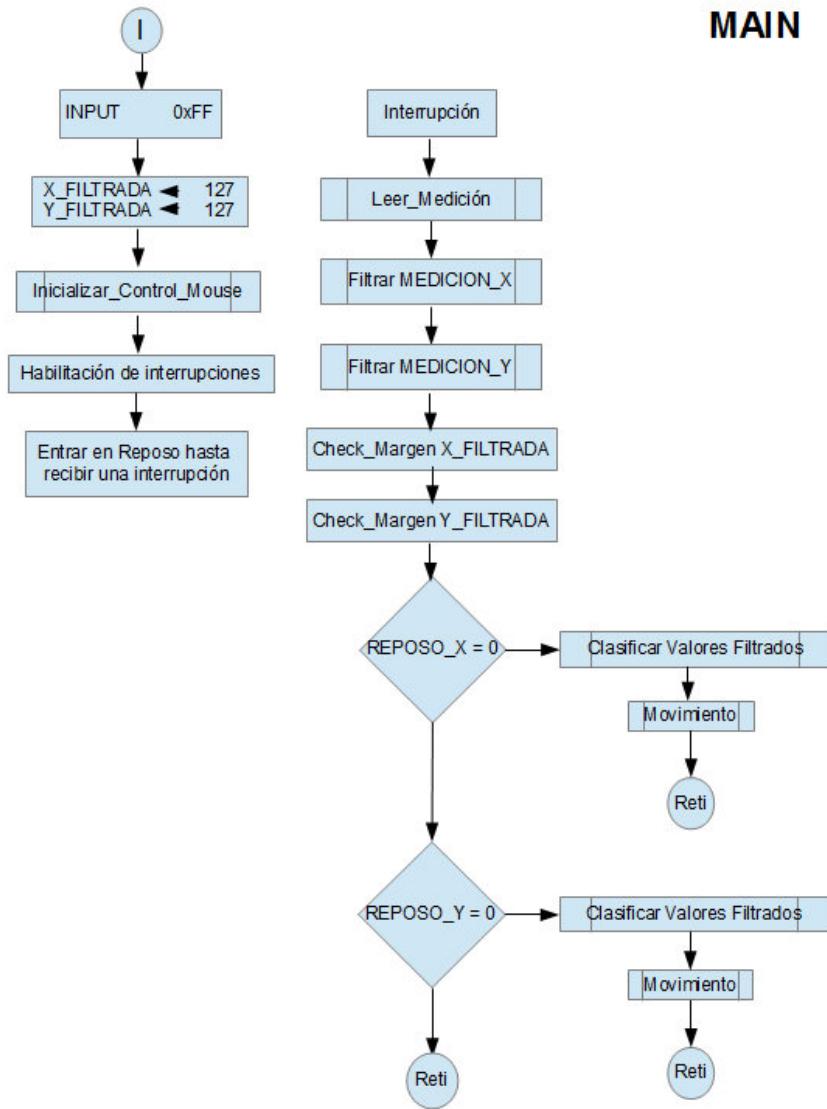


Figura 10: Diagrama de flujo del main

Dentro de la interrupción se lee la medición de la siguiente forma:

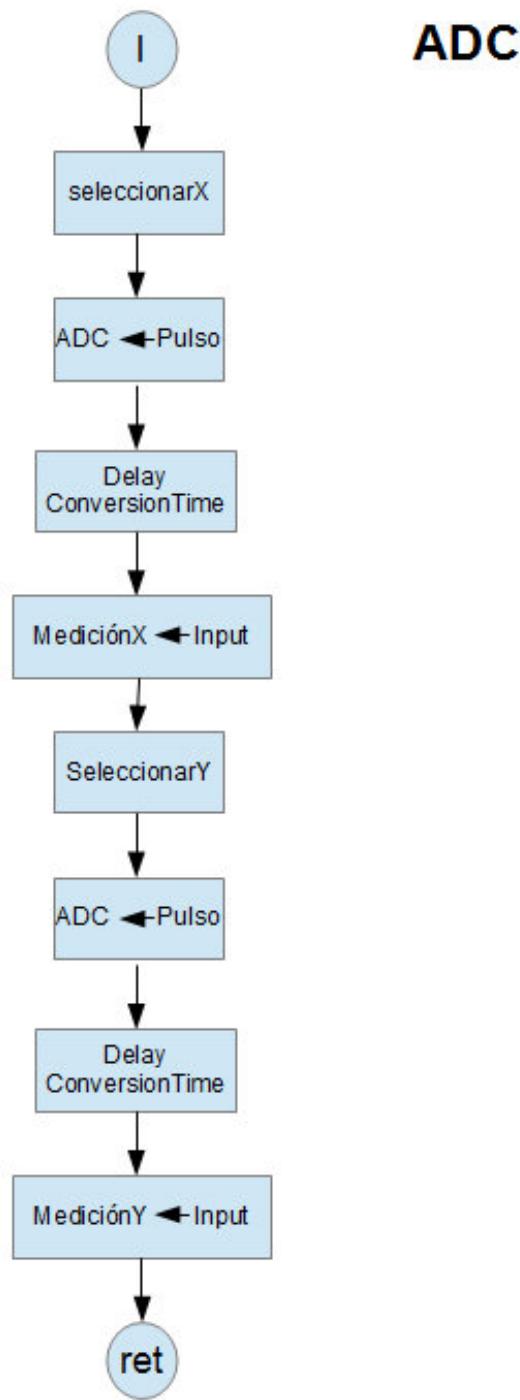


Figura 11: Diagrama de la lectura de la medición

Luego se determina que tipo de aceleración es:

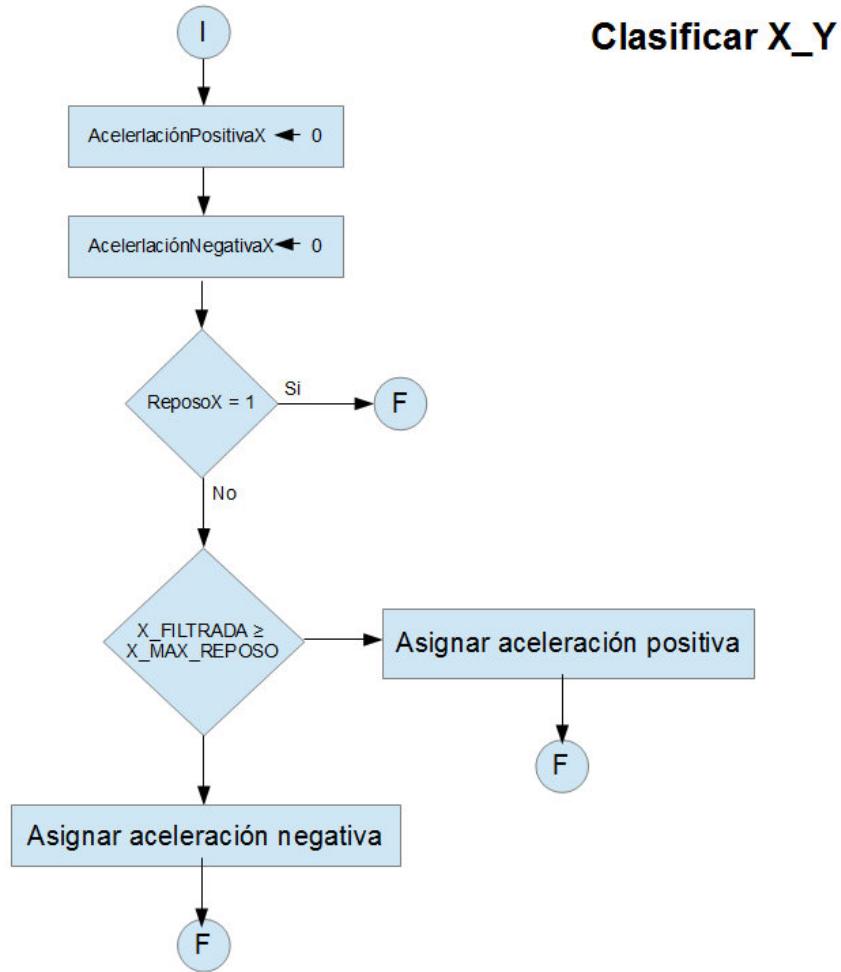


Figura 12: Diagrama de la clasificación de la aceleración

Y luego, si existe aceleración, se ejecuta una función que se encarga de mover el mouse:

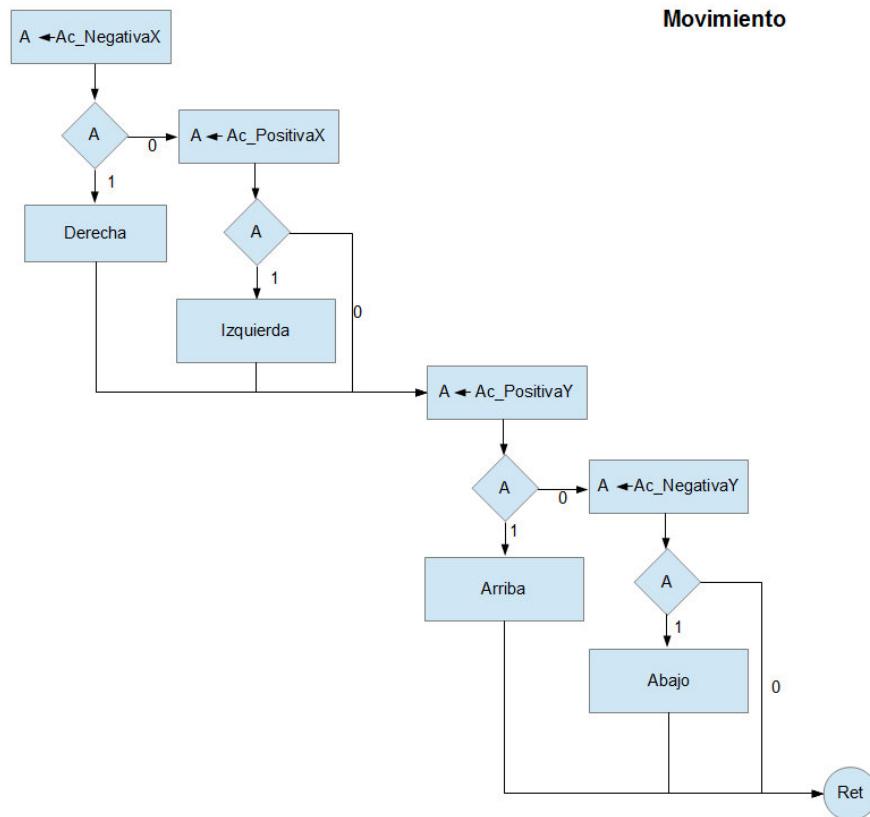


Figura 13: Diagrama de la función Movimiento

Para los movimientos hacia la derecha o hacia arriba, se utiliza la señal activa derecha que se puede denominar como “señal larga”, ya que coloca un pulso de duración larga entre las señales de stand-by. En cambio, para los movimientos hacia la izquierda y hacia abajo, se utiliza la señal activa izquierda que se puede denominar como “señal corta”

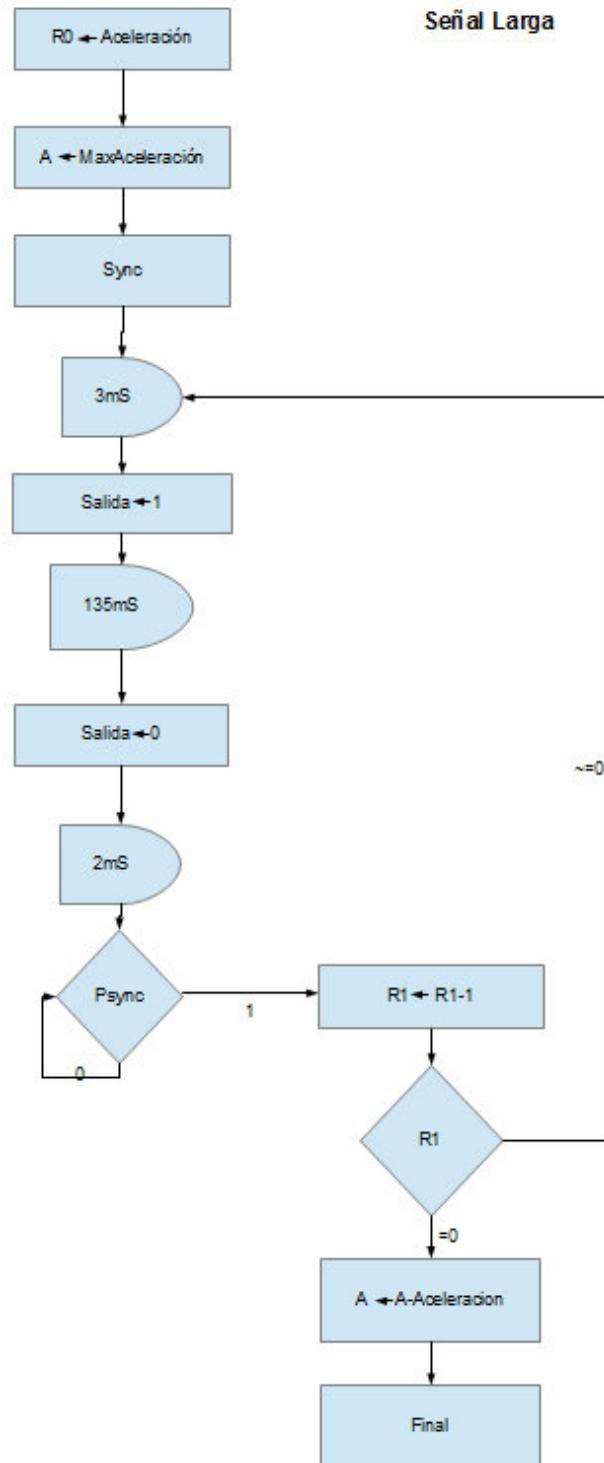


Figura 14: Diagrama de la señal larga

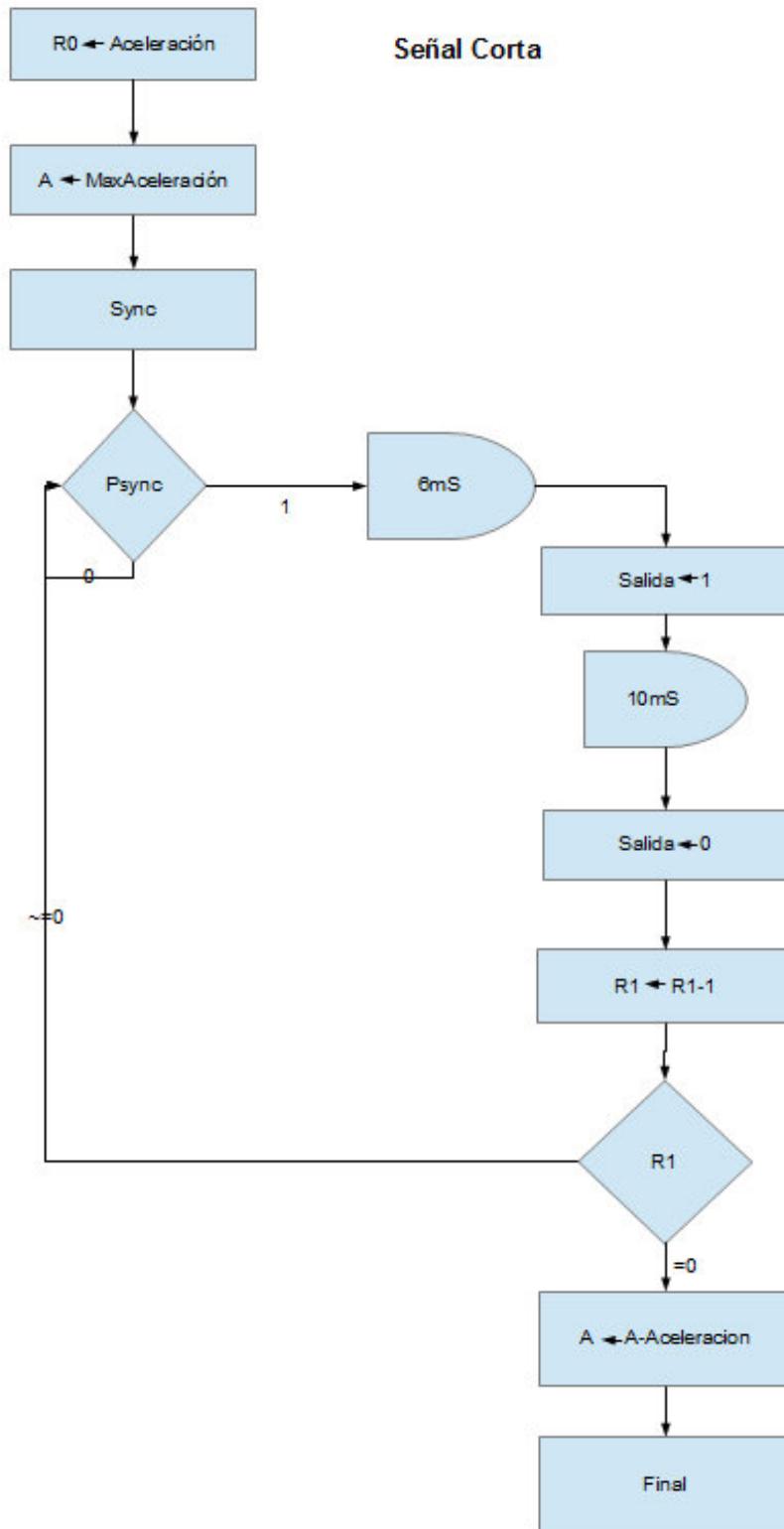


Figura 15: Diagrama de la señal corta

Ambas señales activas terminan de la misma manera, a continuación se muestra un diagrama de flujo de la función final

## Final

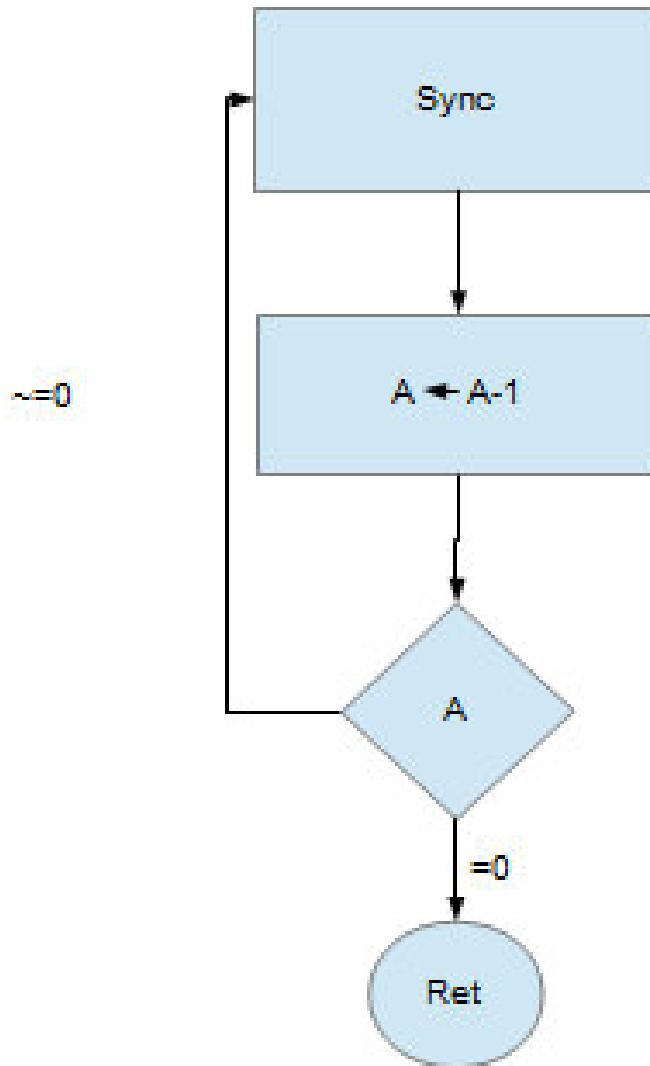


Figura 16: Diagrama de final

## 5. Circuito Esquemático

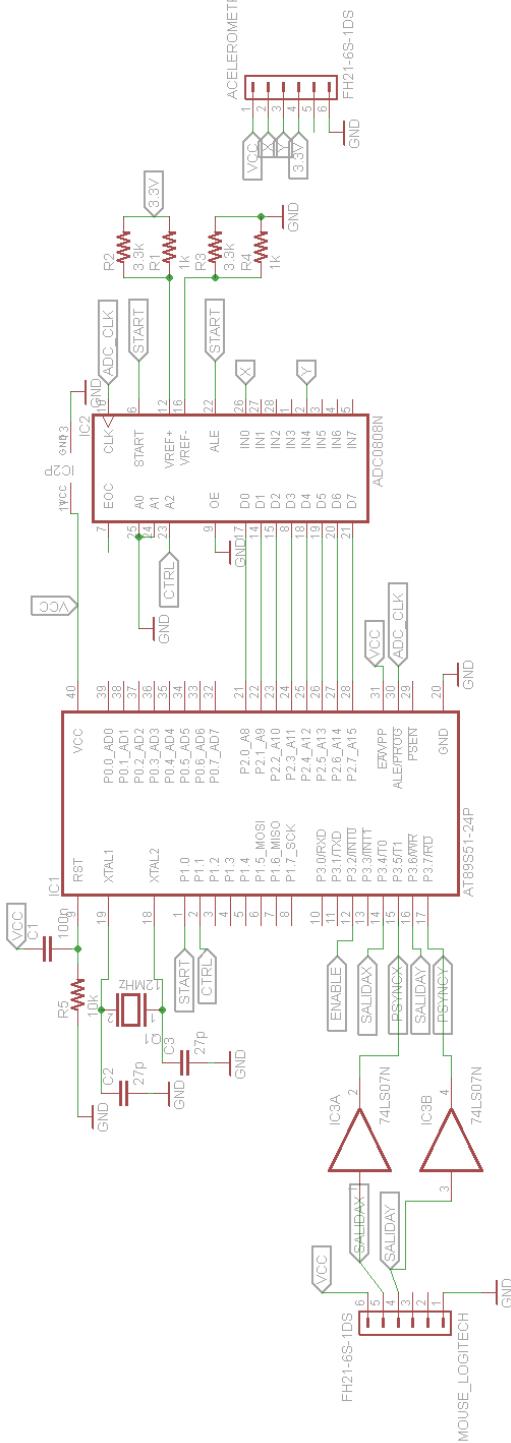


Figura 17: Resultados del filtro implementado

## 6. Listado de Componentes y Costos

A continuación se muestra un cuadro con el resumen de costos del proyecto y la cantidad de horas hombre que se invirtieron en el proceso de desarrollo y construcción del Guante-Mouse.

Componente	Costo	Actividad	Horas-hombre
Acelerómetro	\$80	Medición del mouse	20
Conversor A/D	\$40	Medición del acelerómetro	20
Buffer Digital	\$8.50	Calibración del conversor AD	5
8051	\$20	Diseño del filtro	5
Zocalo+Cables	\$2	Programación	20
Placa Experimental	\$25	Armado del prototipo	5
		Armado del informe	20

Cuadro 1: Listado de componentes y costos

## 7. Resultados

Se cumplieron varias de las expectativas presentadas en el anteproyecto. Durante el desarrollo del proyecto se cambiaron especificaciones y modos de funcionamiento del mouse.

El cambio más notable en relación a lo propuesto, es el movimiento necesario para mover el cursor.

Inicialmente se propuso que el cursor se mueva a partir de movimientos horizontales y verticales de la mano, pero por el funcionamiento del acelerómetro, se decidió cambiar estos movimientos por rotaciones de la mano.

El cursor se mueve hacia la derecha con una inclinación en el sentido horario y hacia la izquierda con una inclinación hacia el lado opuesto. Para mover el cursor hacia arriba, se tiene que subir la muñeca.

Se consiguió calibrar el acelerómetro para que la posición de reposo resulte cómoda cuando se está escribiendo con el teclado.

Ademas se agregó un *enable* para poder mover el cursor de la PC cuando se deseé. Cuando no se está utilizando el mouse, el 8051 entra en modo IDLE por lo que el consumo baja notoriamente.

Se optó por no colocar una rueda física porque resulta molesta para el que este usando el guante. Por lo tanto el guante tiene únicamente los dos botones básicos de un mouse.

Se logró implementar un mouse preciso con una buena sensibilidad. La no linealidad de la aceleración permite una mayor sensibilidad a leves inclinaciones y evita movimientos bruscos e incontrolables para inclinaciones mayores.

El proyecto se alimenta a partir del puerto USB de una PC. Además se debe conectar el mouse *LOGITEC* a un puerto PS2 que es el que permite la comunicación con la computadora.

La conexión por cables de las 2 partes del circuito no permiten que el usuario se aleje mucho de la PC y generalmente resultan ser molestos.

La implementación del movimiento del cursor tiene leves defectos en los movimientos diagonales. En algunos casos se puede notar que los movimientos en los ejes no son simultáneos. esto se ve como que la diagonal tiene forma de escalera. Esto se pudo corregir hasta hacerlo casi imperceptible en la mayoría de las velocidades.

## 8. Conclusiones

Se llegó entonces a un guante mouse cómodo y de precisión aceptable. La programación y construcción del mismo estuvieron condicionadas por el circuito existente de mouse, lo que ayudó por la facilidad de comunicación PC – mouse, pero limita la reproducción del proyecto así como su portabilidad. En un futuro, se podrían operar ciertos cambios que mejoraría el uso del aparato:

- Estudiar en profundidad cuales son los mejores tiempos de envío de señal activa (Tcom), para lograr buena precisión y fluidez .
- Implementación de la rueda de *scrolling* con el acelerómetro. Se podría agregar una parte de código que permita hacer *scrolling* inclinando la mano. Para desasociarlo del movimiento del cursor, se podría activar la rueda juntando los dedos mayor y anular.
- Implementación de una comunicación *wireless* entre el guante y la PC. Se podría desarrollar una forma de transmitir los datos del acelerómetro al 8052, o del 8052 a la PC, de forma inalámbrica, mejorando la comodidad del mouse.
- Utilización de un solo puerto. Actualmente, el proyecto requiere de un puerto PS2 y un USB para funcionar. Se podría pensar en una forma de utilizar solo uno de ellos, distribuyendo la alimentación de un puerto a la totalidad del circuito.
- Optimización de los tiempos. Si se logra optimizar el tiempo de conversión del ADC, así como mejorar el programa del proyecto para que sea más eficiente a nivel temporal, se podría lograr una mayor fluidez en el cursor.

## 9. Apéndices

### 9.1. Filtro pasa-bajos

Un promedio se define como:

$$\bar{X} = \frac{X_1 + X_2 + X_3 + \cdots + X_N}{N}$$

Esta expresión puede reescribirse de la siguiente manera definiendo un subpromedio:

$$\begin{aligned}\bar{X}_{1,2} &\triangleq \frac{X_1 + X_2}{2} \\ \Rightarrow \bar{X} &= \frac{2 \cdot \bar{X}_{1,2} + X_3 + \cdots + X_N}{N}\end{aligned}$$

Agrupando más términos en este subpromedio se obtiene:

$$\bar{X} = \frac{\bar{X}_{1,2,\dots,N-1} \cdot (N-1) + X_N}{N}$$

De esta manera se deduce la fórmula para calcular un promedio progresivamente, en la cual se pondrá en mayor medida el promedio anterior ( $\bar{X}_{i-1}$ ) que el último valor obtenido:

$$\bar{X}_i = \frac{\bar{X}_{i-1} \cdot (i-1) + X_i}{i}$$

Interpretando este resultado como un sistema con entrada  $X_i$  y salida  $\bar{X}_i$  y fijando un valor constante para  $i$  se obtiene la siguiente ecuación en diferencias:

$$y(n) = y(n-1) \cdot \frac{N-1}{N} + x(n) \cdot \frac{1}{N}$$

donde  $N$  da una idea de la cantidad de muestras que se estaría “promediando” de forma ficticia (ya que no es estrictamente un promedio al haber fijado  $i = N = cte$ ).

## 9.2. Selección de Velocidad

Esta función evita errores en la aceleración que se envía a CTRL\_MOUSE que pueden provocar un mal funcionamiento de la función.

Se decidió utilizar esta función cuando, al variar la cantidad de periodos que se utilizan para mover el mouse, se notó que ante aceleraciones grandes existía un *overflow* que hacia que el movimiento del cursor sea entrecortado.

Se dividió la medición por un numero fijo para evitar el *overflow*, pero esto provoco que la sensibilidad baje para cualquier inclinación. La solución consiste en considerar una pendiente mayor para pequeños valores de aceleración y una pendiente menor para valores que se acercan a la aceleración máxima.

Para adecuar el valor enviado por el acelerómetro con la cantidad de periodos utilizados por vez para mover el mouse, se implementó una función que según el valor medido devuelve el valor que recibirá la función CTRL\_MOUSE.

La función busca aumentar la sensibilidad del mouse para inclinaciones leves y controlar la velocidad para aceleraciones grandes.

Se declaró una función partida continua compuesta por tres rectas distintas.

$$f(x) = \begin{cases} a_1x & 0 < x < m_1 \\ a_2x + b_2 & m_1 < x < m_2 \\ a_3x + b_3 & m_2 < x < m_3 \\ ac_{max} & m_3 < x \end{cases}$$

Definiendo todos los margenes (m) y las pendientes (a) se pueden calcular el resto de los parámetros para que sea una función continua.

En la versión actual del mouse se utilizó la función dada por los siguientes valores.

$$f(x) = \begin{cases} \frac{x}{2} & 0 < x < 20 \\ \frac{x}{4} + 5 & 20 < x < 60 \\ \frac{x}{5} + 8 & 60 < x < 80 \\ 25 & 85 < x \end{cases}$$