

UNIVERSIDAD DEL VALLE DE GUATEMALA
FACULTAD DE INGENIERÍA



Dispositivo de Interacción
Humano-Computador montado
sobre la mano con enlace
inalámbrico

Trabajo de graduación presentado por Kurt
Emmanuel Kellner Juárez para optar al grado de
Licenciado en Ingeniería Mecatrónica

Guatemala

2011

Dispositivo de Interacción
Humano-Computador montado
sobre la mano con enlace
inalámbrico

UNIVERSIDAD DEL VALLE DE GUATEMALA
FACULTAD DE INGENIERÍA



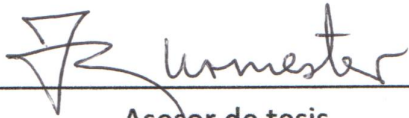
Dispositivo de Interacción
Humano-Computador montado
sobre la mano con enlace
inalámbrico

Trabajo de graduación presentado por Kurt
Emmanuel Kellner Juárez para optar al grado de
Licenciado en Ingeniería Mecatrónica

Guatemala

2011

Vo. Bo.



Asesor de tesis
Ing. Juan Carlos Burmester

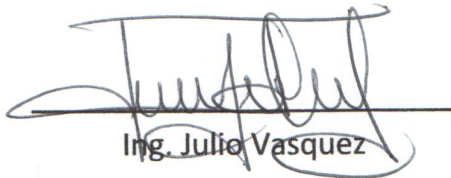
Terna Calificadora:



Ing. Roberto Delgado



Ing. Carlos Esquit



Fecha de aprobación:

Guatemala, 9 de diciembre de 2011

CONTENIDO

CONTENIDO	iv
LISTA DE CUADROS	vi
LISTA DE ILUSTRACIONES	vi
RESUMEN	viii
I. INTRODUCCIÓN	1
II. OBJETIVOS	2
A. Generales	2
B. Específicos	2
III. MARCO TEÓRICO	3
A. Acelerómetros	3
B. Sensor Flex	6
C. Bus de comunicación	9
1. UART – Universal Asynchronous Receiver/Transmitter	9
2. Serial Peripheral Interface (SPI)	10
D. ZigBee/IEEE 802.15.4 LR-WPAN	12
1. XBee Serie 1	15
E. Micro-controladores	16
1. Arduino UNO	16
2. LilyPad Arduino	18
F. E-Textiles	19
1. Hilo conductor	19
2. Pintura dimensional	20
G. Interfaz gráfica	20
1. Processing	21
2. Robot Class	22
IV. DESCRIPCIÓN DE LA FUNCIONALIDAD DEL GUANTE	23

V.	MOVIMIENTOS DE LA MANO ÚTILES APLICABLES EN UN ESPACIO GRÁFICO	27
VI.	MEDICIÓN DE INCLINACIÓN.....	31
VII.	MEDICIÓN DE FLEXIÓN DE LOS DEDOS Y SEÑALES DIGITALES	35
VIII.	RECOLECCIÓN DE DATOS MEDIDOS Y COMUNICACIÓN INALÁMBRICA.....	40
IX.	PROCESAMIENTO DE DATOS Y MANEJO DEL AMBIENTE GRÁFICO	45
X.	CONCLUSIONES	53
XI.	RECOMENDACIONES.....	54
XII.	BIBLIOGRAFÍA.....	55

LISTA DE CUADROS

Tabla 1 - Resumen de acciones de botones, sensores flex y acelerómetro	26
Tabla 2 – Atajos del teclado y sus acciones en el sistema operativo MS Windows 7	27
Tabla 3 - Movimientos de la mano asignados a acciones de la computadora	29
Tabla 4 - Rangos de resistencias de los sensores flex montados.....	36
Tabla 5 - Valores de salida del divisor de voltaje	37
Tabla 6 - Catalogación de flexión de los dedos.....	38

LISTA DE ILUSTRACIONES

Figura 1 - Acelerómetro capacitivo con un capacitor diferencial	5
Figura 2 - Circuito convertidor de capacitancia en voltaje	5
Figura 3 - Sensor flex.....	7
Figura 4 - Bus SPI entre maestro y esclavo	11
Figura 5 - Bus SPI entre maestro y tres esclavos	11
Figura 6 - Modelos de topología de redes ZigBee	14
Figura 7 - XBee con antena expuesta.....	15
Figura 8 - Arduino UNO.....	17
Figura 9 - LilyPad Arduino	18
Figura 10 - LilyPad FTDI.....	18
Figura 11 - Hilo conductor de cuatro fibras	19
Figura 12 - Pintura dimensional	20
Figura 13 - Sensores implementados sobre el guante.....	23
Figura 14 - Inclinationes máximas del guante	24
Figura 15 – Aceleración de la mano al realizar movimientos usuales con un mouse	31
Figura 16 - Implementación del acelerómetro BMA180	32
Figura 17 - Convertidor de niveles lógicos.....	33
Figura 18 - Circuito de un canal del convertidor lógico	33
Figura 19 - Conexión entre el convertidor lógico y el acelerómetro	34
Figura 20 - Sensor flex.....	35
Figura 21 - PCB para aplicaciones textiles.....	35
Figura 22 - Sensor flex soldado sobre PCB para aplicaciones textiles	36
Figura 23 - Circuito divisor de voltaje para medir la flexión del dedo	36
Figura 24 - Botón tipo push diseñado para aplicaciones textiles	38

Figura 25 - Conexión de los botones push	39
Figura 26 - Conexiones del LilyPad Arduino.....	40
Figura 27 - Ambiente de desarrollo integrado (IDE, por sus siglas en inglés) de Arduino.....	41
Figura 28 - Diagrama de flujo del LilyPad Arduino.....	42
Figura 29 - Diagrama de flujo de LilyPad Arduino (cont)	43
Figura 30 - Diagrama de flujo del Arduino UNO	44
Figura 31 - Ambiente de desarrollo integrado (IDE, por sus siglas en inglés) de Processing	45
Figura 32 - Aplicación de prueba	46
Figura 33 - Aplicación final.....	46
Figura 34 - Aplicación final con colores de muestra	48
Figura 35 - Diagrama de flujo de la aplicación en la computadora (1)	49
Figura 36 - Diagrama de flujo de la aplicación en la computadora (2)	50
Figura 37 - Diagrama de flujo de la aplicación en la computadora (1)	51
Figura 38 - Diagrama de flujo de la aplicación de la computadora (4)	52

RESUMEN

Los dispositivos de entrada de las computadoras son esenciales para que un usuario pueda operarlas. Existen dispositivos como el *mouse* o ratón que se encarga de controlar un apuntador en un espacio gráfico. Lo hace a través de un equipo de *hardware* que el usuario desplaza sobre una mesa. Otro dispositivo común es el teclado que maneja la entrada de datos a través de un arreglo de teclas que representan diferentes letras, números o funciones. El dispositivo propuesto en este trabajo de graduación combina todas las funciones de un ratón con algunas de las funciones de un teclado creando así una solución innovadora para poder operar un ambiente gráfico en una computadora.

Es posible medir suficientes variables de la posición de una mano (como la inclinación y flexión de los dedos) utilizando un acelerómetro, tres sensores flex y cuatro botones tipo *push* con el objeto de controlar un ambiente gráfico en una computadora. Además, el sistema propuesto está configurado para que este control se lleve a cabo con movimientos lo más naturales posibles, por lo que utilizar el dispositivo es relativamente sencillo.

Se llegó a la conclusión que para diseñar un dispositivo práctico y funcional es necesario restringir la cantidad de funciones que este puede ejecutar. Además, se concluyó también que es imposible alcanzar el reposo absoluto con la mano; esto debe tomarse en cuenta al momento de desarrollar dispositivos que sean manejados por ella.

I. INTRODUCCIÓN

Este trabajo de graduación propone el diseño de un dispositivo de interacción humano-computador que mediante la inclinación de la mano y la realización de ciertos gestos con la misma, sea capaz de controlar un ambiente gráfico en una computadora. No pretende sustituir los actuales medios de control tales como el teclado y el ratón, sino que presenta una alternativa innovadora.

Para realizar este proyecto se llevaron a cabo investigaciones en diferentes áreas. La primera se enfocaba en las funciones propias de un teclado y un ratón en un ambiente gráfico específico: MS Windows 7 de Microsoft. Luego se hicieron experimentos para conocer los movimientos más naturales de la mano y una referencia cruzada para determinar qué movimientos podían ser utilizados para qué funciones. Una vez definidas las funciones a implementar se procedió a diseñar un guante que incluyera un módulo que midiera la inclinación de la mano con respecto al plano horizontal y con estos datos mover el puntero de un ambiente gráfico. La utilización de sensores de posición montados sobre los dedos pulgar, índice y medio permiten a un micro-controlador reconocer los gestos de la mano. Además, se montaron cuatro botones en los dedos índice y medio para poder controlar la funcionalidad del guante. Para mejorar la operación del guante se diseñó un módulo de comunicación inalámbrico para transferir los datos de los sensores y botones a una computadora en la que se creó una aplicación para interpretarlos y aplicarlos.

Durante la realización de este proyecto se llegó a la conclusión de que para desarrollar un dispositivo innovador es necesario restringir la funcionalidad del mismo. Además, al implementar algún dispositivo sobre la mano hay que tomar en cuenta que es imposible lograr reposo absoluto con la mano; esto ocasiona ruido en las mediciones.

II. OBJETIVOS

A. Generales

1. Diseñar un prototipo de interacción humano-computador montable en la mano por medio de un guante que permita el control remoto e inalámbrico de funciones de manipulación del ambiente gráfico de una computadora por medio de movimientos y gestos de la mano.

B. Específicos

2. Desarrollar una lista de movimientos y gestos naturales de la mano que sean útiles y puedan ser implementados para el control de una computadora
3. Diseñar un sistema de detección de inclinación de la mano utilizando un acelerómetro
4. Diseñar un sistema de reconocimiento de los gestos de la mano utilizando sensores flex
5. Diseñar un sistema de botones para la selección de funciones específicas en la computadora
6. Diseñar un sistema inalámbrico para la transmisión de datos desde la mano hasta la computadora
7. Diseñar una plataforma para el montaje de todos los sensores sobre la mano
8. Diseñar una interfaz en la computadora que reciba los datos de los sensores y aplique los cambios correspondientes en el ambiente gráfico

III. MARCO TEÓRICO

A. Acelerómetros

La aceleración es una característica dinámica de un objeto y de acuerdo a la segunda ley de Newton, se debe esencialmente a la aplicación de una fuerza externa. La posición, velocidad y aceleración de un objeto están relacionadas matemáticamente entre sí: la velocidad es la primera derivada de la función que describe su posición y la aceleración es la segunda derivada. La medición electrónica de los datos de aceleración o velocidad de un objeto por medio de derivadas en un ambiente con mucho ruido eléctrico o mecánico puede ocasionar errores grandes. Para reducir estos errores se utilizan sensores especiales. (Fraden, 2003)

Como regla de oro, en aplicaciones con movimientos de baja frecuencia (con un ancho de banda del orden de 1Hz) medir la posición o desplazamiento es suficientemente preciso. En aplicaciones de frecuencias medias (menores a 1kHz) se favorece a la medición de velocidad. Aplicaciones en las que existen movimientos de altas frecuencias e incluso niveles de ruido apreciables se utiliza la medición de la aceleración. (Fraden, 2003)

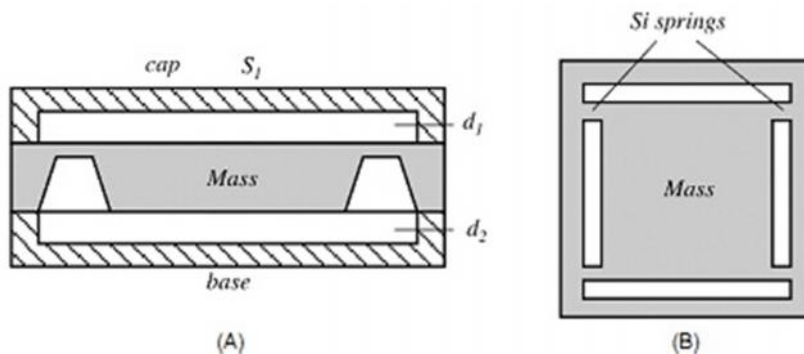
Un acelerómetro es un dispositivo electromecánico que mide fuerzas de aceleración. Estas pueden ser estáticas, como por ejemplo la fuerza de gravedad, o dinámicas causadas al mover o hacer vibrar el acelerómetro. La vibración es un fenómeno mecánico dinámico que involucra un movimiento oscilatorio periódico alrededor de una posición de referencia. Se podría especificar a un acelerómetro como un dispositivo de un grado de libertad que tiene algún tipo de masa sísmica (muchas veces llamada *masa prueba*), un sistema de soporte por muelles o similar y una estructura con propiedades de amortiguación. Una visualización de esto se puede apreciar en la Figura 1. (Andrejašić, 2008) (Fraden, 2003)

Existen varios tipos de acelerómetros que utilizan diferentes tecnologías: capacitancia, piezo-resistencia, piezo-electricidad, temperatura, etc. Sin embargo, todos son manufacturados utilizando la tecnología llamada *Sistemas Micro Electro Mecánicos* (MEMS, por sus siglas en inglés). En el año 1989 el Profesor R. Howe utilizó este término para describir un campo de investigación en auge en el cual elementos mecánicos eran manufacturados a una escala micrométrica. (Andrejašic, 2008) (Fraden, 2003)

Un acelerómetro requiere de un componente especial cuyo movimiento se retrase con respecto al alojamiento del acelerómetro. Un transductor de desplazamiento puede ser utilizado para generar una señal eléctrica como una función de la aceleración. Este componente es la *masa prueba* o también llamada *masa inercial* o *sísmica*. Sin importar el diseño del sensor o la técnica de conversión el objetivo de la medición es la detección del desplazamiento de esta masa con respecto del alojamiento del acelerómetro. Por lo tanto, cualquier transductor de desplazamiento capaz de medir desplazamientos microscópicos bajo vibraciones fuertes o aceleración lineal puede ser utilizado en un acelerómetro. (Fraden, 2003)

Una conversión de desplazamiento capacitivo es uno de los métodos probados y confiables. Un sensor que utiliza este método contiene por lo menos dos componentes: una placa *estacionaria* (conectada al alojamiento) y otra placa unida a la masa inercial. Esta última es libre de desplazarse dentro del alojamiento. Estas placas forman un capacitor cuyo valor es una función de la distancia entre ellas; se dice entonces que el valor de capacitancia es modulado por la aceleración. Como raras veces se excede una medición de desplazamiento mayor que $20\ \mu\text{m}$, se requiere de una compensación confiable de desvíos e interferencias varias. Esto se logra usualmente mediante el uso de una técnica diferencial, agregando un capacitor en la misma estructura. El valor de este segundo capacitor tiene que ser cercano al del primero y debe estar sometido a cambios con un desfase de 180° , así, la aceleración puede ser representada por la diferencia entre los valores de ambos capacitores. (Fraden, 2003)

Figura 1 - Acelerómetro capacitivo con un capacitor diferencial: (A) vista lateral seccionada; (B) vista superior de la masa sísmica apoyada sobre cuatro resortes de silicón

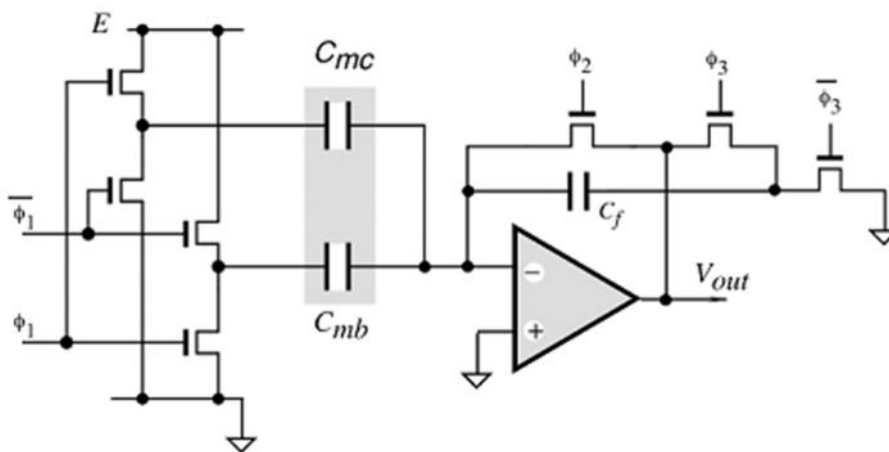


(Fraden, 2003)

(Fraden, 2003)

La Figura 1A muestra un diagrama lateral seccionado de un acelerómetro capacitivo en donde la masa interna se encuentra entre la tapa superior y la base. La masa está apoyada sobre cuatro resortes de silicón. Las placas superior e inferior están separadas de la masa por distancias d_1 y d_2 respectivamente. Las tres piezas son micro-maquinadas a partir de un *wafer* de silicón. (Fraden, 2003)

Figura 2 - Circuito convertidor de capacitancia en voltaje



(Fraden, 2003)

Un circuito simplificado de un convertidor de capacitancia en voltaje se muestra en la Figura 2. El capacitor C_{mc} de placas paralelas de área S_1 se encuentra entre la masa y los electrodos de la tapa superior. El espaciamiento d_1 puede ser reducido una cantidad Δ cuando la masa se mueve en dirección a la tapa superior. El área de las placas del

segundo capacitor C_{mb} es diferente pues aparece entre la masa y la tapa inferior o base. Cuando la masa se mueve hacia arriba alejándose de la base el espaciamento d_2 incrementa en Δ . El valor de Δ es igual a la fuerza mecánica F_m que actúa sobre la masa dividido la constante k de los resortes de silicón:

$$\Delta = \frac{F_m}{k} \quad (1)$$

Estrictamente hablando, el circuito equivalente del acelerómetro únicamente es válido cuando fuerzas electroestáticas no afectan la posición de la masa (esto es, cuando los capacitores dependen linealmente de F_m). Cuando un acelerómetro se utiliza como el capacitor de entrada de un amplificador sumador el voltaje de salida depende del valor de los capacitores y, por consiguiente, de la fuerza:

$$v_{out} = 2E \frac{C_{ms} - C_{mb}}{C_f} \quad (2)$$

Esta ecuación es verdadera para cambios pequeños en las capacitancias del sensor. La salida del acelerómetro también es una función de temperatura y desajustes capacitivo por lo que es recomendable que la calibración se haga sobre un rango completo de temperaturas y que se haga una corrección apropiada durante el procesamiento de la señal. (Fraden, 2003)

B. Sensor Flex

La medición de posición y desplazamiento de algún objeto físico es esencial para aplicaciones como el control de realimentación de procesos, evaluaciones de desempeño, control de transporte, robótica, sistemas de seguridad, etc. La *posición* hace referencia a la determinación de las coordenadas, lineales o angulares, de un objeto con respecto a una referencia seleccionada. *Desplazamiento* se refiere a mover el objeto de una posición a otra una distancia o ángulo determinado. En otras palabras, el desplazamiento es la distancia que un objeto se mueve tomando como referencia su posición anterior. (Fraden, 2003)

Una distancia crítica es medida por sensores de *proximidad*, que no se trata de más que un sensor de posición con umbral. Un sensor de posición es usualmente un

dispositivo lineal cuya señal de salida representa una distancia hacia un objeto desde algún punto de referencia. (Fraden, 2003)

Un sensor de desplazamiento muchas veces es parte de un sensor más complejo en el que la detección de movimiento es uno de varios pasos en una conversión de señal. Un ejemplo de esto es un sensor de presión en el que la presión es traducida en el desplazamiento de un diafragma y el desplazamiento de este diafragma se convierte en una señal eléctrica que representa la presión. Los sensores de posición y desplazamiento son dispositivos estáticos cuya velocidad de respuesta usualmente no es crítica para su desempeño. (Fraden, 2003)

De acuerdo a SensorWiki, los sensores de flexión miden la cantidad de deflexión causada por alguna fuerza ejercida sobre el sensor. Para poder medir esta deflexión existen varios métodos, pero los tres más comunes son: mediciones a base de tinta conductiva, utilización de fibra óptica y mediciones a base de tela conductiva. (SensorWiki, 2011)

Figura 3 - Sensor flex



(SensorWiki, 2011)

La Figura 3 muestra un sensor flex que utiliza tinta conductiva. Al doblar este sensor la resistencia aumenta hasta cierto punto y cuando el sensor se encuentra completamente extendido no hay variaciones en la resistencia. Una propiedad de estos sensores que debe ser notada es que doblar el sensor únicamente desde un extremo no es la mejor forma de utilizarlo. Tampoco el doblarlo a más de 90° en un único punto ya que puede ser dañado. La mejor manera de utilizar este tipo de sensor es curvarlo en

toda su extensión; mientras más pequeño sea el radio de curvatura y mayor extensión del sensor sea curvada mayor será la resistencia. (SensorWiki, 2011)

Las especificaciones básicas de un sensor típico de flexión son el rango de deflexión, mediciones uni- o bi-direccionales, mediciones uni- o bi-polares y el rango de resistencia. El rango de deflexión determina el ángulo máximo de deflexión que puede ser medido, pero este no equivale al ángulo máximo al cual el sensor puede ser doblado. Un sensor unidireccional aumenta su valor de resistencia si se flexiona en una dirección. Si la flexión ocurre en sentido contrario del diseñado el cambio de resistencia es muy bajo o nulo. Otros sensores cambian su valor resistivo tanto si se flexionan en una dirección como en la otra (sensores bi-direccionales). En el caso de un sensor unipolar no importa la dirección de la flexión el cambio de resistencia para esa flexión será el mismo. Sin embargo, un sensor bipolar depende del sentido de la flexión: podría ser que una misma flexión en sentidos opuestos ocasione un cambio resistivo diferente. Por último, el rango de resistencia de un sensor flex es el rango conformado por los valores máximos y mínimos que pueden ser medidos con el sensor. (SensorWiki, 2011)

Los sensores flex que utilizan tinta conductiva son dispositivos de resistencia pasivos. Típicamente son fabricados colocando una capa de tinta resistiva sobre un sustrato de plástico flexible en forma de una tira delgada de extensión entre 1" y 5". Cuando el sensor se encuentra completamente extendido presenta una resistencia intrínseca mientras que cuando se dobla los materiales resistivos que se encuentran en la tinta se separan más el uno del otro lo cual aumenta la resistencia. Usualmente la resistencia nominal intrínseca de un sensor oscila entre los 10k Ω y los 50k Ω e incrementa hasta en un factor de 10 en su máxima deflexión. (SensorWiki, 2011)

Dentro de las capas del sustrato de un sensor flex se encuentra impreso un patrón de tinta conductiva. Para poder conducir electricidad esta tinta contiene pequeñas partículas de carbono o plata mezcladas con los pigmentos de la tinta. Las partículas de carbono se encuentran suspendidas en tinta para evitar la despigmentación con el tiempo. (SensorWiki, 2011)

C. Bus de comunicación

Un bus de comunicación es un medio que transfiere señales eléctricas de control, señales de datos y de temporización entre subsistemas de procesadores. (McGraw-Hill Companies, Inc., 2003)

1. UART – Universal Asynchronous Receiver/Transmitter. El UART (Transmisor/Receptor Universal Asíncrono, por sus siglas en inglés) es el componente electrónico responsable de las comunicaciones entre computadoras, sistemas, micro-controladores y dispositivos que ocurren de forma serial. Transforma información dada de forma paralela en datos en serie que pueden ser enviados sobre una línea de comunicación. Un segundo UART es capaz de recibir y traducir la información que transmite el primero. El componente se encarga de todas las tareas pertinentes al intercambio de información: temporización, chequeo de paridad, *bits* de parada e inicio, velocidad del puerto, etc. Los únicos dispositivos adjuntos a un UART son los chips *drivers* capaces de transformar las señales TTL en voltajes de línea y viceversa. (Lammert Bies, 2011)

Las primeras implementaciones de UART's no incluían un *buffer* por lo que cada dato que el dispositivo recibía o enviaba debía ser manejado por el procesador. Sin embargo esto dio como resultado una velocidad de comunicación limitada razón por la que se desarrollaron nuevas series de UARTs. El 16550A, que incluía dos *buffers FIFO* (*First In First Out*; Primero que llega Primero que sale, por sus siglas en inglés). Cada uno de estos *buffers* era capaz de retener hasta 16 *bytes*; uno se utilizaba para entradas y el otro para salidas. Más adelante se desarrolló el UART 16650 que tenía dos *buffers* con capacidad de 32 *bytes* cada uno. (Lammert Bies, 2011)

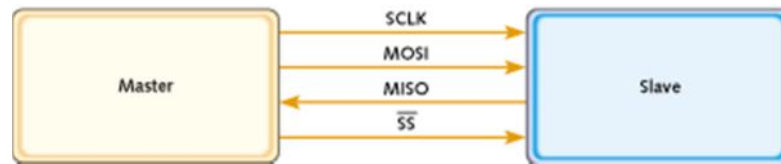
2. Serial Peripheral Interface (SPI). La *Interfaz Serial Periférica* (SPI, por sus siglas en inglés) es un bus de comunicación utilizado entre dispositivos que maneja una velocidad relativamente lenta y es de bajo costo. Provee un buen soporte para la comunicación con dispositivos periféricos accedidos de manera intermitente y tiene un buen desempeño en aplicaciones en las que la comunicación entre los dispositivos tiende más a ser un *stream*. Un ejemplo de una aplicación de este tipo es la comunicación entre microprocesadores o procesadores de señales digitales. Si además se utiliza la capacidad de comunicación en ambas direcciones del bus se maximiza la eficiencia del mismo. (Kalinsky & Kalinsky, 2002)

El protocolo SPI se comunica utilizando una relación maestro-esclavo sin direccionamiento para cada esclavo, razón por la cual requiere de más esfuerzo y recursos de *hardware* para comunicarse con varios esclavos. Sin embargo, si se trata de una comunicación punto-a-punto el bus es realmente eficiente y simple de utilizar. (Kalinsky & Kalinsky, 2002)

El estándar serial SPI fue establecido por Motorola y está soportado por una alta gama de circuitos integrados. La interfaz está disponible en procesadores populares de comunicación como el MPC8260 y micro-controladores como el M68HC 11. (Kalinsky & Kalinsky, 2002)

Los dispositivos se comunican en una relación maestro-esclavo en donde el maestro es el que siempre inicia la comunicación y elige con qué esclavo comunicarse. Al haber iniciado los pulsos de reloj y seleccionado con qué esclavo se va a comunicar la transferencia de datos se realiza en ambas direcciones; es decir, se transmite tanto un *byte* desde el maestro al esclavo como desde el esclavo al maestro. Depende del maestro y esclavo reconocer si el *byte* transmitido es significativo. Si el dispositivo sólo quiere transmitir debe descartar el *byte* recibido y si quiere sólo recibir debe crear un *byte* nulo. (Kalinsky & Kalinsky, 2002)

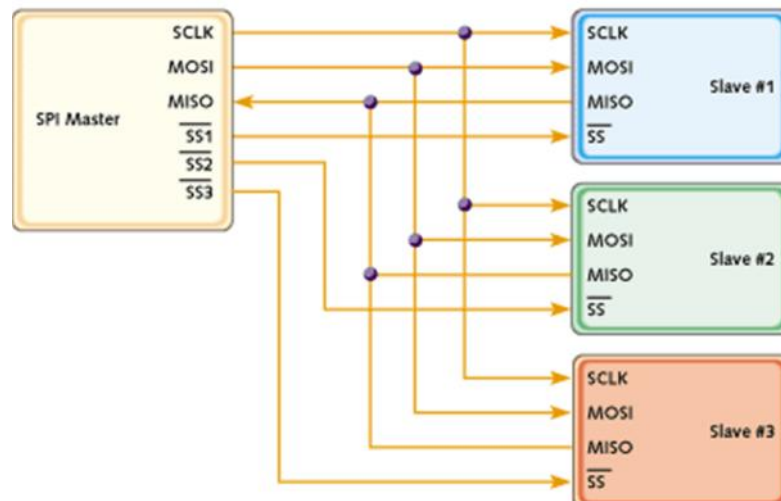
Figura 4 - Bus SPI entre maestro y esclavo



(Kalinsky & Kalinsky, 2002)

La Figura 4 muestra las señales del bus SPI entre el maestro y un solo esclavo. La señal de reloj es transmitida por el maestro vía la terminal SCLK, los datos del maestro hacia el esclavo a través de MOSI (Salida del Maestro Entrada del Esclavo, por sus siglas en inglés), los datos del esclavo hacia el maestro a través de MISO (Entrada del Maestro Salida del Esclavo, por sus siglas en inglés) y la selección del esclavo a través de SS (Selección de Esclavo, por sus siglas en inglés). (Kalinsky & Kalinsky, 2002)

Figura 5 - Bus SPI entre maestro y tres esclavos



(Kalinsky & Kalinsky, 2002)

Si existen varios dispositivos esclavos, como se muestra en la Figura 5, el maestro genera una señal diferente para cada uno de los esclavos. Estas señales generadas se caracterizan por ser señales digitales de propósito general. Es importante mantener el control del tiempo para no activar una señal de esclavo a la mitad de transmisión de un *byte*. (Kalinsky & Kalinsky, 2002)

Existen dos parámetros importantes que hay que configurar en un bus SPI: la polaridad del reloj (CPOL, por sus siglas en inglés) y la fase del reloj (CPHA, por sus siglas en inglés). La polaridad del reloj determina si su señal debe estar en alto o en bajo en los momentos en los que no está transmitiendo. La fase del reloj determina el momento en el pulso del reloj en el que debe hacerse la toma y propagación de datos. Cada uno de los parámetros tiene dos posibles combinaciones lo que da un total de cuatro combinaciones en total; todas incompatibles entre sí. Tanto el maestro como el esclavo deben estar funcionando con los mismos parámetros y, en el caso de que existan varios esclavos con configuraciones diferentes, el maestro debe hacer el ajuste de los parámetros para poder comunicarse con los diferentes esclavos. (Kalinsky & Kalinsky, 2002)

El bus SPI no tiene un mecanismo de confirmación de transmisión de datos, por esta razón es necesario implementar un protocolo de comunicación para tener mayor seguridad en la transferencia de los mismos. Sin embargo, debido a la simpleza del sistema y el hecho de que el bus está diseñado para operaciones a bajas velocidades sobre una distancia corta, muchas veces no es necesaria la implementación de un protocolo. (Kalinsky & Kalinsky, 2002)

D. ZigBee/IEEE 802.15.4 LR-WPAN

El grupo IEEE 802.15.4 se formó para crear un estándar de *Red de Área Local Inalámbrica* (WLAN, por sus siglas en inglés). En vista de la necesidad de una red con cobertura menor y para una densidad de usuarios mayor, se establece la creación de un estándar para las *Redes de Área Personal Inalámbricas* (WPAN, por sus siglas en inglés). El enfoque de las WPAN's es bajo costo, baja potencia, alcance corto y tamaño muy pequeño. Actualmente se han definido tres clases de WPAN's que difieren en velocidad de transferencia de datos, consumo de batería y calidad de servicio (QoS, por sus siglas en inglés). La WPAN de alta velocidad de transferencia de datos (IEEE 802.15.3) es adecuada para aplicaciones multimedia que requieren una alta QoS. Las WPAN's de media velocidad de transferencia de datos, también conocida como IEEE 802.15.1/Bluetooth, maneja una variedad de tareas desde celulares hasta comunicación

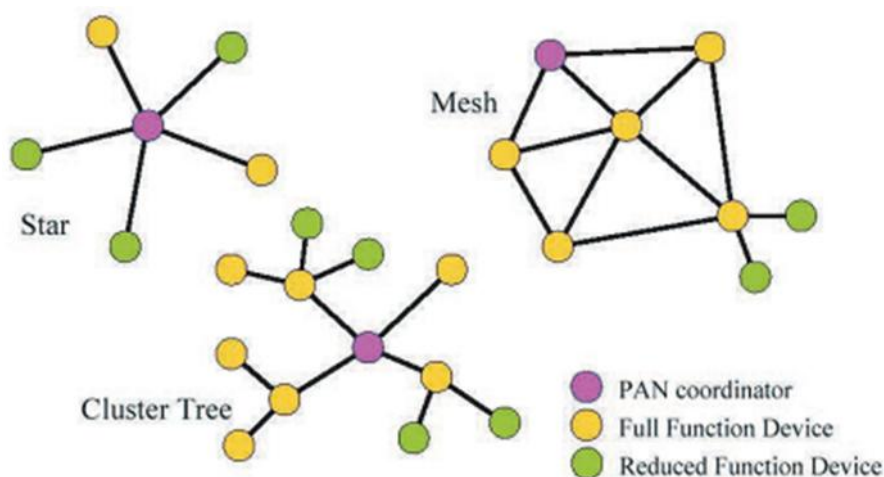
entre PDA y tienen una QoS adecuada para la comunicación de voz. La WPAN de baja velocidad, IEEE 802.15.4/LR-WPAN, es creada para aplicaciones industriales, residenciales y médicas con muy bajo consumo de potencia y costos bajos en comparación con las otras dos redes. Además la necesidad de QoS es más relajada al igual que la velocidad de transferencia de datos. (Coleri Ergen, 2004)

La tecnología ZigBee es un protocolo de redes inalámbricas de velocidad de transferencia baja y bajo consumo de potencia enfocado en la automatización y control remoto de aplicaciones. El comité IEEE 802.15.4 empezó a trabajar en un estándar similar poco tiempo después. De allí, la ZigBee Alliance y la IEEE decidieron unirse y hoy en día ZigBee es el nombre comercial de esta tecnología. (Coleri Ergen, 2004)

Las características principales de las WPAN's de estándar 802.15.4 son: la flexibilidad de red, bajo costo, consumo de potencia muy bajo y tasa de datos baja. Todo en una red *ad hoc* auto-organizativa entre dispositivos fijos, portables y móviles. (Coleri Ergen, 2004)

Un sistema ZigBee está conformado por varios componentes de los cuales el más básico es el dispositivo que puede ser completamente funcional (FFD, por sus siglas en inglés) o de funcionalidad reducida (RFD, por sus siglas en inglés). Una red incluirá por lo menos un FFD operando como el coordinador de la *Red de Área Personal* (PAN, por sus siglas en inglés). Además de este modo de operación un FFD puede operar también como dispositivo o coordinador simple. Un RFD está diseñado para aplicaciones extremadamente sencillas y que no necesitan enviar muchos datos. Un FFD puede hablar con RFD's o con FFD's mientras que un RFD sólo puede hablar con un FFD. (Coleri Ergen, 2004)

Figura 6 - Modelos de topología de redes ZigBee



(Coleri Ergen, 2004)

La Figura 6 muestra los diferentes tipos de topologías que una red ZigBee soporta: Topología en estrella, *peer-to-peer* y árbol de *clusters*. (Coleri Ergen, 2004)

En la topología de estrella la comunicación se establece entre todos los dispositivos y un único controlador central, el coordinador PAN. Usualmente la alimentación del coordinador PAN es fija y estable mientras que la alimentación de todos los demás dispositivos es provista por baterías. Las aplicaciones que se benefician de esta topología van desde la automatización de una casa, periféricos de una computadora hasta juguetes y juegos. (Coleri Ergen, 2004)

En contraste con la topología de estrella, en la topología *peer-to-peer* cualquier dispositivo se puede comunicar con cualquier otro siempre y cuando se encuentren en rango. Una red de este tipo puede ser *ad hoc*, auto-organizativa y auto-curativa. Aplicaciones como control industrial y monitoreo, redes inalámbricas de sensores e incluso rastreo de inventario se pueden beneficiar de una topología así. (Coleri Ergen, 2004)

Las redes tipo *cluster-tree* son un caso especial de una red *peer-to-peer* en la que la mayoría de dispositivos son FFD y los RFD pueden ser conectados como nodos hoja al final de cada rama. Cualquiera de los FFD puede actuar como coordinador y proveer

servicios de sincronización para otros dispositivos y coordinadores, sin embargo, sólo uno de ellos puede ser el coordinador PAN. (Coleri Ergen, 2004)

1. XBee Serie 1. Los XBee's Serie 1 son módulos de radiofrecuencia diseñados sobre los estándares de la IEEE 802.15.4 apoyando sus necesidades de bajo costo y consumo de potencia bajo en una red de sensores inalámbrica. Operan dentro de la banda de frecuencia ISM 2.4GHz. (Digi International Inc., 2011)

Figura 7 - XBee con antena expuesta



(Digi International Inc., 2011)

En un ambiente urbano pueden llegar a tener un alcance de hasta 30 metros mientras que en un ambiente exterior hasta 90 metros. Transmiten con una potencia de 1mW y la sensibilidad del receptor es de -92dBm. (Digi International Inc., 2011)

Utiliza una técnica de reintentos y reconocimientos para confirmar que los datos han sido transmitidos correctamente. Pueden conectarse en una red punto-a-punto, punto-a-multipunto e incluso en una topología *peer-to-peer*. Una de sus más notables características es la capacidad de trabajar en *modo transparente*. En este modo los datos que ingresen al XBee vía la terminal de entrada serán transmitidos a través de radio-frecuencia. Asimismo, los datos que un XBee reciba estando en *modo transparente* serán comunicados vía la terminal de salida. Es posible también programar el módulo para que utilice las entradas digitales o las líneas de convertidores analógicos-digitales y enviar esta información. (Digi International Inc., 2011)

E. Micro-controladores

Un micro-controlador se puede describir como una computadora-sobre-un-chip o, si lo prefieren, una computadora de chip único. (Axelson, 1997)

Un micro-controlador es similar a un micro-procesador en una computadora personal. Tanto los micro-controladores como los micro-procesadores contienen una unidad de procesamiento central (CPU, por sus siglas en inglés). Esta unidad se encarga de ejecutar instrucciones de operaciones de lógica básica, matemática y movimiento de datos. Además del CPU, para completar la definición de *computadora*, hace falta agregar una memoria que permita guardar datos y programas y por último periféricos de entrada y salida. (Axelson, 1997)

Micro sugiere que es un dispositivo pequeño y *controlador* da la idea de que es capaz de controlar objetos, procesos o eventos. Otro término para describir a un micro-controlador es *controlador-embebido*, ya que muchas veces tanto el micro-controlador como sus sistemas de soporte están contruidos sobre el dispositivo que controlan. (Axelson, 1997)

1. Arduino UNO. Arduino es una plataforma de desarrollo electrónico *open-source* basada en *hardware* y *software* flexible y de fácil uso. Está destinada a artistas, diseñadores, aficionados y cualquiera que esté interesado en la creación de objetos y/o ambientes interactivos. Puede manejar datos de entrada de una variedad de *switches* o sensores y controlar una variedad de luces, motores y otros tipos de salidas físicas. (Arduino, 2011)

El Arduino UNO es una tarjeta basada en el micro-controlador ATmega328P de la compañía Atmel. Tiene 14 pines de entrada/salida digitales. Seis de ellos pueden ser utilizados como salidas de ancho de pulso modulable (PWM, por sus siglas en inglés). Cuenta con seis entradas digitales, un cristal oscilador de 16MHz, una conexión USB, un conector de alimentación, pines de conexión para ICSP, un botón de *reset*, y un chip ATmega8U2 programado como un convertidor USB a serial. (Arduino, 2011)

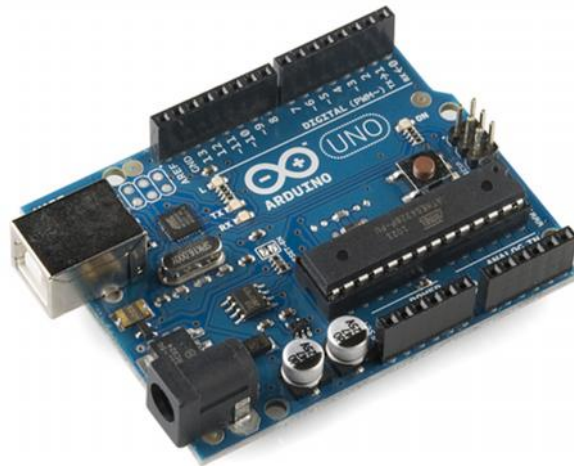
Figura 8 - Arduino UNO

Imagen tomada de: <http://www.sparkfun.com/products/9950>

El Arduino UNO puede ser alimentado tanto con el cable USB conectado a una computadora como por un transformador AC-DC que provea 7-12V. También se puede conectar una fuente de alimentación externa en los pines marcados como V_{in} y Gnd . (Arduino, 2011)

El micro-controlador ATmega328P cuenta con 32kB de memoria *flash*. También cuenta con 2kB de memoria SRAM y 1kB de memoria EEPROM. (Arduino, 2011)

Cada una de las terminales de entradas/salidas puede alimentar o recibir un máximo de 40mA. Todas operan a 5V y están conectadas a una resistencia *pull-up* de 20-50k Ω . Los pines 0 y 1 están conectados al módulo UART del ATmega328P y al chip ATmega8U2 como RX y TX, respectivamente, para poder comunicarse de manera serial con la computadora o con cualquier otro dispositivo que cuente con un módulo UART. Los pines 2 y 3 pueden ser configurados con interrupciones externas tanto por cambio de valor o por un cambio en la señal. Los pines 3, 5, 6, 9, 10 y 11 pueden ser utilizados como un PWM de 8-bits. Los pines 10, 11, 12 y 13 se pueden utilizar con el módulo SPI y los pines A4 y A5 con el módulo I²C del micro-controlador. (Arduino, 2011)

2. LilyPad Arduino. El LilyPad Arduino es una tarjeta de desarrollo que implementa un micro-controlador ATmega328P y está destinado a aplicaciones textiles. Tanto el LilyPad como diferentes dispositivos de alimentación, sensores y actuadores pueden ser cosidos a la ropa y conectados utilizando hilo conductor. (Arduino, 2011)

Su voltaje de alimentación se encuentra entre 2.7 y 5.5V. Cuenta con 14 pines de entrada/salida digitales (de los cuales seis pueden proveer salidas PWM) y seis pines de entradas analógicas. Cada terminal puede administrar hasta 40mA tanto de salida como de entrada. (Arduino, 2011)

Figura 9 - LilyPad Arduino

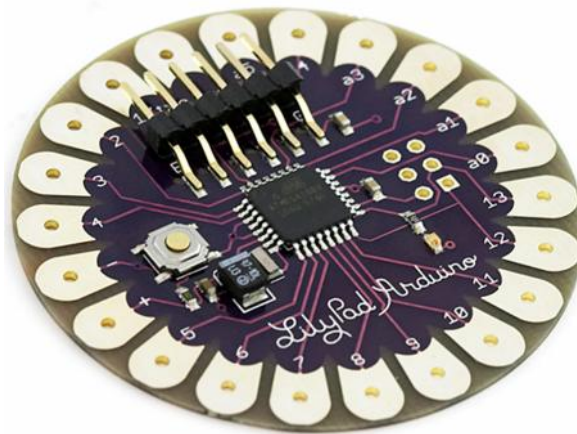


Imagen tomada de: <http://www.sparkfun.com/products/9266>

No cuenta con un chip ATmega8U2 como el Arduino UNO y es necesario utilizar un integrado adicional para poder programarlo y comunicarse vía serial a la computadora. (Arduino, 2011)

Figura 10 - LilyPad FTDI



Imagen tomada de: <http://www.sparkfun.com/products/10275>

El LilyPad FTDI es un módulo que permite comunicarse con la computadora a través de un chip integrado FTDI FT232RL USB-a-serial. Este mapea al micro-controlador como un puerto serial en la computadora y permite la comunicación entre ambos. (Arduino, 2011)

F. E-Textiles

Los textiles electrónicos (E-Textiles, por sus siglas en inglés) son tejidos que tienen componentes electrónicos e interconexiones cocidas en ellos. Poseen una flexibilidad física y un tamaño que no pueden ser alcanzados con las técnicas de manufactura de electrónicos actual. (Virginia Tech, 2009)

1. Hilo conductor. El hilo conductor es fabricado con cuatro fibras de acero inoxidable que permiten la transmisión tanto de señales como de potencia. El hecho de que el material sea acero inoxidable lo hace muy resistente a la oxidación, sin embargo también lo hace muy difícil de soldar. El grosor del hilo lo hace a veces impráctico al momento de coser, por lo que se puede colocar el hilo conductor por encima de la tela y cocer con hilo común sobre él para mantenerlo en su lugar. Ofrece una resistencia de aproximadamente 4 ohmios por pie lineal. (Sparkfun Electronics, 2010)

Figura 11 - Hilo conductor de cuatro fibras



Imagen tomada de <http://www.sparkfun.com/products/10120>

1. Pintura dimensional. La pintura dimensional es un tipo de pintura que se expande cuando seca creando así un patrón alzado sobre la tela. Se le conoce también como pintura *puff* o *puffy*. Originalmente diseñada para usarse sobre tela, soporta bastante bien el desgaste por lavado y secado; más aún si el lavado es a mano y la prenda se guarda plana. Sin embargo puede rajarse o pelarse con el uso continuo. La exposición a altas temperaturas tampoco es recomendable. (wiseGEEK)

Figura 12 - Pintura dimensional



Imagen tomada de: <http://www.hofcraft.com/plaid-dimensional-fashion-fabric-paint.html>

G. Interfaz gráfica

Una interfaz de usuario gráfica (GUI, por sus siglas en inglés) es una *interfaz humano-computador* (esto es, una manera para que humanos interactúen con computadores) que utiliza ventanas, íconos y menú y que puede ser manipulada por un *mouse* (y hasta cierto límite con un teclado también). (The Linux Information Project, 2004)

Las GUI's contrastan con las *interfaces de línea de comando* (CLI, por sus siglas en inglés) que utilizan únicamente texto y sólo se pueden acceder por medio del teclado. El ejemplo más familiar de una CLI es MS-DOS. Otro ejemplo más reciente es Linux, cuando solía existir únicamente en modo *consola*. (The Linux Information Project, 2004)

Una ventana es (usualmente) una porción rectangular de la pantalla de un monitor que puede desplegar contenido (esto es, programas, íconos, un archivo de texto o imagen) aparentemente independiente del resto de la pantalla. Cada ventana puede mostrar una aplicación diferente o cada una puede mostrar diferentes archivos (esto es

texto, imagen u hojas de cálculo) que han sido abiertas o creadas con una única aplicación. (The Linux Information Project, 2004)

Un ícono es una pequeña foto o símbolo en una GUI que representa un programa, un archivo, un directorio o un dispositivo. Los íconos se usan tanto en el *Escritorio* como dentro de aplicaciones. Pequeños rectángulos que representan archivos, folders que representan directorios, un bote de basura que representa un lugar en donde colocar archivos y directorios no deseados y botones en un navegador de la web son sólo algunos ejemplos. (The Linux Information Project, 2004)

Los comandos sobre una GUI los dictamina un *mouse*, *trackball* o *touchpad* que primero mueve un puntero en la pantalla hasta estar sobre un ícono, ítem de un menú o ventana de interés para poder *seleccionar* este objeto. Se pueden mover ventanas e íconos, por ejemplo, moviendo el mouse al mismo tiempo que se mantiene presionado el *click* izquierdo o se pueden abrir aplicaciones con un doble *click*. (The Linux Information Project, 2004)

1. Processing. Processing es un lenguaje de programación *open-source* y un ambiente para personas que quieren crear imágenes, animaciones e interacciones. Inicialmente fue desarrollado para servir como un software para realizar borradores de programas más complicados y para enseñar los fundamentos de la programación por computadora dentro de un contexto visual. Ha evolucionado hasta llegar a convertirse en una herramienta para generar trabajos profesionales terminados. (Processing, 2001)

Processing es una herramienta libre y *open-source* a diferencia de otras herramientas de *software* cuyas licencias de uso son relativamente caras. Esto hace que esta sea una herramienta accesible para escuelas y estudiantes individuales. Su estado *open-source* fomenta la participación y colaboración de una comunidad lo cual es vital para el crecimiento de Processing. Hay contribuyentes que comparten programas, código y responden a preguntas en foros de discusión y construyen librerías para extender las posibilidades del *software*. (The Linux Information Project, 2004)

2. Robot Class. La clase *Robot* del lenguaje de programación *Java* se utiliza para generar eventos de entrada nativos del sistema con el propósito de realizar comprobaciones de automatización, demostraciones que se ejecutan solas y otras aplicaciones en las cuales se necesita el control sobre el *mouse* y el teclado. (Oracle, 2010)

IV. DESCRIPCIÓN DE LA FUNCIONALIDAD DEL GUANTE

Las acciones que pueden realizarse con el guante van en función de los sensores y botones montados sobre el mismo. Para mantener la simplicidad de uso sin sacrificar la cantidad de funciones a implementar, se tomó la decisión de montar cuatro botones tipo *push*, tres sensores flex y un acelerómetro.

Figura 13 - Sensores implementados sobre el guante



La Figura 13 muestra la ubicación de todos los componentes del guante: los sensores flex están colocados sobre los dedos pulgar, índice y medio. Los botones se encuentran sobre los dedos índice y medio y están dispuestos de esa manera para fácil acceso con el pulgar. El acelerómetro se encuentra sobre la mano entre los dedos índice

y pulgar. La ubicación del acelerómetro está diseñada para permitir el movimiento en dos ejes con mayor fluidez. La Figura 14 muestra las diferentes posiciones máximas de la mano.

Las conexiones entre los diferentes componentes fueron hechas con hilo conductor de cuatro capas en lugar de cable o alambre. Se cosieron directas entre los componentes manteniendo siempre cuidado de no permitir que dos hilos hicieran contacto. Para aislar el hilo se recubrió con pintura dimensional. El hilo ofrece una resistencia de 4 ohmios por pie lineal que para fines de esta aplicación es despreciable.

Figura 14 - Inclinationes máximas del guante: (A) Inclinatione hacia abajo (B) Inclinatione hacia la derecha (C) Inclinatione hacia arriba (D) Inclinatione hacia la izquierda



(A)



(B)



(C)



(D)

El guante tiene tres modos de operación:

1. Control del *mouse*
2. Ejecución de una acción
3. Modo presentación

El modo *Control del Mouse* es el que está activo por defecto. En este modo el guante controla el movimiento del mouse, *clicks* izquierdo y derecho y el *scroll*. El botón interno del dedo índice se utiliza para activar y desactivar el control del movimiento del mouse. Los botones externos de los dedos índice y medio controlan los *clicks* izquierdo y derecho, respectivamente. Para activar el *scroll* del mouse se debe dejar presionado el botón interno del dedo medio y flexionar el dedo índice para avanzar hacia arriba o abajo. Para realizar una función de *click&arrastre* (click&drag) se debe presionar el botón externo del dedo índice y luego mantenerlo presionado por aproximadamente 1.5 segundos. De esta manera se *fija* el *click* y es posible dejar de presionar el botón sin que en el espacio gráfico se *suelte* el objeto bajo el apuntador. Para *soltar* lo que se está arrastrando basta con hacer *click* una vez más. El botón derecho tiene una funcionalidad extra: si se mantiene presionado el botón y se flexiona el dedo índice se puede circular las opciones que el menú desplegado ofrezca (modo “menú derecho”). Al soltar el botón se seleccionará automáticamente la opción marcada.

Si se presiona y suelta el botón interno del dedo medio se entra en el modo de *Ejecución de una acción*. En este modo los botones y el acelerómetro quedan deshabilitados. Los sensores *flex* colocados en los tres dedos son los que dictaminan la acción a ejecutar.

Una vez presionado el botón el dispositivo espera aproximadamente un segundo antes de tomar los datos de los sensores flex. Los datos pueden ser tanto una posición fija como un movimiento de los dedos. Después de ejecutada la acción o si transcurren más de dos segundos sin actividad el dispositivo regresa al modo de operación por defecto. Una lista de acciones posibles de ejecutar se encuentra en la siguiente sección.

Si se deja presionado el botón interno del dedo índice por más de dos segundos se entra en el modo *Presentación*. En este modo el acelerómetro y los sensores flex quedan deshabilitados y los botones externos de los dedos índice y medio controlan el paso de diapositivas. Para regresar al modo de operación por defecto se presiona el botón interno del dedo índice por dos segundos.

La Tabla 1 contiene un resumen de las funciones de los botones, sensores flex y acelerómetro en el dispositivo:

Tabla 1 - Resumen de acciones de botones, sensores flex y acelerómetro

Función	Botones				Flex			Acelerómetro
	Índice externo	Índice interno	Medio externo	Medio interno	Flex índice	Flex medio	Flex pulgar	
Mouse On/Off	x	Presionar	x	x	x	x	x	x
Mover mouse en X	x	x	x	x	Control Vel	x	x	(+/-) X
Mover mouse en Y	x	x	x	x	x	Control Vel	x	(+/-) Y
Click izquierdo	Presionar	x	x	x	x	x	x	x
Click derecho	x	x	Presionar	x	x	x	x	x
Scroll	x	x	x	Mantener	Control Arriba/Abajo	x	x	x
Click & Drag	Click, Click&Mantener	x	x	x	x	x	x	x
Menú click derecho	x	x	Mantener/Soltar para seleccionar	x	Arriba/Abajo en Menú	x	x	x
Ejecutar acción	x	x	x	Presionar	x	x	x	x
Modo presentación On/Off	x	Mantener por 2 seg	x	x	x	x	x	x

x = No importa

V. MOVIMIENTOS DE LA MANO ÚTILES APLICABLES EN UN ESPACIO GRÁFICO

Uno de los objetivos específicos de este trabajo es el desarrollo de una lista de movimientos naturales de la mano que puedan ser medidos por diferentes sensores. Estos datos luego serán procesados e interpretados de manera que una computadora pueda utilizarlos en un espacio gráfico. Es importante que los movimientos sean lo más naturales posibles para evitar tanto problemas de salud como de implementación.

Se investigó qué *atajos* del teclado existen en el sistema operativo MS Windows 7 para poder implementarlos por medio del guante. La lista que se obtuvo es la siguiente:

Tabla 2 – Atajos del teclado y sus acciones en el sistema operativo MS Windows 7

Combinación de teclas	Acción ejecutada
Win + Home	Minimizar todas las ventanas menos la actual
Win + Barra Espaciadora	Todas las ventanas se vuelven transparentes
Win + Flecha arriba	Maximiza la ventana actual
Win + Flecha abajo	Minimiza la ventana actual
Win + Flecha derecha	Coloca la ventana del lado derecho de la pantalla
Win + Flecha izquierda	Coloca la ventana del lado izquierdo de la pantalla
Alt + Tab	Circula los programas abiertos
Ctrl + F4	Cerrar la ventana actual
Win + '+'	Acercamiento de pantalla
Win + '-'	Alejamiento de pantalla
Shift + Win + arriba	Maximiza la ventana actual verticalmente

Cont. Tabla 2 - Atajos del teclado y sus acciones en el sistema operativo MS Windows 7

Combinación de teclas	Acción ejecutada
Shift + Win + derecha	Mueve la ventana actual al monitor de la derecha
Shift + Win + izquierda	Mueve la ventana actual al monitor de la izquierda
Win + número (1-9)	Ejecuta la aplicación colocada en ese espacio en la barra de tareas
Shift + Win + número (1-9)	Ejecuta una nueva instancia de la aplicación colocada en ese espacio
Ctrl + Win + número (1-9)	Circula a través de las ventanas abiertas de la aplicación en ese espacio

Datos de (lifehacker.com, 2009)

Es importante notar que las combinaciones de teclas presentadas en la Tabla 2 no necesariamente tienen el mismo efecto en otros sistemas operativos. Por lo tanto, este dispositivo se limita al control del *mouse* y teclado en el sistema operativo MS Windows 7.

Existen dos tipos de acciones posibles de ejecutar con la mano: la primera se basa en la posición inicial y final de los dedos y la segunda únicamente en la posición inicial. Se definió una acción con el guante para algunas de las funciones de los atajos arriba descritos. No se implementaron todas las funciones ya que no todas son eficientes de realizar con un movimiento de la mano.

Tabla 3 - Movimientos de la mano asignados a acciones de la computadora

Movimiento de la mano	Combinación de teclas	Acción ejecutada
<p>Posición inicial: los tres dedos flexionados completamente</p> <p>Posición final: los tres dedos extendidos completamente</p>	Win + '+'	Acercamiento de pantalla
<p>Posición inicial: tres dedos extendidos completamente</p> <p>Posición final: tres dedos flexionados completamente</p>	Win + '-'	Alejamiento de pantalla
<p>Posición inicial: tres dedos flexionados completamente</p> <p>Posición final: índice y medio extendido completamente; pulgar flexionado</p>	Win + Flecha arriba	Maximiza la ventana actual
<p>Posición inicial: índice y medio extendidos; pulgar flexionados</p> <p>Posición final: tres dedos flexionados</p>	Win + Flecha abajo	Minimiza la ventana actual

Cont. Tabla 3 – Movimientos de la mano asignados a acciones de la computadora

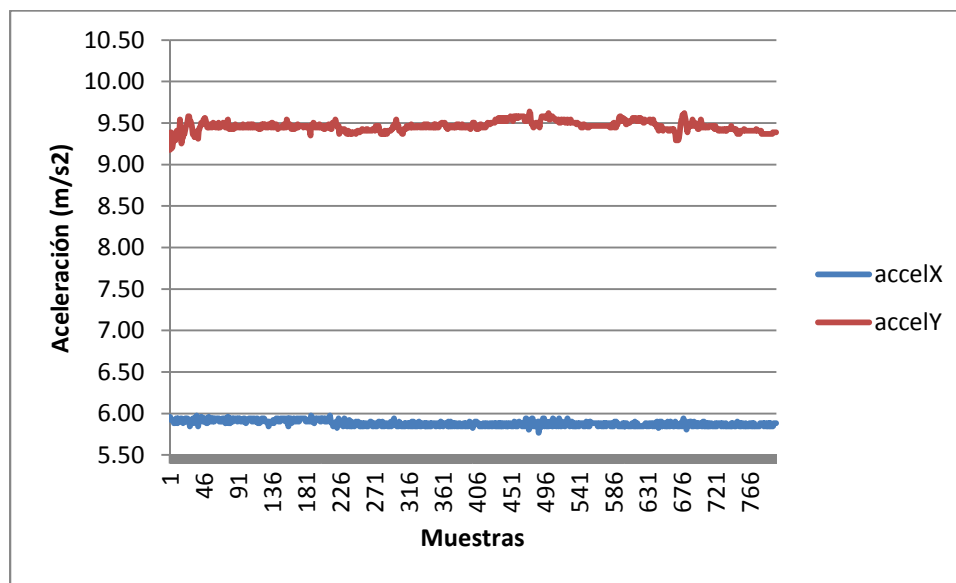
Movimiento de la mano	Combinación de teclas	Acción ejecutada
<p>Posición inicial: pulgar y medio flexionados, índice a la mitad. Con el movimiento del índice se cambia de ventana</p> <p>Posición final: pulgar y medio flexionados, índice extendido</p>	Alt + TAB	Circula las ventanas abiertas
<p>Posición inicial: tres dedos flexionados</p> <p>Posición final: pulgar extendido</p>	Ctrl + F4	Cierra la ventana actual

VI. MEDICIÓN DE INCLINACIÓN

Existen diferentes metodologías para medir el movimiento de una mano. Es posible medir la posición con respecto a una referencia, el desplazamiento, la velocidad, la velocidad angular, la aceleración, la inclinación con respecto de un plano, etc. Cada uno de estos datos requiere de diferentes técnicas y la utilización de diferentes sensores.

Se midió la aceleración de la mano mientras se realizaban movimientos usuales con un mouse.

Figura 15 – Aceleración de la mano al realizar movimientos usuales con un mouse



La Figura 15 muestra que hay mucho ruido en las mediciones debido a la vibración de la mano. Además, la mayoría de movimientos que se hicieron se reflejan en la grafica con una velocidad relativamente constante por lo que resulta muy impráctico utilizar estas mediciones como fuente de datos. Por lo tanto se optó por medir la inclinación de la mano con respecto al campo gravitatorio de la tierra.

El acelerómetro que se escogió es el BMA180 de Bosch (ver Figura 16). Es un acelerómetro digital tri-axial con sensor de temperatura integrado. Utiliza un convertidor analógico-digital (ADC, por sus siglas en inglés) con una resolución de 12 ó 14 bits. Puede comunicarse tanto por un bus SPI de cuatro cables como un bus I²C y cuenta también con un pin de interrupción. Tiene la opción de medir $\pm 1g$, $\pm 1.5g$, $\pm 2g$,

$\pm 3g$, $\pm 4g$, $\pm 8g$, $\pm 16g$ y un rango de temperatura de -40°C - $+85^{\circ}\text{C}$, además de contar con varios filtros digitales y modos de operación. La potencia eléctrica consumida es de aproximadamente $650\mu\text{A}$ incluso cuando se encuentra realizando una operación de 14 bits. Su voltaje de alimentación es de: $+1.62\text{V} - 3.6\text{V}$ para la terminal VDD y $+1.2\text{V} - 3.6\text{V}$ para VDDIO. Cuenta con una memoria tipo EEPROM de 256 bytes para almacenar datos calibración y datos del consumidor. (Bosch Sensortec, 2010)

La configuración que se utilizó es la que viene por defecto de la fábrica: el rango de medición de gravedad de $\pm 2g$ junto con un filtro pasa-bajas digital tipo *Butterworth* de 150Hz. Además, el modo de operación es de bajo ruido; lo que implica el máximo consumo de corriente ($650\mu\text{A}$). El ADC está configurado para funcionar a 14 bits.

Figura 16 - Implementación del acelerómetro BMA180. (A) Vista superior (B) Vista posterior

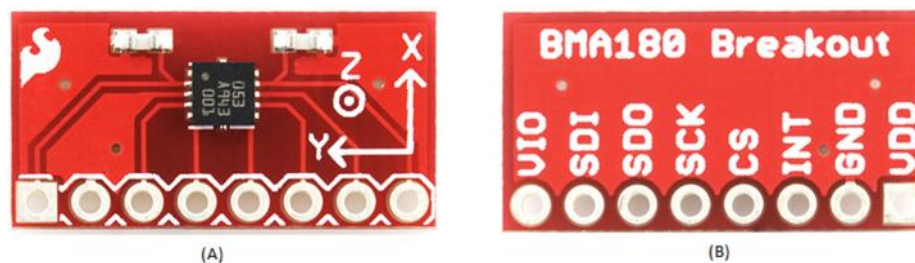


Imagen tomada de: <http://www.sparkfun.com/products/9723>

La Figura 16 muestra la implementación del acelerómetro sobre una pequeña placa (Sparkfun Electronics, 2010). El acelerómetro cuenta con dos capacitores conectados a las alimentaciones VDD y VDDIO para evitar ruido en la señal. En el reverso se pueden apreciar las entradas y salidas del sensor.

Para obtener los datos medidos del sensor es necesario comunicarse con él vía SPI o I^2C . Se utilizó el bus SPI de 4 cables para hablar con el sensor. Se implementó una red *peer-to-peer* de únicamente un maestro y un esclavo, micro-controlador y sensor respectivamente. Las mediciones del sensor se encuentran almacenadas en los registros del mismo. Debido a que la conversión es de 14 bits se necesitan dos registros para almacenar la información. Sin embargo, el sistema no requiere de una resolución tan alta, por lo que se utilizaron únicamente los bits más significativos de la conversión. Esta

información se encuentra en los registros 03h y 05h para los ejes X y Y, respectivamente. (Bosch Sensortec, 2010)

El sensor debe ser alimentado con +3.6V máximo y el micro-controlador conectado al acelerómetro funciona a un diferente potencial (+5v). Por esta razón, fue necesario ajustar los voltajes del micro-controlador utilizando dos tipos de circuitos convertidores.

Figura 17 - Convertidor de niveles lógicos (A) Visto superior (B) Vista posterior

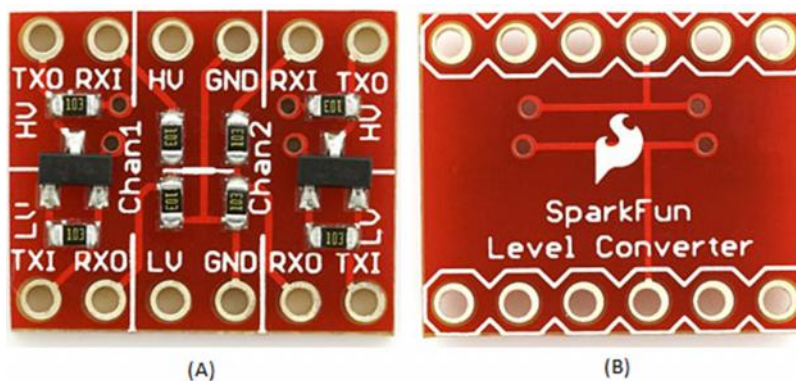


Imagen tomada de: <http://www.sparkfun.com/products/8745>

La Figura 17A muestra la parte superior del convertidor lógico mientras que la Figura 17B la parte posterior.

Figura 18 - Circuito de un canal del convertidor lógico; (A) Amplificador de señales con un FET; (B) Divisor de voltaje

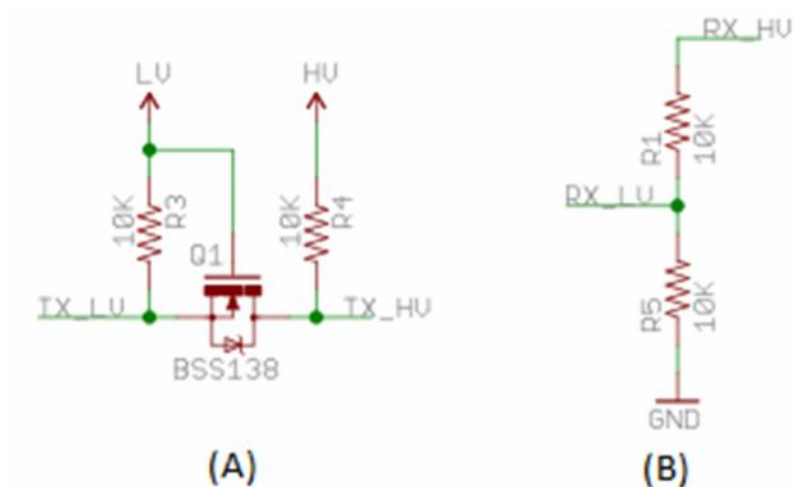
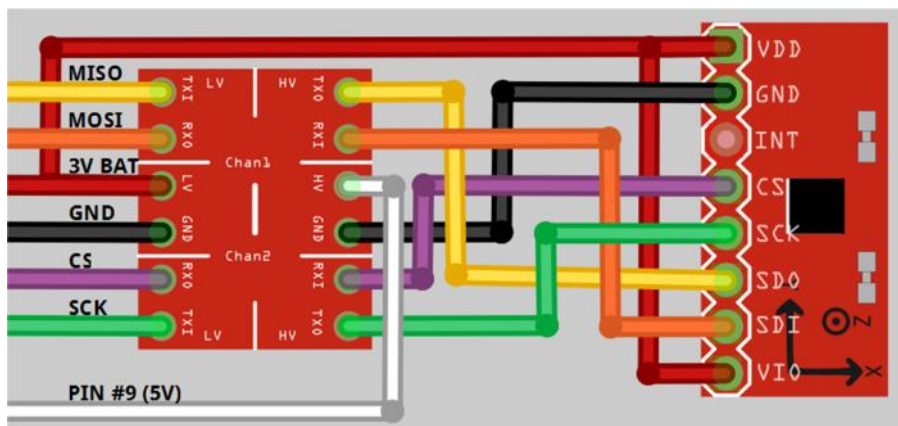


Imagen tomada de: <http://www.sparkfun.com/datasheets/BreakoutBoards/Level-Converter-v10.pdf>

El convertidor lógico cuenta con dos canales idénticos para realizar las conversiones. La Figura 18A muestra un circuito amplificador utilizando un transistor tipo FET canal N y la Figura 18B muestra un divisor de voltaje. El transistor utilizado es un BSS138 especialmente diseñado para convertir señales lógicas (Fairchild Semiconductor, 2005). Conectado a él se encuentran las señales *LV* y *HV*. Estas son la alimentación del circuito. El FET amplifica las señales del voltaje bajo *LV* al voltaje alto *HV* y atenúa las señales de *HV* a *LV*. El divisor de voltaje únicamente atenúa las señales.

Figura 19 - Conexión entre el convertidor lógico y el acelerómetro



La Figura 19 muestra la conexión entre el convertidor lógico y el acelerómetro. Es importante notar que las líneas de comunicación SPI y el voltaje alto (*HV*) provienen de la plataforma de desarrollo mientras que el voltaje bajo (*LV*) y la tierra provienen de una batería CR2032 de 3V.

VII. MEDICIÓN DE FLEXIÓN DE LOS DEDOS Y SEÑALES DIGITALES

Los sensores flex que se encuentran sobre los dedos pulgar, índice y medio se utilizan para medir el ángulo de flexión de cada dedo.

Figura 20 - Sensor flex

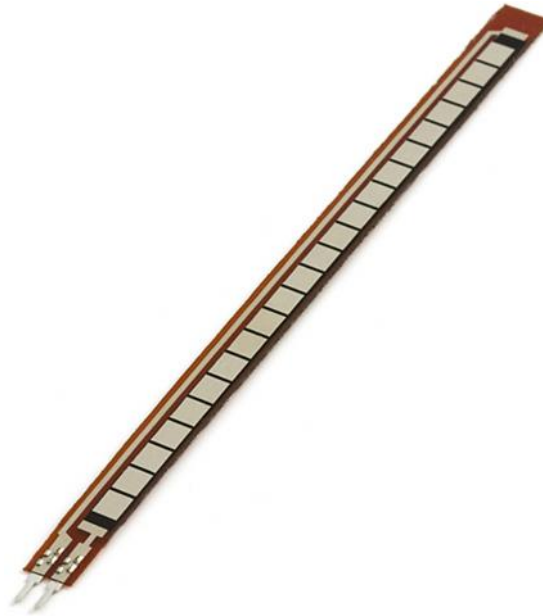


Imagen tomada de: <http://www.sparkfun.com/products/8606>

Figura 21 - PCB para aplicaciones textiles



Imagen tomada de: <http://www.sparkfun.com/products/10071>

Las Figuras 20 y 21 muestran un sensor flex y un PCB diseñado para aplicaciones textiles respectivamente. El sensor flex tiene patas para soldarlo a una superficie, pero es muy difícil montarlo sobre tela. Se utilizó el PCB especializado para solucionar este problema. Se soldó el sensor flex al PCB y se procedió a coser el PCB al guante. La Figura 22 muestra el sensor flex soldado.

Figura 22 - Sensor flex soldado sobre PCB para aplicaciones textiles



Imagen tomada de: <http://www.sparkfun.com/products/10071>

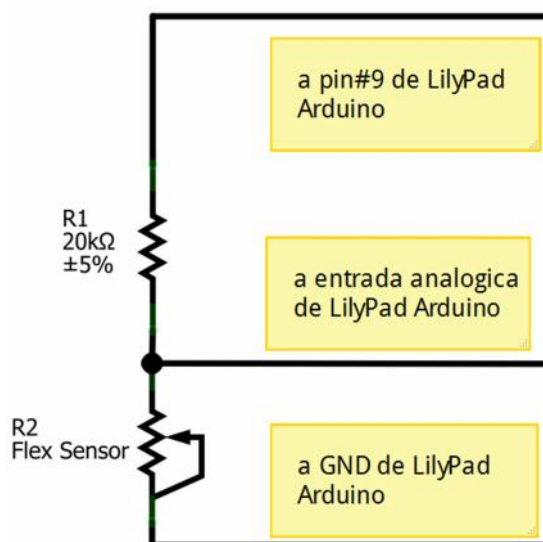
Se midió la resistencia de los sensores flex una vez montados sobre la mano. Los resultados se encuentran en la Tabla 4.

Tabla 4 - Rangos de resistencias de los sensores flex montados

Dedo	Flexión	
	Completamente extendido	Completamente flexionado
Pulgar	~10k Ω	~25k Ω
Índice	~10k Ω	~30k Ω
Medio	~10k Ω	~30k Ω

El sensor flex se colocó como parte de un divisor de voltaje. La Figura 23 muestra el circuito resultante.

Figura 23 - Circuito divisor de voltaje para medir la flexión del dedo



La alimentación provista por el pin#9 del LilyPad Arduino es de +5V. La Tabla 5 muestra los resultados del divisor cuando el sensor flex cambia de resistencia en pasos de 1k Ω . Se seleccionó una resistencia $R_1 = 20k\Omega$ ya que con este valor el rango de voltajes obtenido es el más amplio. La ecuación del divisor de voltaje es:

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (3)$$

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \cdot 20$$

Sea $V_{in} = 5V$ y $R_1 = 20 \text{ k}\Omega$

Sea $V_{in} = 5V$ y $R_1 = 2$

$$V_{out} = (5V) \frac{R_{Flex}}{(20 \text{ k}\Omega) + R_{Flex}} \quad (4)$$

Tabla 5 - Valores de salida del divisor de voltaje

R_{Flex}	V_{out}
10k Ω	1.67V
11k Ω	1.77V
12k Ω	1.88V
13k Ω	1.97V
14k Ω	2.06V
15k Ω	2.14V
16k Ω	2.22V
17k Ω	2.30V
18k Ω	2.37V
19k Ω	2.44V
20k Ω	2.50V
21k Ω	2.56V
22k Ω	2.62V
23k Ω	2.67V
24k Ω	2.73V
25k Ω	2.78V
26k Ω	2.83V
27k Ω	2.87V
28k Ω	2.92V
29k Ω	2.96V
30k Ω	3.00V

El micro-controlador ATmega328P sobre el LilyPad Arduino registra la señal de voltaje de cada uno de los divisores de voltaje y lo mapea a un valor entre cero y cuatro. Este valor luego es transmitido a la interfaz gráfica para conocer la flexión del dedo. No

es necesario utilizar más de cinco posiciones de los dedos para poder realizar las funciones y acciones requeridas del dispositivo.

Tabla 6 - Catalogación de flexión de los dedos

Valor	Flexión
0	Dedo completamente extendido
1	Dedo parcialmente flexionado
2	Flexión media
3	Dedo parcialmente flexionado
4	Dedo completamente flexionado

Para registrar señales digitales en el dispositivo se utilizaron botones tipo *push* como los mostrados en la Figura 24.

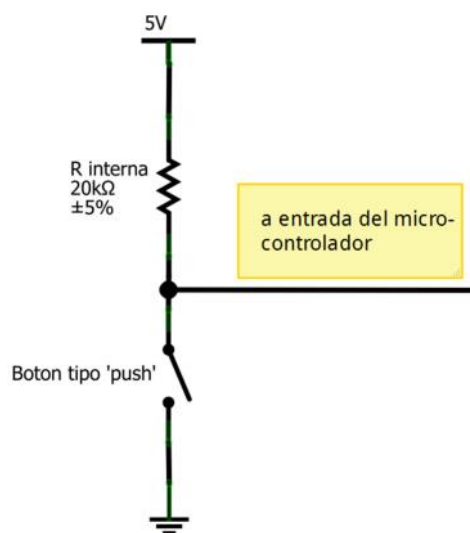
Figura 24 - Botón tipo push diseñado para aplicaciones textiles; (A) Vista superior; (B) Vista inferior



Imagen tomada de: <http://www.sparkfun.com/products/8776>

De acuerdo a la hoja de datos del micro-controlador ATmega328P, los pines de entrada deben estar conectados ya sea a voltaje o a tierra o neutro vía una resistencia *pull-up* o *pull-down*; incluso aquellos pines que no estén en función. (Atmel, 2009) Para poder registrar correctamente los valores de los botones *push* se activaron las resistencias *pull-up* internas del micro-controlador y los botones se conectaron al pin y a tierra. La Figura 25 ilustra el circuito.

Figura 25 - Conexión de los botones push utilizando la resistencia interna del micro-controlador

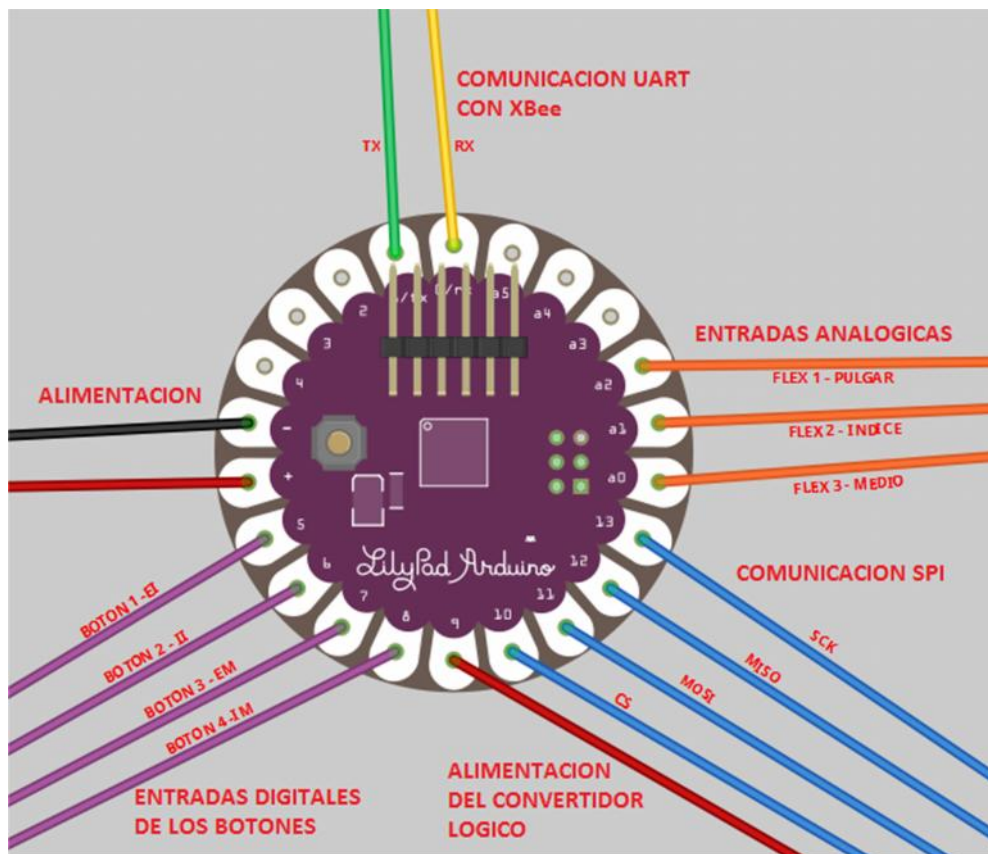


VIII. RECOLECCIÓN DE DATOS MEDIDOS Y COMUNICACIÓN INALÁMBRICA

Los botones *push*, los sensores flex y el acelerómetro están conectados a la plataforma de desarrollo LilyPad Arduino. Sobre esta plataforma se encuentra un micro-controlador ATmega328P encargado de recopilar los datos y luego enviarlos vía UART a un módulo XBee para transmisión inalámbrica.

La interconexión del LilyPad Arduino con los distintos sensores y actuadores se muestra en la Figura 26.

Figura 26 - Conexiones del LilyPad Arduino



La estructura mínima del código para el micro-controlador se muestra en la Figura 27.

Figura 27 - Ambiente de desarrollo integrado (IDE, por sus siglas en inglés) de Arduino



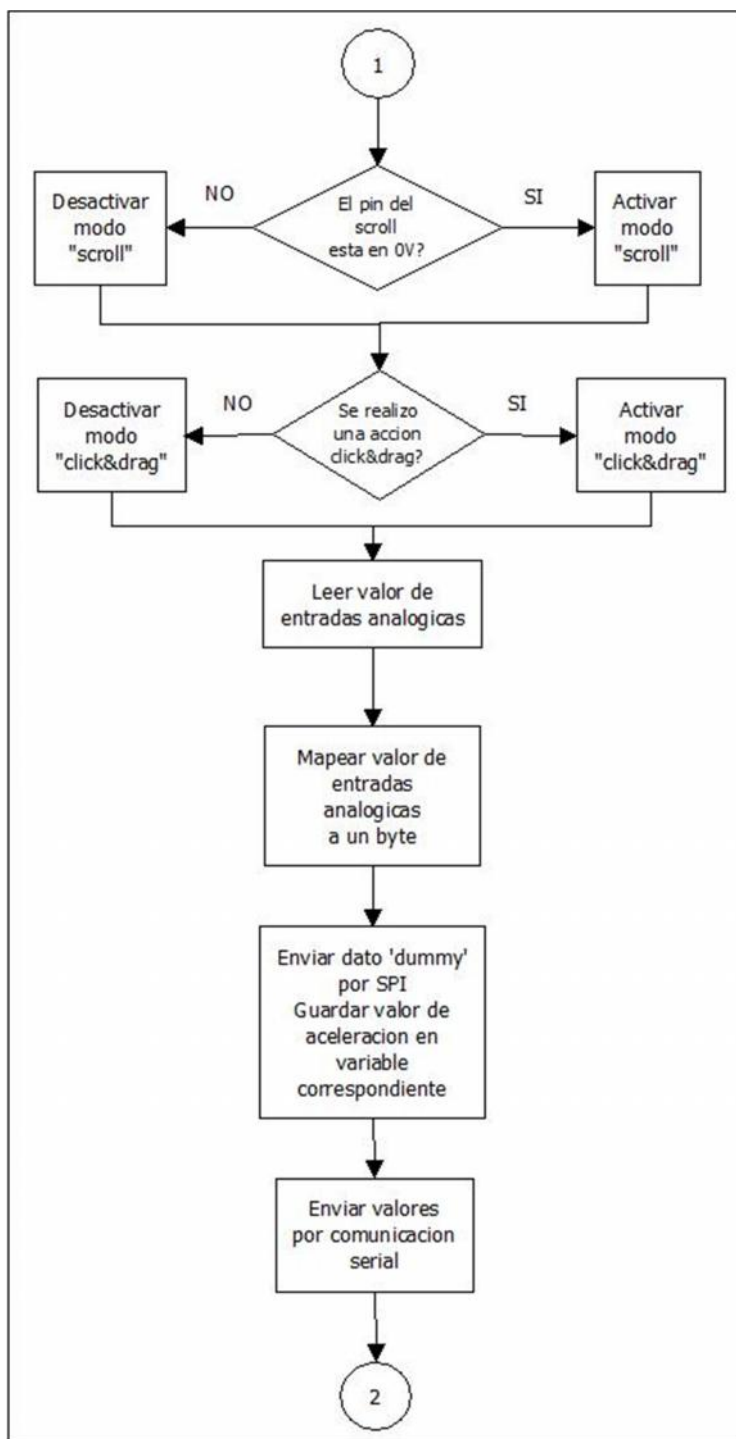
Es importante mencionar que un programa que no contenga las funciones *setup()* y *loop()* no se podrá compilar y, por lo tanto, tampoco se podrá quemar en el microcontrolador.

Previo a la definición de las funciones *setup()* y *loop()* es posible declarar las variables y constantes que se utilizarán en la programación. Se instanciaron variables para cada uno de los botones digitales (variables tipo *byte*), para las tres entradas analógicas (tipo *int*) y para los valores del acelerómetro (tipo *byte*). Las entradas analógicas regresan un valor entre 0-1023, pero es necesario enviar únicamente un valor entre 0-4. Para tal fin, se instanciaron tres variables más tipo *byte*.

La función *setup()* ejecuta una vez el código contenido en ella. Su función es configurar el sistema. Dentro de la función *setup()* se tiene que incluir:

- La inicialización de la librería *Serial* a un *baud-rate* apropiado
- La inicialización de la librería *SPI* con los parámetros: *MSBFIRST*, *SPI_MODE3* y *SPI_CLOCK_DIV8*
- La inicialización de los pines 9-13 como salidas
- La fijación del pin 9 en *alto* (es decir, a +5V)

Figura 29 - Diagrama de flujo de LilyPad Arduino (cont)



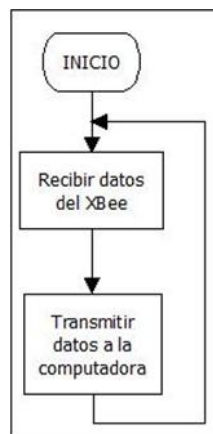
El LilyPad Arduino está conectado vía UART a un XBee Serie 1. Éste está configurado de manera que los datos que ingresan por la línea DI (ingreso de datos, por sus siglas en inglés) son transmitidos inmediatamente por una radiofrecuencia; este modo de

operación se conoce como “modo transparente”. El XBee conectado al LilyPad únicamente transmite datos; nunca recibe.

El XBee receptor se encuentra montado sobre un Arduino UNO. Está configurado para operar en “modo transparente” también, por lo que los datos que él reciba por medio de radiofrecuencias los comunica al micro-controlador inmediatamente. El Arduino UNO está conectado a la computadora vía un cable USB y esta conexión se realiza a través del UART del micro-controlador. Por lo tanto es necesario utilizar un UART implementado en *software* para la comunicación con el XBee. La librería *SoftwareSerial* permite la simulación de este UART en los pines #2 y #3. La desventaja de utilizar el UART *virtual* es que éste no puede interrumpir la ejecución del programa del micro-controlador. Por este motivo, cuando se utiliza la función de leer un dato la ejecución del programa se detiene hasta que se haya recibido. La cantidad de datos que el XBee receptor le envía al Arduino UNO es siempre la misma: cuatro *bytes* que representan el estado de los botones, tres *bytes* que representan la posición de los sensores flex, dos *bytes* que representan la inclinación/aceleración del acelerómetro, un *byte* para el modo presentación, un *byte* para el modo “click&drag”, un *byte* para el modo “menú derecho” y un *byte* para el modo “scroll”; un total de 13 *bytes*.

La función del Arduino UNO es únicamente comunicar los datos recibidos por el XBee y transmitirlos a la computadora.

Figura 30 - Diagrama de flujo del Arduino UNO

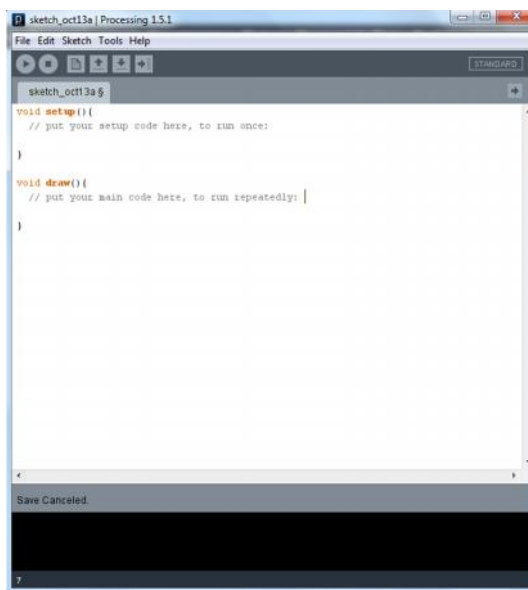


IX. PROCESAMIENTO DE DATOS Y MANEJO DEL AMBIENTE GRÁFICO

Los datos que el Arduino UNO envía a la computadora vía USB tienen que ser interpretados por alguna aplicación que interactúe con el sistema operativo. Esta aplicación tiene que ser capaz de interpretar los valores numéricos recibidos a través de un puerto serial e influir en el control del *mouse* y de varias funciones del teclado.

Para lograr este objetivo se utilizó el lenguaje de programación Processing que tiene la capacidad de recibir los datos enviados por el Arduino UNO e interpretarlos. Además, gracias a que está basado en Java, es posible incluir la librería *Robot* que permite influir en la posición y los botones del *mouse* y también simular pulsaciones de teclas.

Figura 31 - Ambiente de desarrollo integrado (IDE, por sus siglas en inglés) de Processing



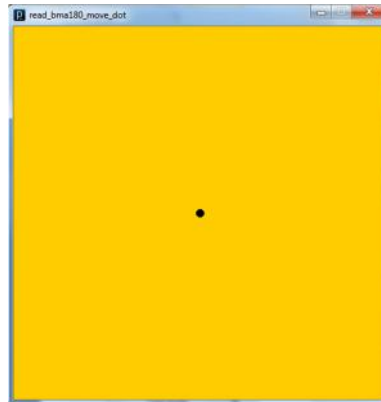
Similar a la programación de Arduino, hay dos funciones que se utilizan en cualquier aplicación: *setup()* y *draw()*.

La función *setup()* tiene exactamente la misma aplicación en Processing que en Arduino: define las propiedades iniciales del ambiente. Se cargan imágenes, selecciona el color de fondo e incluso el tamaño de la aplicación. Se ejecuta sólo una vez y antes de

la función *draw()* que ejecuta constantemente todas las instrucciones que se incluyan dentro de ella.

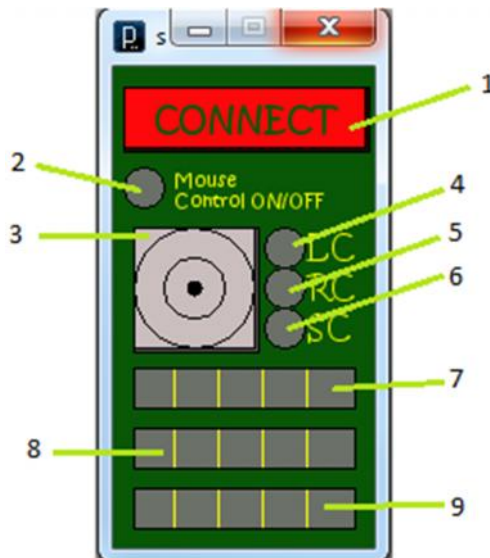
Para realizar pruebas se desarrolló una aplicación en la que un punto negro se mueve de acuerdo a la inclinación del acelerómetro (Figura 32).

Figura 32 - Aplicación de prueba



La aplicación final muestra indicadores para todos los sensores y también un botón para conectar la aplicación al guante.

Figura 33 - Aplicación final



La Figura 33 muestra la aplicación final. A continuación una explicación breve de cada parte de la misma:

1. Botón para conectar. Se torna verde si el guante está conectado.
2. Indicador del control del guante. Si el guante tiene el control del mouse se torna de color verde. Si el guante se encuentra en modo presentación se torna de color naranja.
3. Indicador de la inclinación del acelerómetro. El punto negro en el centro indica la inclinación del acelerómetro. Si se inclina la mano a la derecha el punto negro se moverá hacia la derecha.
4. Indicador del botón izquierdo del *mouse*. Se torna de color amarillo si se presiona el botón.
5. Indicador del botón derecho del *mouse*. Se torna de color rojo si se presiona el botón.
6. Indicador del *scroll* / ejecución de acción. Si se está en modo *ejecutar acción* el indicador se torna de color naranja. Si se está haciendo uso del *scroll* el indicador se torna de color verde. Si el *scroll* es hacia arriba sólo la mitad superior del indicador se torna verde. Lo inverso ocurre si es hacia abajo. Si se está en el centro el indicador completa se torna verde.
7. Indicador de flexión del dedo pulgar. La barra se llena desde la izquierda en las 5 posiciones predeterminadas. Su color es naranja.
8. Indicador de flexión del dedo índice. Misma barra de llenado que la descrita en el punto 7, pero de color verde.
9. Indicador de flexión del dedo medio. Misma barra de llenado que las dos anteriores, pero de color azul.

Figura 34 - Aplicación final con colores de muestra

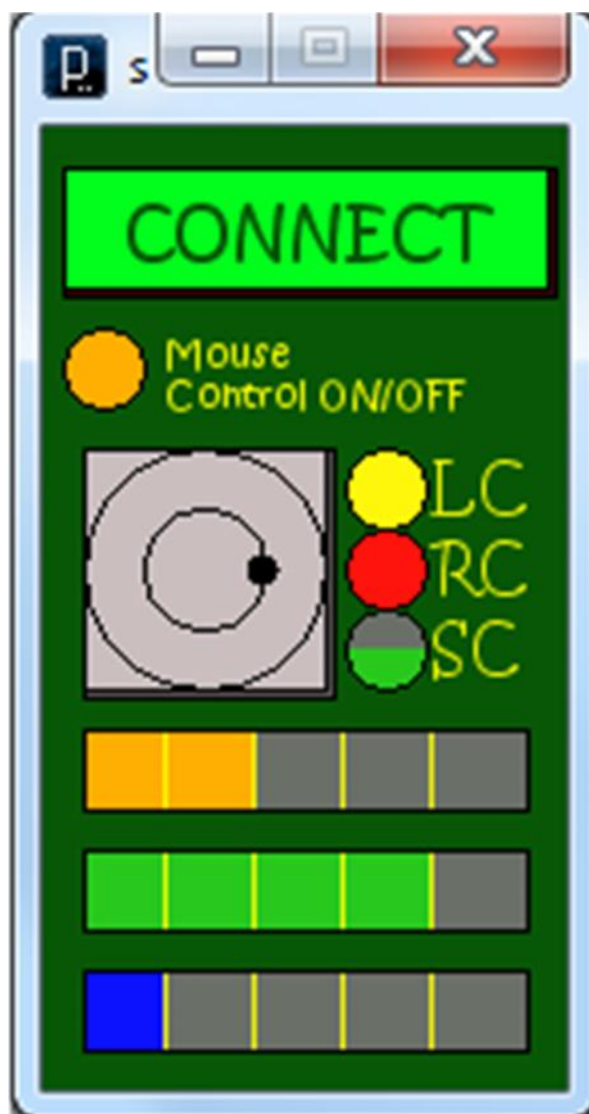


Figura 35 - Diagrama de flujo de la aplicación en la computadora (1)

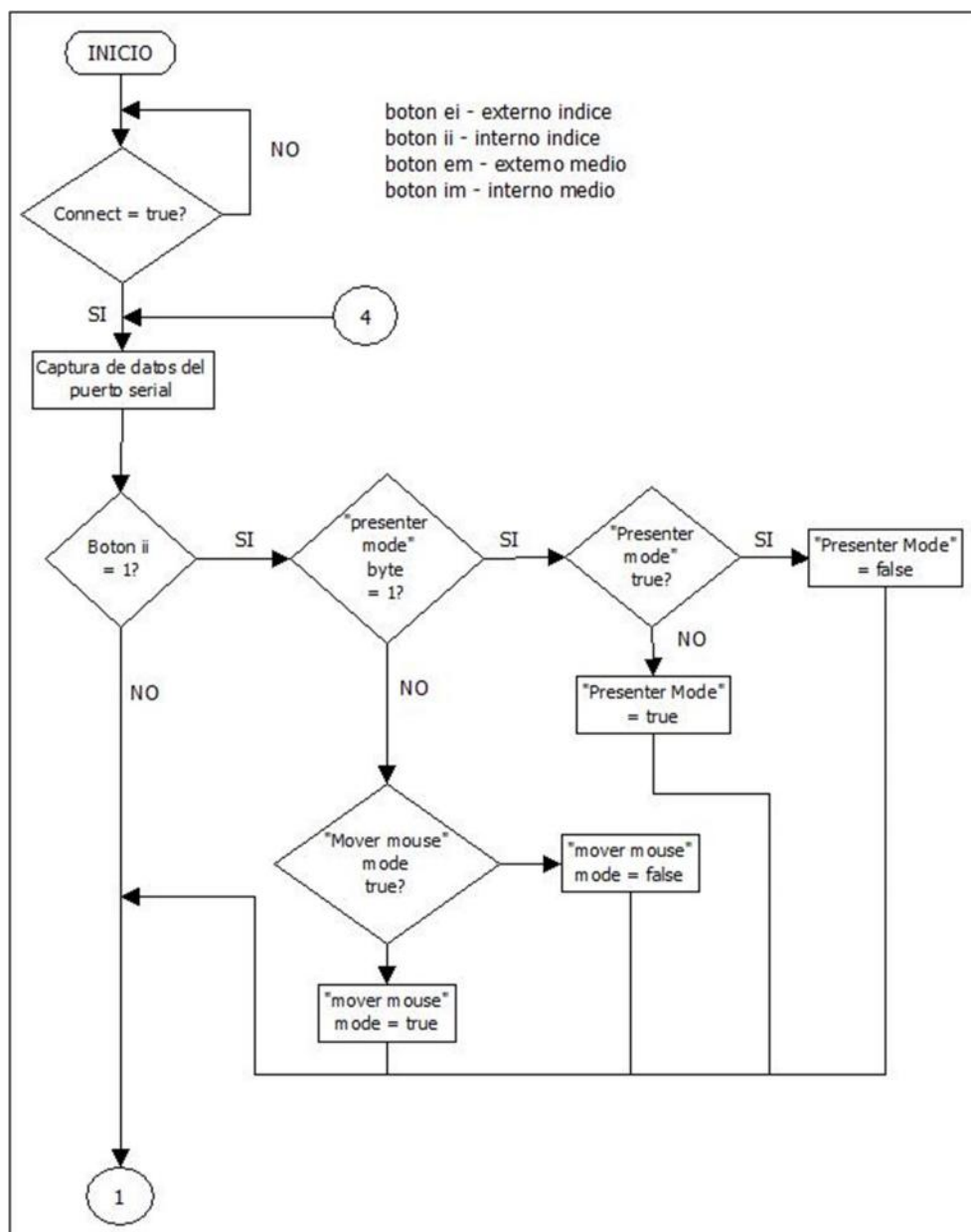


Figura 36 - Diagrama de flujo de la aplicación en la computadora (2)

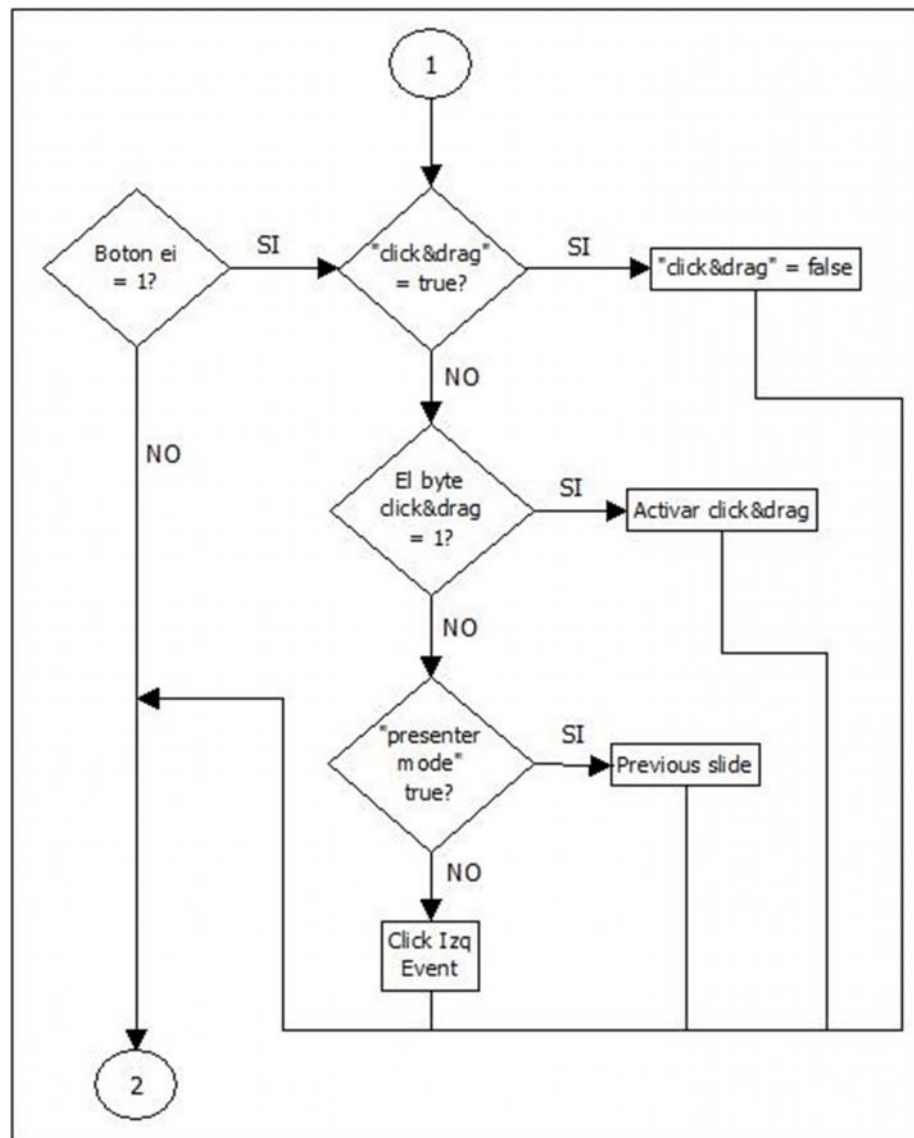


Figura 37 - Diagrama de flujo de la aplicación en la computadora (1)

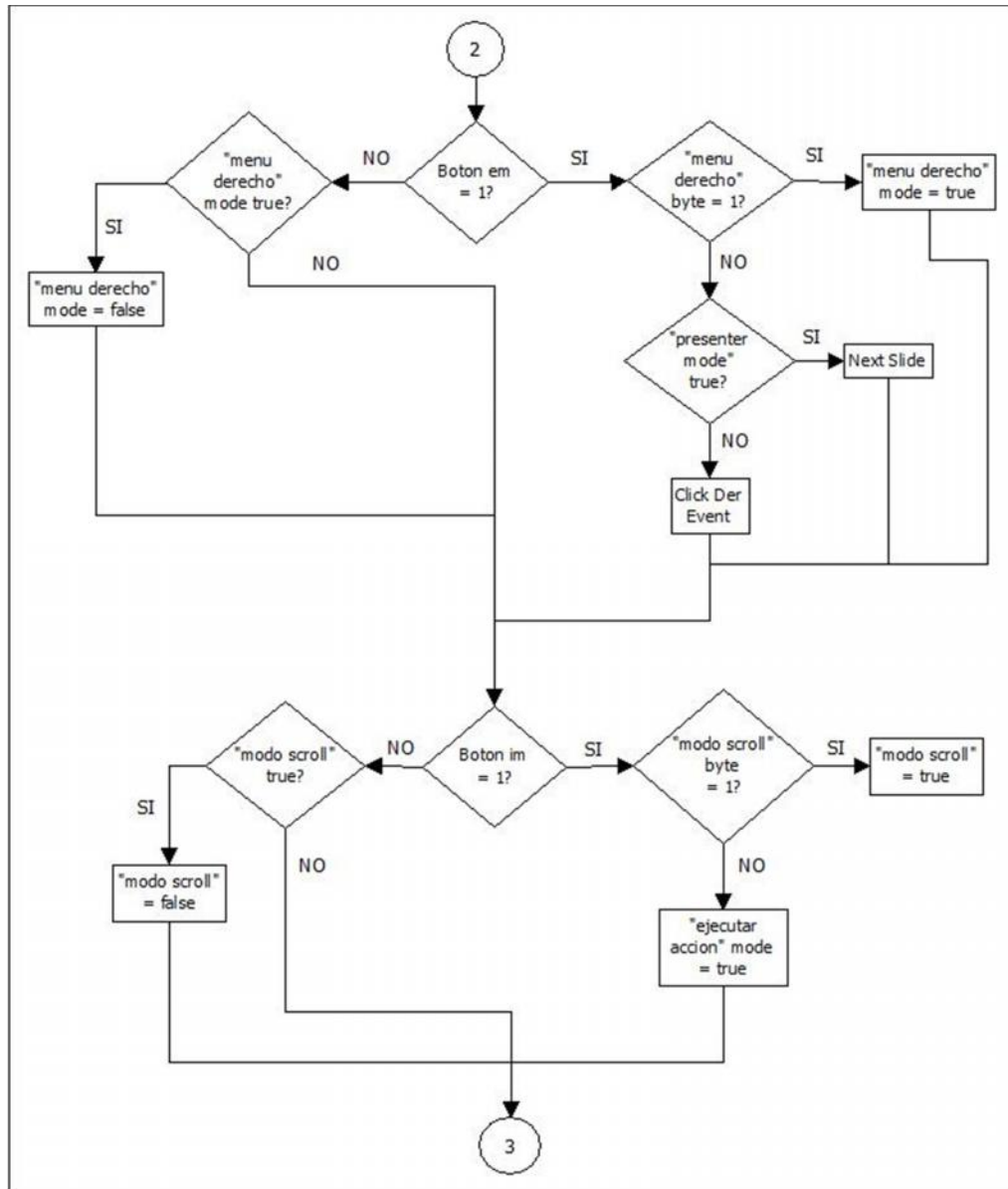
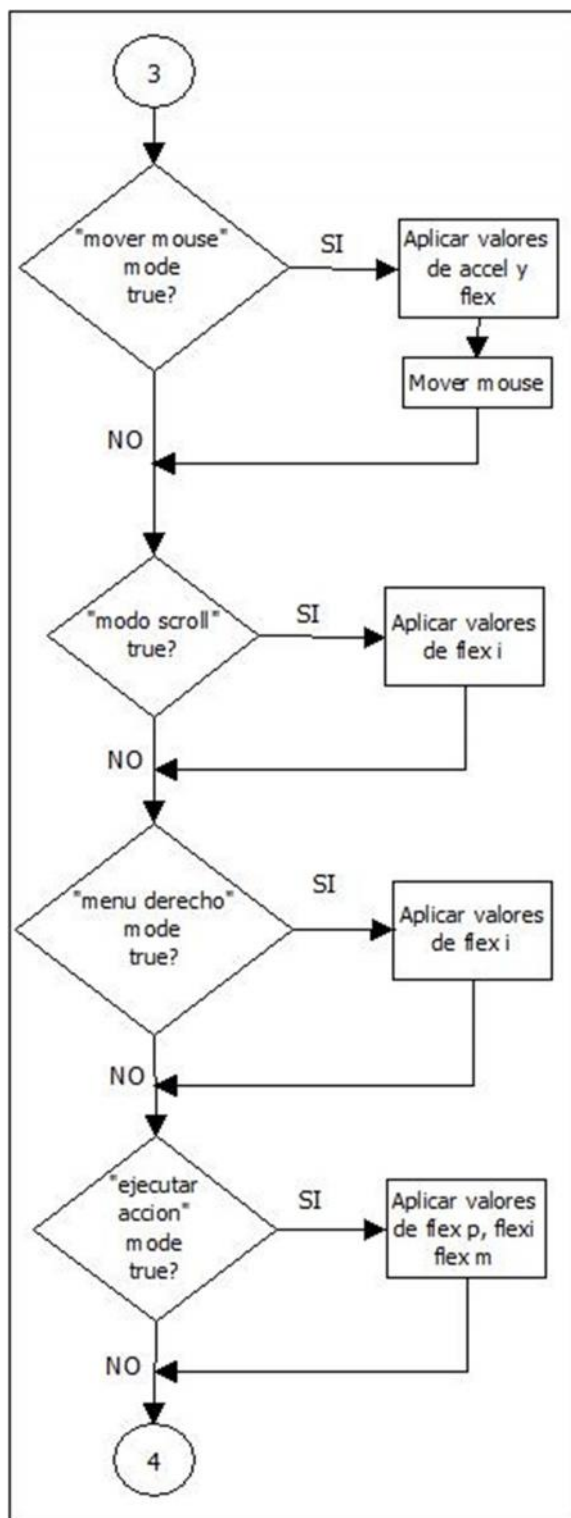


Figura 38 - Diagrama de flujo de la aplicación de la computadora (4)



X. CONCLUSIONES

1. Debido a la naturaleza de los movimientos de la mano relacionados con el control de un puntero, la aceleración que la mano alcanza es muy baja como para utilizar estas mediciones de una manera precisa por lo que la medición de inclinación con respecto al campo gravitatorio de la tierra es más apropiada
2. Un único sensor flex es capaz de medir de manera exacta una y sólo una articulación por lo que sólo es posible devolver una aproximación de la flexión de un dedo humano
3. El hilo conductor es práctico para realizar conexiones en tela debido a su manejabilidad, pero presenta grandes problemas de corto circuito si los componentes se encuentran muy cerca uno del otro o si la trama de hilo se trata de un bus de comunicación
4. La mejor locación para colocar un acelerómetro para realizar mediciones de inclinación de la mano es justo detrás del pulgar entre este y el dedo índice ya que permite un rango de inclinación bastante simétrico entre derecha e izquierda

XI. RECOMENDACIONES

1. Utilizando un adaptador USB – Serial (como por ejemplo un chip de FTDI) es posible comunicar un módulo XBee directamente con la computadora lo que elimina la necesidad de un micro-controlador intermediario
2. Es posible utilizar el módulo XBee en su modo API y conectarlo directamente a un acelerómetro con salidas analógicas en dos ejes, tres sensores flex y cuatro botones; de esta manera se elimina la necesidad del micro-controlador en el guante
3. Una medición más exacta de la flexión de un dedo se puede lograr utilizando dos sensores flex de dos pulgadas colocados uno en la articulación entre el hueso metacarpiano y la falange proximal y otro entre la falange proximal y la falange medial
4. Es posible medir el desplazamiento de la mano en base a la aceleración por medio de una doble integral, pero este procedimiento escapa el alcance del presente trabajo)

XII. BIBLIOGRAFÍA

- [1]. Andrejašic, M. (Marzo de 2008). *MEMS ACCELEROMETERS*. Recuperado el 8 de Octubre de 2011, de http://mafija.fmf.uni-lj.si/seminar/files/2007_2008/MEMS_accelerometers-koncna.pdf
- [2]. Arduino. (2011). *Arduino*. Recuperado el 14 de Octubre de 2011, de Arduino: <http://arduino.cc/>
- [3]. Atmel. (Octubre de 2009). Datasheet. *8-bit AVR Microcontroller with 4/8/16/32k bytes In-System Programmable Flash* .
- [4]. Axelson, J. (1997). *The Microcontroller idea book*. Madiso, Wisconsin: Lakeview Research.
- [5]. Boca Research. (s.f.). *Glossary*. Recuperado el 8 de Octubre de 2011, de [www.bocaresearch.com: http://www.bocaresearch.com/technologies/glossary.html](http://www.bocaresearch.com/technologies/glossary.html)
- [6]. Bosch Sensortec. (7 de Diciembre de 2010). Datasheet. *BMA180 - Digital, triaxial acceleration sensor* .
- [7]. bTrade. (2011). *Glossary*. Recuperado el 8 de Octubre de 2011, de [www.btrade.com: http://www.btrade.com/secure/resource-center/glossary](http://www.btrade.com/secure/resource-center/glossary)
- [8]. Coleri Ergen, S. (10 de Septiembre de 2004). ZigBee/IEEE 802.15.4 Summary. Berkeley, California, United States of America.
- [9]. Digi International Inc. (16 de Septiembre de 2011). Manual. *XBee/XBee-PRO RF Modules* .
- [10]. Fairchild Semiconductor. (Octubre de 2005). Datasheet. *BSS138 - N-Channel Logic Level Enhancement Mode Field Effect Transistor* .
- [11]. Fraden, J. (2003). *Handbook of Modern Sensors - Physics, Designs and Applications*. San Diego, California: Springer.
- [12]. Kalinsky, D., & Kalinsky, R. (1 de Febrero de 2002). *Introduction to Serial Peripheral Interface*. Recuperado el 9 de Octubre de 2011, de [www.eetimes.com: http://www.eetimes.com/discussion/beginner-s-corner/4023908/Introduction-to-Serial-Peripheral-Interface](http://www.eetimes.com/discussion/beginner-s-corner/4023908/Introduction-to-Serial-Peripheral-Interface)

- [13]. Lammert Bies. (Agosto de 2011). *Serial UART Information*. Recuperado el 9 de Octubre de 2011, de www.lammertbies.nl: <http://www.lammertbies.nl/comm/info/serial-uart.html>
- [14]. lifehacker.com. (26 de Octubre de 2009). *The Master List of Windows 7 Shortcuts*. Recuperado el 11 de Octubre de 2011, de www.lifehacker.com: <http://lifehacker.com/5390086/the-master-list-of-new-windows-7-shortcuts>
- [15]. McGraw-Hill Companies, Inc. (2003). *McGraw-Hill Science & Technology Dictionary*. McGraw-Hill.
- [16]. Microsoft. (2011). *TechNet*. Recuperado el 8 de Octubre de 2011, de <http://technet.microsoft.com>: <http://technet.microsoft.com/en-us/library/bb742416.aspx>
- [17]. Oracle. (2010). *Class Robot*. Recuperado el 15 de Octubre de 2011, de <http://download.oracle.com/>: [http://download.oracle.com/javase/1,5.0/docs/api/java/awt/Robot.html#getPixelColor\(int, int\)](http://download.oracle.com/javase/1,5.0/docs/api/java/awt/Robot.html#getPixelColor(int, int))
- [18]. Princeton University. (s.f.). *Communications Protocol*. Recuperado el 8 de Octubre de 2011, de <http://wordnet.princeton.edu/>: <http://wordnetweb.princeton.edu/perl/webwn?s=communications%20protocol>
- [19]. Processing. (2001). *Processing*. Recuperado el 15 de Octubre de 2011, de www.processing.org: <http://processing.org/>
- [20]. SensorWiki. (7 de Junio de 2011). *Flexion*. Recuperado el 8 de Octubre de 2011, de [SensorWiki.org](http://sensorwiki.org): <http://sensorwiki.org/doku.php/sensors/flexion>
- [21]. Sparkfun Electronics. (10 de Mayo de 2010). Recuperado el 11 de Octubre de 2011, de www.sparkfun.com: <http://www.sparkfun.com/products/9723>
- [22]. The Linux Information Project. (1 de Octubre de 2004). *GUI Definition*. Recuperado el 15 de Octubre de 2011, de www.linfo.org: <http://www.linfo.org/gui.html>
- [23]. theComputerFolks. (s.f.). *The Computer Folks Glossary*. Recuperado el 8 de Octubre de 2011, de www.thecomputerfolks.com: <http://www.thecomputerfolks.com/c.htm>
- [24]. Virginia Tech. (2009). <http://www.ccm.ece.vt.edu/etextiles/>. Recuperado el 13 de 03 de 2012, de <http://www.ccm.ece.vt.edu>: <http://www.ccm.ece.vt.edu/etextiles/>

MS Windows 7, Arduino UNO, Arduino LilyPad, Bosch Sensortec BMA180, Sparkfun y otros nombres mencionados en este trabajo son propiedad de sus compañías respectivas.

Todos los demás productos y nombres de corporaciones que aparecen en este documento pueden o no ser marcas registradas o poseer derechos de autor de sus respectivas compañías y sólo son usadas para identificación o explicación y para el beneficio del lector sin ninguna intención de infringir.