

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“Diseño de un traductor de gestos por medio de un guante  
electrónico sensitivo al movimiento”

**TESIS DE GRADO**

Previo la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

Presentada por:

David Ricardo Caicedo Fernández

Enrique Xavier Solórzano García

GUAYAQUIL – ECUADOR

Año: 2006

## AGRADECIMIENTO

En primer lugar a DIOS el cual por su gran misericordia, nos da salud, vida y nos dio la posibilidad de estudiar, a nuestro amigo, Ing. César Buri por la paciencia y el tiempo dedicado, a nuestros amigos los cuales nos daban consejos o palabras de apoyo, a nuestros padres que con sus sacrificios fueron pilares fundamentales para que pudiéramos hacer este trabajo.

“Encomienda al Señor tu camino. Y confía en El y El hará. Exhibirá tu justicia con La luz, y tu derecho como el mediodía.”,  
Salmos 37:5-6.

## TRIBUNAL DE GRADUACIÓN

---

Ing. Holger Cevallos  
SUB-DECANO DE LA FIEC  
PRESIDENTE

---

Ing. Germán Vargas L.  
DIRECTOR DE TESIS

---

Ing. Dennys Paillacho  
VOCAL

---

Ing. Sara Ríos O.  
VOCAL

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponden Exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

---

David Caicedo

---

Enrique Solórzano

## RESUMEN

El presente trabajo desarrolla el Diseño de un Traductor de Gestos basado en un Guante Electrónico Sensitivo al Movimiento, tratando de interpretar las señas más importantes empleadas por las personas privadas de hablar. Este proyecto esta enfocado a establecer un mecanismo de comunicación entre estas personas y el resto de la sociedad.

Debido a que este traductor de gestos esta basado en un guante electrónico sensitivo al movimiento, la primera parte de este trabajo se centrará en la adquisición de datos sobre la posición de los dedos.

Estos datos serán enviados a un microprocesador para que las señales sean comparadas, y decodificadas en lenguaje escrito o hablado.

# ÍNDICE GENERAL

RESUMEN

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

CAPÍTULO 1

1. ADQUISICIÓN DE DATOS

- 1.1. Fundamentos de la adquisición de datos
- 1.2. Sensores piezoeléctricos, de aceleración e inclinación
- 1.3. Multiplexado y encapsulado de la información a transmitir
  - 1.3.1.1. Multiplexado de señales
  - 1.3.1.2. Convertidor Analógico – Digital

1.3.1.3. Encapsulado de información

1.4. Transmisión y recepción de información

## CAPÍTULO 2

### 2. DISEÑO DEL TRADUCTOR

2.1. Construcción del guante electrónico

2.1.1.1. Parámetros de construcción

2.1.1.2. Funcionamiento de la unidad de control

2.2. Decodificador de los paquetes recibidos

2.3. Base de datos de las señas más usadas.

### 3. INTERFASES

3.1. Interfases

3.1.1. Presentación de datos

3.1.2. Manual de usuario

3.2. Resultados obtenidos

## CONCLUSIONES Y RECOMENDACIONES

## APÉNDICES

## BIBLIOGRAFÍA

## ÍNDICE DE FIGURAS

Figura 1.1 Etapas de la adquisición de datos

Figura 1.2 Diagrama circuital de contactos metálicos

Figura 1.3 Diagrama de conexión de los sensores de flexibilidad

Figura 1.4 Voltaje de salida vs. Grado de flexión del sensor

Figura 1.5 Diagrama de conexión del acelerómetro LIS3L02AS

Figura 1.6 Diagrama de conexión del sensor de inclinación Z2205

Figura 1.7 Código convolucional

Figura 1.8 Diagrama de conexión del conector serial DB9

Figura 1.9 Diagrama de conexión del MAX232CP

Figura 2.1 Posicionamiento de dispositivos en el guante

Figura 2.2 Diseño de la unidad de alimentación

Figura 2.3 Diagrama simplificado de la unidad de control del PIC

Figura 2.4 Diagrama ASM de la subrutina de calibración

Figura 2.5 Diagrama ASM de la subrutina de Formar Bytes de sensores piezoeléctricos

Figura 2.6 Diagrama ASM de la subrutina de Formar Byte de sensores de inclinación y contactos

Figura 2.7 Diagrama ASM de la subrutina de Formar Bytes de sensores de aceleración

Figura 2.8 Diagrama general del programa

Figura 2.9 Diagrama de recepción de datos en el computador

Figura 2.10 Diagrama de procesamiento de datos

Figura 2.11 Diagrama de estados de la función de aceleración

Figura 2.12 Diagrama de estados para definir el movimiento

Figura 2.13 Diagrama de búsqueda en la base de datos

Figura 3.1 Menú del software traductor

Figura 3.2 Antes de dar clic sobre la opción Nuevo

Figura 3.3 Ventana de ingreso de contraseña

Figura 3.4 Ventana indicadora de contraseña correcta

Figura 3.5 Cuadro de dialogo de contraseña incorrecta

Figura 3.6 Barra del menú Opciones

Figura 3.7 Ventana de Agregar con sus partes

Figura 3.8 Mensajes de palabra agregada

Figura 3.9 Ventana indicadora de Sonido Guardado

Figura 3.10 Ventana de la opción Quitar (Eliminar palabra)

Figura 3.11 Ventana indicadora de palabra eliminada

Figura 3.12 Ventana de la opción Editar

Figura 3.13 Ventana de la opción Depurar (Datos)

Figura 3.14 Ventana de la opción Sonido

Figura 3.15 Ventana de Ayuda

Figura 3.16 Conexión del dispositivo de interfase al puerto COM1

Figura 3.17 Icono indicador de conexión

## ÍNDICE DE TABLAS

Tabla 1.1 Distribución de entradas para el multiplexor CD4067BE

Tabla 1.2 Nomenclatura adoptada para los contactos metálicos

Tabla 1.3 Codificación de los sensores piezoeléctricos

Tabla 1.4 Codificación de las señales emitidas por el acelerómetro

Tabla 2.1 Representación del código de movimientos por eje

# INTRODUCCIÓN

El presente trabajo trata del “Diseño de un traductor de gestos por medio de un guante electrónico sensitivo al movimiento”, enfocado a ayudar a las personas privadas de hablar, con el fin de establecer una comunicación más fácil y rápida con sus semejantes; sabiendo que ellos tienen que aprender la comunicación manual (mímica dactilológica) o aprender exclusivamente la lectura labio facial como medio de comunicación con otras personas, sin embargo la gran mayoría de personas con la capacidad de hablar no entienden su mímica dactilológica.

Actualmente en nuestro medio no se experimenta mucho con el uso de sensores o algunos dispositivos parecidos para la lectura de señales y la interpretación de las mismas a través de un sistema con alguna finalidad. Para nuestro caso en particular, se tratara de adquirir las señales provenientes de los dedos y muñeca para poder procesarlos.

En este proyecto se analiza el Diseño de un traductor de gestos por medio de un guante electrónico sensitivo al movimiento, considerando la portabilidad del guante y los datos necesarios para realizar la traducción.

Este guante estará constituido principalmente por sensores de inclinación, curvatura y aceleración. Los datos brindados por estos sensores serán recogidos por un microcontrolador para ser transmitidos de manera inalámbrica a un computador.

En el computador estos datos serán comparados con una base de datos de posiciones, que pueden significar una letra, palabra, número o frase, que posteriormente será presentada a través de la pantalla del computador.

# CAPÍTULO 1

## 1. ADQUISICIÓN DE DATOS

En este capítulo se mostrará todo lo referente a la adquisición y transmisión de los datos desde los sensores, hasta cuando estos datos codificados llegan al computador.

Para comprender esto de una mejor manera se hizo el siguiente bosquejo de las etapas concebidas para realizar este proyecto. (Figura 1.1)

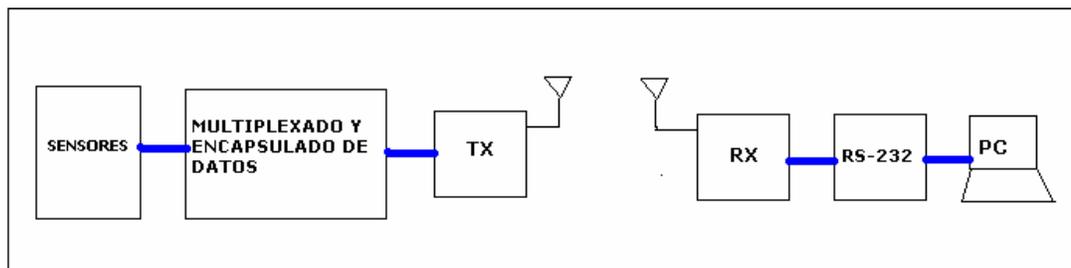


Figura 1.1 Etapas de la adquisición de datos

## 1.1 Fundamentos de la adquisición de datos

Sin duda alguna, la etapa que fue diseñada de mejor manera considerando los parámetros a medir es la etapa de adquisición de datos. En ella se hizo un listado de parámetros (posiciones y movimientos) a sensar, para poder buscar algún instrumento o dispositivo que lo haga de la mejor manera. Partiendo de ello se determinó que se debía sensar lo siguiente:

- Uniones entre los dedos
- Flexión de los dedos
- Grado de inclinación de la mano
- Aceleración de la mano
- Postura de la muñeca

Realizando un análisis de los requerimientos, la facilidad de construcción, el ensamblaje y el reducido tamaño que deberían tener estos dispositivos se resolvió utilizar lo siguiente:

- Uniones entre los dedos

Se utilizó contactos metálicos para determinar si los dedos estaban unidos o separados. (Figura 1.2)

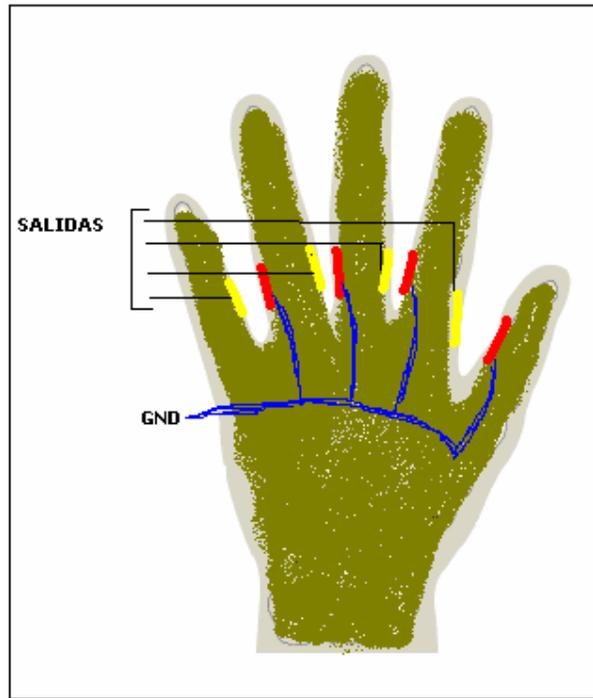


Figura 1.2 Diagrama circuital de contactos metálicos

Si algún contacto entre los dedos está cerrado, el microcontrolador (dispositivo controlador) sensorá un 0 lógico (0V) y para el caso en que esté abierto, entonces el microcontrolador lo tomará como un 1 lógico (5V).

- Flexión de los dedos

Se utilizó sensores piezoeléctricos para determinar el grado de flexión de los dedos. Se empleó 2 sensores por cada dedo, debidamente colocados en las falanges.

- Grado de inclinación de la mano

Para ello se empleó dos sensores de inclinación con dos grados de libertad, colocados sobre el metacarpo.

- Aceleración de la mano

Se utilizó un acelerómetro de 3 ejes colocado encima del metacarpo, para sensar los movimientos de la mano en el espacio.

- Postura de la muñeca

Por medio de sensores piezoeléctricos se pudo determinar la posición relativa de la muñeca con respecto al antebrazo (radio y el cúbito). Para ello se necesitaron 2 sensores.

## **1.2 Sensores piezoeléctricos, de aceleración e inclinación**

Los sensores piezoeléctricos tienen la particularidad, que al ser flexionados ocurre un cambio en la resistencia eléctrica entre sus terminales. Son transductores pasivos, es decir necesitan alguna excitación o polarización para poder convertir un tipo de energía en otra. Para nuestro caso es necesario utilizar un divisor de tensión, de modo que al flexionarlos se obtenga un voltaje proporcional al grado de flexión. (Figura 1.3)

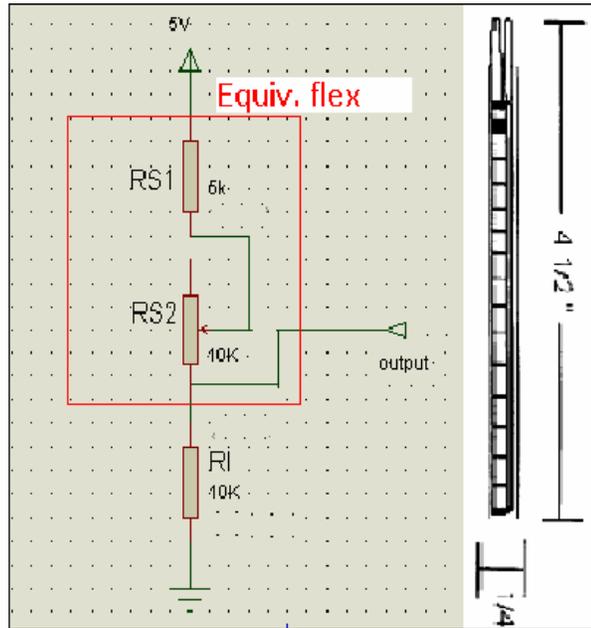


Figura 1.3 Diagrama de conexión de los sensores de flexibilidad

Partiendo de la premisa, que al doblar el sensor aumenta su resistencia podemos determinar que el máximo voltaje sentido se establecerá cuando no se flexiona el sensor. (Figura 1.4)

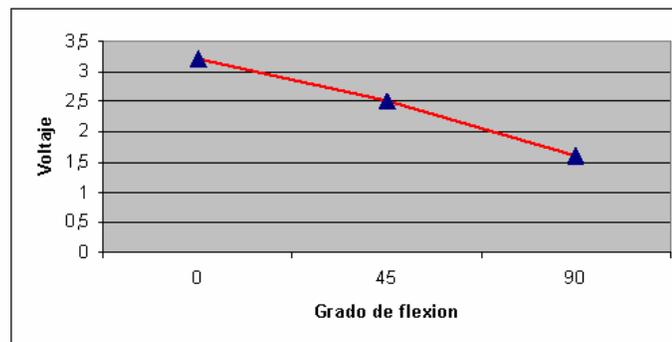


Figura 1.4 Voltaje de salida vs. Grado de flexión del sensor

Características de los sensores piezoeléctricos FLX 01:

- Resistencia máxima: 15K a 20K, proporcional al grado de flexión o curvatura
- Resistencia nominal: 5k ohms
- Aplicaciones: guantes de realidad virtual, robótica, sistemas biométricos, física e ingeniería

Por el tamaño de estos sensores, se procedió a dividirlos en dos, utilizando cada mitad en cada coyuntura de los dedos, logrando así una optimización en la cantidad de sensores requeridos y del tamaño de los mismos.

Los acelerómetros son dispositivos que nos permiten medir la aceleración en uno, dos o tres ejes. Para nuestro caso, se empleó el acelerómetro LIS3L02AS de tres ejes. (Figura 1.5)

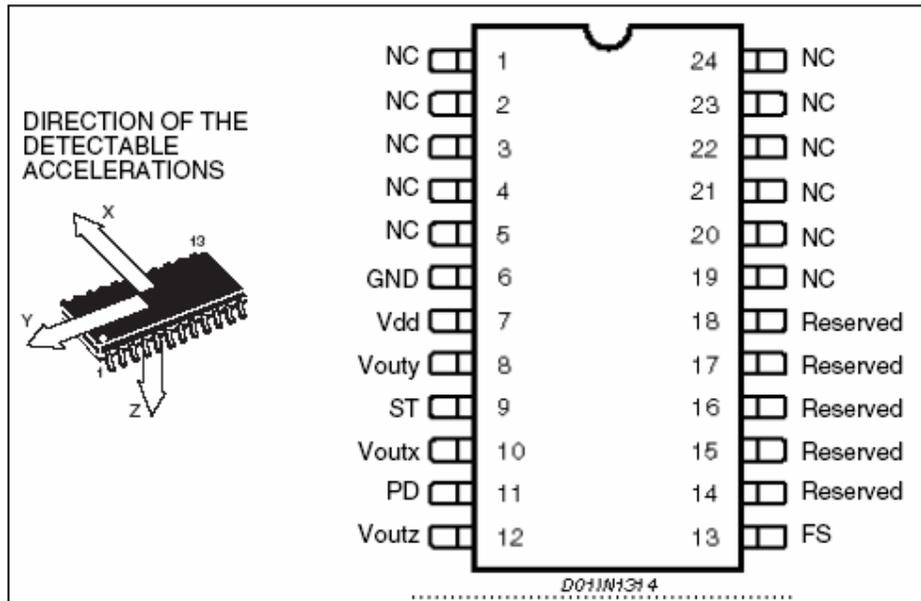


Figura 1.5 Diagrama de conexión del acelerómetro LIS3L02AS

Características de los sensores de aceleración LIS3L02AS:

- Voltaje de operación entre 2.4V y 3.6V
- Bajo consume de potencia
- +/-2g/+6g escalas ajustables
- 0.5mg resolución sobre los 100Hz
- Voltaje de salida depende de la sensibilidad de la alimentación y de un voltaje
- Empaquetamiento SOIC

Es importante destacar que nuestro acelerómetro necesita trabajar en conjunto con el sensor de inclinación Z2205. Sensor que nos ayuda a determinar el grado de inclinación de la mano. (Figura 1.6)

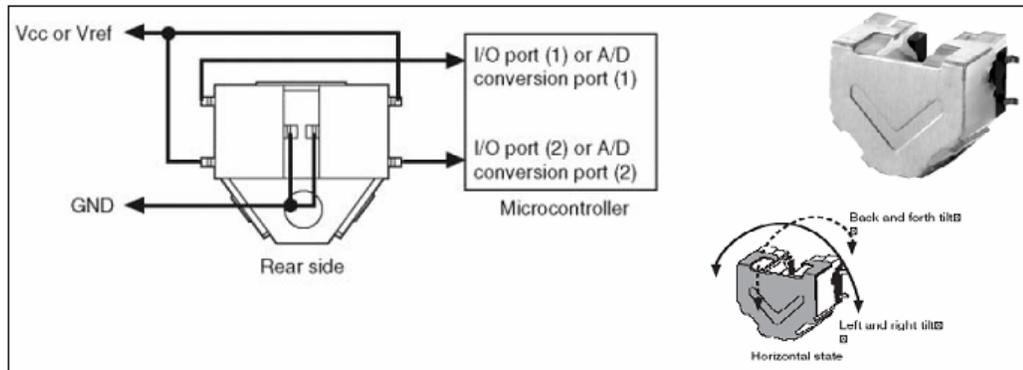


Figura 1.6 Diagrama de conexión del sensor de inclinación Z2205

Características de los sensores de inclinación Z2205:

- Detección de inclinación entre los 45° y 75° (a la izquierda y a la derecha)
- Bajo nivel de consumo de potencia
- Temperatura de operación: -10°C ~ 60°C
- Voltaje de alimentación: 2.7V ~ 3.3V
- Alto nivel de salida de voltaje: Vdd – 0.5V mínimo
- Bajo nivel de salida de voltaje: 0.5V máximo

Utilizando estos tres dispositivos se puede estimar la posición de la mano y su movimiento relativo con respecto al usuario.

### **1.3 Multiplexado y encapsulado de la información a transmitir**

En este subcapítulo se tratará sobre la forma en que transmite las señales provenientes de los distintos sensores, así como también el encapsulado de los datos.

#### **1.3.1 Multiplexado de señales**

Se realizó el multiplexado a través de 2 dispositivos: el microcontrolador PIC 16F876A y del multiplexor CD4067BE de 16 canales.

El CD4067BE, nos permitió recolectar y multiplexar debidamente las señales analógicas conectadas a él (véase tabla 1.1):

Las características fundamentales del CD4067BE son:

- Multiplexor analógico de 16 canales
- Voltaje de polarización entre 5V y 15V
- Cuatro entradas selectoras

Entrada	Señales
Analógica	
0	Acelerómetro eje A (AA)
1	Acelerómetro eje B (AB)
2	Acelerómetro eje C (AC)
3 (pulgar)	Sensor dedo 1A (S1A)
4	Sensor dedo 1B (S1B)
5 (índice)	Sensor dedo 2A (S2A)
6	Sensor dedo 2B (S2B)
7 (medio)	Sensor dedo 3A (S3A)
8	Sensor dedo 3B (S3B)
9 (anular)	Sensor dedo 4A (S4A)
10	Sensor dedo 4B (S4B)
11 (meñique)	Sensor dedo 5A (S5A)
12	Sensor dedo 5B (S5B)
13	Sensor muñeca A (SMA)
14	Sensor muñeca B (SMB)
15	No conectado

Tabla 1.1 Distribución de entradas para el multiplexor CD4067BE

Las entradas selectoras son controladas por el microcontrolador PIC16F876A, el cual además obtendrá las señales provenientes de los sensores de inclinación (IA1, IB1, IA2Y IB2), así como también las provenientes de los contactos metálicos entre los dedos (estos dos últimos grupos de señales irán conectadas directamente al PIC 16F876A) que son señales digitales. (Tabla 1.2)

Nomenclatura	Posición
C1	Entre pulgar e índice
C2	Entre índice y medio
C3	Entre medio y anular
C4	Entre anular y meñique

Tabla 1.2 Nomenclatura adoptada para los contactos metálicos

Características del PIC 16F876A:

- Su tamaño de código de programa es de 8K.
- Cinco entradas analógicas, con una resolución de 10bits
- Velocidad de ejecución es de 5 millones de instrucciones por segundo a 20 MHz y 1 millón de instrucciones por segundo a 4MHz.
- 22 pines de entrada/salida.
- Capacidad de interrupción: múltiples fuentes.

### 1.3.2 Convertidor Analógico – Digital

Dado que cualquiera de las 15 señales a la salida del multiplexor (12 provenientes de los sensores piezoeléctricos más las tres señales de los acelerómetros) es analógica, es necesaria digitalizarla y asignarle un código binario en base a lo que esta podría representar.

El PIC 16F876A cuenta con 5 entradas analógicas de las cuales utilizaremos el AN0 (Pin 2). Este convertidor tiene una resolución de 10 bits y se le asignó un código según su procedencia (sensores piezoeléctricos FLEX o del acelerómetro) y valor.

Para los sensores piezoeléctricos FLEX se capturó los límites de cada uno (esta fase es en la cual se toma los valores máximos y mínimos de cada sensor y se lo almacena en variables, se hablará de esto en mayor detalle en el capítulo II), se procedió a determinar tres rangos y en base a ellos se los codificó en dos bits de la siguiente manera (Tabla 1.3):

Rango	Estado del sensor	Código
1	Estirado	00
2	Medio doblado	01
3	Doblado	10

Tabla 1.3 Codificación de los sensores piezoeléctricos

Aunque no es necesario calibrar los ejes del acelerómetro, se debe marcar cuatro rangos que indiquen si esta sometido o no a alguna aceleración, de ser verdadero indicar si esta es positiva o negativa, de la siguiente manera:

(Tabla 1.4)

Rango	Aceleración	Código
0	Negativa	000
1	-g	001
2	Cero	010
3	+g	011
4	Positiva	100

Tabla 1.4 Codificación de las señales emitidas por el acelerómetro

### 1.3.3 Encapsulado de información

El encapsulado de los datos se lo realizó a través del PIC formándose 7 bytes de información. Los bytes fueron constituidos de la siguiente manera:

Debemos recordar que cada sensor piezoeléctrico se lo ha codificado en dos bits, y dado que son 12 sensores, entonces se necesitará tres bytes (byte 1 – byte 3) para encapsular toda la información de estos sensores.

Byte 1

S2B	S2B	S2A	S2A	S1B	S1B	S1A	S1A
-----	-----	-----	-----	-----	-----	-----	-----

Byte 2

S4B	S4B	S4A	S4A	S3B	S3B	S3A	S3A
-----	-----	-----	-----	-----	-----	-----	-----

Byte 3

SMB	SMB	SMA	SMA	S5B	S5B	S5A	S5A
-----	-----	-----	-----	-----	-----	-----	-----

En el siguiente byte va contenida la información referente al estado de las señales provenientes de las uniones y de los sensores de inclinación.

Byte 4

C7	C3	C2	C1	T11	T12	T21	T22
----	----	----	----	-----	-----	-----	-----

A continuación se almacena la información de dos ejes del acelerómetro (3 bits por cada eje).

Byte 5

0	0	AB	AB	AB	AA	AA	AA
---	---	----	----	----	----	----	----

En el siguiente se almacena la información del último eje del acelerómetro.

Byte 6

0	0	0	0	0	AC	AC	AC
---	---	---	---	---	----	----	----

El último byte indica el fin de la trama y sincroniza la transmisión, para facilidad se ha enviado la trama FFxH o en binario '11111111'.

Byte 7

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

#### **1.4 Transmisión y recepción de información**

Debido a que se desea enviar los datos del guante de manera inalámbrica se hace necesaria la modulación. Para ello se utilizó un transmisor conectado en el guante (TLP 434A), y su par receptor en el computador (RLP 434), los cuales permiten la comunicación entre el dispositivo y el computador a una velocidad de 2400 bps.

Características del transmisor TLP 434A y del receptor RLP 434:

- Frecuencia de operación 433.9 MHz
- Modulación ASK para optimizar la eficiencia de la señal
- Bajo nivel de corriente

Además, dado que existe la presencia de ruido en el ambiente se decidió implementar un algoritmo simple de corrección de errores. Se utilizó el siguiente código convolucional, para codificar la información transmitida (Figura 1.7):

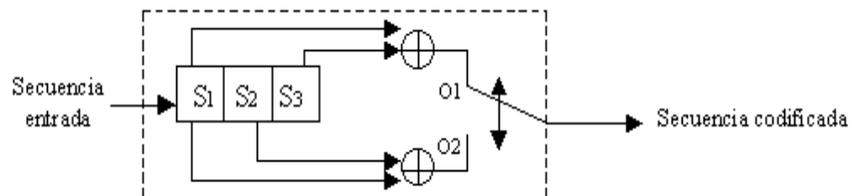


Figura 1.7 Código convolucional

Del mismo modo se implementó en el lado del receptor el decodificador para éste código convolucional, obteniendo la información que originalmente se estaba transmitiendo.

Este código le da una mayor seguridad y confiabilidad a la transmisión inalámbrica, dado que nos permite corregir algunos errores (máximo tres), sin

embargo disminuye la tasa efectiva de transmisión a la mitad, es decir a 1200 bps.

Luego de recibir los datos en el RLP 434, es necesario enviar los datos al computador a través del puerto serial.

Las comunicaciones paralela todavía son usadas en nuestros días. Antiguamente se las ocupaban para trabajar con equipos remotos (todavía hay algunos), pero hoy en día se trabaja con el puerto serial ya que es mucho más simple de implementar que por medio del puerto paralelo (por la cantidad de cables), además la mayoría de los computadores (por no decir casi todos), tienen este puerto. Para objeto del presente trabajo se utilizará el puerto serial de 9 pines DB9, de los cuales se pueden ocupar solamente tres, transmisión (Tx), recepción (Rx) y tierra (Ground), cada uno de estos pines van insertados en la plaqueta del receptor. Antes de ser enviados los datos de la salida del receptor (datos remodulados) es necesario utilizar un IC MAX232CPE. (Figura 1.8)

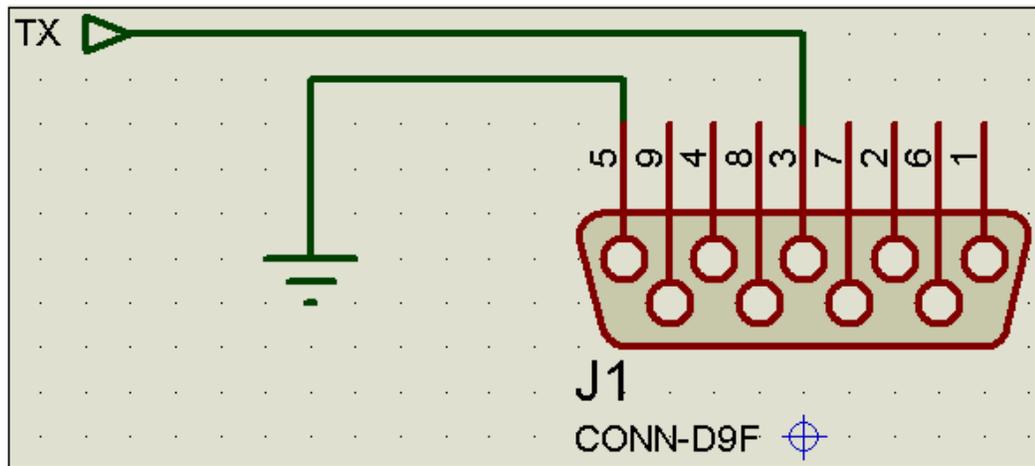


Figura 1.8 Diagrama de conexión del conector serial DB9

El IC MAX232CPE (Figura 1.9) aplica los protocolos EIA/TIA-232E, el mismo que nos ayudará a convertir de la lógica TTL ('1' entre 3.5 y 5.5 V y '0' entre 0 y 0.2 V dependiendo del voltaje de polarización) a los estándares eléctricos para la transmisión serial de datos ('1' entre -3 y -25 V y '0' entre +3 y +25 V).

A continuación mencionaremos sus principales características:

- Alimentación de 5 V
- Cumple las especificaciones de los protocolos EIA/TIA-232E y V.28
- Dos entradas y dos salidas
- Tasa máxima de datos de 120Kbps

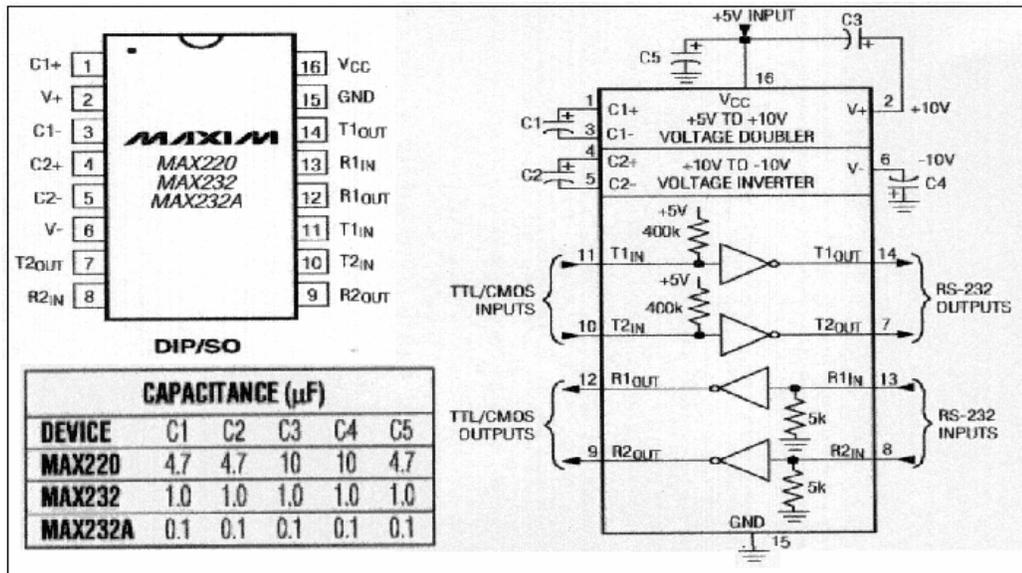


Figura 1.9 Diagrama de conexión del MAX232CP

# **CAPÍTULO 2**

## **2. DISEÑO DEL TRADUCTOR**

En este capítulo se tratará sobre los parámetros de construcción del guante electrónico, así como de la etapa de interfase entre el Guante Electrónico y la PC.

### **2.1 Construcción del guante electrónico**

Podríamos decir que en este subcapítulo analizaremos la parte estética y el funcionamiento de la unidad de control del guante, claro sin olvidar los requerimientos del mismo

#### **2.1.1 Parámetros de construcción**

Para la construcción del guante se analizó los siguientes parámetros en el guante:

- Temperatura teniendo puesto el guante
- Capacidad de flexión puesto el guante
- Tamaño reducido
- Ergonómico
- Posicionamiento de los sensores y de los integrados en la mano incluyendo la fuente de alimentación

Teniendo en cuenta todos los parámetros anteriormente mencionados, se confeccionó el guante de la tela licra Bussin y se hizo el posicionamiento de dispositivos (Figura 2.1).

Dado los requerimientos de energía que debían tener los sensores piezoeléctricos, de inclinación, aceleración, el multiplexor analógico y del microcontrolador se determinó que se debía dotar de una fuente de alimentación de 5Vdc / 320mA.

También se consideró que la fuente debería ser recargable y pequeña debido a las dimensiones y características propias del guante.

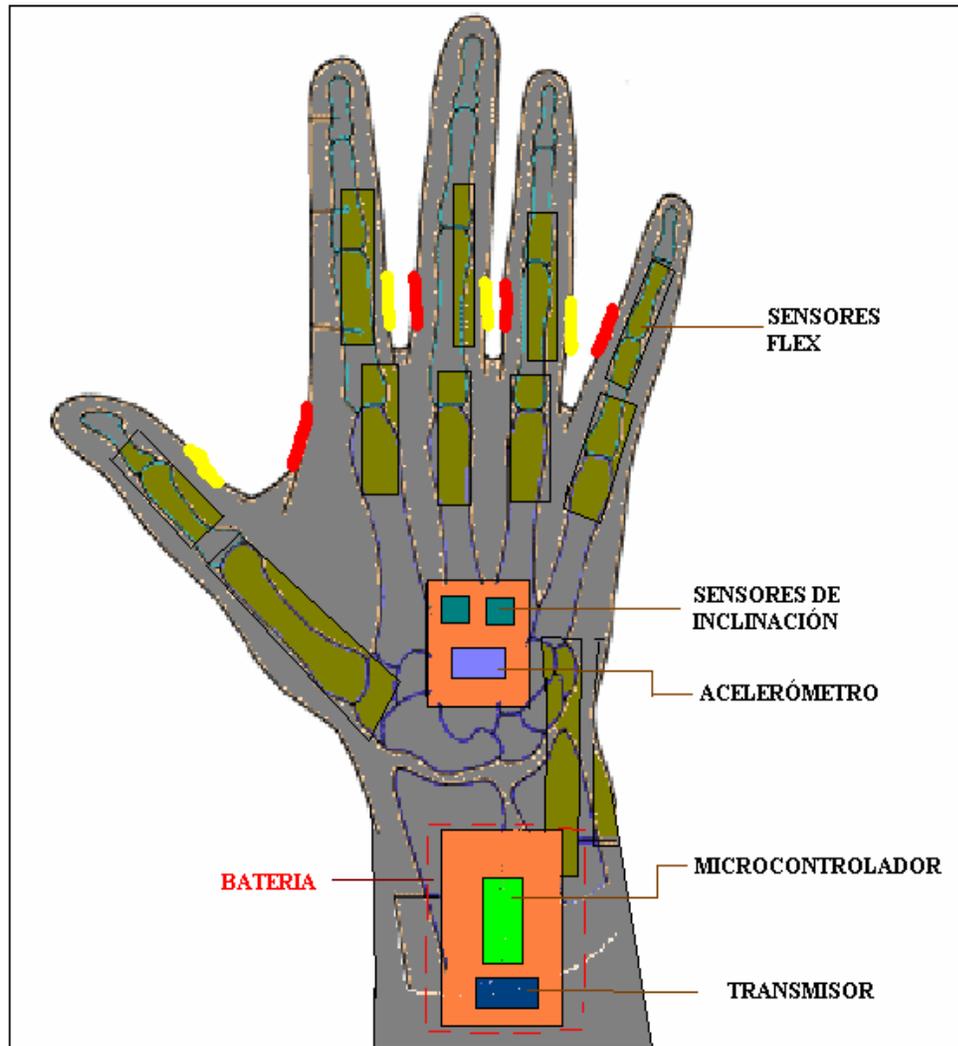


Figura 2.1 Posicionamiento de dispositivos en el guante

Después de este análisis se determinó que era factible utilizar dos baterías de Li-ion de 3.7Vdc y de 600mA/h en serie, obteniendo 7.4Vdc. Luego de ello agregar un regulador de voltaje de 5V LM7805, el cual alimentará a los

sensores piezoeléctricos, inclinación y aceleración, así como también al PIC, al multiplexor analógico CD4067BE y al transmisor. (Figura 2.2)

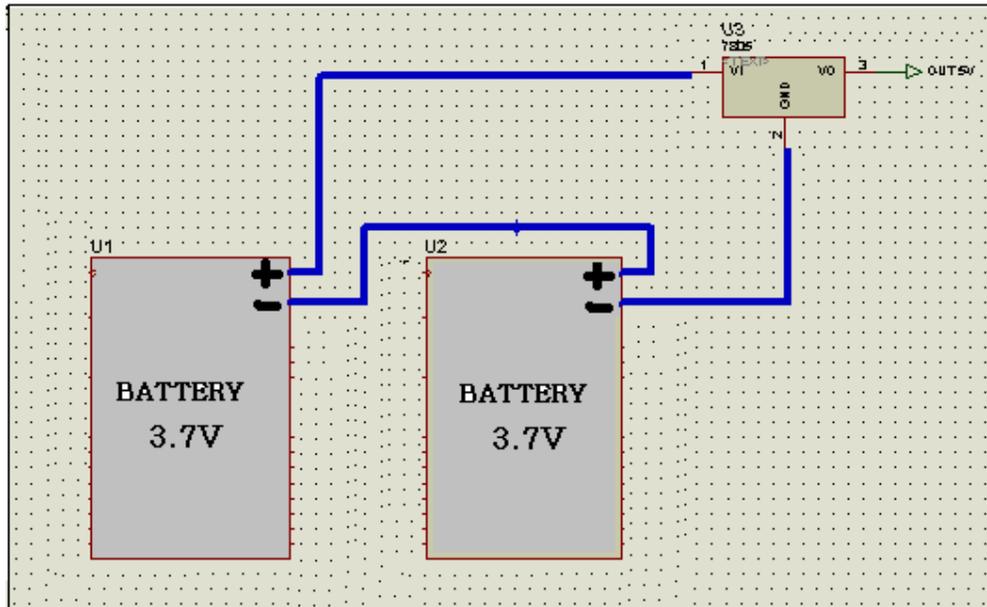


Figura 2.2 Diseño de la unidad de alimentación

Se resolvió trabajar con baterías de Li-ion ya que ofrecen ventajas frente a las de NiMh y a las de Ni-Cd.

Principales características de las baterías Li- Ion:

- ❖ Mayor rendimiento

La carga de la batería dura más tiempo sin afectar su desempeño cuando son puestas nuevamente en operación. Además permite mayor tiempo de recargas antes de requerir un reemplazo.

❖ Menor peso

Las baterías de Iones de Litio pesan 35% menos que las baterías de Ni – Cd comparadas con baterías de capacidad similar.

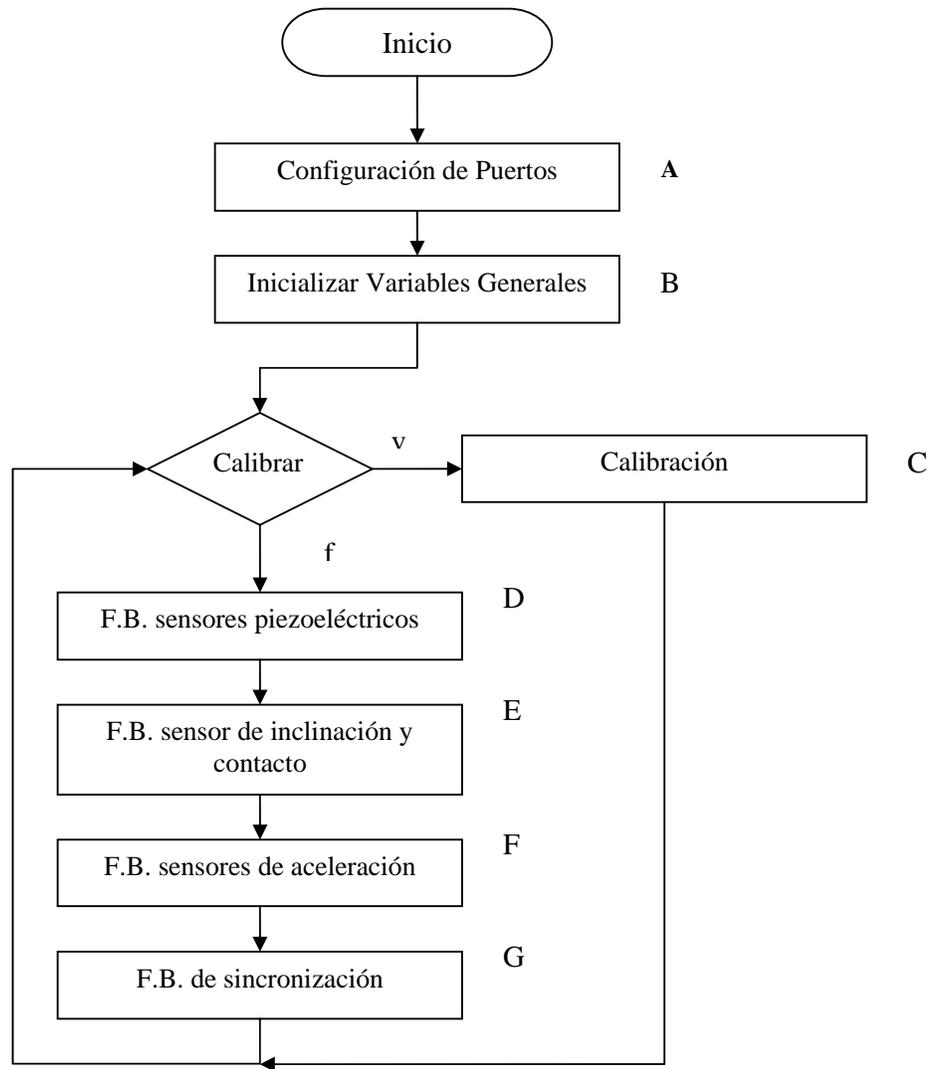
❖ Sin efecto memoria

A diferencia de otras aleaciones, las baterías de Li-ión no ‘memorizan’ el período de recarga anterior y, por lo tanto, la duración de carga en este tipo de baterías no disminuye conforme se realizan más descargas.

### **2.1.2 Funcionamiento de la unidad de control**

En este subcapítulo se abordará todo lo concerniente a la manera estructural que el microcontrolador PIC 16F876A, se realizará las distintas rutinas de adquisición de datos de los sensores y el envío de datos al computador.

Para la mejor explicación del mismo se lo hará a través de los diagramas ASM. A continuación un diagrama general sobre las rutinas implementadas en el microcontrolador.(Figura 2.3)



Nota: F.B. = Formar Byte

Figura 2.3 Diagrama simplificado de la unidad de control del PIC

- Configuración de Puertos

En esta parte del programa se asignó las entradas y salidas de cada puerto tal como se indica a continuación:

El puerto A se configuró moviendo el literal `b'00000001'` (para la configuración de los puertos la asignación de un 1 significa entrada y 0 salida) al TRISA, registro encargado de la configuración del puerto A. El pin RA0 esta configurado como entrada y servirá para que el PIC detecte los datos analógicos multiplexados, el resto de pines están configurados como salidas, dado que no tienen función primaria alguna.

El puerto B se configuró moviendo el literal `b'11110000'` al TRISB, registro encargado de la configuración del puerto B. Los pines RB7:RB4 son las entradas correspondientes a los sensores de inclinación y los de RB3:RB0 son las salidas que controlaran las entradas selectoras del multiplexor analógico.

El puerto C se configuró moviendo el literal `b'10011111'` al TRISC, registro encargado de la configuración del puerto C. El pin RC0 es la entrada calibrar, RC4:RC1 son las entradas correspondientes a los contactos metálicos (C4:C1) y el pin RC7 es la entrada de recepción que no se usa. Las salidas

son RC6 la cual es el pin transmisor de datos de manera serial y el pin RC5 es una salida no utilizada.

- Inicializar Variables Generales

Aquí se inicializan las variables de cada sensor piezoeléctrico con los valores máximos y mínimos por defecto (00h y FFh respectivamente).

- Calibración

La calibración de sensores es necesaria debido a que todas las personas no poseen las mismas características físicas de la mano, refiérase a longitud de los dedos o grosor de los mismos, así como también todos no flexionan los dedos hasta un mínimo o máximo de curvatura común. Partiendo de ello y del hecho de que para poder generar los rangos para la codificación de los sensores piezoeléctricos se partió del mayor y menor voltaje sentido, se hace imprescindible personalizar el guante de tal manera que todos lo puedan utilizar.

A continuación se muestra la calibración de cada sensor piezoeléctricos FLEX ( $i = 4..15$ ), esto se va realizando en secuencia y trabaja conjuntamente con el multiplexor de señales.(Figura 2.4)

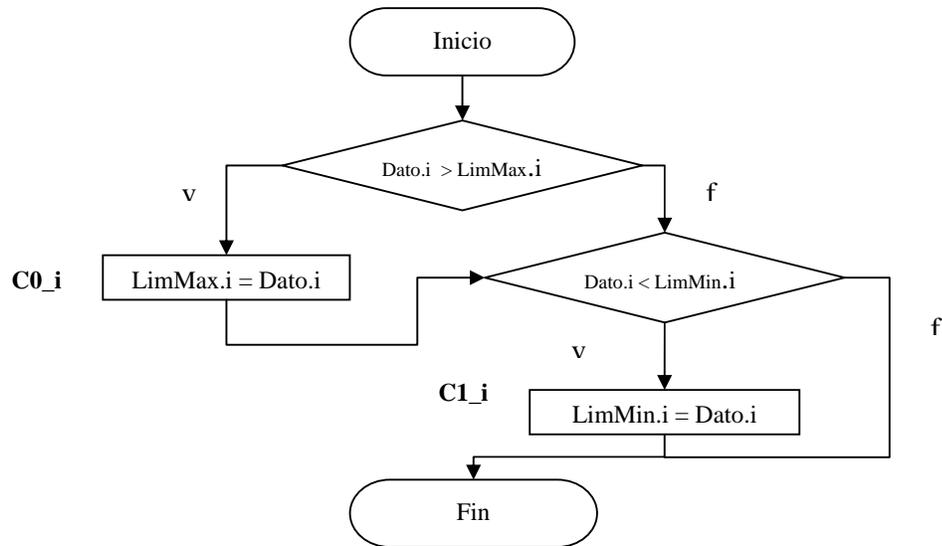


Figura 2.4 Diagrama ASM de la subrutina de calibración

- Formar Bytes de los sensores piezoeléctricos

En esta fase se hizo la selección secuencial de cada una de las señales analógicas, provenientes de los sensores piezoeléctricos, cargando el valor a las salidas del puerto B. Con ello obtendremos la señal escogida en la entrada RA0, se procederá a realizar la conversión analógica – digital y a codificar el dato en 2 bits. Luego, estos valores son encapsulados en grupos de 4, para ser posteriormente transmitidos y volver a realizar toda la secuencia hasta que se hayan transmitido todos los valores de los 12 sensores piezoeléctricos (en 3 bytes). (Figura 2.5)

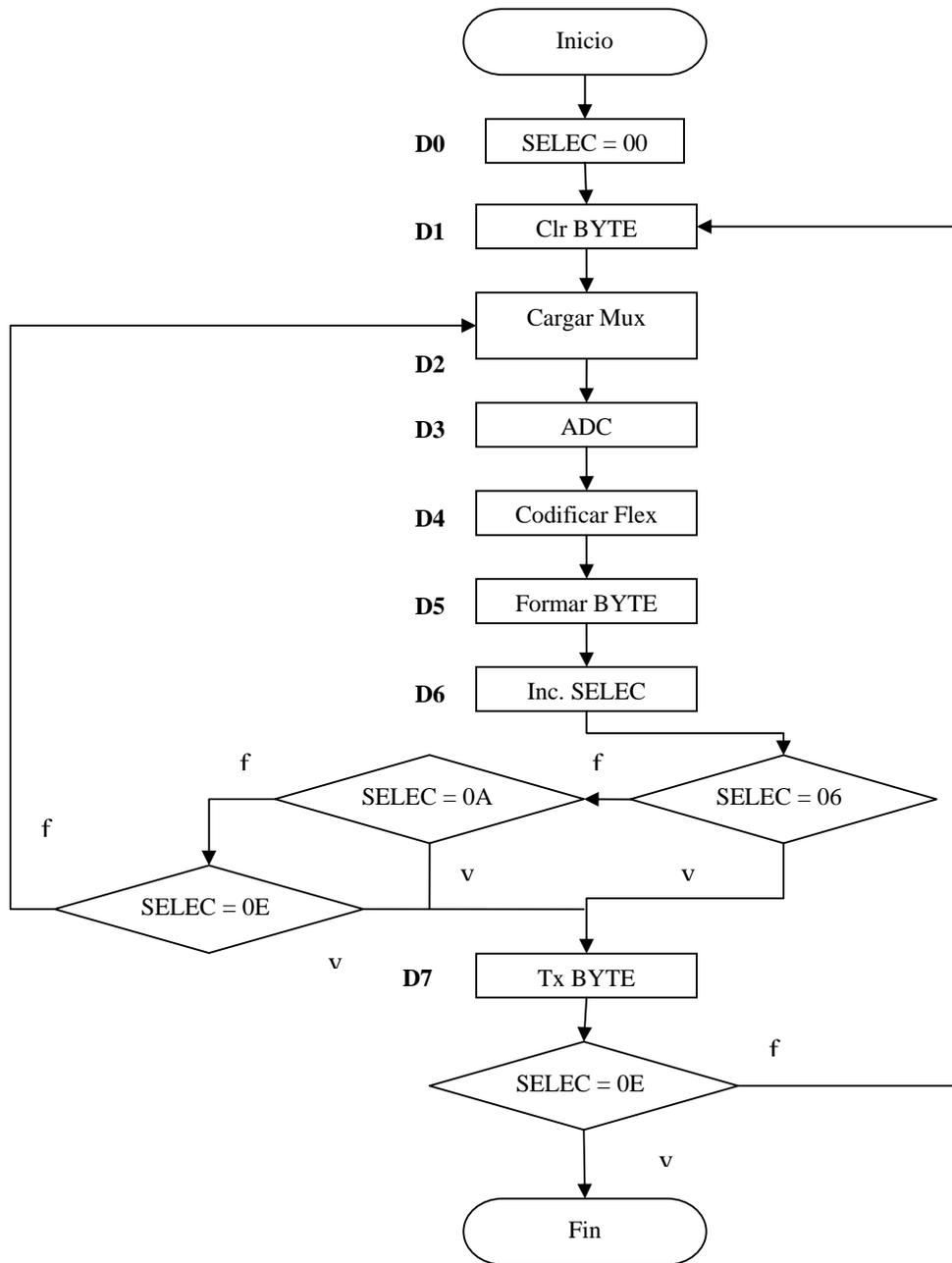


Figura 2.5 Diagrama ASM de la subrutina de Formar Bytes de sensores piezoeléctricos

- Formar Byte originado por el sensor de inclinación y por los contactos metálicos

Esta parte se refiere a la manera de capturar la información proveniente de los sensores de inclinación (RB7:RB4) y de los contactos metálicos (RC4:RC1) para unirla y transmitir el byte con la información anteriormente mencionada. (Figura 2.6)

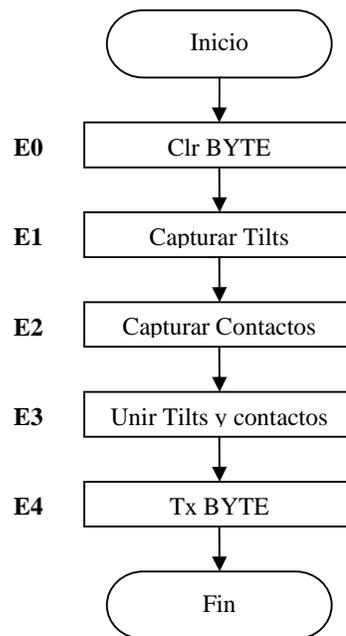


Figura 2.6 Diagrama ASM de la subrutina de Formar Byte de sensores de inclinación y contactos

- Formar Bytes originados por el sensor de aceleración

Al igual que para los sensores piezoeléctricos, se realizó la selección secuencial de cada una de las señales analógicas provenientes de cada uno de los tres ejes de nuestro acelerómetro y se convirtió en una señal digital. Luego de ello se hace necesario codificarla en 3 bits la información ofrecida por el acelerómetro para luego ser unida y transmitida en 2 bytes (Figura 2.7).

- Formar Byte de sincronización

Esta fase tan solo consiste en enviar un byte con todos sus bits seteados (FFh), para hacer referencia que todos los bytes de datos ya fueron transmitidos. Además este byte, nos sirve para sincronizar la trama de datos en el lado del receptor.

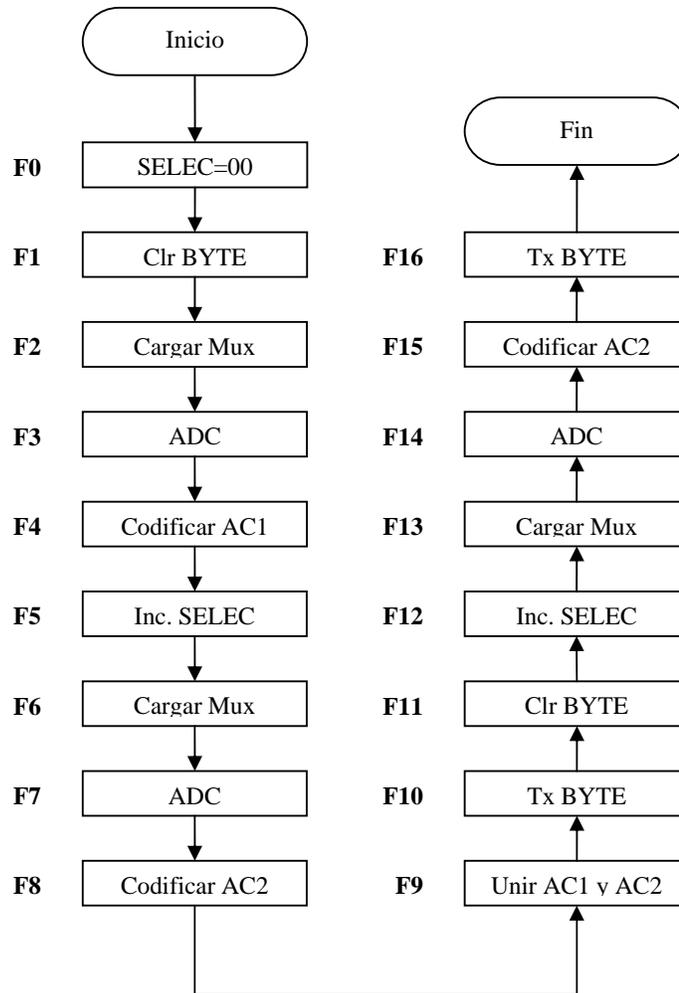


Figura 2.7 Diagrama ASM de la subrutina de Formar Bytes de sensores de aceleración

## **2.2 Decodificador de los paquetes recibidos**

Luego de todo el proceso de adquisición de datos, calibración, formateo y transmisión de los mismos, desde el transductor (guante), se procede a recibir los paquetes, decodificarlos, procesarlos y mostrar la información obtenida.

Para esto se utilizó un programa basado en Java, debido a que es un lenguaje:

- De objetos
- Independiente de la plataforma

Algunas características notables del mismo:

- Robusto
- Gestiona la memoria automáticamente
- No permite el uso de técnicas de programación inadecuadas
- Multihilo
- Mecanismos de seguridad incorporados
- Herramientas de documentación incorporadas

- Básicamente el programa consta de 3 subprogramas que corren paralelamente (Multihilo), lo que provee una transparencia para el usuario del procesamiento realizado. (Figura 2.8)

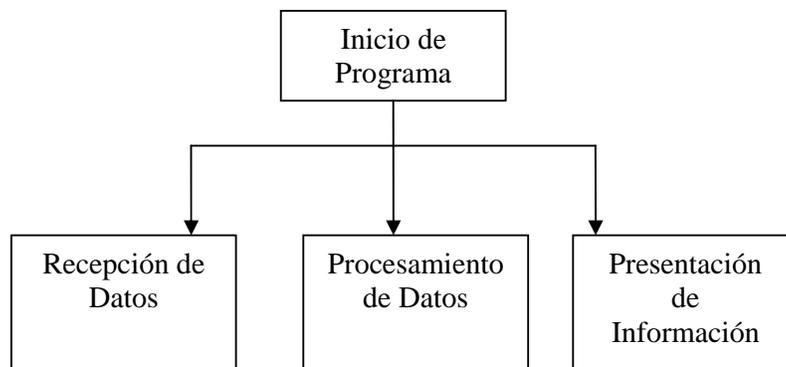


Figura 2.8 Diagrama general del programa

Ahora procedamos a analizar cada subprograma del mismo:

- Recepción de datos

Como ya se mencionó en el capítulo anterior, los datos son recibidos de modo serial, a través del puerto COM1.

Para asegurarnos de que el buffer de recepción del puerto no se llene y haya bytes descartados, los datos del buffer son extraídos y puestos en una cola de bytes de modo continuo, para su posterior procesamiento. (Figura 2.9)

De este modo nos aseguramos de recibir todos los bytes enviados por el transmisor sin que se pierda alguno ni la sincronía de los mismos.

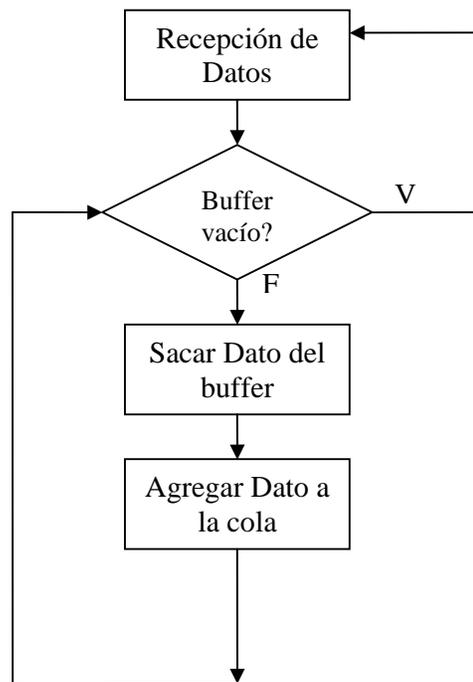


Figura 2.9 Diagrama de recepción de datos en el computador

- Procesamiento de datos

Este subprograma trabaja con la cola de datos, que se está llenando con el subprograma anterior.

Al iniciarse, verifica que haya datos en la cola, si no existen datos, espera 100 ms para que se ingresen algunos datos y vuelve a revisar. A partir de estos datos se decodificará y procesará la información transmitida.

El proceso se inicia con la obtención de dato por dato de la cola, se lo revisa y se lo almacena en variables locales, correspondientes a cada byte esperado. De estos datos se obtiene el valor esperado por repetición del dato transmitido (Por ej. Se recibe 0xFF 0xFE 0xFF 0xFF -> se decodificará 0xFF).

Adicionalmente si los datos corresponden a los de aceleración son también agregados a una cola especializada para determinar el movimiento.

(Figura 2.10)

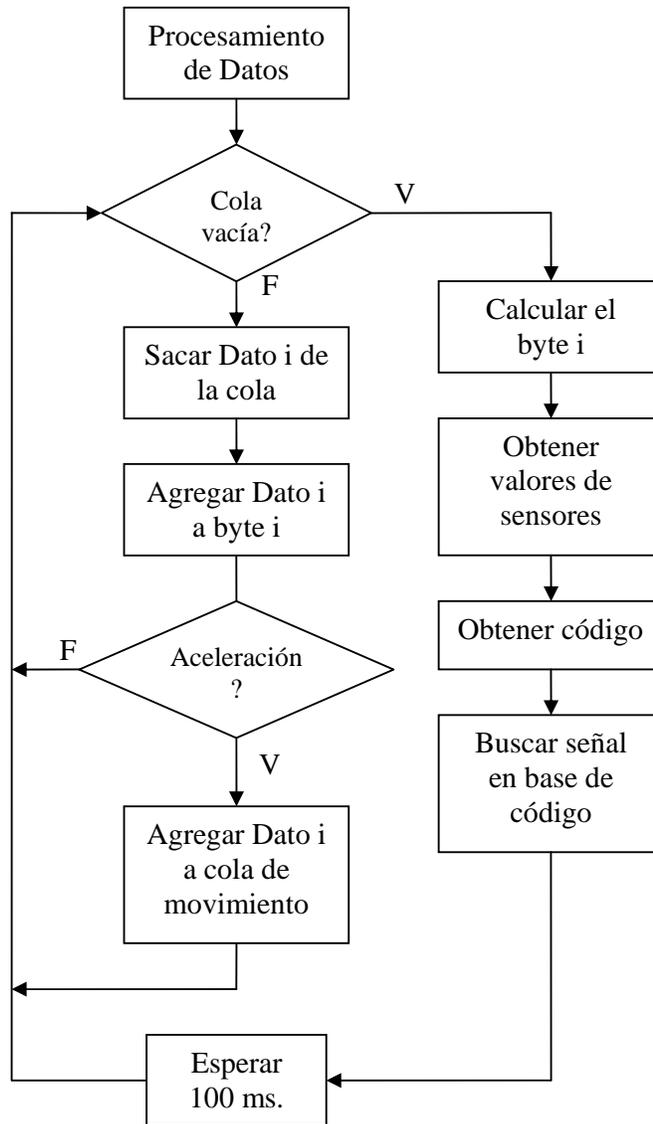


Figura 2.10 Diagrama de procesamiento de datos

Posteriormente, con los bytes obtenidos, se procede a decodificarlos para obtener los valores de los sensores de curvatura, inclinación, acelerómetros y las uniones en un momento determinado.

De estos datos decodificados, se procederá a obtener el código que representan, de la siguiente manera:

$$Cod1 = s_0 + 3s_1 + 3^2 s_2 + \dots + 3^{11} s_{11}$$

$$Cod2 = t_0 + 2t_1 + 4t_2 + 8t_3$$

$$Cod3 = u_0 + 2u_1 + 4u_2 + 8u_3$$

donde,

si = sensores de curvatura

ti = valor del sensor de inclinación

ui = valor de las uniones

Además existen tres códigos adicionales que se pueden agregar o no, dependiendo si el gesto incluye un movimiento.

$$AcelX = f(t, posición)$$

$$AcelY = f(t, posición)$$

$$AcelZ = f(t, posición)$$

donde, cada uno representa la aceleración o movimiento de la mano en cada eje del espacio.

De ahí que los valores que puede tomar cada código corresponden a: (Tabla 2.1)

Código	AcelX	AcelY	AcelZ
0	Alternado	Alternado	Alternado
1	Izquierda	Atrás	Abajo
2	Derecha	Adelante	Arriba
3	Quieto	Quieto	Quieto

Tabla 2.1 Representación del código de movimientos por eje

Cabe resaltar que la función  $f(t, posición)$  depende de dos parámetros  $t =$  tiempo y  $posición =$  eje de ubicación de la mano en un momento determinado. Esta función es independiente en cada eje y tiene el siguiente comportamiento: (Figura 2.11)

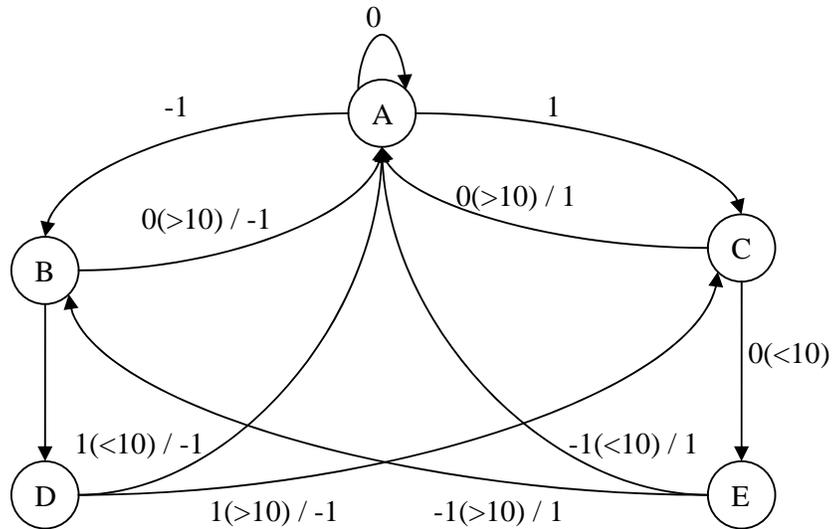


Figura 2.11 Diagrama de estados de la función de aceleración

donde,

Un “-1” significa que hay un valor de aceleración menor a 2, un “1” un valor mayor a 2 y un “0” un valor igual a 2.

La representación “x(>y)”, indica que se recibió el código “x”, una cantidad de veces mayor a “y” de modo consecutivo.

La representación “x(<y)”, indica que se recibió el código “x”, una cantidad de veces menor a “y”, antes de recibir otro valor.

El “/z” indica el valor que arroja la función en ese instante de tiempo al cambiar de estado. Ese valor es ingresado en otra cola que se procesa para obtener finalmente el movimiento ejecutado. A continuación el funcionamiento de dicho procesamiento: (Figura 2.12)

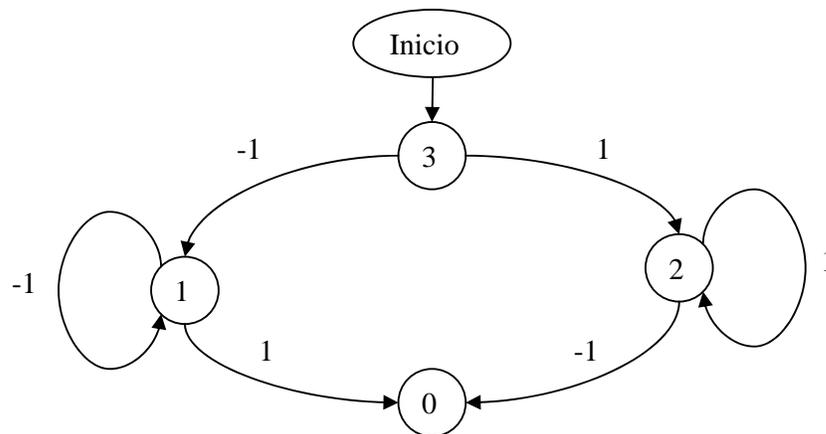


Figura 2.12 Diagrama de estados para definir el movimiento donde, las entradas provienen de la cola generada por la función anterior

En este diagrama de estado, el código del estado (0, 1, 2, 3) representa el código del movimiento en el eje, con su respectivo significado (véase Tabla 2.1). Este será el valor asignado al gesto, si depende del movimiento.

Como se puede observar, cada gesto o señal hecha con el guante consta de 3 códigos si es estático y de 4 a 6 códigos si es un gesto dinámico, los cuales

hacen único al gesto y nos permite diferenciarlos entre sí y facilitar su búsqueda.

Estos códigos son finalmente comparados con una base de códigos, para obtener la letra, palabra o frase que significan.

### **2.3 Base de datos de las señas más usadas**

La búsqueda en la base de datos se realiza de la siguiente manera por prioridad en los códigos:

Primero se busca el código producido por los sensores de curvatura.

Segundo, el código producido por las uniones.

Tercero, el código producido por los sensores de inclinación.

De aquí en adelante los códigos son opcionales, dependiendo si la palabra fue ingresada con movimiento o no.

Cuarto, el código producido por el acelerómetro en el eje X.

Quinto, el código producido por el acelerómetro en el eje Y.

Sexto, el código producido por el acelerómetro en el eje Z.

Con esto se puede obtener la palabra asociado a este conjunto de códigos, si se encuentra definida (Figura 2.13).

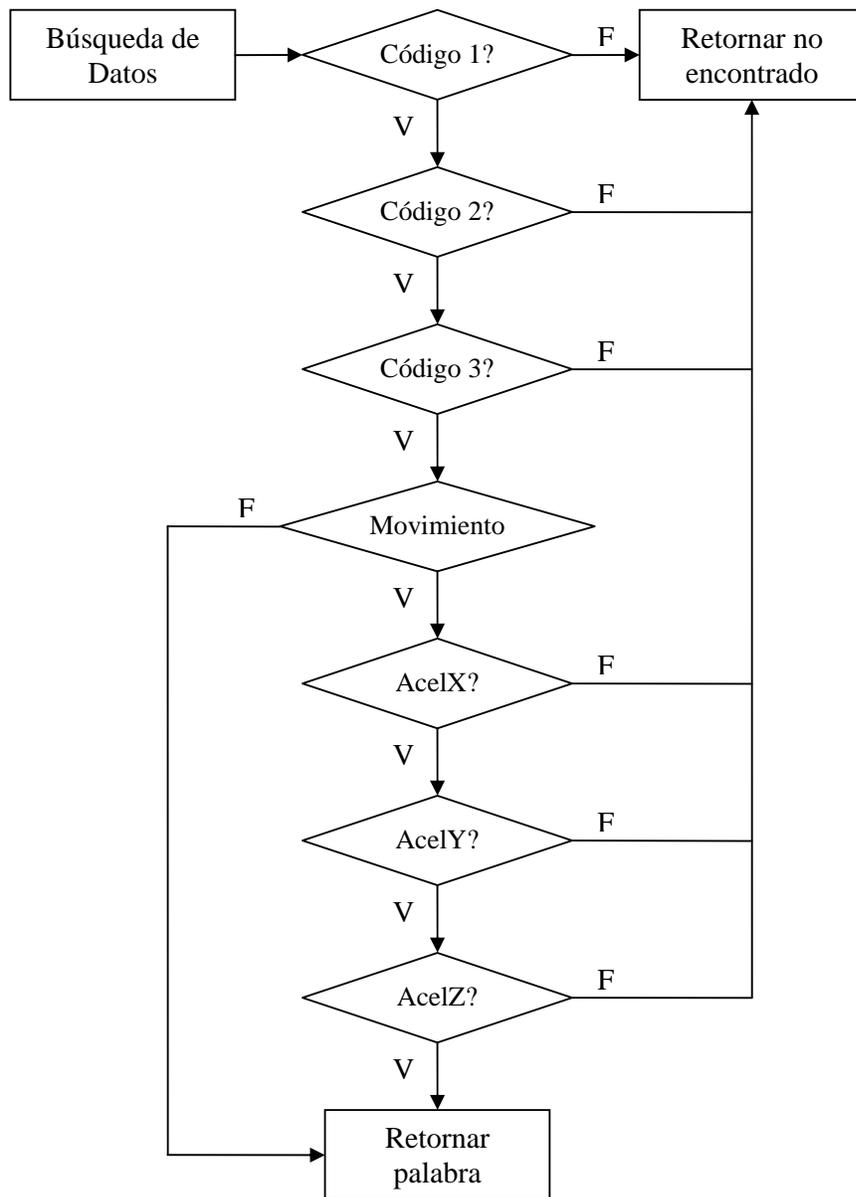


Figura 2.13 Diagrama de búsqueda en la base de datos

# **CAPÍTULO 3**

## **3. INTERFASES**

En el capítulo 3 se mostrará y se explicará lo referente a la presentación de datos en el computador, así como también los resultados obtenidos en este proyecto.

### **3.1 Interfases**

Este subcapítulo mostrará la manera en que los datos serán presentados en el computador y el manual de usuario como una herramienta de ayuda para el uso del traductor; tanto hardware (guante electrónico) como software (Traductor de Gestos C.S.).

#### **3.1.1 Presentación de la información de los datos**

Luego de que se ha obtenido los datos de los sensores, la señal es representada en forma de mensajes en la pantalla del ordenador.

Para la mejor utilización del programa se han diseñado tres menús: Archivo, Opciones y Ayuda; a continuación veremos más en detalle cada uno de ellos. (Figura 3.1)

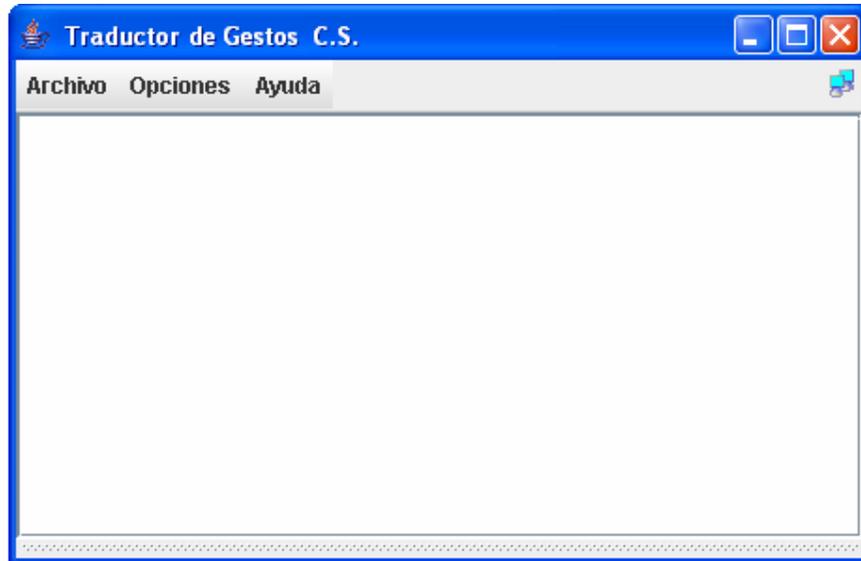


Figura 3.1 Menú del software traductor

### **Archivo**

Esta pestaña ubicada en la barra de menú, despliega otras opciones si damos un clic sobre ella. Estas opciones son:

**Nuevo.-** esta opción borra de la pantalla todas las traducciones hechas cada vez que se da clic sobre ella. La pantalla quedara como en la figura anterior.

(Figura 3.2)

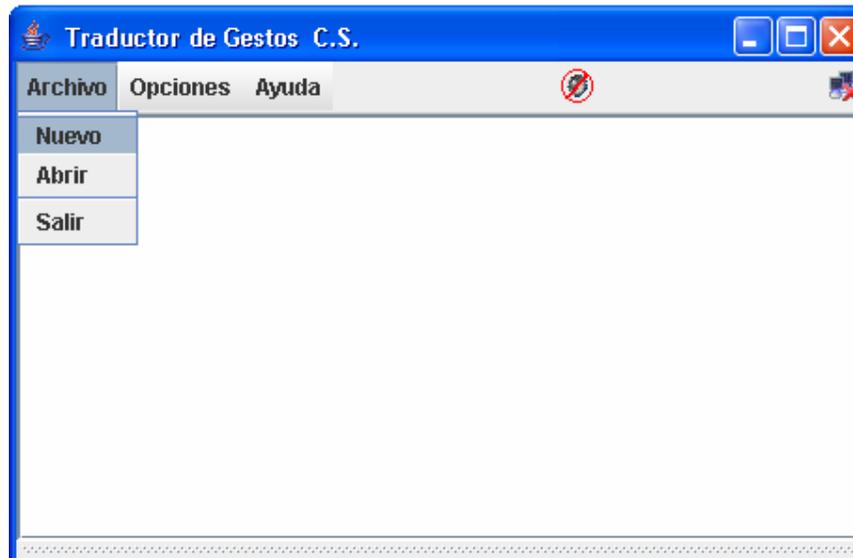


Figura 3.2 Antes de dar clic sobre la opción Nuevo

**Salir.-** cuando damos clic sobre esta opción salimos del programa traductor.

### **Opciones**

Este menú contiene varias funciones, las que podemos apreciar cuando damos clic sobre el mismo (Figura 3.3).

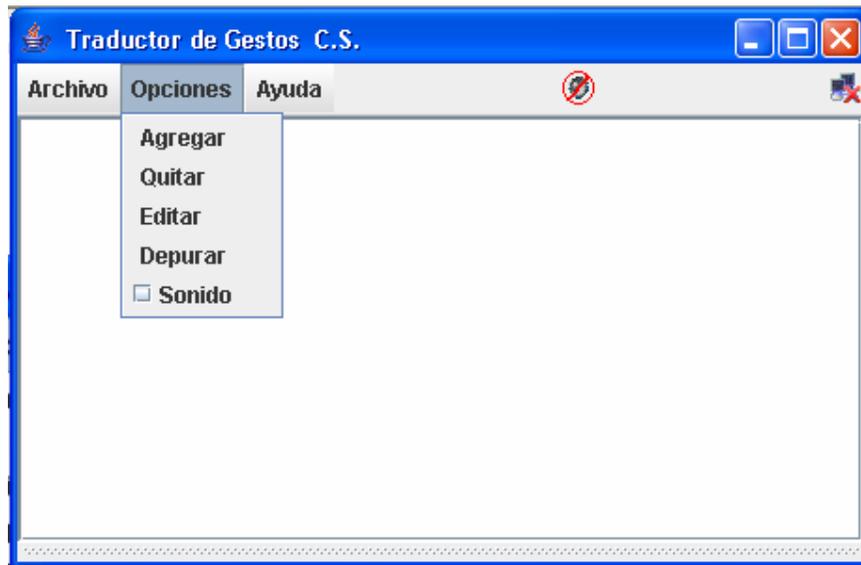


Figura 3.3 Barra del menú Opciones

El programa tiene dos modos de uso:

**Usuario.-** Permite solamente la utilización del traductor.

**Administrador.-** Permite agregar, editar o eliminar palabras dentro del programa. Para acceder a este modo de administrador se requiere ingresar una contraseña.

**Contraseña.-** Nos permite tener acceso a las opciones de Agregar, Quitar y Editar. Cuando damos clic sobre alguna de estas opciones se pedirá la clave para que el usuario acceda a dichas opciones. (Figura 3.4)

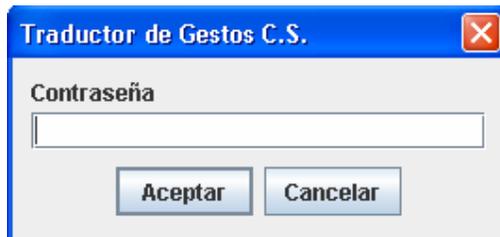


Figura 3.4 Ventana de ingreso de contraseña

Si la clave ingresada es correcta aparecerá un mensaje que indicara que ha ingresado la contraseña correcta y automáticamente se habilitará las opciones de Agregar, Quitar y Editar. (Figura 3.5)

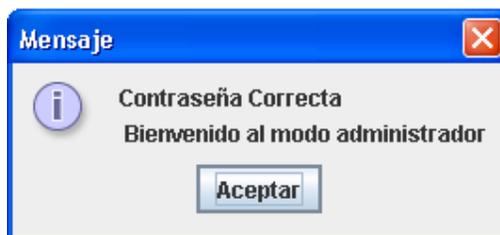


Figura 3.5 Ventana indicadora de contraseña correcta

Por otro lado si hubo algún error en el ingreso entonces el programa genera un mensaje indicando el ingreso incorrecto de la contraseña. (Figura 3.6)

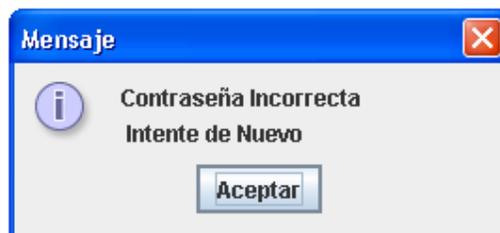


Figura 3.6 Cuadro de dialogo de contraseña incorrecta

Nota: Una vez activado el modo administrador este quedará activo hasta que se cierre la aplicación.

**Agregar.-** se utiliza cuando el usuario necesita agregar alguna palabra a el software traductor. Cuando el usuario da clic sobre esta opción es mostrada la siguiente ventana (Figura 3.7):

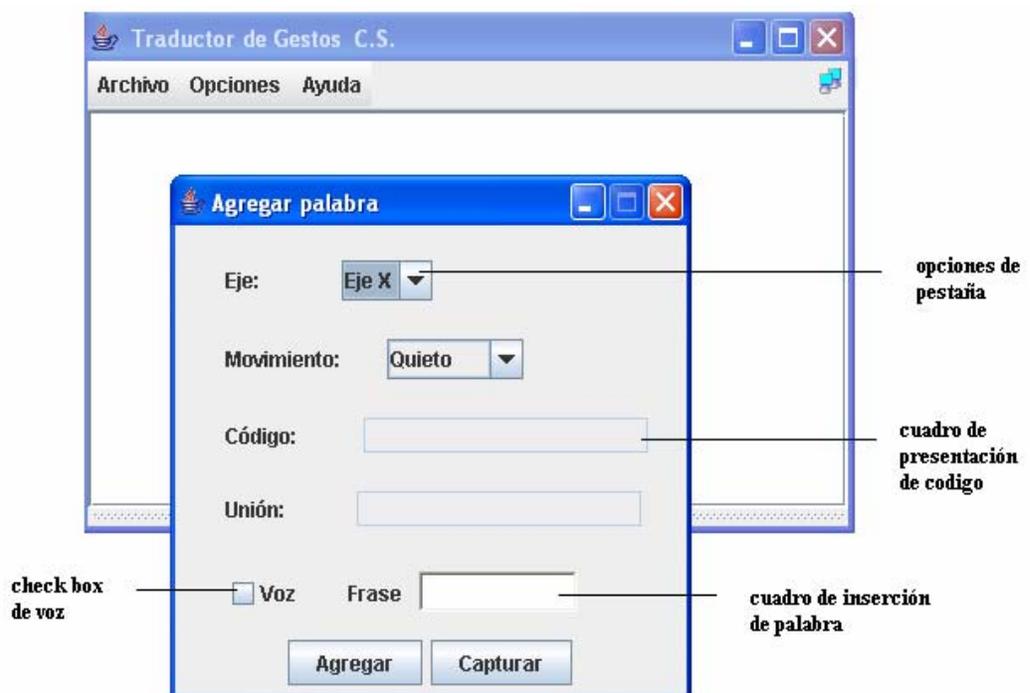


Figura 3.7 Ventana de Agregar con sus partes

En el cuadro de inserción de palabra, el usuario tendrá que introducir el significado de la traducción del código mostrado en pantalla, tomando en consideración los campos señalados por el usuario. Estos campos son: Aceleración en X, Aceleración en Y y la Aceleración en Z. Una vez escogida

las opciones discriminantes y habiendo insertado el significado de la traducción del código se procede a dar clic sobre el botón agregar. Luego de esto aparecerá una ventana indicando que la palabra ha sido agregada (Figura 3.8), o en su defecto indicará que ya se encuentra la palabra en la base de datos.

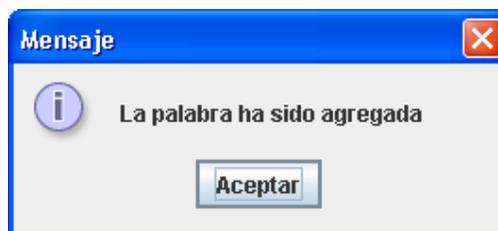


Figura 3.8 Mensajes de palabra agregada

Si se habilitó el check box de sonido entonces se mostrará la ventana para grabar sonido, inmediatamente.

Se debe presionar el botón Grabar para empezar la grabación y Parar para cortarla. Una vez parada la grabación se habilitan las opciones Reproducir y Guardar.

**Reproducir.-** como su nombre lo indica, reproduce el sonido guardado temporalmente hasta cerrar la ventana. Este botón es implementado para que la persona, quien está editando el sonido, pueda verificarlo y grabarlo luego cuando esté totalmente de acuerdo con el mismo.

**Guardar.-** Guarda en una base de datos el sonido asignado para la palabra traducida. Cuando se presiona este botón se muestra una ventana que indica que a la palabra se le ha asignado un archivo de voz (Figura 3.9).

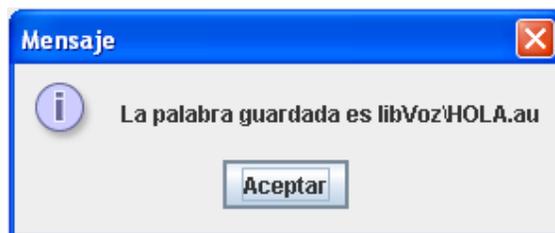


Figura 3.9 Ventana indicadora de Sonido Guardado

### **Quitar**

Esta opción permite eliminar una palabra traducida, tanto del archivo de texto como del archivo de sonido. Al dar clic sobre esta opción se abrirá una ventana, la cual contiene una lista ordenada por orden alfabético de todas las palabras guardadas en la base de datos y se mostrará los códigos que corresponden a la palabra seleccionada. (Figura 3.10)

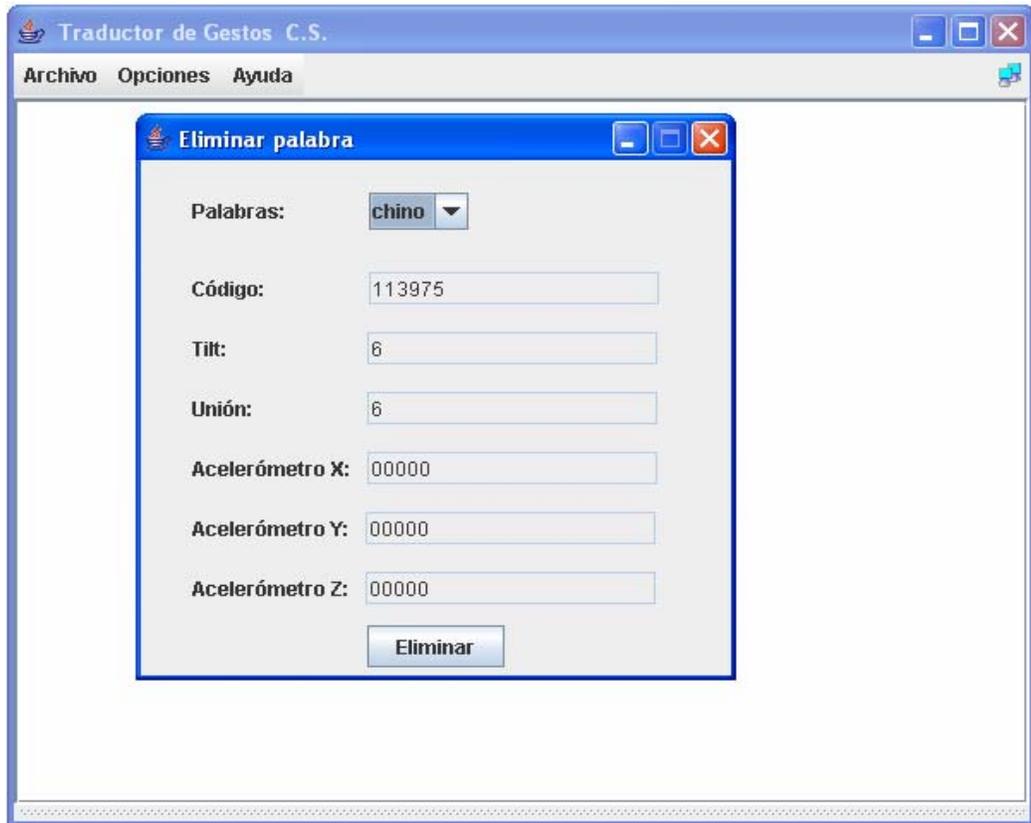


Figura 3.10 Ventana de la opción Quitar (Eliminar palabra)

Luego de haber escogido una palabra para eliminar, debemos presionar el botón eliminar para que ejecute la instrucción, instantáneamente se mostrara una ventana indicando que la palabra ha sido eliminada (Figura 3.11).

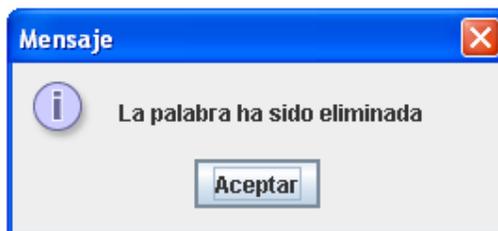


Figura 3.11 Ventana indicadora de palabra eliminada

## Editar

La Opción editar me permite modificar o asignarle sonido a una palabra ya ingresada (Figura 3.12).

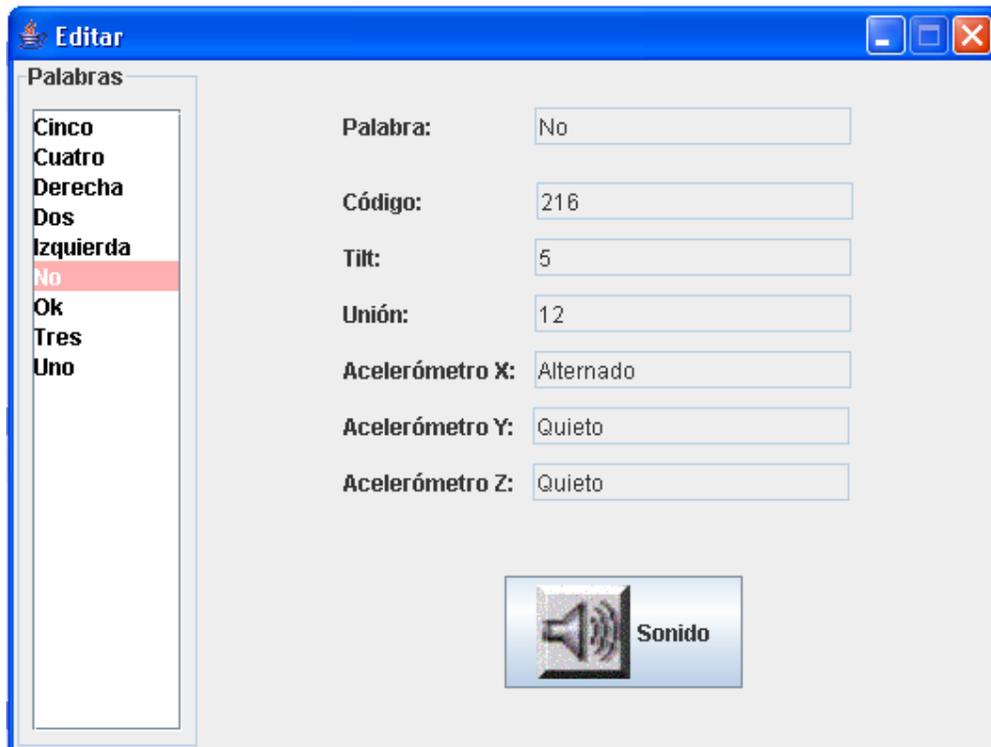


Figura 3.12 Ventana de la opción Editar

La ventana correspondiente presenta una lista de palabras con sus respectivos códigos, una vez escogida la palabra a la cual le deseamos editar el sonido debemos dar clic sobre el botón que tiene la figura de un altavoz.

Luego de esto aparecerá la ventana para grabar sonido y tendrá las mismas características que la ventana de “Grabar sonido” de la opción “Agregar”.

### Depurar

Esta opción muestra por pantalla el monitoreo de los sensores, esta opción fue implementada para que el usuario pueda verificar los valores recibidos por el programa y verificar algún error de recepción o de codificación (Figura 3.13).



Figura 3.13 Ventana de la opción Depurar (Datos)

A través de esta ventana se puede verificar el estado de los sensores, realizando movimientos con la mano y comprobando por pantalla si cambia el estado de los sensores. En el subcapítulo 3.1.2 se tratará sobre como el usuario puede detectar si el guante está en optimas condiciones.

### **Sonido**

Esta casilla de verificación permite escoger dos tipos de modo para la traducción: con sonido o sin sonido. Podemos cambiar de opción simplemente dando clic sobre la casilla correspondiente “Sonido” (Figura 3.14).

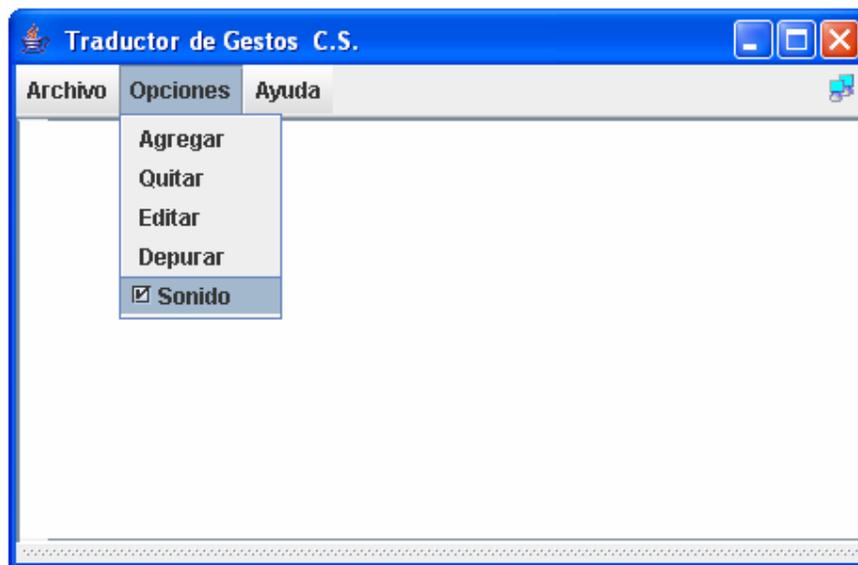


Figura 3.14 Ventana de la opción Sonido

## Ayuda

Muestra información sobre el diseño del software (Figura 3.15).

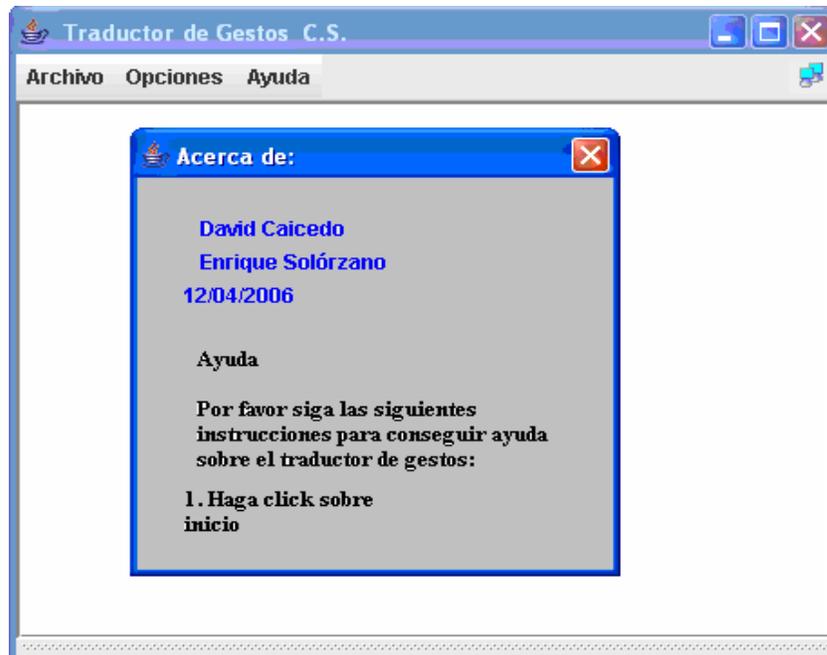


Figura 3.15 Ventana de Ayuda

### 3.1.2 Manual de usuario

Como ya es de conocimiento para nosotros el traductor esta conformado por dos partes primordiales: el guante con sus sensores y el programa traductor de gestos como tal. Es por ello que en esta sección se pretenderá dar una guía de uso del traductor teniendo como base el subcapítulo anterior.

## **Guante Electrónico**

Para un correcto uso del guante, se ha propuesto una lista de pasos que optimizarán el funcionamiento y vida útil del guante electrónico, estos pasos son:

### 1. Chequear que la batería esté cargada

Verificar que las baterías se encuentran cargadas en su valor máximo.

### 2. Chequear que la batería esté conectada al guante

Conectar el hembra y macho del circuito de la batería, al resto del circuito del guante.

### 3. Chequear que los sensores estén conectados correctamente

Verificar que todos los conectores de los sensores se encuentren en el lugar correspondiente.

### 4. Encender el guante electrónico

Poner el SWITCH en el extremo en que está marcado ON.

5. Chequear que el led de estado de los sensores este apagado

Este led indica que hay algún problema en la conexión de los sensores, y por ende deberá ser revisado antes de continuar para evitar un malfuncionamiento del equipo.

6. Chequear que esté conectado el dispositivo de interfase al puerto COM1 del computador.

Debe estar conectado este dispositivo, para que exista transferencia de información entre el receptor inalámbrico (incluye receptor + RS-232) y el computador. (Figura 3.16)

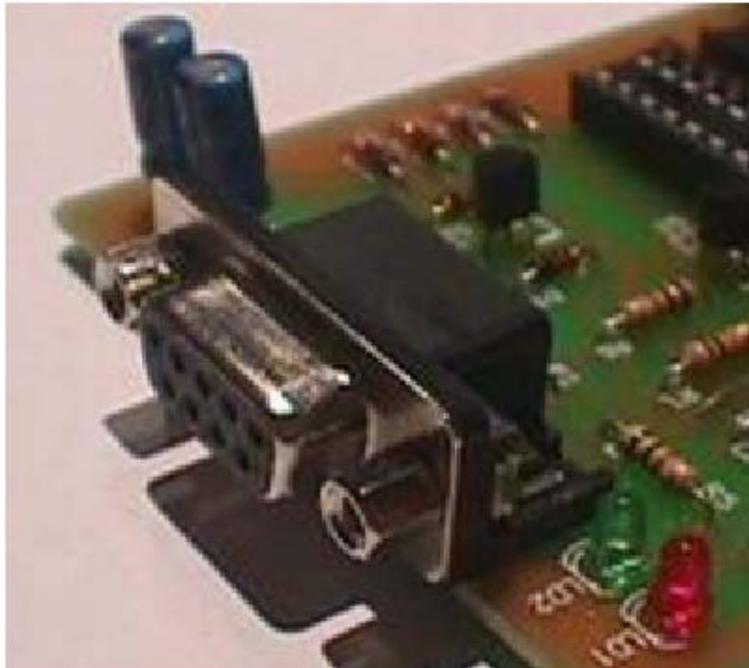


Figura 3.16 Conexión del dispositivo de interfase al puerto COM1

## 7. Calibrar el guante electrónico

Se necesita calibrar el guante al inicio de cada uso. Esto se hace necesario debido a que todas las personas no flexionan sus dedos hasta un mismo ángulo, ni tampoco sus dedos poseen la misma longitud. Haciendo esta calibración se asegura la mayor precisión para la medición de las posiciones de los dedos.

La manera de calibrar el guante es mantener presionado el botón calibrar, por un tiempo no menor a 5 segundos. Durante este tiempo el usuario tendrá que flexionar completamente sus dedos y su muñeca para poder el microcontrolador establecer los extremos de voltajes de cada sensor.

## **Traductor de Gestos C.S.**

Después de seguir estos pasos referentes al guante electrónico seguir los pasos de utilización del Programa “Traductor de Gestos C.S”.

### 1. Abrir el programa Traductor

Esto se lo puede lograr de dos formas básicas: la primera es crear un acceso directo al programa desde escritorio y ejecutarlo desde el mismo o la otra

manera, que consiste en ir y buscar el software en la lista de programas instalados en la maquina.

## 2. Chequear el icono indicador de conexión

Este icono situado en el lado derecho de la barra de herramientas, indica si el guante electrónico está o no transmitiendo datos. (Figura 3.17)

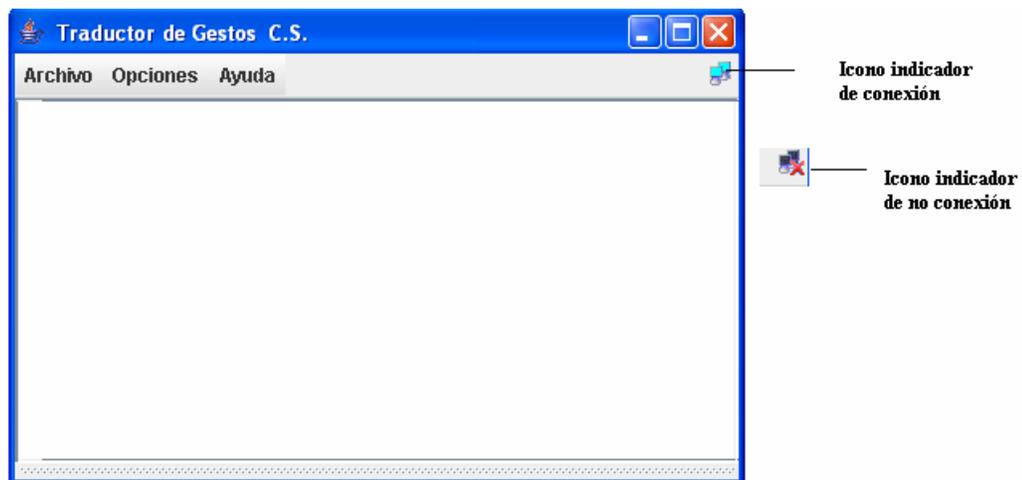


Figura 3.17 Icono indicador de conexión

3. Comenzar a utilizar las opciones de menú indicadas y explicadas en el subcapítulo anterior según los requerimientos del usuario.

Es importante que después de utilizar el guante se cierre el programa y que el guante sea apagado, para evitar posibles problemas.

### **3.2 Resultados obtenidos**

A través de este proyecto se ha obtenido una vía de comunicación simple de una persona privada de hablar. Entrando en mayor detalle se logró diseñar e implementar un traductor con las siguientes características:

#### **Guante electrónico**

- Fácilmente desarmable para mejorar el posicionamiento de los sensores.
- Tamaño reducido y ergonómico para la mano.
- Transmisión inalámbrica con un alcance 20 m.
- Capacidad de montar y desmontar el microcontrolador PIC 16F876A, para mejoras en la programación del mismo.
- Indicadores LEDS del estado del guante.
- Interfase de transmisión de datos de alta velocidad.

## **Traductor de gestos C.S.**

- Capacidad de diferenciar y traducir más de 50 palabras o frases, para cualquier persona mayor a 10 años debido a la localización de los sensores.
- Capacidad de agregar a la base de datos más gestos con su respectivo significado
- Capacidad de eliminar una palabra creada por el usuario.
- Capacidad de depurar el programa para poder apreciar los datos que están llegando al computador.
- Programa bajo la plataforma de JAVA, con una facilidad de cambios en el código fuente para mejoras en el mismo.

## CONCLUSIONES Y RECOMENDACIONES

1. Después de haber culminado este proyecto de investigación y de desarrollo nos atrevemos a decir que nuestro guante electrónico provee de un medio portátil, ligero y fácil de usar para comunicarse.
2. Este guante podría ser utilizado para otras aplicaciones tales como: control remoto de una mano o brazo mecánico (robótico), aplicaciones de realidad virtual, un Mouse indicador para PC entre otros.
3. En la parte concerniente al traductor de gestos (software), se constituye en una herramienta simple para personas discapacitadas del habla para poder transmitir algún mensaje muy simple para nosotros pero con un verdadero problema para ellos.
4. En futuras instancias, se podría incrementar la precisión de este dispositivo con sensores más especializados para detectar la flexión de los dedos, posicionamiento, etc. Además de otras clases de sensores tales como: giroscopio, para sensar la orientación del usuario.

5. Todo este sistema podría ser simplificado al implementarse en un controlador más sofisticado, incrementando su capacidad de procesamiento y disminuyendo el volumen total. Esto nos permitiría incorporarlo como un dispositivo totalmente portátil, al adicionarle una pantalla o incluso un sintetizador de voz para su funcionamiento.

## ANEXO

- ANEXO1: LISTA DE MATERIALES:

1 Transmisor TLP 434A

1 Receptor RLP 434

1 Acelerómetro LIS3L02AS

6 Sensores piezoeléctricos de curvatura FLEX01

2 Sensor de inclinación Z2205

1 Microcontrolador Pic 16F876A

1 Multiplexor analógico CD4067BE

2 Reguladores de Voltaje LM7805

2 Baterías de celular de Li-Ión 3.7V

1 Cargador para las baterías

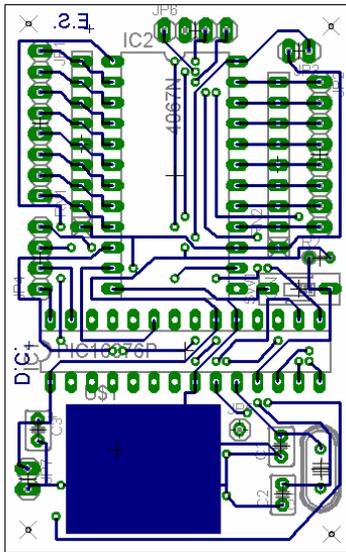
1 Circuito integrado de interfases Max 232CPE

1 Cristal de cuarzo de 4Mhz

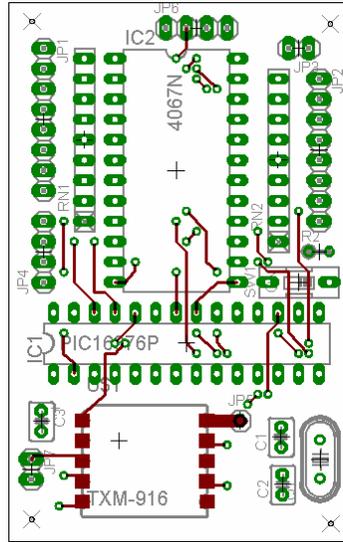
15 Resistencias de 10K

- 1 Botonera
- 1 Diodo led ultra brillante
- 5 Capacitores de 1uF
- 3 Placas de circuito impresos (Acelerómetro, Transmisor y Receptor)
- 7 Conectores de pines de 1 fila \* 8 (hembra y macho)
- 1 Conector DB9 con cable incluido
- 1 Fuente DC de 6V
- 1 metro de cable ELIAX
- 8 Ojalotes para tela
- 10 Broches para tela
- 1 metro de cable UTP
- 1 metro de tela licra

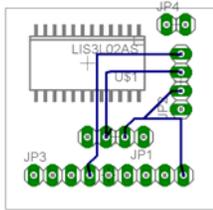
- ANEXO2: CIRCUITOS IMPRESOS



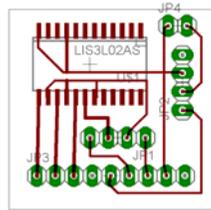
TX Vista Inferior



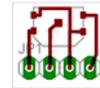
TX Vista Superior



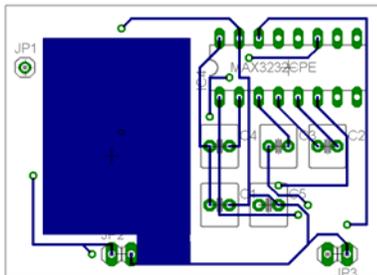
Acelerómetro  
Vista Inferior



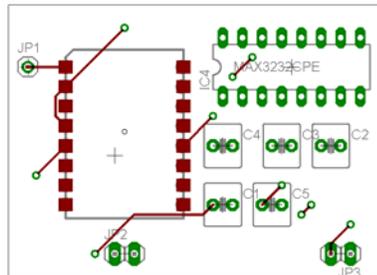
Acelerómetro  
Vista Superior



Inclinación



RX Vista Inferior



RX Vista Superior

## BIBLIOGRAFÍA

### Referencias de libros:

- RAMOS, Guillermo y HERNÁNDEZ, Jorge. Electrónica Industrial y Automatización. CEKIT 2002
- REYES, Carlos, Microcontroladores PIC 16F8XXX. Microchip 2004

### Referencias de documentos Inéditos

- POLO Jesus. *“Arquitectura de microcontroladores”*, Diciembre 1997.

### Referencia de páginas de Internet:

- Sun Microsystems, Inc., <http://www.javasoft.com>, Copyright 1994-2006 , consultada 24 de enero del 2006
- nilda@avantel.net,  
<http://www.adiosalasordera.com/singLang/tabla%20de%20lenguaje.htm>, consultada 15 de marzo del 2006
- HERRAMIENTAS TECNOLÓGICAS PARA LA AYUDA DE LAS PERSONAS SORDAS, <http://lenguajedemanos.info>, consultada 10 de febrero del 2006