



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD INFORMÁTICA Y ELECTRÓNICA

INGENIERÍA ELECTRÓNICA CONTROL Y REDES INDUSTRIALES

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL PARA MANO
ROBÓTICA, UTILIZADA PARA COMUNICACIÓN VISUAL ASL”**

TESIS DE GRADO

Previo a la obtención del título de

“INGENIERO EN ELECTRÓNICA CONTROL Y REDES INDUSTRIALES”

Presentado por:

EDISON FERNANO RUIZ ANDINO

RIOBAMBA-ECUADOR

2013

A mi familia por todo el apoyo que me dieron durante toda mi vida de estudiante.

A mis hermanos, por sus palabras de aliento en momentos de desesperanza.

A mis compañeros y amigos con quienes compartimos buenos momentos en las aulas de estudio y fuera de ellas.

Edison

Este trabajo está dedicado a mis padres Wilson y María, por su paciencia y gran apoyo durante todo el tiempo de mi vida estudiantil en la Politécnica, por ellos llegue a terminar mi carrera, gracias a ellos cambie mi destino.

A mis dos ángeles, que siempre los llevaré en mi mente y corazón.

Edison

FIRMAS RESPONSABLES Y NOTA

NOMBRES	FIRMA	FECHA
Ing. Iván Menes DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Paúl Romero DIRECTOR DE ESCUELA ING. ELECTRÓNICA CONTROL Y REDES INDUSTRIALES	_____	_____
Ing. Paúl Romero DIRECTOR DE TESIS	_____	_____
Dr. Geovanny Vallejo MIEMBRO DEL TRIBUNAL	_____	_____
Tlgo. Carlos Rodríguez DIRECTOR CENTRO DE DOCUMENTACIÓN	_____	_____
NOTA DE LA TESIS	_____	

“Yo, Edison Fernando Ruiz Andino, soy responsable de la ideas, doctrinas y resultados expuestos en esta tesis; y, el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO”

Edison Fernando Ruiz Andino

AUTOR

ÍNDICE DE ABREVIATURAS

ASL	American SignLanguage
CPU	Unidad Central de Proceso
CAD	Conversor Analógico/Digital
CDA	Conversor Digital/Analógico
CMOS	Semiconductor complementario de óxido metálico
CI	CircuitoIntegrado
CMRR	Common Mode Rejection Ratio
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EPROM	Erasable Programmable Read-Only Memory
IEEE	Instituto de Ingenieros Eléctricos y Electrónicos
GDL	Grados de libertad
PC	Computadora Personal
PIC	Programmable Interface Controller
PCB	Printed Circuit Board
PROM	Programmable Read-Only Memory
PWM	Pulse-Width Modulation
RAM	Random Access Memory
RISC	Procesador con un Conjunto Reducido de Instrucciones
ROM	Read-Only Memory
RTD	DispositivoTermoResistivo
SP	Set Point

UCP	Unidad Central de Proceso
μC	Microcontrolador
HMI	Supervisory Control And Data Acquisition
SCADA	Human Machine Interface

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

CAPÍTULO I:

MARCO REFERENCIAL

Introducción

1.1. Antecedentes..... 20

1.2. Justificación..... 21

1.3. Objetivos..... 22

1.3.1 General..... 22

1.3.2 Específicos.....	22
1.4. Hipótesis.....	24

CAPÍTULO II:

MARCO TEÓRICO

Introducción

2.1. Lenguaje de señas.....	25
2.1.1. Clasificación de las lenguas de señas.....	27
2.2. Movimientos de la mano.....	29
2.2.1. Anatomía de la mano.....	29
2.2.2. Dedos.....	30
2.2.3. Articulaciones y huesos de la mano.....	31
2.2.3.1 Función.....	32
2.2.3.2 Estructura de la mano.....	32
2.2.3.3 Huesos de la mano.....	33
2.3 El Microcontrolador ATMEGA16.....	37

2.4 Servo motor HITEC HS-311.....	40
2.5 Modulación de ancho de pulso (PWM).....	41
2.6 Prótesis y manos robóticas.....	42
2.6.1. Imitar el movimiento natural de la mano.....	43
2.6.2. Prótesis estéticas.....	44
2.6.3 Prótesis funcionales.....	44
2.6.4. Prótesis mecánicas.....	45
2.6.5. Prótesis eléctricas.....	46
2.7. Manos robóticas.....	47
2.7.1 La mano NTU.....	47
2.7.2 La mano Stanford/JPL.....	49
2.7.3. Mano Utah/MIT.....	49
2.7.3. Mano Utah/MIT.....	50

CAPÍTULO III:

DISEÑO ELECTRÓNICO

Introducción

3.1. Diseño de bloques.....	50
3.2. Análisis de requerimientos.....	50
3.2.1 Diseño del actuador.....	51
3.2.2 Microcontrolador.....	52
3.2.3 Comunicación RS-232.....	54
3.2.4 Adaptador USB a Serial Omega DB).....	57
3.2.5 MAX 232.....	59
3.2.6. Algoritmos y software de control en el microcontrolador.....	62
3.2.6.1. Proceso en el Atmega16A.....	63
3.2.7. Algoritmos y software de control SCADA.....	65

CAPÍTULO IV:

IMPLEMENTACIÓN

Introducción

4.1. Mano robótica.....	68
-------------------------	----

4.2. Servomotor.....	70
4.3. Armado de la mano.....	71
4.4. Diseño del circuito de control.....	72
4.4.1 Programación en AtmelStudio 6.1.....	75
4.5 Software de control SCADA.....	77
4.5.1 Líneas de código.....	80

CAPÍTULO V:

ANÁLISIS Y RESULTADOS

Introducción

5.1. Pruebas de funcionamiento.....	85
5.2 Análisis y tabulación de los datos.....	77

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

ANEXOS

BIBLIOGRAFÍA

ÍNDICE DE FIGURAS

Figura II.1 Símbolos de cada una de las letras.....	30
Figura II.2 Axisas de una mano.....	31
Figura II.3 Huesos de la mano.....	34
Figura II.4 Huesos de la mano separados.....	35
Figura II.5. Prótesis de la mano.....	43
Figura II.6 Prótesis cosméticas.....	44
Figura II.7. Prótesis por debajo del codo.....	46
Figura II.8 Mano eléctrica MyoHand de la empresa Otto Bock.....	47
Figura II.9 Mano NTU.....	47
Figura II.10 Mano stanford/JPL.....	48
Figura II.11 Mano Utah.....	49
Figura II.12 Mano Marcus.....	49
Figura III.1 Diagrama de bloques del SW y HW.....	50
Figura III.2 Servomotor HS-311.....	51

Figura III.3 Acoples que se utilizarán en el servomotor.....	51
Figura III.4 Atmega16A.....	52
Figura III.5 Comunicación serial micro vs terminal.....	53
Figura III.6 Conector DB9 hembra.....	54
Figura III.7 Adaptador USB a Serial.....	57
Figura III.8 Circuito integrado MAX 232.....	59
Figura III.9 Pines de conexión del MAX 232.....	60
Figura III.10 Diagrama de flujo del microcontrolador.....	64
Figura III.11 Diagrama de flujo SCADA.....	66
Figura III.12 Interfaz de usuario.....	67
Figura IV.1 Partes de la mano robótica.....	69
Figura IV.2 Dimensiones del servomotor HITEC HC-311.....	70
Figura IV.3 Servomotor con accesorios.....	70
Figura IV.4 Prototipo de mano robótica.....	71
Figura IV.5 Prototipo terminado.....	72

Figura IV.6	Circuito de control.....	73
Figura IV.7	PCB.....	74
Figura IV.8	Vista 3D.....	74
Figura IV.9	Controles comunes de Visual Basic 2010.....	78
Figura IV.10	Botones creados.....	79
Figura IV.11	Selección del puerto.....	79
Figura IV.12	Serial Port.....	80
Figura V.1	Ondas generadas por el microcontrolador.....	86
Figura V.2	Letra J generada por el prototipo.....	86
Figura V.3	Letra L generada por el prototipo.....	87
Figura V.4	Letra O generada por el prototipo.....	87
Figura V.5	Letra W generada por el prototipo.....	88

ÍNDICE DE TABLAS

Tabla II.1 Características de los servos HITEC-HS311.....	41
TablaII.2 Señal de ancho de pulso modulado.....	42
Tabla III.1 Pines asignados del DB9.....	56
Tabla III.2 Niveles de voltaje del DB9.....	57
Tabla III.3 Niveles de voltaje del MAX232.....	62
Tabla V.1 Tiempo de respuesta.....	88

INTRODUCCIÓN

La mano humana constituye un medio de comunicación en personas con deficiencia auditivas, pudiendo expresar letras, palabras o con la ayuda de gestos pueden expresar emociones o ideas. Existen muchos medios para el aprendizaje de las personas con esta discapacidad una de las primeras etapas es el conocimiento del lenguaje de señas, llamado también alfabeto dactilológico.

En la provincia de Chimborazo existen 746 personas con discapacidades auditivas y de lenguaje, habiendo en total en el Ecuador 213.000 personas con esta deficiencia.

Esto ha desencadenado una serie de estudios de esta parte del cuerpo humano para determinar las características esenciales y de movilidad.

Debido a la cantidad de huesos y falanges que contiene cada una de las articulaciones es necesario un análisis previo para establecer las pautas que posteriormente ayudaran a la construcción del prototipo.

Del análisis dependerá los grados de libertad que posea el prototipo ha desarrollarse siendo capaz de realizar el alfabeto dactilológico e interactuar con la persona sordomuda.

La presente tesis de grado detalla las etapas del diseño y construcción de la mano robótica que será utilizada en la comunicación con personas especiales que carecen del sentido del oído utilizando el lenguaje ASL; está dividida en seis capítulos:

- ✓ El capítulo uno describe la problematización, la justificación, los objetivos, y la hipótesis planteada en la tesis de grado.
- ✓ El capítulo dos detalla la revisión bibliográfica y linkográfica de todos los elementos mecánicos y electrónicos que se utilizaron en el diseño y la construcción de la mano robótica.
- ✓ El capítulo tres resume las fases del diseño electrónico de la mano robótica.
- ✓ En el capítulo cuatro se hace especial énfasis en la implementación de la etapa de control electrónico de la mano robótica.
- ✓ Y finalmente el capítulo cinco detalla el procedimiento utilizado para la demostración de la hipótesis planteada en la presente tesis de grado.

CAPÍTULO I

MARCO REFERENCIAL

En este capítulo se presenta a grandes rasgos el problema que existe en la comunicación entre personas especiales que carecen del sentido del oído, los antecedentes, la justificación, los objetivos, la hipótesis y la solución a este problema mediante el diseño y la construcción de la mano robótica.

1.1. ANTECEDENTES

En el mundo en el que vivimos todos nos encontramos inmersos en diversos tipos de problemas, entre los cuales se encuentran los de tipo social, refiriéndose exactamente a las discapacidades que poseen algunas personas, ya sean estas sensoriales, psicológicas, mentales o físicas, de tal manera que las personas se encuentran limitadas en sus actividades cotidianas.

Hoy en día, gracias al pensamiento inteligente de muchas personas de gran importancia

surgió la idea de denominar de una manera diferente a este gran grupo de personas que posee discapacidad de tal forma que se evite la negatividad y el impacto psicológico que surge al llamarlos discapacitados, por esto se propuso llamarlos hombres y mujeres con diversidad funcional, tomando en cuenta que estas personas forman parte del desarrollo del país, dándoles así la oportunidad de colaborar en el crecimiento económico del país.

Por esta razón se han creado centros especializados para estas personas con el fin de apoyarles en el desarrollo de sus capacidades físicas, psicológicas y académicas, las cuales acogen a hombres y mujeres con diversidad funcional, estos institutos cuentan con profesionales capacitados para tratar estos casos de discapacidad.

La sordera es la dificultad o la imposibilidad de usar el sentido del oído debido a una pérdida de la capacidad auditiva parcial (hipoacusia) o total (cofosis), y unilateral o bilateral. Así pues, una persona sorda será incapaz o tendrá problemas para escuchar, ésta puede ser un rasgo hereditario o puede ser consecuencia de una enfermedad, traumatismo, exposición a largo plazo al ruido, o medicamentos agresivos para el nervio auditivo.

El lenguaje ASL, son formas estandarizadas de lenguas de señas han sido usadas en Italia desde el siglo XVII y en Francia desde el siglo XVIII con el objetivo de instruir a los sordos. La lengua de señas francesa antigua fue desarrollada y usada en París por el Abbé de l'Épée en su escuela para sordos. Estas lenguas siempre fueron modeladas siguiendo las lenguas de señas naturales ya en uso por las culturas sordas en el área de origen, frecuentemente, con adiciones gramaticales que reflejaban las lenguas habladas. Los pueblos indígenas de las planicies norteamericanas usaban una lengua de señas para

intercomunicarse, ya que no compartían la misma lengua hablada. Su influencia en ASL, si existe, es desconocida. Fuera de la costa de Massachusetts, en la isla de Martha's Vineyard en el siglo XVIII, la población tenía un porcentaje mayor de sordera que en el resto de Estados Unidos continental debido al efecto fundador y el aislamiento geográfico. La lengua de señas de Martha's Vineyard era bien conocida por casi todos los isleños ya que muchas familias tenían miembros sordos. Eso permitía a casi todos la oportunidad de tener contacto frecuente con la lengua de señas a la edad en que aprender idiomas es más fácil.

1.2. JUSTIFICACIÓN

El presente proyecto de tesis tiene como fin, implementar un sistema de control para manipular una mano robótica capaz de interpretar el lenguaje a señal ASL, para ayudar a las personas con discapacidad auditiva, ya que no existe muchas personas que conozcan este tipo de lenguaje y por esta razón se le hace complicado la comunicación con las demás personas tomando en cuenta que en las instituciones de educación no existe ayuda para personas con esta discapacidad.

El proyecto basará su funcionamiento en el análisis de las necesidades de las personas que tienen esta discapacidad auditiva, es decir que cualquier palabra que se necesite transmitir hacia la persona que tiene esta discapacidad, constará con una interfaz para escribir palabras, que a su vez serán traducidas a lenguaje ASL por la mano robótica.

El sistema acopla el uso de microcontroladores, los cuales se encargan de dar órdenes a la parte de potencia e interactuar con la interfaz del computador mediante el puerto de comunicación DB9.

1.3. OBJETIVOS

1.3.1 Objetivo general

- ✓ Diseñar e implementar un sistema de control para mano robótica, utilizada en comunicación visual mediante lenguaje ASL.

1.3.2Objetivos específicos

- ✓ Investigar la estructura fisiológica de la mano, para determinar la capacidad del movimiento que tiene.
- ✓ Estudiar los tipos de elementos y dispositivos que se pueden utilizar para imitar el movimiento de la mano robótica.
- ✓ Implementar la Mano Robótica.
- ✓ Diseñar los circuitos e interfaces físicas, para controlar la recepción de las señales que servirán para controlar los movimientos de la mano robótica.
- ✓ Diseñar la interfaz SCADA, para la manipulación de la mano robótica mediante el PC
- ✓ Realizar las pruebas de funcionamiento.
- ✓ Documentar las capacidades y limitaciones que tendrá todo el sistema de control.

1.4. HIPÓTESIS

Con el diseño de un sistema de control para manejar de forma automática una mano robótica se dispondrá de una herramienta adicional para la comunicación mediante lenguaje ASL, la cual se utilizará para facilitar la comunicación entre personas especiales carentes de sentido auditivo

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se narra una introducción a cerca del lenguaje de señas, la definición de las mismas, se hará énfasis en el lenguaje ASL, tipos de manos que se han construido.

También incluye una breve descripción de los elementos con mayor relevancia en cuanto al diseño y construcción de la mano robótica.

2.1. Lenguaje de señas.

Aún cuando hoy en día las lenguas de señas se utilizan casi exclusivamente entre personas con sordera, su origen es tan antiguo como el de las lenguas orales o incluso más, en la

historia de la Humanidad, y también han sido y siguen siendo empleadas por comunidades de oyentes. De hecho, los amerindios de la región de las Grandes Llanuras de América Norte, usaban una lengua de señas para hacerse entender entre etnias que hablaban lenguas muy diferentes con fonologías extremadamente diversas. El sistema estuvo en uso hasta mucho después de la conquista europea. Otro caso, también amerindio, se dio en la isla de Manhattan, donde vivía una tribu única en la que un gran número de sus integrantes eran sordos, debido a la herencia de desarrollo de un gen dominante, y que se comunicaban con una lengua gestual. Un caso similar se desarrolló en la isla de Martha's Vineyard al sur del estado de Massachusetts, donde debido al gran número de sordos se empleó una lengua de señas que era de uso general también entre oyentes, hasta principios del siglo XX. Pese a esto, no existen referencias documentales sobre estas lenguas antes del siglo XVII. Los datos que se poseen tratan, sobre todo, de sistemas y métodos educativos para personas sordas.

En el siglo XVI Jerónimo Cardano, médico de Padua, en la Italia norteña, proclamó que las personas sordas podrían hacerse entender por combinaciones escritas de símbolos asociándolos con las cosas a que ellos se referían. En el año 1620 Juan de Pablo Bonet publica su *Reducción de las letras y Arte para enseñar á hablar los Mudos*, considerado como el primer tratado moderno de Fonética y Logopedia, en el que se proponía un método de enseñanza oral de los sordos mediante el uso de señas alfabéticas configuradas unimanualmente, divulgando así en toda Europa, y después en todo el mundo, el alfabeto manual, útil para mejorar la comunicación de los sordos y mudos. En 1817 Gallaudet fundó la primera escuela de la nación para las personas sordas, en Hartford, Connecticut, y

Clercse convirtió en el primer maestro sordo de lengua de señas de los Estados Unidos. Pronto las escuelas para las personas sordas empezaron a aparecer en varios estados. Entre ellos la Escuela de Nueva York que abrió sus puertas en 1818. En 1820 otra escuela se abrió en Pennsylvania, y un total de veintidós escuelas se habían establecido a lo largo de los Estados Unidos por el año 1863.

En esta obra aparece un abecedario ilustrado mediante grabados calcográficos de los signos de las manos que representan las letras del alfabeto latino. Del tratado sobre Reducción de las letras y Arte para enseñar a hablar los Mudos se hicieron traducciones a las principales lenguas de cultura. Sobre la base del alfabeto divulgado por Bonet, Charles-Michel de l'Épée publica el siglo XVIII su alfabeto, que básicamente es el que ha llegado hasta la actualidad, siendo conocido internacionalmente como alfabeto manual español.

2.1.1. Clasificación de las lenguas de señas

Las lenguas de señas modernas, al igual que las lenguas orales, están sujetas al proceso universal de cambio lingüístico que hace que evolucionen con el tiempo y eventualmente una misma lengua puede evolucionar en lugares diferentes hacia variedades diferentes. De hecho, muchas de las lenguas modernas de señas pueden ser clasificadas en familias:

Lenguas originadas en la antigua lengua de señas de Kent, usada durante el siglo XVII, que dio lugar a la lengua de señas usada en Martha's Vineyard (Massachusetts) y que influyó de manera importante en la lengua de señas americana (ASL).

Lenguas originadas en la antigua lengua de señas francesa. Estas lenguas se remontan a las

formas estandarizadas de lenguas de señas usadas en España, Italia y Francia desde el siglo XVIII en la educación de los sordos. En concreto, la antigua lengua de señas francesa se desarrolló en el área de París, gracias a los esfuerzos del abad Charles Michel de l'Épée en su escuela de sordos. En tiempos modernos esta lengua ha dado lugar a otras varias, como la lengua de señas americana (ASL), la lengua de señas mexicana (LSM), la moderna lengua de señas francesa (LSF), la lengua de señas italiana (LIS), la lengua de señas de Irlanda (IRSL) y las lenguas de señas ibéricas (que muestran similitudes con la antigua lengua de señas francesa, pues no en vano los educadores españoles de sordos del siglo XIX se formaron en el Instituto Nacional de Sordomudos de París), derivándose a dos o tres lenguas diferentes con cierta inteligibilidad mutua, la lengua de señas española (LSE), la lengua de señas catalana (LSC) y la lengua de señas valenciana (LSCV).

Lenguas originadas en la lengua de señas británica (BSL), que se diversificó durante el siglo XIX dando lugar a la lengua de señas australiana (Auslan), la lengua de señas de Nueva Zelanda (NZSL) y la lengua de señas de Irlanda del Norte (NIRSL).

Lenguas originadas en la lengua de señas alemana (DGS), que se considera relacionada con la lengua de señas de la Suiza alemana (DSGS), la lengua de señas austríaca (ÖGS) y probablemente la lengua de señas israelí (ISL).

ASL también es usada (a veces juntamente con lenguas de señas indígenas) en Filipinas, Malasia, Singapur, Hong Kong, República Dominicana, El Salvador, Haití, Puerto Rico, Costa Rica, Costa de Marfil, Burkina Faso, Ghana, Togo, Benín, Nigeria, Chad, Gabón, República Democrática del Congo, República Centroafricana, Mauritania, Kenia,

Madagascar y Zimbabue.



Figura II.1 Símbolos de cada una de las letras

Como toda lengua de señas, su gramática y sintaxis son distintas de cualquier lengua hablada en su área de influencia. Aunque no ha habido una encuesta fiable para determinar el número de personas que tienen ASL como lengua materna, se estima que hay entre 500 000 y 2 000 000 sólo en Estados Unidos.

2.2. Movimientos de la mano

Las manos forman parte de las extremidades del cuerpo humano, siendo el cuarto segmento del miembro superior o torácico, están localizadas en los extremos de los antebrazos, son prensiles y tienen cinco dedos cada una. Abarcan desde la muñeca hasta la yema de los dedos en los seres humanos.

2.2.1. Anatomía de la mano

La mano humana consiste en una palma central (cuyos huesos forman el metacarpo) de la que surgen cinco dedos, está unida al antebrazo por una unión llamada muñeca (cuyos

huesos forman el carpo). Además, la mano está compuesta de varios, músculos y ligamentos diferentes que permiten una gran cantidad de movimientos y destreza.

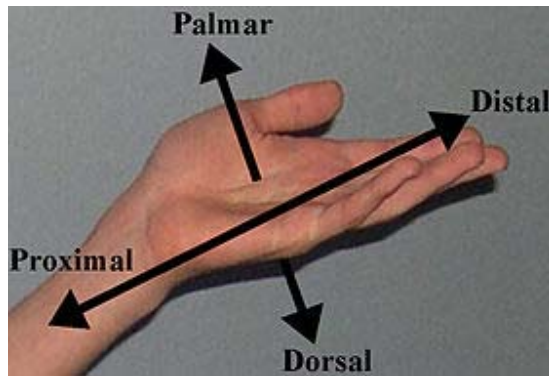


Figura II.2 Axisas de una mano

2.2.2. Dedos

Nombre de los cinco dedos de fuera hacia adentro, con la palma hacia arriba:

- ✓ Pulgar, también conocido como dedo gordo de la mano o primer dedo de la mano, está en el lado interno de la mano, paralelo al brazo. El pulgar puede rotar fácilmente 90° , no como el resto de dedos que solamente pueden rotar cerca de 45° . Los pulgares se diferencian por poder oponerse al resto de los dedos en una acción muscular conocida como oposición. Los otros cuatro dedos de la mano se localizan en el borde exterior de la palma. Estos cuatro dedos pueden ser plegados hacia la palma, esto permite sostener objetos y además agarrar otros más pequeños.
- ✓ Índice, también conocido como segundo dedo de la mano.
- ✓ Corazón, también conocido como tercer dedo de la mano, dedo medio, mayor,

cordial o grosero.

- ✓ Anular, también conocido como cuarto dedo de la mano; se le llama anular por ser el que porta el anillo de matrimonio en la cultura occidental.
- ✓ Meñique, también conocido como quinto dedo de la mano o dedo pequeño de la mano.

2.2.3. Articulaciones y huesos de la mano

La mano humana realiza funciones muy importantes como la mecánica, sensitiva o hasta comunicación. La mano es el primer órgano para la manipulación física del entorno, La punta de los dedos contiene una de las zonas con más terminaciones nerviosas del cuerpo, son la principal fuente de información táctil, es por eso que el sentido del tacto se asocia inmediatamente con las manos.

2.2.3.1 Función

La mano humana permite a las personas mantener los objetos. Se utiliza en el comer, conducir y dar la mano, e incluso se utiliza como medio de comunicación y gesto de saludo. Una mano se utiliza para las actividades críticas de todos los días.

2.2.3.2 Estructura de la mano.

La mano es una estructura de altísima complejidad, una verdadera maravilla de la bioarquitectura.

2.2.3.3 Huesos de la mano.

La mano humana tiene al menos 27 huesos: el Carpo o muñeca tiene 8; el metacarpo o palma tiene 5 y los 14 huesos restantes son digitales.

La muñeca tiene ocho huesos (los huesos carpianos), dispuestos en dos grupos de cuatro. Estos huesos encajan en una pequeña cavidad formada por los huesos del antebrazo el radio y el cúbito, si bien es de resaltar que el cúbito no se articula verdaderamente con ninguno de los huesos de la muñeca. Bajo la cara inferior del cúbito se encuentra el ligamento triangular de la muñeca, que sí se articula con los huesos.

- ✓ Los huesos de la fila proximal son, de fuera hacia adentro: el escafoides, el semilunar, el piramidal y el pisiforme
- ✓ Los huesos de la fila distal son, de fuera hacia adentro: el trapecio, el trapecoide, el grande y el ganchoso.

La palma de la mano tiene cinco huesos (los huesos metacarpianos), uno por cada dedo.

Las manos humanas contienen catorce huesos digitales, también llamados falanges: dos en el pulgar, y tres en cada uno de los otros cuatro dedos. Estos son:

- ✓ la falange distal
- ✓ la falange media y
- ✓ la falange proximal.

En conjunto forman un canal de concavidad anterior por el que se deslizan los tendones de

los músculos flexores de los dedos.

Los 8 huesos del carpo se organizan en dos filas o hileras, una superior y otra inferior. De radial a cubital la fila superior compuesta de los huesos escafoides, semilunar, piramidal y pisiforme. La fila inferior la forman el trapecio, trapezoide, hueso grande y hueso ganchoso.

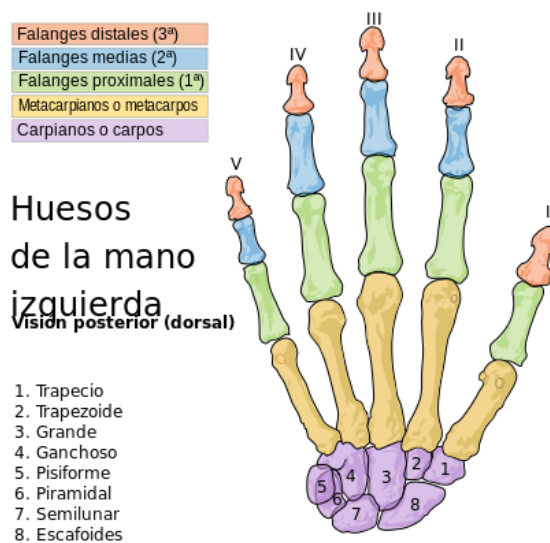


Figura II.3 Huesos de la mano

Fila superior.

Hueso escafoides: Es el más lateral, alargado de la superior e inferior y de medial a lateral.

Hueso semilunar: situado entre el escafoides y el piramidal.

Hueso piramidal: presenta una forma de pirámide cuadrangular.

Hueso pisiforme: hueso irregularmente redondeado.

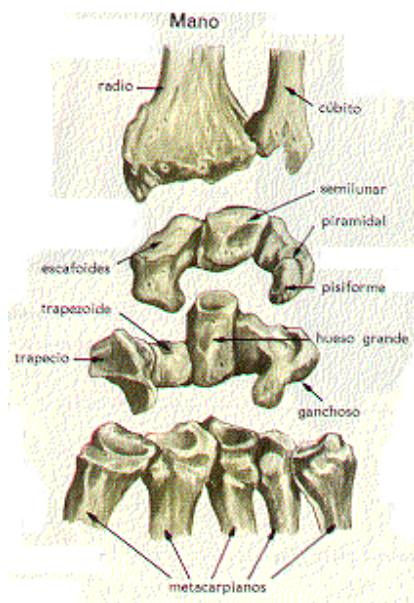


Figura II.4 Huesos de la mano separados

Fila inferior.

Hueso trapecio: el más lateral de la segunda fila.

Hueso trapezoide: situado entre el hueso trapecio y el hueso grande.

Hueso grande: es el más voluminoso de los huesos del carpo. Se describen en él una parte superior redondeada cabeza, un cuerpo y una zona intermedia denominada cuello.

Hueso ganchoso: presenta la forma de un prisma triangular. Consta de cinco caras: dos bases no articulares, una anterior y otra posterior, y tres caras articulares.

Macizo óseo carpiano, Canal carpiano y conducto carpiano.

Los ocho huesos del carpo forman en conjunto un macizo óseo que presenta 4 caras

(anterior, posterior, superior e inferior) y dos bordes (lateral y medial).

Canal Carpiano: Este canal está limitado lateralmente por los tubérculos de los huesos escafoides y trapecio, y medialmente por la eminencia del hueso pisiforme y el hueso ganchoso.

Conducto Carpiano: Por donde discurren el nervio mediano, y los tendones de los músculos flexor superficial de los dedos, flexor profundo de los dedos y flexor radial del carpo.

Metacarpo

Constituye el esqueleto de la palma y del dorso de la mano, se compone de 5 huesos largos.

Los espacios limitados entre ellos se denominan espacios interóseos.

De lateral a medial reciben el nombre de primero, segundo, tercero, cuarto y quinto metacarpiano.

Las falanges son huesos largos, presentan un cuerpo y dos extremos: La base y la cabeza de la falange.

Las falanges pueden ser llamadas: falange 1, falange 2, falange 3 o también falange, falangina y falangeta.

No obstante, las dos falanges del dedo pulgar son más voluminosas que las de los otros dedos y solo tiene dos, haciendo falta la falange 2 o falangina.

Huesos sesamoideos

Se da el nombre de huesos sesamoideos a unos pequeños huesos que presentan la forma de sésamo. En la mano existe un número variable de huesos sesamoideos, todos situados en la cara palmar. Dos son constantes y se encuentran en la cara palmar de la articulación metacarpo falángica del de los dedos índice y meñique .Con menos frecuencia se observan en las articulaciones metacarpo falángicas del dedo medio y del anular y en la articulación inter-falángica del dedo pulgar.

2.3 El Microcontrolador ATMEGA16

Microcontrolador AVR de 8 bit de alto rendimiento y bajo consumo, arquitectura RISC avanzada.

- ✓ 131 instrucciones. La mayoría de un simple ciclo de clock de ejecución.
- ✓ 32 x 8 registros de trabajo de propósito general.
- ✓ Capacidad de procesamiento de unos 16 MIPS a 16 MHz.
- ✓ Funcionamiento estático total.
- ✓ Multiplicador On-Chip de 2 ciclos

Memorias de programa y de datos no volátiles.

- ✓ 16K bytes de FLASH auto programable en sistema.

Resistencia: 1.000 ciclos de escritura / borrado, sección de código añadida opcional con bits de bloqueo independientes, programación en sistema con el programa añadido On-Chip.

Operación de lectura durante la escritura.

- ✓ 512 bytes de EEPROM.

Resistencia: 100.000 ciclos de escritura / borrado.

- ✓ 1K bytes de SRAM interna.
- ✓ Bloqueo (cerradura) programable para la seguridad del software.

Interface JTAG.

- ✓ Mantenimiento de eliminación de errores On-Chip.

Programación de FLASH, EEPROM, fusibles y bits de bloqueo a través de la interface JTAG.

- ✓ Capacidades de BoundaryScan de acuerdo con el Standard (norma) JTAG.

Características de los periféricos.

- ✓ Dos Timer/Contadores de 8 bits con prescaler separado y modo comparación.
- ✓ Un Timer/Contador de 16 bits con prescaler separado, modo comparación y modo de captura.
- ✓ Comparador analógico On-Chip.
- ✓ Timerwatchdog programable con oscilador separado On-Chip.
- ✓ Interface serie SPI maestro/esclavo.
- ✓ USART serie programable.

- ✓ Contador en tiempo real con oscilador separado.
- ✓ ADC de 10 bit y 8 canales.
- ✓ 8 canales de terminación simple
- ✓ 7 canales diferenciales sólo en el encapsulado TQFP.
- ✓ 2 canales diferenciales con ganancia programable a 1x, 10x o 200x.
- ✓ 4 canales de PWM.
- ✓ Interface serie de dos hilos orientada a byte.

Características especiales del microcontrolador.

- ✓ Reset de Power-on y detección de Brown-out programable.
- ✓ Oscilador RC interno calibrado.
- ✓ Fuentes de interrupción externas e internas.
- ✓ 6 modos de descanso: Idle, reducción de ruido ADC, Power-save, Power-down, Standby y Standby extendido.

I/O y encapsulados

- ✓ 32 líneas de I/O programables.
- ✓ PDIP de 40 pines, TQFP y MLF de 44 pines.

Tensiones de funcionamiento.

- ✓ 2.7 - 5.5V (ATmega16L).
- ✓ 4.5 - 5.5V (ATmega16).

Niveles de velocidad.

- ✓ 0 - 8 MHz (ATmega16L).
- ✓ 0 - 16 MHz (ATmega16).

2.4 Servo motor HITEC HS-311

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición

Un servomotor es un motor eléctrico que tiene la capacidad de ser controlado, tanto en velocidad como en posición.

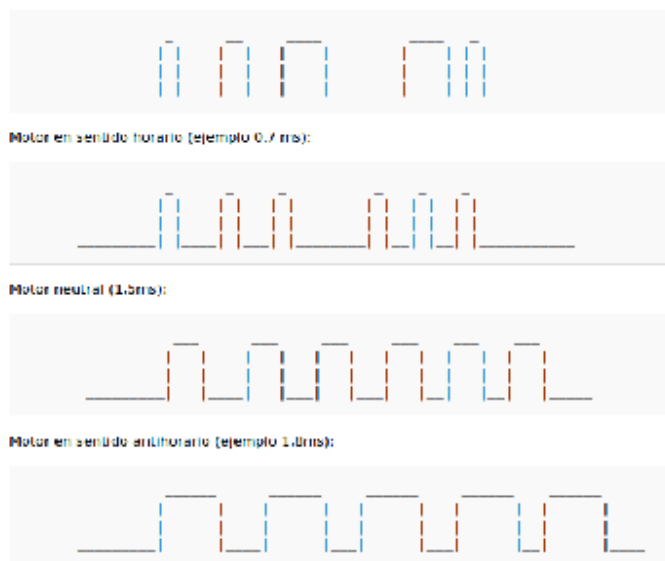
Los servos se utilizan frecuentemente en sistemas de radio control y en robótica, pero su uso no está limitado a éstos. Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

Basic Information	
Modulation:	Analog
Torque:	4.8V: 42.0 oz-in (3.02 kg-cm) 6.0V: 48.6 oz-in (3.50 kg-cm)
Speed:	4.8V: 0.19 sec/60° 6.0V: 0.15 sec/60°
Weight:	1.52 oz (43.0 g)
Dimensions:	Length: 1.57 in (39.9 mm) Width: 0.78 in (19.8 mm) Height: 1.43 in (36.3 mm)
Motor Type:	3-pole
Gear Type:	Plastic
Rotation/Support:	Bushing
Additional Specifications	
Rotational Range:	? (add)
Pulse Cycle:	20 ms
Pulse Width:	900-2100 μ s
Connector Type:	? (add)

Tabla II.1 Características de los servos HITEC-HS311

2.5 Modulación de ancho de pulso (PWM)

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los 50Hz, así las señales PWM tendrán un período de veinte milisegundos. La electrónica dentro del servomotor responderá al ancho de la señal modulada. Si los circuitos dentro del servomotor reciben una señal de entre 0,5 a 1,4 milisegundos, éste se moverá en sentido horario; entre 1,6 a 2 milisegundos moverá el servomotor en sentido anti horario; 1,5 milisegundos representa un estado neutro para los servomotores estándares.



TablaII.2 Señal de ancho de pulso modulado

2.6 Prótesis y manos robóticas

La pérdida de función en una extremidad o la amputación de la misma pueden ejercer un efecto desmesuradamente negativo en la calidad de vida. Asimismo, la asimilación de una disfunción o una discapacidad comporta un proceso igualmente traumático y a menudo la recuperación resulta larga y ardua, pues los cambios en el estilo de vida a los que obliga una situación así no se producen de la noche a la mañana.

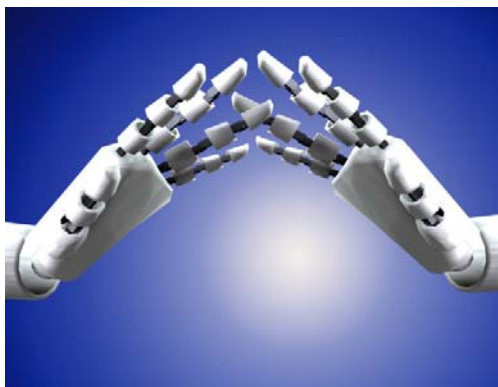


Figura II.5. Prótesis de la mano

La ingeniería ha recogido el guante y en los últimos años se han desarrollado manos robóticas con un nivel extraordinario de destreza, fuerza y sensibilidad.

2.6.1. Imitar el movimiento natural de la mano

La teoría de las sinergias analiza todos los movimientos complejos que se pueden realizar como producto de una combinación de otros básicos, por ejemplo cerrar los dedos para agarrar un objeto. Estas unidades de movimiento, o sinergias, generadas por una configuración concreta de los músculos, dependen de las características anatómicas del ser humano.

2.6.2. Prótesis estéticas

Son denominadas también prótesis pasivas, dado que sólo cubre el aspecto estético, de donde toman su nombre. Para fabricarlas se emplean de manera recurrente tres materiales: PVC rígido, látex flexible o silicona.

Estos materiales se caracterizan por ser más livianos y requieren menos mantenimiento por que tienen menos piezas móviles que otras opciones protésicas. En la figura II.6 se muestran algunas prótesis cosméticas.



Figura II.6 Prótesis cosméticas

2.6.3 Prótesis funcionales

Las prótesis o implantes funcionales, a diferencia de las anteriores, proveen cierta movilidad intentando suplir las funciones del miembro natural amputado. En cuanto a las prótesis funcionales, éstas se pueden clasificar en mecánicas o mioeléctricas, siendo estas últimas las que ofrecen una mejor solución en cuanto al número y fuerza de los agarres, aunque tiene un mayor peso y precio.

2.6.4. Prótesis mecánicas

Las manos mecánicas son dispositivos que se utilizan con la función de apertura o cierre voluntario por medio de un arnés el cual se sujeta alrededor de los hombros, parte del pecho y parte del brazo controlado por el usuario. Su funcionamiento se basa en la extensión de una liga por medio del arnés para su apertura o cierre, y el cierre o apertura se efectúa

sólo con la relajación del músculo respectivo gracias a un resorte. Estos elementos se recubren con un guante para dar una apariencia más estética, sin embargo se limita al agarre de objetos relativamente grandes y redondos ya que el guante estorba al querer sujetar objetos pequeños. El tamaño de la prótesis y el número de ligas que se requiera dependiendo de la fuerza y el material para su fabricación varían de acuerdo a las necesidades de cada persona.

La prótesis que se muestra en la figura II.7 es una prótesis mecánica con cable y un gancho doble que funciona de manera relativamente simple. El cable va de la prótesis hacia el muñón y pasa por detrás de la espalda hasta un arnés situado en el otro hombro. Cuando mueve los hombros hacia adelante, se amplía la distancia entre los omóplatos y el cable se estira varios centímetros, abriendo el gancho contra los resortes o bandas elásticas. Cuando relaja los hombros, el cable se afloja y los resortes o bandas elásticas cierran el dispositivo.



Figura II.7. Prótesis por debajo del codo

2.6.5. Prótesis eléctricas

Estas prótesis usan motores eléctricos en el dispositivo terminal, muñeca o codo con una batería recargable. Se controlan de varias formas, ya sea con un servocontrol, control con botón pulsador o botón con interruptor de arnés. En ciertas ocasiones se combinan estas formas para su mejor funcionalidad. Se usa un socket, que es un dispositivo intermedio entre la prótesis y el muñón logrando la suspensión de éste por una succión. Su adquisición y reparación son costosas, existiendo otras desventajas evidentes como son el cuidado a la exposición de un medio húmedo y el peso de la prótesis. En la figura II.8 se muestra una prótesis eléctrica de la compañía Otto Bock.



Figura II.8 Mano eléctrica MyoHand de la empresa Otto Bock

2.7. Manos robóticas

2.7.1 La mano NTU

La mano NTU, desarrollada por la Universidad Nacional de Taiwan, es una mano con 5 dedos equipados con sensores para detectar fuerza y posición, y con 17 GDL. El

dedopulgar y el índice tienen 4 GDL, mientras que los otros dedos presentan 3 articulaciones activas. Los motores, reducciones y sensores se encuentran instalados en la mano. En la figura II.9 se presenta la mano NTU



Figura II.9 Mano NTU

2.7.2 La mano Stanford/JPL

La mano Stanford/JPL (Jet Propulsion Laboratory), presenta 9 grados de libertad en los dedos y un pulgar oponente. La mano es actuada por 12 servomotores DC. Sus articulaciones son impulsadas por cables. En la figura II.10 se presenta la mano Stanford.



Figura II.10 Mano stanford/JPL

2.7.3. Mano Utah/MIT

La mano Utah/MIT, desarrollada por el grupo de investigación en agarre y manipulación de dedos, dispone de 4 grados de libertad en cada uno de sus cuatro dedos incluyendo el pulgar. La geometría de la mano robótica es antropomorfa a pesar de que el dedo pulgar es permanente en su posición. Los 16 GDL se accionan mediante un enfoque de tendones que son accionados por un sistema de 32 tendones independientes poliméricos y actuadores neumáticos. En la figura II.11 se presenta la mano Utah.



Figura II.11 Mano Utah

2.7.4 Mano Marcus

La mano Marcus, se compone de tres dedos, pulgar, índice y corazón, 2 GDL y está equipada con dos motores. La mano tiene sensores de efecto Hall para determinar la posición y un sensor de fuerza en el dedo pulgar. En la figura II.12 se presenta la mano Marcus.

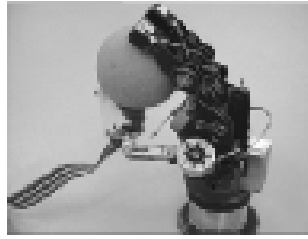


Figura II.12 Mano Marcus

CAPÍTULO III

DISEÑO ELECTRÓNICO

Después de analizar todos los mecanismos necesarios planteamos una solución, se comienza con la descripción del diseño de bloques para su implementación física. Generando un circuito(hardware) mediante un proceso de síntesis, donde la calidad del resultado dependerá del estilo descriptivo con el cual se explicará todos los componentes que intervienen.

3.1. Diseño de bloques

El siguiente diagrama explica de forma general el funcionamiento del software y hardware que controla la mano robótica.

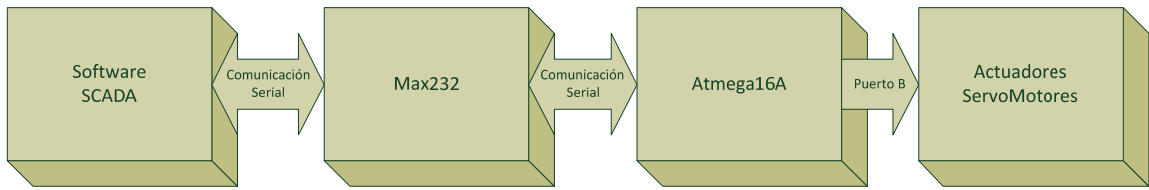


Figura III.1 Diagrama de bloques del SW y HW

3.2. Análisis de requerimientos

A continuación se detalla cada uno de los elementos utilizados en la construcción de la interfaz de comunicación que se utiliza en el proyecto de tesis.

3.2.1. Diseño del actuador

Para optar por el actuador indicado que se utilizará hay que analizar la cantidad de movimiento, la velocidad de reacción y la fuerza que se debe tener para cada uno de los dedos, debido a la finalidad propuesta, se necesita que cada uno de los dedos tenga su propio movimiento; por lo que se opta por utilizar servomotores ya que estos se ajustan a las necesidades de movimiento, fuerza y velocidad requeridas para el prototipo en cuestión.

En el mercado existen varios tipos de motores los mismos que difieren en tamaño, torque, velocidad. Por su utilización y sus prestaciones se utiliza un servomotor de marca HITEC, el modelo HS-311.



Figura III.2 Servomotor HS-311



Figura III.3 Acoples que se utilizarán en el servomotor

3.2.2. Microcontrolador

Para el diseño del controlador se implementará en un Microcontrolador ATMEGA16A, con transmisión SERIAL, del cual se va a utilizar sus interrupciones para desarrollar el controlador de 5 canales.



Figura III.4 Atmega16A

Como sabemos para variar la posición del servomotor debemos variar el ancho de pulso, el mismo que debe oscilar dentro de la frecuencia a 50Hz.

Para desarrollar el algoritmo en primera instancia se calcula el periodo de cada servomotor:

$$T = \frac{1}{F} = \frac{1}{50} = 20ms$$

Esto quiere decir que cada 20ms debe existir el tiempo en alto necesario para la posición que se requiera, como se sabe el máximo pulso que se le puede dar a un servomotor es de 2.5ms, por esta razón se declaró los siguientes valores dentro del lenguaje de programación:

```
void initialize(){
    DDRB=0B11111111; //Puerto B como salida
    TCCR0=0B00001100; //TC, CLK 256
    OCR0=77; //400Hz
}
```

Para la comunicación con el computador se utiliza el módulo SERIAL interno del Atmega16A, el mismo que se comunica directamente con cualquier programa.

Simulación

La simulación se realiza en PRITEUS 7, este programa de simulación permite cargar el archivo .HEX al Microcontrolador y así poder observar el funcionamiento para luego comprobarlo en la práctica.

Una herramienta útil que incluye este programa es un puerto virtual SERIAL, el mismo que se configura como un puerto físico, en la figura III.5 se muestra la comunicación mediante el terminal y el puerto serial del microcontrolador.

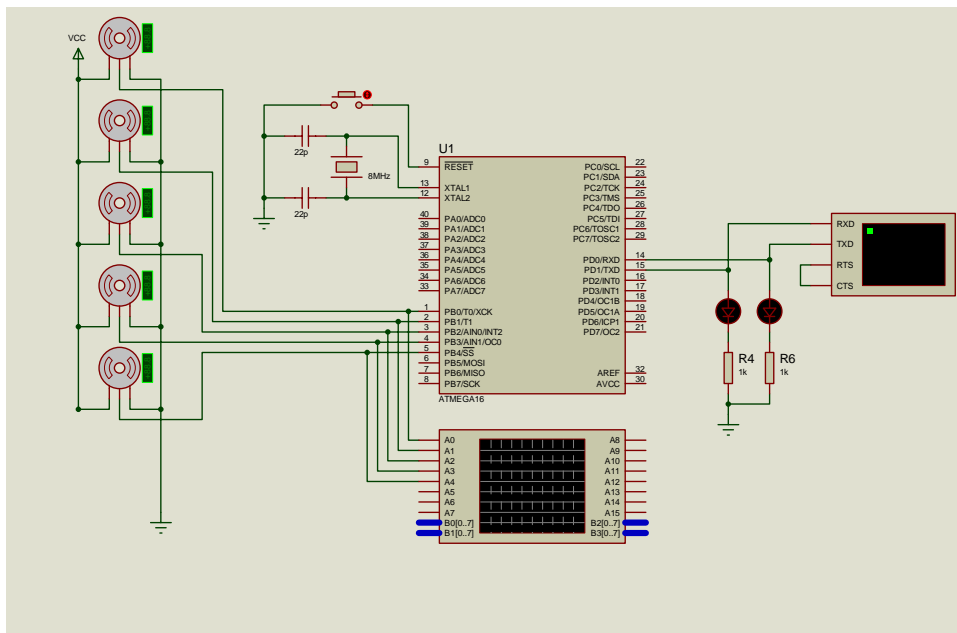


Figura III.5 Comunicación serial micro vs terminal

3.2.3 Comunicación RS-232

RS232 (Recommended Standard 232, también conocido como Electronic Industries Alliance RS232C) es una interfaz que designa una norma para el intercambio de una serie de datos binarios

entre un DTE (Equipo terminal de datos) y un DCE (*Data Communication Equipment*, Equipo de Comunicación de datos), aunque existen otras en las que también se utiliza la interfaz RS-232.

En particular, existen ocasiones en que interesa conectar otro tipo de equipamientos, como pueden ser computadores. Evidentemente, en el caso de interconexión entre los mismos, se requerirá la conexión de un DTE (*Data Terminal Equipment*) con otro DTE. Para ello se utiliza una conexión entre los dos DTE sin usar módem, por ello se llama: null módem ó módem nulo.

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DE-9, o popularmente también denominados DB-9), más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC)



Figura III.6 Conector DB9 hembra

La interfaz RS-232 está diseñada para imprimir documentos para distancias cortas, de hasta 15 metros según la norma, y para velocidades de comunicación bajas, de no más de 20 kbps. A pesar de esto, muchas veces se utiliza a mayores velocidades con un resultado aceptable. La interfaz puede trabajar en comunicación asíncrona o síncrona y tipos de canal

simplex, halfduplex o full duplex. En un canal simplex los datos siempre viajarán en una dirección, por ejemplo desde DCE a DTE. En un canal halfduplex, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos puedan viajar en la otra dirección. En un canal full duplex, los datos pueden viajar en ambos sentidos simultáneamente. Las líneas de handshaking de la RS-232 se usan para resolver los problemas asociados con este modo de operación, tal como en qué dirección los datos deben viajar en un instante determinado.

Si un dispositivo de los que están conectados a una interfaz RS-232 procesa los datos a una velocidad menor de la que los recibe deben de conectarse las líneas handshaking que permiten realizar un control de flujo tal que al dispositivo más lento le dé tiempo de procesar la información. Las líneas de "handshaking" que permiten hacer este control de flujo son las líneas RTS y CTS. Los diseñadores del estándar no concibieron estas líneas para que funcionen de este modo, pero dada su utilidad en cada interfaz posterior se incluye este modo de uso.

Señal		DB-25	DE-9 (DB-9, TIA-574)	EIA/TIA 561	Host	RJ-50	MMJ
Common Ground	G	7	5	4	4,5	6	3,4
Transmitted Data	TD	2	3	6	3	8	2
Received Data	RD	3	2	5	6	9	5
Data Terminal Ready	DTR	20	4	3	2	7	1
Data Set Ready	DSR	6	6	1	7	5	6
Request To Send	RTS	4	7	8	1	4	-
Clear To Send	CTS	5	8	7	8	3	-
Carrier Detect	DCD	8	1	2	7	10	-
Ring Indicator	RI	22	9	1	-	2	-

Tabla III.1 Pines asignados del DB9

Las UART o U(S)ART (Transmisor y Receptor Asíncrono Universal) se diseñaron para convertir las señales que maneja la CPU y transmitir las al exterior. Las UART deben resolver problemas tales como la conversión de voltajes internos del DCE con respecto al DTE, gobernar las señales de control, y realizar la transformación desde el bus de datos de señales en paralelo a serie y viceversa. Debe ser robusta y deberá tolerar circuitos abiertos, cortocircuitos y escritura simultánea sobre un mismo pin, entre otras consideraciones. Es en la UART en donde se implementa la interfaz.

Cuando se requiere conectar un microcontrolador (con señales típicamente entre 3.3 y 5 V) con un puerto RS-232 estándar, se utiliza un driver de línea, típicamente un MAX232 o compatible, el cual mediante dobladores de voltaje positivos y negativos, permite obtener la señal bipolar (típicamente alrededor de +/- 6V) requerida por el estándar.

Para los propósitos de la RS-232 estándar, una conexión es definida por un cable desde un dispositivo al otro.

Hay 25 conexiones en la especificación completa, pero es muy probable que se encuentren menos de la mitad de éstas en una interfaz determinada. La causa es simple, una interfaz *full duplex* puede obtenerse con solamente 3 cables.

Voltaje	Señal	Nivel Lógico	Control
+3 a +15	Espacio	0	On
-3 a -15	Marca	1	Off

Tabla III.2 Niveles de voltaje del DB9

3.2.4 Adaptador USB a Serial Omega DB9



Figura III.7 Adaptador USB a Serial

Cable USB a serial RS-232 DB-9 macho soporta velocidad de transferencia de datos de hasta 230Kbps. Ofrece una solución rápida, sencilla y rentable, y es ideal para la comunicación de las distintas aplicaciones y la automatización. Esto hace que la interfaz USB transparente a los periféricos de serie, lo que les permite interactuar fácilmente con USB de la computadora. Este dispositivo también es compatible con el ahorro de energía suspender y reanudar operación

- ✓ Compatible con USB 1.1 y USB 2.0
- ✓ Funciona con los dispositivos periféricos interfaz serial, convierte la conexión USB a interfaz serial.
- ✓ Conector de los Terminales: USB A Macho a dB9 Macho
- ✓ Longitud del cable 2 pies

- ✓ Soporta el modo automático handshake
- ✓ Hasta 230Kbps tasa de transferencia de datos
- ✓ Compatible con activación remota y administración de energía
- ✓ Fácil instalación
- ✓ Soporta Windows 98/2000/XP/Vista/7/2003 Server/2008 Server
- ✓ MAC 8.6 o superior
- ✓ Linux RedHat 7.3, 8 y 9
- ✓ Conector de los Terminales: USB A Macho a dB9 Macho

3.2.5. MAX 232

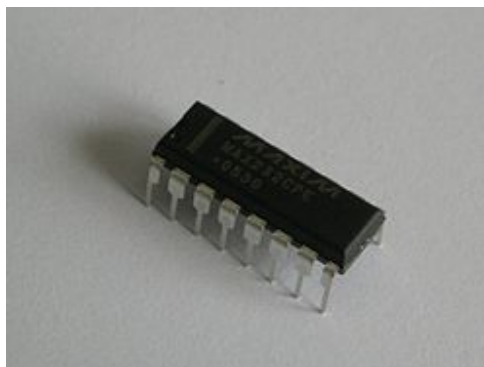


Figura III.8 Circuito integrado MAX 232

El MAX232 es un circuito integrado que convierte las señales de un puerto serie RS-232 a señales compatibles con los niveles TTL de circuitos lógicos. El MAX232 sirve como interfaz de transmisión y recepción para las señales RX, TX, CTS y RTS.

El circuito integrado tiene salidas para manejar niveles de voltaje del RS-232 (aprox. ± 7.5 V) que las produce a partir de un voltaje de alimentación de + 5 V utilizando multiplicadores de voltaje internamente en el MAX232 con la adición de condensadores externos. Esto es de mucha utilidad para la implementación de puertos serie RS-232 en dispositivos que tengan una alimentación simple de + 5 V.

Las entradas de recepción de RS-232 (las cuales pueden llegar a ± 25 V), se convierten al nivel estándar de 5 V de la lógica TTL. Estos receptores tienen un umbral típico de 1.3 V, y una histéresis de 0.5 V.

La versión MAX232A es compatible con la original MAX232, y tiene la mejora de trabajar con mayores velocidades de transferencia de información (mayor tasa de baudios), lo que reduce el tamaño de los condensadores externos utilizados por el multiplicador de voltaje, 0.1 μ F en lugar del 1.0 μ F usado en el dispositivo original.

Una versión más nueva de este circuito integrado, el MAX3232 también es compatible con el original, pero opera en un rango más amplio, de 3 a 5.5 V

Conexiones

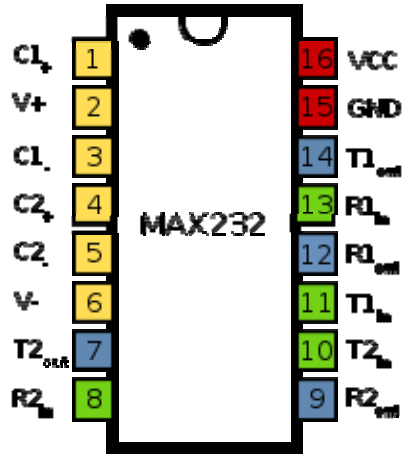


Figura III.9 Pines de conexión del MAX 232

C1₊

Conexión positiva del condensador C1 del doblador de voltaje de +5V a +10V.

C1₋

Conexión negativa del condensador C1 del doblador de voltaje de +5V a +10V.

C2₊

Conexión positiva del condensador C2 del inversor de voltaje de +10V a -10V.

C2₋

Conexión negativa del condensador C2 del inversor de voltaje de +10V a -10V.

V-

Conexión de salida del voltaje de -10V.

V+

Conexión de salida del voltaje de +10V.

T1_{in}, T2_{in}, R1_{out}, R2_{out}

Conexiones a niveles de voltaje de TTL o CMOS.

T1_{out}, T2_{out}, R1_{in}, R2_{in}

Conexiones a niveles de voltaje del protocolo RS-232.

VCC

Alimentación positiva del MAX232

GND

Alimentación negativa del MAX232

Niveles de voltaje

El circuito integrado MAX232 recibe un nivel TTL lo convierte, cambia un nivel lógico TTL de 0 a un nivel comprendido entre +3 y +15 V, y cambia un nivel lógico TTL 1 a un nivel comprendido entre -3 a -15 V, y viceversa, para convertir niveles de RS232 a TTL.

Tipo de línea RS232 y Nivel lógico	Voltaje RS232	Voltaje TTL hacia o desde el MAX232
Transmisión de datos (Rx/Tx) Nivel lógico 0	+3 V a +15 V	0 V
Transmisión de datos (Rx/Tx) Nivel lógico 1	-3 V a -15 V	5 V
Señales de control (RTS/CTS/DTR/DSR) Nivel lógico 0	-3 V a -15 V	5 V
Señales de control (RTS/CTS/DTR/DSR) Nivel lógico 1	+3 V a +15 V	0 V

Tabla III.3 Niveles de voltaje del MAX232

3.2.6. Algoritmos y software de control en el microcontrolador

Para la adquisición y caracterización se implementó un programa que permite testear las entradas digitales que entregan la orden para mover las articulaciones, generar señales de PWM para poder variar la velocidad de los motores, se debe realizar un promedio de 100 muestras y efectuar la comunicación serial con el computador. El código del programa se encuentra en el anexo...

3.2.6.1. Proceso en el Atmega16A

Para la programación del Atmega16 se utilizó el lenguaje ATMELSTUDIO 6.1, que facilita la codificación del programa ya que utiliza instrucciones de lenguaje C. Además permite la

simulación del programa, que es una herramienta muy útil para ahorrar tiempo en la realización.

El diagrama de flujo de la programación del Atmega16 se puede observar en la figura III.7

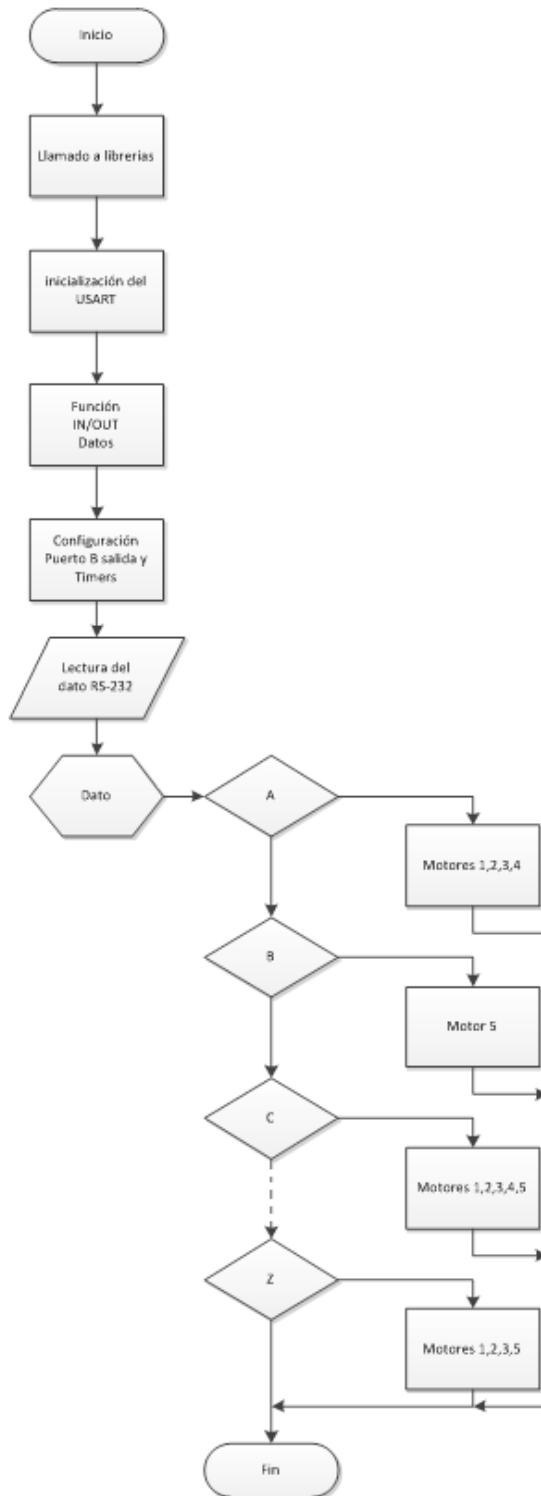


Figura III.10 Diagrama de flujo del microcontrolador

3.2.7. Algoritmos y software de control SCADA

Para la programación de la interfaz SCADA se utilizó el lenguaje de programación Visual Basic 2010, que facilita la codificación al utilizar una interfaz gráfica orientada a objetos.

El diagrama de flujo de la programación del SCADA se puede observar en la figura III.8

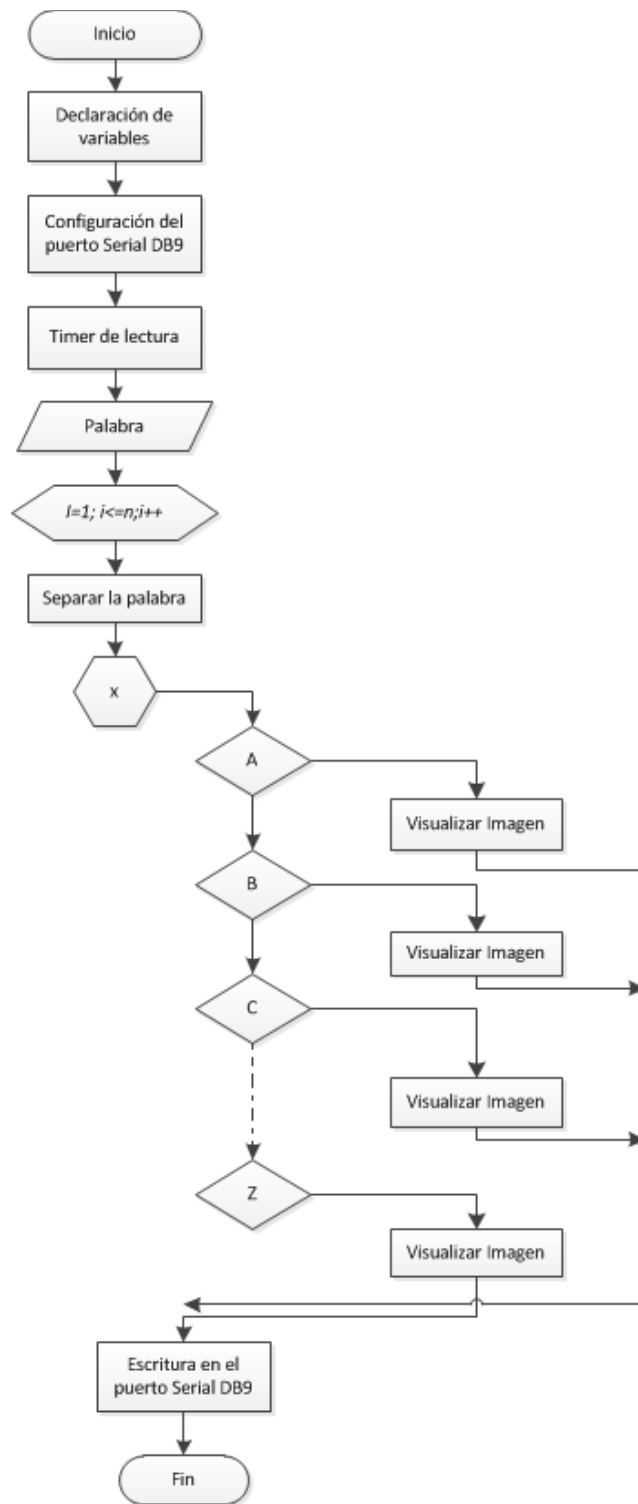


Figura III.11 Diagrama de flujo SCADA

La figura III.12 muestra la interfaz que será utilizada en la comunicación del software con el circuito de control que moverá cada uno de los dedos

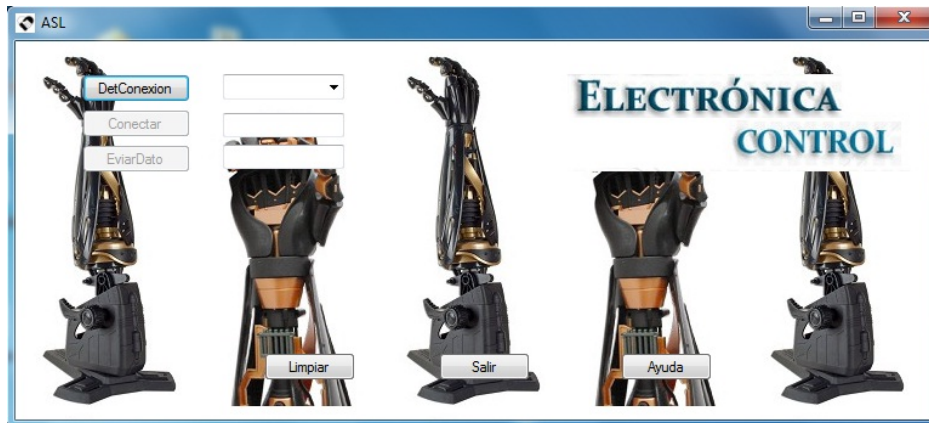


Figura III.12 Interfaz de usuario

La ventana consta de los siguientes elementos principales:

- ✓ “DetConexión”, en esta opción se podrán buscar todos los puertos disponibles que tenemos en el computador.
- ✓ “Conectar”, aquí podemos ya establecer la comunicación entre el software y el circuito de control.
- ✓ En el “textbox” que se encuentra junto al botón de “Conectar” escribimos la palabra que vamos a traducir a lenguaje ASL.
- ✓ “Limpiar”, si necesitamos enviar otra palabra utilizaremos esta opción para dejar en blanco el “textbox”

CAPÍTULO IV

IMPLEMENTACIÓN

Una herramienta CAD brinda un entorno muy preciso de acabados perfectos y propiedades físicas ideales, pero en la realidad nos encontramos con diferencias en estos aspectos, Los materiales utilizados son: Servomotores genéricos HITEC HS-311, nylon (0.5mm), calibradores, forros para cables y mano robótica de juguete (DarthVaderRoboticArm).

Para el diseño del controlador se implementará el Atmega16A de 40 pines, el mismo que nos brindará los módulos requeridos para realizar el controlador de los 5 servomotores, y sobre todo el módulo de comunicación SERIAL RS 232 para la comunicación con el computador lo que hace posible la interacción entre el software y el prototipo.

En el presente capítulo se detallará los pasos y materiales para la adaptación de cada una de las piezas que componen la mano robótica, además del procedimiento de construcción del controlador, el mismo que será el encargado del correcto funcionamiento de todos los servos.

4.1 Mano robótica

A continuación se detalla cada uno de los elementos de los que está compuesta la mano robótica.

Los elementos se encuentran detallados en la figura IV.1

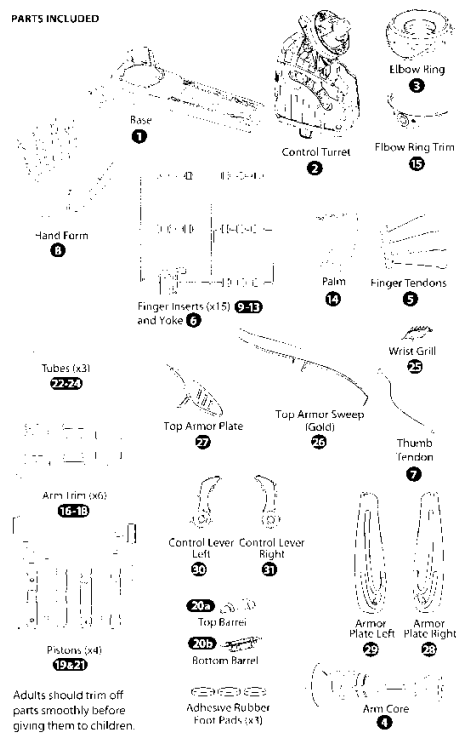


Figura IV.1Partes de la mano robótica

4.3 Armado de la mano

La mano está constituida por 27 huesos que se agrupan en tres áreas distintas:

- ✓ Huesos del carpo
- ✓ Huesos del metacarpo
- ✓ Huesos de los dedos

Las uniones o juntas de las falanges están sujetas por una lámina de plástico entre sí para evitar que salgan de su sitio y además para darle una mayor seguridad de agarre. Éstas a su vez ya ensambladas están sujetas por un dispositivo de fijación denominado leva de fijación como se muestra en la Figura IV.4.



Figura IV.4 Prototipo de mano robótica

Una vez ensambladas las falanges se procede a instalar el mecanismo que proporciona la fuerza de desplazamiento, que a su vez mueve las falanges, produciendo un movimiento en los dedos, como se muestra en la Figura IV.5.



Figura IV.5 Prototipo terminado

Cada uno de los motores genera el movimiento para cada uno de los dedos, el nylon se encuentra conectado en el extremo superior de cada uno de los dedos, de esta manera genera la fuerza necesaria para contraer cada uno de los dedos.

4.4 Diseño del circuito de control

El circuito de control consta de un microcontrolador ATMEGA16 de 40 pines, un MAX 232 que nos ayuda a establecer la comunicación entre el computador y el microcontrolador, conector DB9-F y elementos varios.

El diseño se realizó en el simulador Proteus ISIS 7, el circuito fue probado en un protoboard para establecer su correcto funcionamiento y evitar errores al momento de crear el PCB en Proteus ARES 7, el diseño se encuentra en la Figura IV.6

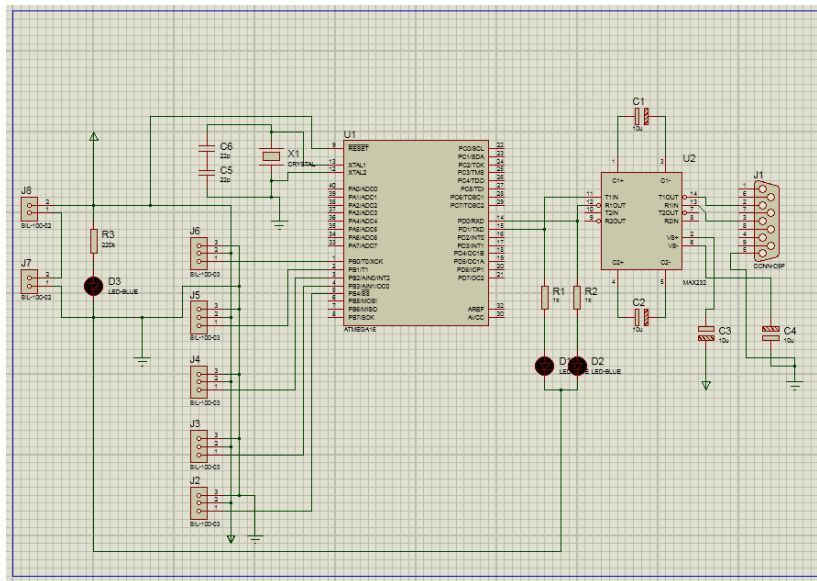


Figura IV.6 Circuito de control.

Para tener un diseño compacto se ha construido un PBC (Véase figura IV.7), el mismo que se lo realiza en una sola cara para disminuir el costo, cabe anotar que el controlador es para 5 servomotores.

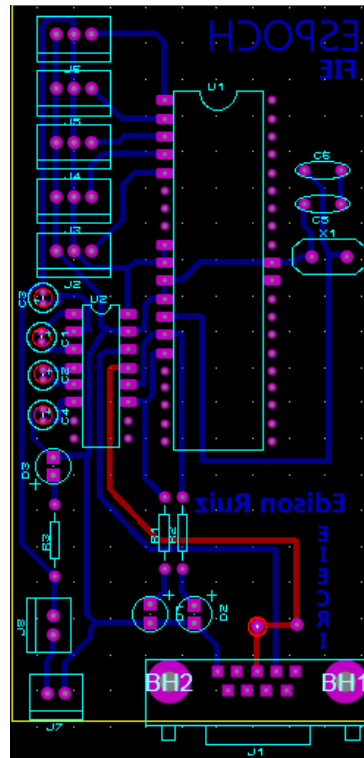


Figura IV.7PCB

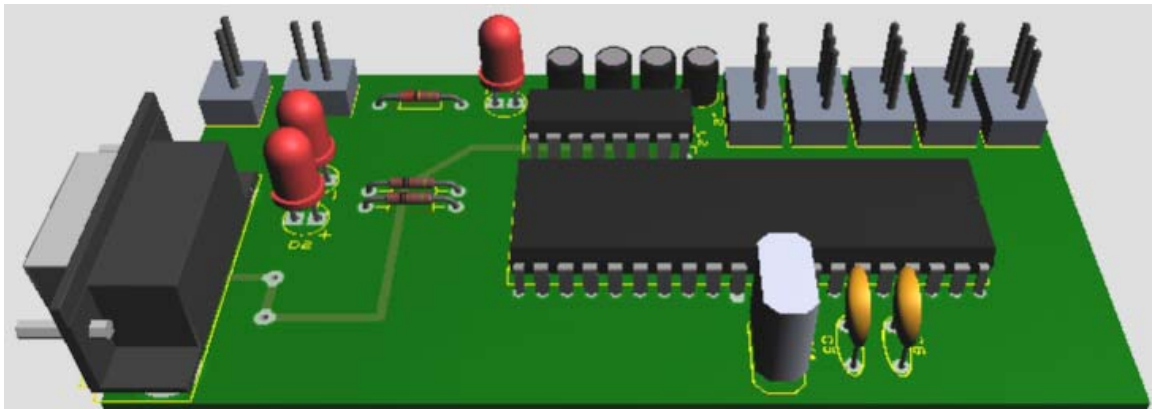


Figura IV.8 Vista 3D

4.4.1 Programación en AtmelStudio 6.1

La codificación se realizó en AtmelStudio 6.1, a continuación se detalla las líneas de programación más importantes del microcontrolador y se detallará cada uno de los módulos utilizados para la comunicación.

Comunicación serial:

```
void InitUART(unsigned char baudrate){

    UBRRL=baudrate;

    UCSRB=(UCSRB|_BV(RXEN)|_BV(TXEN));

}

unsigned char ReceiveByte(void){

    while(!(UCSRA&(1<<RXC)));

    return UDR;

}

void TransmitByte(unsigned char data){

    while(!(UCSRA&(1<<UDRE)));

    UDR=data;

}
```

Movimiento de servomotor

```
voidzero(){  
  
    for(i=1;i<=100;i++)  
  
        {  
  
            PORTB=0b00011111;  
  
            _delay_us(1500);  
  
            PORTB=0b00000000;  
  
            _delay_us(20000);  
  
        }  
  
}
```

Selección de cada uno de los caracteres y ejecución del servomotor

```
dato=ReceiveByte();  
  
    switch(dato)  
  
        {  
  
            case'A':  
  
                for(i=1;i<=100;i++)  
  
                    {
```

```
PORTB=0b00011111;  
  
_delay_us(2500);  
  
PORTB=0b00000000;  
  
_delay_us(20000);  
  
}  
  
TransmitByte(dato);  
  
zero();  
  
break;
```

4.5 Software de control SCADA

En cada proceso y desarrollo de mecanismos robóticos existen interfaces para el manejo de los mismos, ya sea para el control, monitoreo, mantenimiento, etc., siendo muy importante el desarrollo de programas para la comunicación entre máquina – hombre HMI. Existen programas con muchas similitudes pero la diferencia entre ellos es la complejidad al manipularlos.

Visual Basic 2010 incorpora herramientas que permite fácilmente la programación de un sistema. En este presente capítulo se detalla los componentes que forman parte del sistema

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación, tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de

desarrollo web como ASP.NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Controles comunes

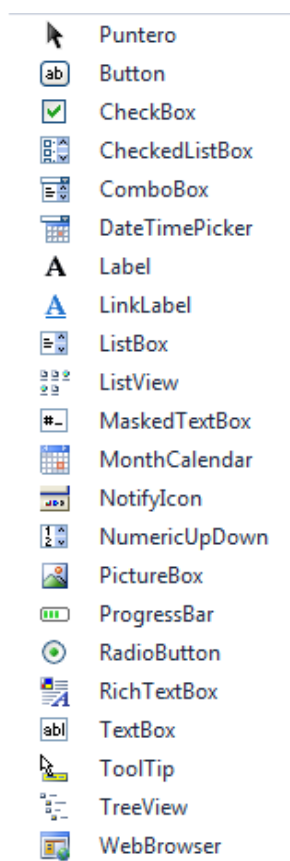


Figura IV.9 Controles comunes de Visual Basic 2010

Los componentes comunes utilizados en esta interfaz son:

Botton.- Este elemento se utilizó para crear cada uno de los botones que sirven para ejecutar las acciones de cada una de las funciones, por ejemplo para habilitar la comunicación entre el computador y el controlador de la mano.

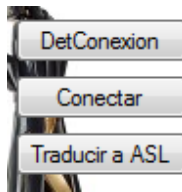


Figura IV.10 Botones creados

ComboBox.- Esta herramienta sirve para listar cada uno de los puertos de comunicación que se encuentran disponibles para establecer la comunicación

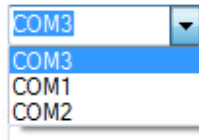
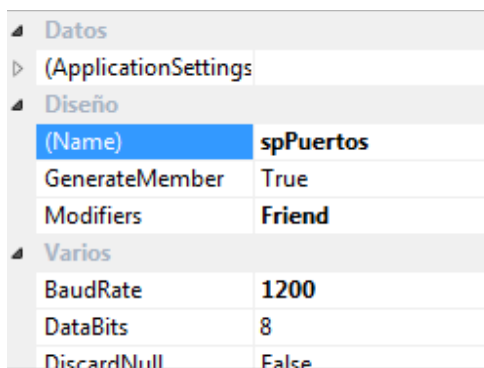


Figura IV.11 Selección del puerto

SerialPort.- Este componente simplifica la configuración de la comunicación entre los dispositivos ayudando a ajustar la velocidad de transmisión, bits de parada, etc.



Datos	
(ApplicationSettings)	
Diseño	
(Name)	spPuertos
GenerateMember	True
Modifiers	Friend
Varios	
BaudRate	1200
DataBits	8
DiscardNull	False

Figura IV.12 Serial Port

4.5.1 Líneas de código

Lectura de puertos

En las siguientes líneas de código se detalla cada una de las acciones que se van a realizar

```
PrivateSubbtnDeterminarConexion_Click(ByVal sender AsSystem.Object, ByVal e AsSystem.EventArgs)
```

```
HandlesbtnDeterminarConexion.Click
```

```
cboPuertos.Items.Clear()
```

```
ForEachPuertoDisponibleAsStringInMy.Computer.Ports.SerialPortNames
```

```
cboPuertos.Items.Add(PuertoDisponible)
```

```
Next
```

```
IfcboPuertos.Items.Count> 0 Then
```

```
cboPuertos.Text = cboPuertos.Items(0)
```

```
DimresultAsDialogResult
```

```
result = MessageBox.Show("Seleccione el puerto a trabajar", "Puerto Serial DB9",  
    MessageBoxButtons.OKCancel)
```

```
If result = DialogResult.OK Then
```

```
    btnConectar.Enabled = True
```

```
EndIf
```

```
Else
```

```
    MessageBox.Show("Ningun puerto encontrado", "Puerto Serial DB9", MessageBoxButtons.OK)
```

```
    btnConectar.Enabled = False
```

```
    btnEnviarDato.Enabled = False
```

```
    cboPuertos.Items.Clear()
```

```
    cboPuertos.Text = (" ")
```

```
EndIf
```

```
EndSub
```

Envío de datos desde el software al microcontrolador

```
Private Sub btnEnviarDato_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles btnEnviarDato.Click
```

```
    txtBufferIn.Text = ""
```

```
    txtBufferIn.ForeColor = Color.Black
```

```
txtBafferIn.Font = NewFont("Microsoft Sans Serif", 8, FontStyle.Regular)
```

```
    a = txtBafferOut.Text
```

```
    n = Len(a)
```

```
    a = UCase(a)
```

```
For i = 1 To n Step 1
```

```
    x = GetChar(a, i)
```

```
spPuertos.DiscardOutBuffer()
```

```
spPuertos.Write(x)
```

```
Next
```

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\electro.jpg")
```

```
EndSub
```

Manejo de caracteres y visualización de imágenes

```
Private Sub txtBafferIn_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles txtBafferIn.TextChanged
```

```
    y = txtBafferIn.Text
```

```
If StrBafferIn <> "" Then
```

```
    Select Case y
```

```
        Case "A"
```

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\A.jpg")
```

```
PictureBox1.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox2.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox3.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox4.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox5.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

Case"B"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\B.jpg")
```

```
PictureBox1.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
```

```
PictureBox2.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox3.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox4.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

```
PictureBox5.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

CAPÍTULO V

ANÁLISIS Y RESULTADOS

Las pruebas que se realizaron están divididas en: pruebas de funcionamiento del controlador, movimiento de los servomotores y pruebas tiempo de respuesta.

5.1 Pruebas de funcionamiento del controlador

Una parte esencial del proyecto es su controlador de servomotores, gracias a este se puede enviar al prototipo a diferentes grados de posicionamiento a sus salidas. Cabe señalar que el controlador puede manejar más de cinco servos a la vez.

Estas pruebas se realizaron en el simulador con la ayuda de los instrumentos de media que se encuentran dentro de Proteus, la frecuencia que se le entrega al servomotor es de 49.9919Hz y sabiendo que el servomotor trabaja a 50Hz se concluye que el controlador es muy eficiente y exacto(véase la figura V.1).

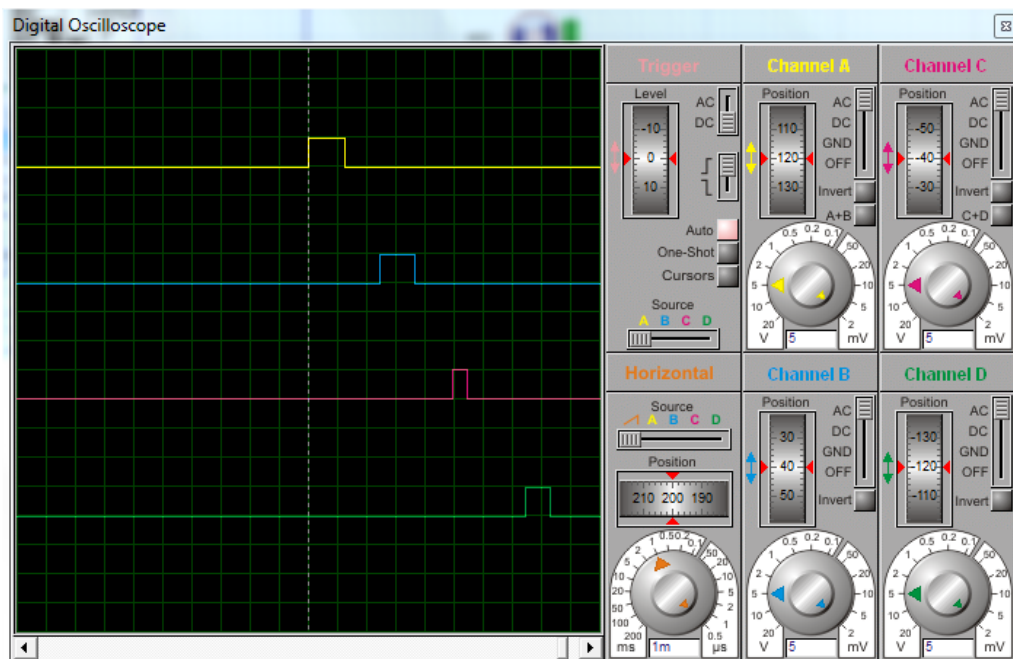


Figura V.1 Ondas generadas por el microcontrolador

Cada uno de los servos se mueven entre 0° y 120° para generar cada uno de los movimientos, con excepciones por ejemplo de la letra “C” que necesita un movimiento entre 0° y 50° .

Las pruebas se realizaron con las siguientes letras:

Letra J



Figura V.2 Letra J generada por el prototipo

Letra L



Figura V.3 Letra L generada por el prototipo

Letra O



Figura V.4 Letra O generada por el prototipo

Letra W



Figura V.5 Letra W generada por el prototipo

Se escogió estas 4 letras para medir el tiempo de respuesta de la mano al momento de generar los símbolos, a continuación se muestra la tabla con los tiempos que se tomaron al realizar las pruebas.

Letra	Tiempo-respuesta
J	2,3 seg
L	2,2 seg
O	2,3 seg
W	2,31 seg

Tabla V.1 Tiempo de respuesta

Limitación:

- ✓ La limitación que tenemos en este prototipo, es que la mano de juguete no puede representar de manera idéntica a los movimientos de la mano humana, ya que está compuesta por varias articulaciones que permite realizar varios movimientos en un solo dedo, a diferencia del prototipo que posee únicamente movimientos de flexión y extensión y carece de movimientos de aducción y abducción.

CONCLUSIONES

- ✓ La mano es una estructura compleja ya que puede realizar varios movimientos en un solo dedo y tratar de imitar estos movimientos tiene un grado de complejidad, con las herramientas CAD disponibles se puede hacer una réplica casi exacta de cada uno de los dedos, pero al momento de crear el miembro tenemos una limitación por que las herramientas para crear estos elementos son poco precisas y existe un rango de error que limita su funcionamiento.
- ✓ Los elementos que se pueden utilizar son plásticos como sorbetes, tubos, partes internas del disco duro, juguetes. Para el prototipo se utilizó un juguete que se asemeja a una mano real, ya que esta podía sujetar elementos.
- ✓ Al implementar la mano se tuvo que hacer varias modificaciones en el juguete, fue necesario retirar los accesorios que se encontraban dentro de la mano y sujetar los tendones con nylon de 0.5mm y estos a su vez son sujetados a los servomotores que se encuentran en la parte superior de la base.
- ✓ El diseño de los circuitos fueron creados en el Proteus ISIS 7 el cual facilita la ubicación y enrutamiento de las pistas del circuito, hay que tomar en cuenta la separación que existe entre las pistas ya que al momento de pasar a la placa pueden quedar pegadas y provocar corto circuito.
- ✓ Al momento de implementar la interfaz SCADA se tomó en cuenta que sea de fácil manejo y que no ocupe un gran tamaño en el disco.
- ✓ Con las pruebas se pudo regular de mejor manera los movimientos de cada uno de los dedos asemejándose cada vez más a los símbolos que genera la mano humana.

- ✓ Una de las capacidades adicionales de la mano robótica es sostener objetos, en cuanto a las limitaciones tenemos en la representación de varios símbolos porque estos necesitan de una estructura mecánica diferente, por ejemplo que los dedos se puedan cruzar entre ellos o movimiento de la muñeca en caso de la letra “J” y “Z”

RECOMENDACIONES

- ✓ Se debe trabajar con herramientas compatibles con el sistema operativo, si existe una incompatibilidad se puede crear un error al momento del funcionamiento y crear complicaciones donde no las hay.
- ✓ Verificar que los componentes del prototipo sean de un material resistente, ya que este va a ser sometido a varias pruebas y esfuerzos hasta llegar al movimiento deseado.
- ✓ Al momento de realizar las adaptaciones en el prototipo se debe tener mucho cuidado de no alterar la resistencia del material por que puede ocasionar un daño grave e irreparable.
- ✓ Para la elaboración del circuito se debería tomar en cuenta la necesidad de conectar un MAX232, que estará ubicado entre el microcontrolador y el PC, ya que este funciona como acoplador e inversor de los datos que se están transmitiendo, de no estar el circuito antes mencionado se obtendrá símbolos y no los datos que se enviaron.
- ✓ Se diseñó una interfaz hardware/software que permite conectar el prototipo de mano para obtener la realimentación y enviar órdenes para realizar posturas de los símbolos y tal vez agarre de elementos.
- ✓ Para tener movimientos más naturales es necesario que las articulaciones de los dedos sigan una trayectoria a través de algoritmos de control.

RESUMEN

Diseño e implementación de un sistema de control para mano robótica, utilizada para comunicación visual ASL, para la Escuela de Ingeniería Electrónica en Control y Redes Industriales.

Para llevar a cabo un robot con dicha característica es necesario trabajar el sistema mecánico del mismo, teniendo en cuenta la cantidad de articulaciones que permite la movilidad, el control del prototipo, se trabaja con la electrónica, que admite controlar el robot y procesamiento de información para obtener la respuesta correcta.

Este prototipo tiene como principal objetivo ayudar a las personas a comunicarse con otras que tienen deficiencia en el sentido auditivo a través del lenguaje ASL, mediante el diseño y construcción de la mano, circuitos de control e interfaces, utilizando para esto un ATMEGA16, ATMEL STUDIO 6.1, SCADA y VISUAL BASIC 2010

El prototipo tiene una efectividad del 75% al momento de generar los símbolos del lenguaje ASL, el actuador tiene un tiempo promedio de 2.29seg al generar los movimientos.

Se concluyó que el prototipo ayudará a las personas a comunicarse con otras que tienen deficiencia del sentido auditivo a través del lenguaje de señas ASL.

Se recomienda el funcionamiento del prototipo dentro de una institución que tenga estudiantes con dicho tipo de discapacidad, así los estudiantes se sentirán en un ambiente más amigable que ayudará a mejorar su rendimiento académico.

SUMMARY

Design and implementation of a control system for robotic hand to be used in visual communication ASL, in the Escuela de Ingeniería Electrónica en Control y Redes Industriales.

In order to carry out a robot with such characteristic is necessary to work on a mechanical system for itself, taking into account the quantity of articulations that lead movement, prototype control, working with electronics, that let to control the robot and processing of information in order to get the correct answer.

This prototype has a main objective to help people with hearing sickness to communicate each other's through ASL language, by means of the design and building of "a hand", interfaces and control circuits, using ATMEGA16, ATMEL STUDIO 6.1, SCADA and VISUAL BASIC 2010.

The prototype has an effectiveness of 75% at the moment of generating language symbols ASL, the actor has a mean time of 2.29 sec when generating movements.

It was concluded the prototype will constitute a real help for people with hearing deficiency to communicate each other by means of miming language ASL.

It is recommended the functioning of the prototype into an institution including students with such unable problems, so students are going to feel into a more friendly environment that will help them to improve their academic render.

ANEXOS

ANEXO 1

CÓDIGO DEL ATMEGA 16

```
#include<avr/io.h>
#include<util/delay.h>
#include<avr/interrupt.h>
floatlag0,lag1,lag2,lag3,lag4;
char dato;
inti;
void InitUART(unsigned char baudrate){
    UBRRL=baudrate;
    UCSRB=(UCSRB|_BV(RXEN)|_BV(TXEN));
}
unsigned char ReceiveByte(void){
    while(!(UCSRA&(1<<RXC)));
    return UDR;
}
void TransmitByte(unsigned char data){
    while(!(UCSRA&(1<<UDRE)));
    UDR=data;
}
void initialize(){
    lag0=lag1=lag2=lag4=0.82;
    TCCR0=0B00001100;
    OCR0=77;
}
void Wait()
{
    uint8_t i;
    for(i=0;i<14;i++)
    {
        _delay_loop_2(0);
        _delay_loop_2(0);
    }
}
void zero(){
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(1500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
}
int main(void)
{
    InitUART(51);
    initialize();
    DDRC=0xFF;
    while(1)
    {
        dato=ReceiveByte();
        switch(dato)
```



```

{
    case'A':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'B':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'C':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'D':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    case'E':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'F':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;

```

```

        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'G':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'H':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'I':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'J':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'K':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    }

```

```

TransmitByte(dato);
zero();
break;
case'L':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'M':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'N':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'O':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'P':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'Q':

```

```

for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'R':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'S':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'T':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'U':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
    PORTB=0b00000000;
    _delay_us(20000);
}
TransmitByte(dato);
zero();
break;
case'V':
for(i=1;i<=100;i++)
{
    PORTB=0b00011111;
    _delay_us(2500);
}

```

```

        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'W':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'X':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'Y':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    case'Z':
    for(i=1;i<=100;i++)
    {
        PORTB=0b00011111;
        _delay_us(2500);
        PORTB=0b00000000;
        _delay_us(20000);
    }
    TransmitByte(dato);
    zero();
    break;
    }
}
//TODO:: Please write your application code
}
}

```

ANEXO 2

PROGRAMACIÓN SCADA

```
Public Class formAsI
    Dim StrBafferOut As String
    Dim StrBafferIn As String
    Dim a As String
    Dim y As String
    Dim n As Integer
    Dim x, b As Char
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\electro.jpg")
        StrBafferOut = ""
        StrBafferIn = ""
        btnConectar.Enabled = False
        btnEnviarDato.Enabled = False
        tmrTimer.Enabled = False
    End Sub

    Private Sub btnDeterminarConexion_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles btnDeterminarConexion.Click
        cboPuertos.Items.Clear()
        For Each PuertoDisponible As String In My.Computer.Ports.SerialPortNames
            cboPuertos.Items.Add(PuertoDisponible)
        Next
        If cboPuertos.Items.Count > 0 Then
            cboPuertos.Text = cboPuertos.Items(0)
            Dim result As DialogResult
            result = MessageBox.Show("Seleccione el puerto a trabajar", "Puerto Serial DB9", MessageBoxButtons.OKCancel)
            If result = DialogResult.OK Then
                cboPuertos.Text = cboPuertos.Items(0)
                btnConectar.Enabled = True
            End If
        Else
            MessageBox.Show("Ningun puerto encontrado", "Puerto Serial DB9", MessageBoxButtons.OK)
            btnConectar.Enabled = False
            btnEnviarDato.Enabled = False
            cboPuertos.Items.Clear()
            cboPuertos.Text = (" ")
        End If
    End Sub

    Private Sub btnConectar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles btnConectar.Click
        If btnConectar.Text = "Conectar" Then
            spPuertos.PortName = cboPuertos.Text
            btnConectar.Text = "Desconectar"
            txtBafferIn.Font = New Font("Microsoft Sans Serif", 8, FontStyle.Italic)
            txtBafferOut.Font = New Font("Microsoft Sans Serif", 8, FontStyle.Italic)
            txtBafferOut.ForeColor = Color.Gray
            txtBafferIn.ForeColor = Color.Gray
            txtBafferIn.Text = "Respuesta del Circuito"
            txtBafferOut.Text = "Escriba el Texto"
            btnDeterminarConexion.Enabled = False
        End If
    End Sub
End Class
```

```

btnSalir.Enabled = False
cboPuertos.Enabled = False
btnEnviarDato.Enabled = True
tmrTimer.Enabled = True
spPuertos.Open()
ElseIf btnConectar.Text = "Desconectar" Then
btnConectar.Text = "Conectar"
txtBufferIn.Text = ""
txtBufferOut.Text = ""
btnDeterminarConexion.Enabled = True
btnSalir.Enabled = True
cboPuertos.Enabled = True
btnEnviarDato.Enabled = False
'tmrTimer.Enabled = False
'spPuertos.Close()
EndIf
EndSub

```

```

Private Sub btnEnviarDato_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnEnviarDato.Click
txtBufferIn.Text = ""
txtBufferIn.ForeColor = Color.Black
txtBufferIn.Font = New Font("Microsoft Sans Serif", 8, FontStyle.Regular)
    a = txtBufferOut.Text
    n = Len(a)
    a = UCase(a)
For i = 1 To n Step 1
    x = GetChar(a, i)
spPuertos.DiscardOutBuffer()
spPuertos.Write(x)
Next
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\electro.jpg")
EndSub

```

```

Private Sub tmrTimer_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tmrTimer.Tick
StrBufferIn = spPuertos.ReadExisting
If StrBufferIn <> "" Then
txtBufferIn.Text = StrBufferIn
'StrBufferIn = ""
spPuertos.DiscardInBuffer()
EndIf
EndSub

```

```

Private Sub btnLimpiar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLimpiar.Click
txtBufferIn.Text = ""
txtBufferOut.Text = ""
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\electro.jpg")
PictureBox1.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
PictureBox2.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
PictureBox3.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
PictureBox4.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
PictureBox5.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
EndSub

```

```

Private Sub txtBufferIn_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtBufferIn.TextChanged
y = txtBufferIn.Text
If StrBufferIn <> "" Then

```

SelectCase y

Case"A"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\A.jpg")
PictureBox1.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox2.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox3.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox4.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox5.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

Case"B"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\B.jpg")
PictureBox1.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\rojo.png")
PictureBox2.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox3.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox4.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
PictureBox5.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\verde.png")
```

Case"C"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\C.jpg")
```

Case"D"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\D.jpg")
```

Case"E"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\E.jpg")
```

Case"F"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\F.jpg")
```

Case"G"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\G.jpg")
```

Case"H"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\H.jpg")
```

Case"I"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\I.jpg")
```

Case"J"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\J.jpg")
```

Case"K"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\K.jpg")
```

Case"L"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\L.jpg")
```

Case"M"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\M.jpg")
```

Case"N"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\N.jpg")
```

Case"O"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\O.jpg")
```

Case"P"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\P.jpg")
```

Case"Q"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\Q.jpg")
```

Case"R"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\R.jpg")
```

Case"S"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\S.jpg")
```

Case"T"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\T.jpg")
```

Case"U"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\U.jpg")
```

Case"V"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\V.jpg")
```

Case"W"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\W.jpg")
```

Case"X"

```
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\X.jpg")
```



```

Case"Y"
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\Y.jpg")
Case"Z"
ImagenMano.Image = Image.FromFile("C:\Users\The One\Desktop\iconos\Z.jpg")
EndSelect
MessageBox.Show("Letra Traducida " & "" & txtBafferIn.Text & "", "ASL", MessageBoxButtons.OK)
EndIf
EndSub

    PrivateSub txtBafferOut_Click(ByVal sender As Object, ByVal e As System.EventArgs)
        Handles txtBafferOut.Click
txtBafferOut.Text = ""
txtBafferOut.ForeColor = Color.Black
txtBafferOut.Font = NewFont("Microsoft Sans Serif", 8, FontStyle.Regular)
EndSub

    PrivateSub btnSalir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles btnSalir.Click
formSalir.Show()
EndSub

PrivateFunction MessageBoxIndirect() As Integer
ThrowNew NotImplementedException
EndFunction

    PrivateSub ImagenMano_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles ImagenMano.Click

EndSub

    PrivateSub btnAyuda_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles btnAyuda.Click
'y = Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
'Process.Start("WINWORD", "y\iconos\COMUNICACIÓN ASL-AYUDA.docx")
Process.Start("WINWORD", "C:\HELP.docx")
EndSub

    PrivateSub cboPuertos_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cboPuertos.SelectedIndexChanged

EndSub

    PrivateSub spPuertos_DataReceived(ByVal sender As System.Object, ByVal e
As System.IO.Ports.SerialDataReceivedEventArgs) Handles spPuertos.DataReceived

EndSub

    PrivateSub spPuertos_ErrorReceived(ByVal sender As System.Object, ByVal e
As System.IO.Ports.SerialErrorReceivedEventArgs) Handles spPuertos.ErrorReceived

EndSub

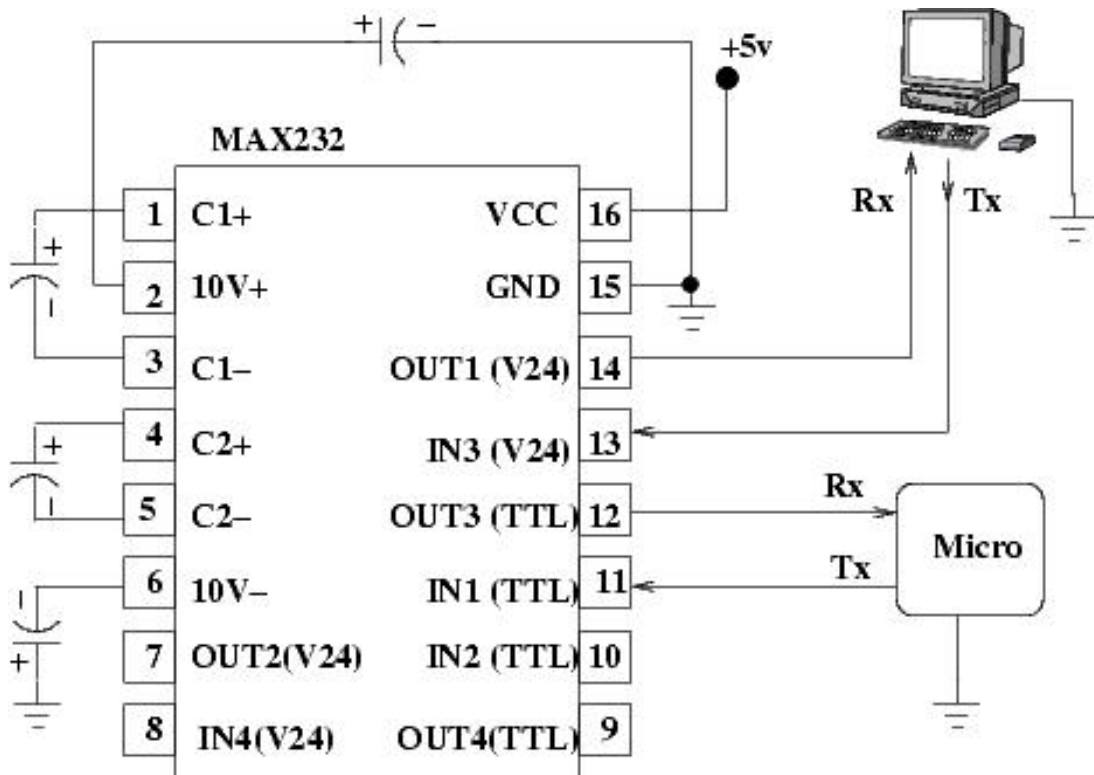
    PrivateSub PictureBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
PictureBox1.Click

EndSub
EndClass

```

ANEXO 3

CONFIGURACIÓN DEL MAX232



ANEXO 4

MANUAL DE USUARIO

COMUNICACIÓN ASL

Ventana Principal

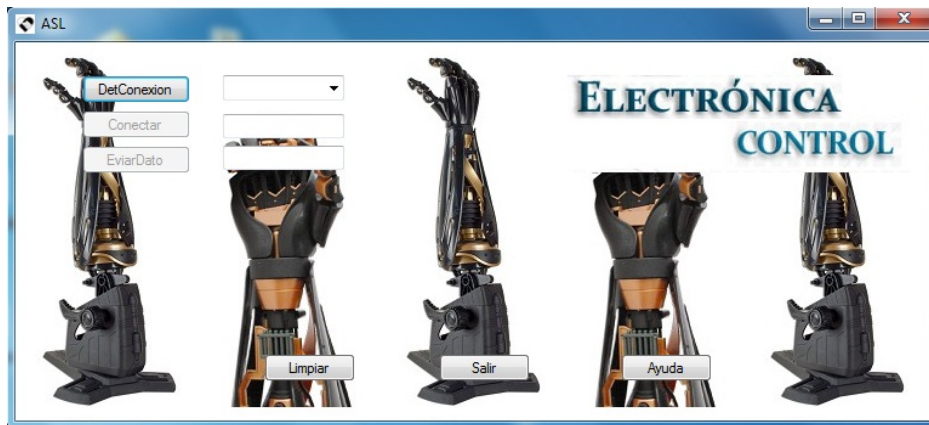


Figura 1, ventana principal

Localizar el puerto de comunicación que se va a utilizar

1. Inicio
2. Click derecho en “Equipo”
3. Click izquierdo en “Administrar”

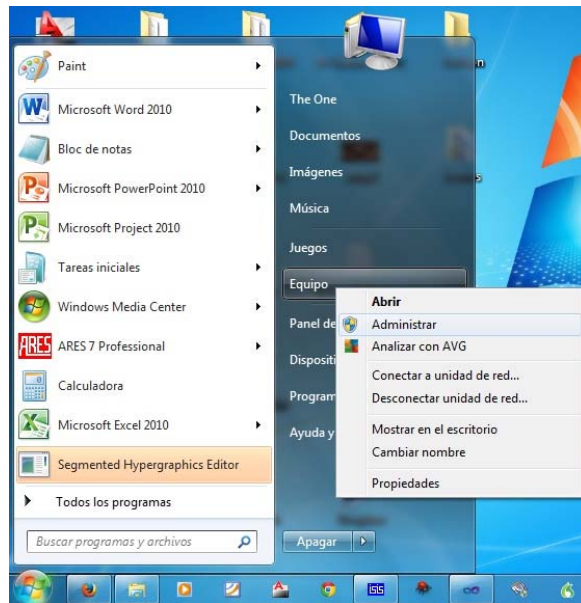


Figura 2, Equipo

- Ingresamos al “Administrador de Dispositivos”
- Buscamos la opción “Puertos (COM y LTP)”
- Verificamos el número del puerto que vamos a utilizar

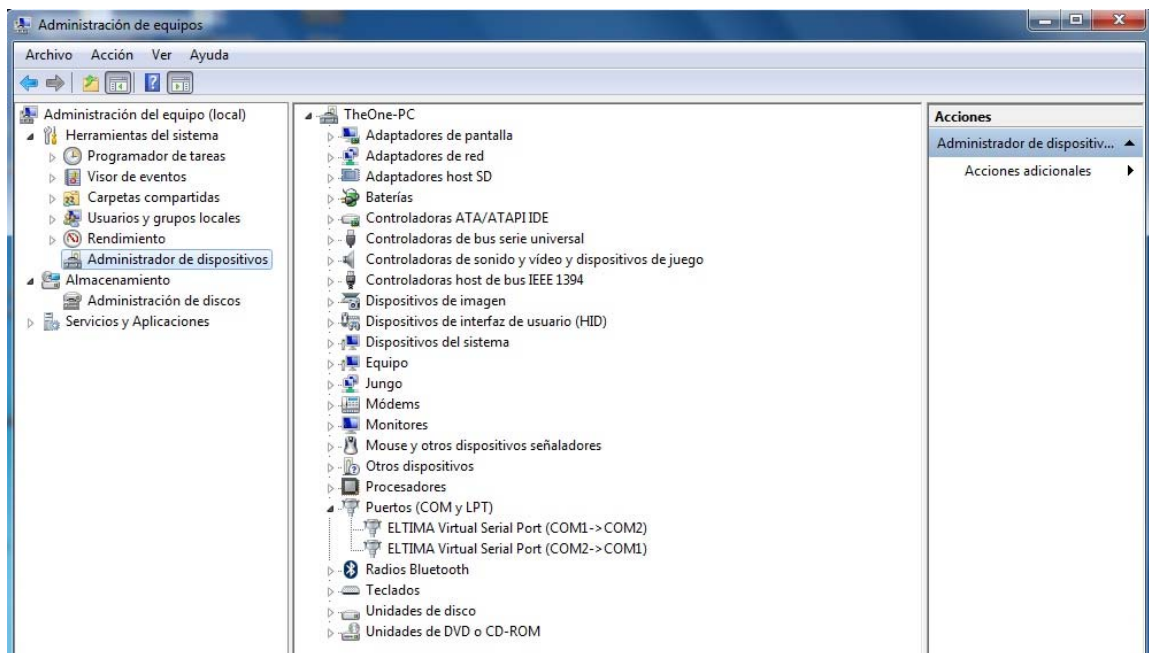


Figura 3, Administrador de Equipos

- Después de localizar el puerto de comunicación procedemos a establecer la comunicación entre el computador y la mano

1. Damos click en el botón “DetConexion” en la ventana principal

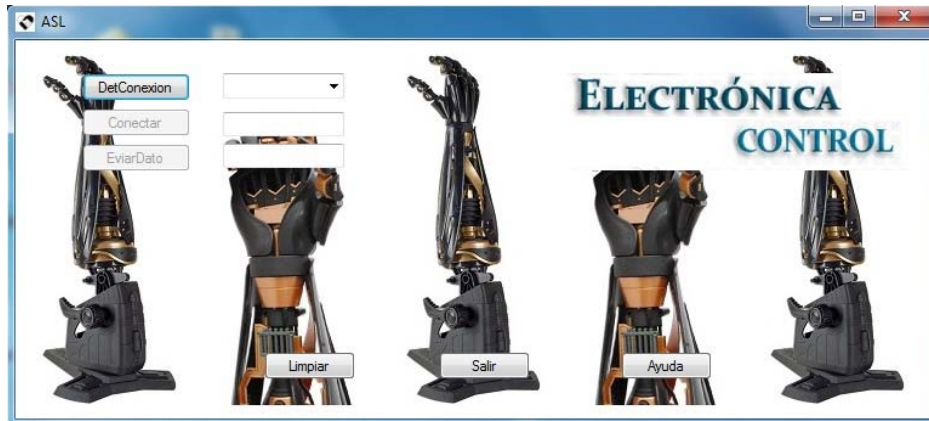


Figura 4, DetConexión

2. Damos click en “Aceptar”



Figura 5, Selección de puerto

3. Seleccionamos el puerto de comunicación que tenemos en la figura 3
4. Damos click en el botón “Conectar”

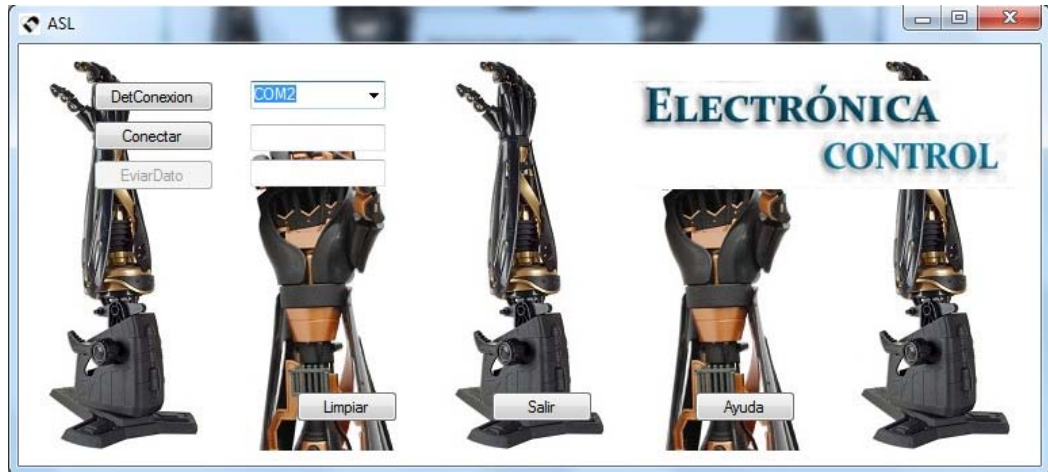


Figura 6, Conectar

5. Escribimos la palabra a traducir
6. Damos click en "Traducir a ASL"

BIBLIOGRAFÍA

- 1. Cox, B.**, Embedded C Programming and the Atmel Avr., 2a ed., NY – USA., Delmar Cengage 2012., Pp. 122-130-132-205.

- 2. Mikelm., J.**, Bascom AVR Programming., 1a ed., Ljubljana – Slovenia., AX electronica 2012., Pp. 20-135-140-150

- 3. Zuñiga., D.**, Mano Robótica para la Enseñanza del Alfabeto Dactolológico., 1a ed., SchlungsdienstLange O.H.G. – Berlin – Alemania., Saarbrücken 2012., 13-14-15-30-35-40-50-70.

4._ CONSEPTOS DE ELEMENTOS UTILIZADOS

- ✓ <http://bibdigital.epn.edu.ec/handle/15000/1774>
2013-07-10
- ✓ <http://www.unocero.com/2013/09/10/mano-robotica-a-partir-de-impresora-3d/>
- ✓ <http://www.ups.edu.ec/documents/2497096/2497486/Art8.pdf>
2013-07-24
- ✓ http://www.servocity.com/html/hs-311_standard.html
2013-08-10

5._ DATASHEET DE ELEMENTOS

- ✓ <http://www.alldatasheet.com>
2013-08-12
- ✓ http://www.datasheetcatalog.com/datasheets_pdf/M/A/X/2/MAX232.shtml
- ✓ <http://juandeg.tripod.com/conectordb9.htm>
- ✓ http://www.servocity.com/html/hs-311_standard.html
- ✓ http://www.datasheetcatalog.com/datasheets_pdf/T/L/4/9/TL494.shtml
- ✓ <http://www.servodatabase.com/servo/hitec/hs-311>
2013-08-14