

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA: INGENIERÍA ELECTRÓNICA**

**Tesis previa a la obtención del título de: INGENIERO ELECTRÓNICO**

**TEMA:  
“DISEÑO E IMPLEMENTACIÓN DE DOS CONTROLADORES PARA LOS  
BRAZOS ROBÓTICOS: ANTROPOMÓRFICO Y SCARA, UTILIZANDO  
MICROCONTROLADORES ATMEL AVR`S PARA EL LABORATORIO DE  
ROBÓTICA DE LA UNIVERSIDAD POLITÉCNICA SALESIANA CAMPUS  
SUR”**

**AUTOR:  
JAIME ROLANDO HEREDIA VELASTEGUI**

**DIRECTOR:  
CARLOS GERMÁN PILLAJO ANGOS**

**Quito, diciembre del 2013**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO  
DEL TRABAJO DE GRADO**

Yo Jaime Rolando Heredia Velasteguí autorizo a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaro que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del autor.

-----  
Jaime Rolando Heredia Velasteguí

CC.: 1721400768

## **DEDICATORIA**

Dedico a mis padres que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento.

## **AGRADECIMIENTOS**

Agradezco de manera especial y sincera al Ingeniero Carlos Pillajo por aceptarme para realizar este trabajo de grado bajo su dirección. Su apoyo y confianza en mi labor y su capacidad para guiar mis ideas siempre enmarcadas en su orientación y rigurosidad, han sido la clave del buen compromiso que hemos realizado juntos, el cual no se puede concebir sin su oportuna participación.

## ÍNDICE

INTRODUCCIÓN .....	1
CAPÍTULO 1 .....	3
HIPÓTESIS.....	3
1.1. Hipótesis de la investigación .....	3
1.2. Planteamiento del problema .....	3
1.3. Justificación e importancia del proyecto .....	4
1.4. Objetivos .....	4
1.4.1. Objetivo General .....	4
1.4.2. Objetivos Específicos.....	4
1.5. Alcances .....	5
CAPÍTULO 2 .....	6
ESTADO DEL ARTE.....	6
2.1.1. ¿Significado de Robótica? .....	6
2.1.2. Historia de la Robótica.....	6
2.2. Mecatrónica.....	8
2.3. Herramientas matemáticas .....	9
2.3.1. Descripción de posición y orientación .....	9
2.3.1.1. Posición.....	10
2.3.1.2. Orientación.....	12
2.4. Sensor.....	12

2.4.1. Tipos de sensores .....	13
2.4.1.1. Según la señal de salida.....	13
2.4.1.1.1. Los sensores digitales.....	13
2.4.1.1.1.1. Sensor de Final de Carrera .....	14
2.5. Encoder Incremental .....	15
2.6. Microcontroladores .....	16
2.6.1. Interrupciones en un microcontrolador (ISR) .....	17
2.6.2 Fuses en un microcontrolador. ....	18
2.7. Visión artificial.....	19
2.8. Motor.....	20
2.8.1. Máquinas de corriente continua .....	20
2.8.2. Aspectos constructivos del motor de corriente continua.....	21
2.8.3. Motor Brushless .....	22
2.9. LabVIEW .....	23
2.9.1. ¿Qué es LabVIEW?.....	23
2.9.2. Panel Frontal .....	23
2.9.3. Diagrama de bloques.....	24
2.10. Matlab .....	24
2.10.1. Guide.....	24
2.10.2. Vrbuild .....	25
CAPÍTULO 3 .....	26
DESARROLLO DE HARDWARE Y SOFTWARE .....	26

3.1. Introducción .....	26
3.2. Hardware .....	28
3.2.1. Módulo L298N Puente H Driver Para Motores .....	29
3.2.1.1. Características Generales Del Módulo De Potencia .....	30
3.2.1.2. Circuito Integrado L298N .....	31
3.2.2. Módulo de control .....	31
3.2.2.1 Fuente de alimentación .....	32
3.2.2.2 Distribución de pines del ATMEGA 128 en el circuito de control.....	33
3.2.2.3 Tarjeta de interfaz USB con el microcontrolador ATMEGA 8 .....	41
3.2.3 Modulo del Teach Pad .....	44
3.2.4 Encoder .....	44
3.3. Software .....	46
3.3.1 Diseño de interfaz gráfica en Guide de Matlab.....	46
3.3.2 Construcción de los brazos Antropomórfico y Scara en 3D utilizando el software vrbuild de Matlab .....	50
3.3.3 Desarrollo del programa en el script de Matlab .....	53
3.3.4 Desarrollo del programa para el control de los brazos antropomórfico y Scara en Labview .....	57
3.3.2.1 Visión artificial en Labview .....	63
CAPÍTULO 4 .....	67
PRUEBAS Y RESULTADOS .....	67

4.1. Análisis del tiempo requerido para realizar una misma trayectoria.....	67
4.1.1 Análisis del tiempo requerido para realizar una misma trayectoria utilizando el brazo scara.....	67
4.1.2 Análisis del tiempo requerido para realizar una misma trayectoria utilizando el brazo Antropomórfico.....	69
4.2. Precisión y Repetitividad .....	70
4.3. Exactitud .....	71
4.4 Resolución Espacial .....	77
4.5 Análisis de costos.....	78
4.5.1 Costo de desarrollo de Hardware .....	78
4.5.2 Costo de desarrollo del software.....	79
CONCLUSIONES .....	80
RECOMENDACIONES .....	82
LISTA DE REFERENCIAS .....	83
ANEXOS .....	86



## ÍNDICE DE FIGURAS

Figura 1 Diagrama de relación entre cinemática inversa y directa .....	9
Figura 2 Gráfica de desplazamiento de brazo antropomórfico .....	11
Figura 3 Gráfica de desplazamiento de brazo scara.....	12
Figura 4 Estructura de un sensor final de carrera.....	14
Figura 5 Amplificador Operacional funcionamiento en corte y saturación.....	15
Figura 6 Grafica de la superficie reflectante .....	16
Figura 7 Diagrama de pines del microcontrolador Atmel Avr 128 .....	17
Figura 8 Diferencia entre motor Brushed y un motor Brushless. ....	22
Figura 9 Diagrama Esquemático De Un Motor Brushless.....	22
Figura 10 Diagrama de bloques del sistema de control del brazo scara .....	26
Figura 11 Diagrama de bloques del sistema de control del brazo antropomórfico....	27
Figura 12 Tarjeta de control.....	29
Figura 13 Conexión de los pines del driver de potencia .....	30
Figura 14 Circuito del módulo de potencia con El C.I. L298N .....	31
Figura 15 Diagrama de la fuente de alimentación .....	32
Figura 16 Distribución de pines del conector vista desde el encoder .....	39
Figura 17 Conexión de cada motor a la tarjeta de control en el brazo Scara según su etiqueta .....	40
Figura 18 Conexión de cada motor a la tarjeta de control en el brazo Antropomórfico según su etiqueta .....	40

Figura 19 Conexiones De Pines Del Atmega 128.....	41
Figura 20 Interfaz de comunicación entre el microcontrolador ATMEGA128 y la PC .....	42
Figura 21 Circuito del Controlador .....	43
Figura 22 Diseño del circuito de Teach Pad .....	44
Figura 23 Circuito del Encoder .....	45
Figura 24 Tarjeta de control implementada al robot scara y antropomórfico.....	46
Figura 25 Diseño en Guide del <i>teach pad</i> para el Brazo Scara.....	47
Figura 26 Diseño en Guide del <i>teach pad</i> para el Brazo Antropomórfico.....	47
Figura 27 Ubicación de la opción View Callbacks.....	49
Figura 28 Formas para realizar la figura .....	50
Figura 29 Extrusión modificada.....	51
Figura 30 Diseño final del Brazo Antropomórfico .....	52
Figura 31 Diseño final del Brazo Scara .....	52
Figura 32 Movimiento de cada articulación y ubicación de etiquetas en el brazo Scara.....	53
Figura 33 Movimiento de cada articulación y ubicación de etiquetas en el brazo Antropomórfico.....	53
Figura 34 Comando para insertar una imagen .....	54
Figura 35 Comando para incorporar al archivo de vrbuid.....	54
Figura 36 Comando para la incorporación de puntos de vista .....	54

Figura 37 Comandos del Callbacks.....	55
Figura 38 Comando para cargar el archivo del mundo virtual.....	55
Figura 39 Comando para configurar el sentido de giro de un elemento en el mundo virtual .....	56
Figura 40 Comando para notificar un problema .....	56
Figura 41 Mensaje de aviso.....	56
Figura 42 Ubicación de Buttons and Switches .....	57
Figura 43 Diagramas del case structure y sequence structure.....	58
Figura 44 Sub VI para comunicación serial .....	59
Figura 45 Diagrama de bloques del proceso de envío de los dígitos a la tarjeta .....	59
Figura 46 Diagrama del proceso de envío de datos en labview.....	60
Figura 47 Concatenación de caracteres para el envío .....	61
Figura 48 Posición de reset desde Labview .....	62
Figura 49 Constantes según el rango de giro en la articulación Hombro .....	63
Figura 50 Modulo de Visión Acquisition.....	64
Figura 51 Configuración del patrón de color .....	65
Figura 52 Limitación del área de búsqueda del patrón de color .....	65
Figura 53 Limitación de rango y programa de control .....	66
Figura 54 Trayectoria realizada por el brazo scara .....	67
Figura 55 Trayectoria realizada por el brazo Antropomórfico .....	69
Figura 56 Representación gráfica de precisión y repetitividad.....	71

Figura 57 Representación gráfica de la resolución escalar .....	78
---	----

## ÍNDICE DE TABLAS

Tabla 1 Rango de movimiento en cada eje del Brazo antropomórfico y scara.....	11
Tabla 2 Fuses habilitados en el Atmega 128.....	18
Tabla 3: Principales Características del ATMEGA 128 .....	33
Tabla 4 Distribución de pines del ATMEGA 128 .....	34
Tabla 5 Función asignada a cada pin .....	39
Tabla 6 Características De Microcontrolador ATMEGA8 .....	41
Tabla 7 Descripción de componentes del Guide.....	48
Tabla 8 Resultado de cocientes .....	59
Tabla 9 Tabla de dígitos a enviar .....	60
Tabla 10 Designación de letras para el control de sentido de giro del brazo Antropomórfico.....	61
Tabla 11 Designación de letras para el control de sentido de giro del brazo Scara...	62
Tabla 12 Resultados de los tiempos utilizados por el brazo scara para cumplir una misma trayectoria el brazo Scara .....	68
Tabla 13 Resultados y tiempos en que cumple una trayectoria el brazo Antropomórfico.....	70
Tabla 14 Rangos de valores obtenidos en la prueba de exactitud para el brazo Antropomórfico.....	72
Tabla 15 Tabulación de datos obtenidos del desplazamiento (mm) en eje X del brazo antropomórfico.....	73
Tabla 16 Tabulación de datos obtenidos del desplazamiento (mm) en el eje Y del brazo antropomórfico .....	73

Tabla 17 Tabulación de datos obtenidos del desplazamiento (mm) en el eje Z del brazo antropomórfico .....	74
Tabla 18 Resultado de errores absoluto y relativo en el movimiento del brazo antropomórfico en el eje X.....	74
Tabla 19 Resultado de errores absoluto y relativo en el movimiento del brazo antropomórfico en el eje Y.....	74
Tabla 20 Resultado de errores absoluto y relativo en el movimiento del brazo antropomórfico en el eje Z.....	75
Tabla 21 Resultados totales de errores absolutos y relativos en la trayectoria cumplida por el brazo antropomórfico .....	75
Tabla 22 Rangos de valores obtenidos en la prueba de exactitud para el brazo Scara .....	75
Tabla 23 Tabulación de datos obtenidos del desplazamiento en (mm) del brazo Scara en el eje X.....	76
Tabla 24 Tabulación de datos obtenidos del desplazamiento en (mm) del brazo Scara en el eje Y.....	76
Tabla 25 Resultado de errores absolutos y relativos en el movimiento del brazo Scara en el eje X.....	77
Tabla 26 Resultado de errores absoluto y relativo en el movimiento del brazo Scara en el eje Y.....	77
Tabla 27 Resultados totales de errores absolutos y relativos en la trayectoria cumplida por el brazo scara.....	77
Tabla 28 Resolución Espacial del brazo Scara y Antropomórfico encontradas según pruebas realizadas .....	78
Tabla 29 Costos de hardware .....	79



## ÍNDICE DE ECUACIONES

Ecuación 1 Rango de desplazamiento de coordenadas (X Y Z) .....	10
Ecuación 2 Rango en el que puede desplazarse el brazo scara en los diferentes ejes	10
Ecuación 3 Rango en el que puede desplazarse el brazo antropomórfico en los diferentes ejes.....	11
Ecuación 4 Nomenclatura de un sistema en tres dimensiones .....	12
Ecuación 5 Formula del valor medio .....	72
Ecuación 6 Formula de la desviación.....	72
Ecuación 7 Formula de error absoluto .....	72
Ecuación 8 Formula de error relativo.....	72



## ÍNDICE DE ANEXOS

Anexo 1 Programa del Microcontrolador para el control de los motores.....	86
Anexo 2 Programa realizado en Bascom para el control de los robots mediante un teach pad.....	101
Anexo 3 Datasheet del Micro controlador ATMEGA 128 .....	129
Anexo 4 Especificaciones del driver de potencia .....	132

## **RESUMEN**

El trabajo tiene como propósito dotar a la Universidad Politécnica Salesiana de dos controladoras para los brazos robóticos Rhino Scara y XR-3 Antropomórfico. Se realizó algunas pruebas técnicas en las controladoras originales de cada brazo robótico y se llegó a la conclusión de que eran antiguas y se encontraban en mal estado. Se propuso la innovación para el control de los brazos utilizando un micro controlador ATMEGA AVR 128, con el cual se puede optimizar el peso, consumo de energía y espacio.

El uso del Micro controlador AVR también permite la implementación de otras aplicaciones muy útiles dentro de la robótica, como es la visión artificial, integrada al brazo XR-3 antropomórfico por medio de una cámara web, para la clasificación de piezas blancas u otros colores cambiando el patrón de matiz. Los brazos robóticos fueron dotados de un teach pad gráfico utilizando una pantalla GLCD y un micro controlador ATMEGA AVR 164.

## **ABSTRACT**

This paper aims to provide the Salesian Polytechnic University two controllers for robotic arms, and Scara Rhino XR-3 Anthropomorphic. We performed some testing techniques in original controllers each robotic arm and concluded that they were old and in poor condition. Innovation was proposed for the control arms using a microcontroller ATMEGA AVR 128, with which you can optimize the weight, power consumption and space.

Using the AVR Microcontroller also allows the implementation of other useful applications in robotics, such as computer vision, integrated into XR-3 anthropomorphic arm through a webcam, for the classification of white pieces or other colors changing shade pattern. The robotic arms were equipped with a graphic pad teach using a GLCD display and AVR ATMEGA 164 microcontroller.

## INTRODUCCIÓN

La empresa Robots Rhino de origen Estadounidense se dedica al desarrollo de una amplia gama de robots desarrollados exclusivamente para la educación, formación e investigación. Robots Rhino proporciona los accesorios para la reparación y restauración de cada uno de sus modelos de robots. Además puede suministrar información sobre mejoras a los brazos de la serie XR existentes.

Existen en el mercado dos modelos de robots que debido a su calidad, eficiencia y manejo se utiliza en todo el mundo, los modelos son: Rhino Scara y Rhino Robotics Model XR-3.

The Rhino Scara es un brazo robot de cuatro ejes con una pinza accionada eléctricamente. Puede ser operado ya sea desde el Controller Mark III o el controlador Mark IV. El robot tipo SCARA Rhino es un relativamente grande. Robots Scara se utilizan principalmente en las operaciones de automatización de montaje de alta precisión.

Rhino Robotics Model XR-3 es un brazo antropomórfico, siendo un robot popular entre los educadores que necesitan un diseño abierto, resistente y adecuado para la introducción de los estudiantes de nivel universitario al campo de la robótica.

El modelo XR-3 Brinda un correcto funcionamiento por años sin problemas. Fue desarrollado con seis servomotores PMDC con caja de reducción y encoders incrementales integrados en la estructura de cada motor. Para obtener la posición de inicio utiliza micro interruptores en todos los ejes y en el actuador. El mantenimiento del brazo es fácil al igual que la reparación de toda la estructura. El brazo antropomórfico XR-3 es un diseño popular, con instalaciones en centros educativos de todo el mundo.

Con el desarrollo del proyecto de grado se deja en desuso la tarjeta Controller Mark III y Controller Mark IV, y mediante un micro controlador actualizamos la tarjeta de control. El nuevo sistema de control es más eficiente en muchos aspectos como son: tamaño, peso, movilidad, energía, etc.

Para el desarrollo del proyecto los lineamientos son los siguientes: diseño mecánico, diseño electrónico, desarrollo del programa en el software Labview y Matlab para el control y simulación y la construcción de un Teach Pad para el control manual de los robots.

Para el desarrollo del proyecto se plantean los siguientes capítulos:

- Hipótesis
- Diseño del hardware
- Diseño del Software
- Pruebas de Implementación

# CAPÍTULO 1

## HIPÓTESIS

### 1.1. Hipótesis de la investigación

Con la intención de dar solución al problema formulado se plantea la siguiente hipótesis:

Diseñando e implementando controladores actuales para los brazos robóticos antropomórfico y Scara, se dotará al Laboratorio de Robótica de la Universidad Politécnica Salesiana Campus Sur de robots con tecnología moderna, garantizando el mejor aprendizaje de los estudiantes de la carrera de ingeniería Electrónica.

### 1.2. Planteamiento del problema

En la actualidad muchas fábricas utilizan robots para el ensamblaje de sus productos, ya que la eficiencia en producción que se logra es muy alta.

El campo de la robótica es desarrollado en Universidades Politécnicas, muchas de estas tienen laboratorios con robots de varios tipos, los cuales sirven para enseñar al alumno de una manera didáctica el funcionamiento de cada uno de estos.

La Universidad Politécnica Salesiana Campus Sur, en su laboratorio de robótica posee algunos brazos robóticos entre estos un robot Scara y un Antropomórfico, los cuales constan de sus controladoras para su manejo. Con el avance de la tecnología, estas tarjetas de control han quedado obsoletas y en desuso.

Con el diseño y la implementación de tarjetas de control actuales para los brazos robóticos Scara y Antropomórfico que se encuentran en La Universidad Politécnica Salesiana Campus Sur, se podrá tener una gama de brazos robóticos en uso para el laboratorio de la Institución, la cual garantizará un mejor aprendizaje para los estudiantes del Campus Sur.

### **1.3. Justificación e importancia del proyecto**

Como se conoce en la actualidad las empresas necesitan invertir grandes recursos económicos para automatizar sus procesos. Este proyecto de investigación se justifica ya que, se presenta como una alternativa para que las PYMES (pequeñas y medianas empresas) puedan acceder a la automatización de sus procesos, utilizando brazos robóticos, sin la necesidad de invertir gran capital.

Los controladores a diseñarse e implementarse, actualizaran y reemplazaran los equipos obsoletos existentes en los laboratorios de la Universidad Politécnica Salesiana Campus Sur, esta implementación mejorará el aprendizaje y trabajos de investigación futura.

### **1.4. Objetivos**

Para cumplir el proyecto se ha planteado los siguientes objetivos:

#### **1.4.1. Objetivo General**

Diseñar e implementar dos controladores para brazos robóticos: antropomórfico y scara, utilizando micro controladores Atmel AVR`s para el control de movimiento y velocidad, con una interfaz USB para su manejo mediante un software de entorno gráfico, constara de un teach pad gráfico para su control manual y un hardware que conste de drivers de potencia y encoders diferenciales que determinaran la posición en cada una de las articulaciones de los robots.

#### **1.4.2. Objetivos Específicos**

- Analizar e investigar el estado del arte de los brazos robots.
- Diseñar y construir el hardware de: los controladores, los drivers de potencia y la interfaz USB.
- Diseñar los programas en los software´s: Matlab y Labview de entorno gráfico.
- Diseñar e implementar la interfaz entre el brazo robot y el Computador.

## 1.5. Alcances

Diseño e implementación de controladoras con tecnología moderna, con la capacidad de:

- Control manual de los robots, con un “teach pad” que utiliza un GLCD gráfico.
- Implementar en cada robot una controladora que usa un micro controlador ATMEGA 128 para el control.
- Diseño e implementación de un programa en el *software Labview y Matlab* para control y simulación respectivamente de los robots scara y antropomórfico.
- Diseño e implementación de circuitos electrónicos de potencia para el control de cada motor PMDC.
- Diseño e implementación de un circuito electrónico con interfaz USB para el control de cada robot mediante el computador.
- Cada robot puede moverse a diferentes posiciones de forma manual según la necesidad del usuario además control se puede realizar con algoritmos de programación los cuales quedan a criterio de futuros tesisistas.



## **CAPÍTULO 2**

### **ESTADO DEL ARTE**

Es un estudio general sobre Mecánica, Robótica, Automatización Industrial y electrónica, con el propósito de documentar información necesaria para el desarrollo del proyecto de grado planteado.

#### **2.1. Robótica**

La Robótica se relaciona en sí con el deseo de sintetizar algunos aspectos de la función humana mediante el uso de mecanismos, sensores, actuadores y computadoras. Obviamente esto representa un enorme compromiso que evidentemente parece requerir una multitud de ideas provenientes de varios campos clásicos. En la actualidad expertos de varios campos trabajan en la investigación de distintos temas de la robótica. Generalmente no es común que un solo individuo domine todos los campos de la robótica, es natural esperar una subdivisión. En un nivel relativamente alto de abstracción parece razonable dividir la robótica en cuatro áreas principales: manipulación mecánica, locomoción, visión computacional e inteligencia artificial. (Craig, 2006, pág. I)

##### **2.1.1. ¿Significado de Robótica?**

El significado de robótica según el Diccionario de la real academia de la lengua Española es el siguiente: “Técnica que aplica la informática al diseño y empleo de aparatos que, en sustitución de personas, realizan operaciones o trabajos, por lo general en instalaciones industriales.” (Real Academia de la Lengua, 2013)

##### **2.1.2. Historia de la Robótica**

La adopción de equipos robóticos empezó a inicios del año 1980 en países como: Estados Unidos de América, Japón y varios países Europeos. La demanda de robots industriales para el control de procesos aumentaba o disminuía según el estado económico de cada país, de la cantidad de robots

que se instalaron en los años 1995-2000, se debe resaltar que Japón casi duplica en número de robots a su inmediato seguidor que es la Unión Europea, esto se debe a que en este país se considera robot a una máquina que en los otros países no es más que una máquina de fábrica.

En la década de 1990 los dueños de las grandes empresas a nivel mundial se empeñaron en comprar una gran cantidad de robots industriales para controlar procesos manuales que tenían en cada una de sus fábricas, cada robots según salía al mercado empezaba a disminuir su costo, lo cual era ventaja ya que también ofrecían muchas otras características como: bajo costo de producción, efectividad, rapidez, flexibilidad y precisión, los precios de los robots empezaron a disminuir mientras que el costo por mano de obra humana empezaba a aumentar significativamente. (Craig, 2006, págs. 1-10)

“Si factorizamos en los números estos ajustes de calidad, el costo por utilizar robots disminuye muchos más rápido que su etiqueta de precio” (Craig, 2006, págs. 1-2).

Si entendemos de forma clara esta idea podemos llegar a la siguiente conclusión: mientras más bajo sea el costo de cada robot útil en una área específica en el proceso industrial y a medida que aumente el costo de mano de obra humana para el mismo proceso industrial, este trabajo será un candidato innato para que se lo realice por un robot industrial, esta ideología a echo que el mercado de robots pueda crecer y afianzarse en el mercado.

La principal idea de reemplazar a un humano por un robot se dio por prevenir posibles riesgos laborales que existen dentro de áreas de procesos industriales.

“Si un dispositivo mecánico que puede programarse para realizar una amplia variedad de aplicaciones, probablemente sea un robot industrial” (Craig, 2006, pág. 2)

El estudio realizado para el control de los brazos robóticos incluyen varios temas necesarios, el desarrollo de situaciones de una maquina según la estática y la dinámica, las ciencias exactas como la matemática entregan información para describir movimientos espaciales y atributos o destrezas que

queremos entregar a cada manipulador, la teoría de control ayuda para que el diseño de cada algoritmo realice los movimientos deseados, la Ingeniería Eléctrica proporciona información sobre la cantidad de corrientes, voltajes y potencia que necesita cada motor para ser controlado efectivamente además nos ayuda a identificar los sensores que debimos utilizar según el trabajo a cumplir de cada brazo robótico y finalmente la parte de Ingeniería Informática nos entrega la base para poder controlar el brazo por medio de una programación limpia y eficiente por medio de uno o varios software . En la programación esta la esencia de cada manipulador, todas las señales de sensores, entregadas según el proceso a realizarse deben ser adquiridas y se debe entregar una respuesta muy eficiente a cada inconveniente que se presente. (Craig, 2006, págs. 1-19)

## **2.2. Mecatrónica**

“La ingeniería mecánica, como práctica profesional extensa, experimentó un crecimiento durante principios del siglo XIX porque proporciono un cimiento necesario para el rápido y exitoso desarrollo de la revolución industrial.” (Alciatore & Hystand, 2008).

Otras definiciones del término “Mecatrónica” es la que se le dio por los Japoneses en la década de los 60, cuya definición se dispersó por toda Europa y en la actualidad se la utiliza comúnmente en los Estados Unidos de América. Las principales disciplinas involucradas en el diseño de sistemas mecatrónicos incluyen a la ingeniería mecánica, electrónica, de control y computación. Un ingeniero que estudio mecatrónica debe tener la capacidad de diseñar y seleccionar circuitos análogos o digitales, componentes basados en microprocesadores, dispositivos mecánicos, sensores y actuadores, y controladores, de modo que el producto final logre la meta deseada.

Los sistemas desarrollados en el ámbito de la mecatrónica son considerados como dispositivos inteligentes, esto se debe a la complejidad para su desarrollo, involucrando elementos tales como la lógica, la retroalimentación

y la computación, el objetivo de desarrollar sistemas complejos es poder llegar a simular el proceso de pensamiento del ser humano.

Actualmente los sistemas constan de una parte electrónica, eléctrica, mecánica y de control computacional, llevando a la conclusión de que un sistema de instalación mecatrónica hoy en día se utiliza comúnmente dentro de las diferentes áreas como por ejemplo la automatización. (Alciatore & Histan, 2008)

### 2.3. Herramientas matemáticas

El subcapítulo de modelo matemático muestra las herramientas necesarias para poder ubicar al efector final de cada robot en una posición deseada. Como se explicó en los antecedentes el movimiento del efector según cualquier algoritmo queda como preámbulo para el desarrollo de nuevos trabajos de grado.

El modelo matemático de un robot es el algoritmo que desarrolla la controladora para cumplir las diferentes tareas que le sean asignadas, esto se refiere a que el funcionamiento de cada robot va a depender de su modelo matemático, al diseñar una controladora podemos utilizar el modelo matemático que cumpla nuestras expectativas.

En este subcapítulo analizaremos el movimiento del robot con respecto a un sistema de referencia situado en la base. Obtendremos una descripción analítica del movimiento espacial y, en particular, de la posición y orientación del extremo final del robot.

Figura 1 Diagrama de relación entre cinemática inversa y directa



Fuente: (Zambrana, 2009)

#### 2.3.1. Descripción de posición y orientación

En el estudio general sobre la robótica siempre existe la incógnita de cómo ubicar a un cuerpo dentro del sistema tridimensional, entonces nos referiremos a dos temas básicos pero de mucho interés como son: la posición y la orientación.

### **2.3.1.1. Posición**

Cuando se desea localizar un cuerpo rígido en el espacio se necesitan herramientas que nos permitan conocer la ubicación espacial de sus puntos. En el plano la localización se describe por dos componentes independientes que comúnmente son las coordenadas  $(x, y)$  o el ángulo y una distancia esto de da según el sistema de coordenadas que estemos utilizando, mientras que en el espacio tridimensional son necesarios tres componentes por ejemplo en el sistema de coordenadas cartesianas la nomenclatura es la siguiente:  $(x,y,z)$ .

#### **2.3.1.1.1. Coordenadas Cartesianas**

El sistema de coordenadas cartesianas  $(x,y,z)$  es el que tomaremos como sistema básico. Este sistema se define como el conjunto de distancias (con signo) a tres planos ortogonales.

La coordenada  $x$  es la distancia al plano  $YZ$ .

La coordenada  $y$  es la distancia al plano  $XZ$ .

La coordenada  $z$  es la distancia al plano  $XY$ .

El signo en estas coordenadas indica a qué lado del plano se encuentra el punto. Las tres coordenadas varían en el mismo rango:

Ecuación 1 Rango de desplazamiento de coordenadas (X Y Z)

$$x \in -\infty, \infty \quad y \in -\infty, \infty \quad z \in -\infty, \infty$$

Fuente: (Tomás, 2010)

Ecuación 2 Rango en el que puede desplazarse el brazo scara en los diferentes ejes

$$x \in 10,45 \quad y \in -45,45 \quad z \in 0,9$$

Elaborado por: Jaime Heredia

Ecuación 3 Rango en el que puede desplazarse el brazo antropomórfico en los diferentes ejes

$$x \in 17,45 \quad y \in -45,45 \quad z \in 0,90$$

Elaborado por: Jaime Heredia

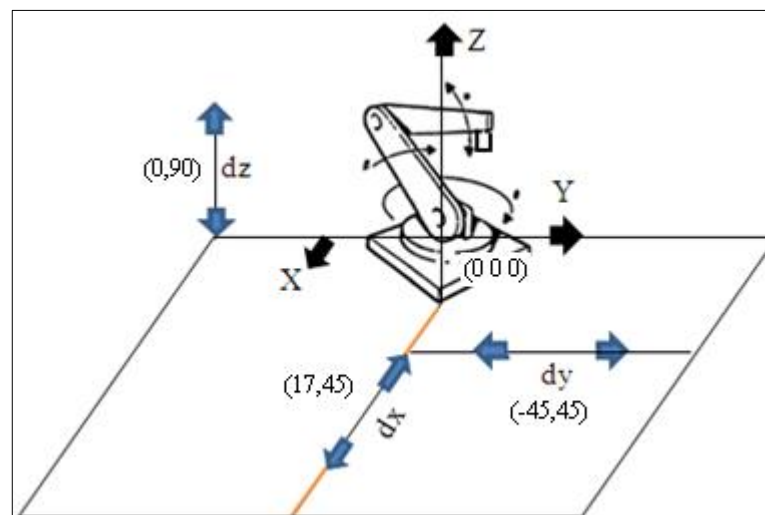
Para el proyecto se ha asignado un rango de movimiento para cada articulación. Cada robot puede desplazarse hasta una máxima distancia para evitar colisiones o roturas de las cadenas que componen cada brazo, el rango fue delimitado según el área de trabajo de cada robot. En la siguiente tabla se indica los rangos máximos de movimiento del efector final.

Tabla 1 Rango de movimiento en cada eje del Brazo antropomórfico y scara

Eje		Scara			Antropomórfico		
		Posición de inicio (cm)	Posición final (cm)	Distancia (cm)	Posición de inicio (cm)	Posición final (cm)	Distancia (cm)
X	dx	10	45	35	17	45	28
Y	dy	-45	45	90	-45	45	90
Z	dz	0	9	9	0	90	90

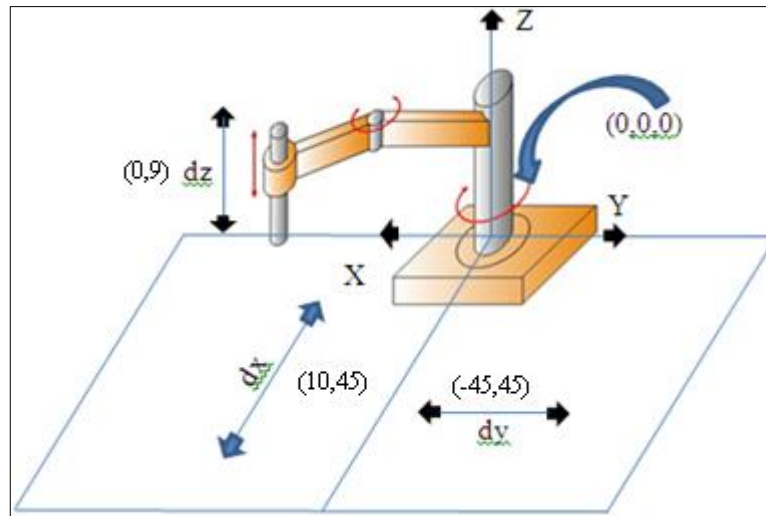
Elaborado por: Jaime Heredia

Figura 2 Gráfica de desplazamiento de brazo antropomórfico



Elaborado por: Jaime Heredia

Figura 3 Gráfica de desplazamiento de brazo scara



Elaborado por: Jaime Heredia

En términos de las coordenadas cartesianas, el vector de posición se escribe como:

Ecuación 4 Nomenclatura de un sistema en tres dimensiones

$$\mathbf{r} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

Fuente: (Tomás, 2010)

### 2.3.1.2. Orientación

Para la representación de un punto es necesario solamente indicar sus coordenadas, dos o tres según el sistema, para un cuerpo es necesario además indicar su orientación en el espacio. Por ejemplo para un robot no basta con conocer su ubicación en el espacio, sino que es necesario también conocer la orientación de su extremo para que realice adecuadamente la operación.

### 2.4. Sensor

“Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, da una señal de salida trasducible que es función de la variable medida” (Areny, 2007, págs. 356-357)

Un sensor es un elemento que obtiene información de alguna magnitud física que es imperceptible a los sentidos.

Una medida física es directa cuando se deduce información cuantitativa acerca de objeto físico o acción mediante comparación directa con una interferencia. A veces se puede hacer simplemente de forma mecánica, como en el caso de una balanza clásica.

En medidas indirectas la cantidad de interés se calcula a partir de otras medidas y de la aplicación de la ecuación que describe la ley que relaciona dichas magnitudes. Los métodos empleados suelen ser siempre eléctricos. Es el caso por ejemplo, de la medida de la potencia transmitida por un eje a partir de la medida del par y de la medida de su velocidad de rotación. (Areny, 2007, págs. 350-356)

### **2.4.1. Tipos de sensores**

Los sensores se han clasificado según criterios de utilidad, debido a la cantidad de sensores disponibles para medir las distintas magnitudes físicas presentes en los campos de procesos industriales. El sensor utilizado en este proyecto es uno de los más simples como es el final de carrera el cual envía una señal TTL en alto cuando el robot busca ubicarse en la posición inicial. Los sensores han sido clasificados de la siguiente manera:

#### **2.4.1.1. Según la señal de salida**

En este grupo los sensores se dividen en analógicas y digitales dependiendo de la forma de señal que se entrega a la salida.

##### **2.4.1.1.1. Los sensores digitales**

En este tipo de sensores la salida oscila en saltos o pasos discretos, esto quiere decir que una señal digital, en contraste con la señal analógica, no varían en forma continua, sino que cambian en pasos o en incrementos discretos, la mayoría de las señales digitales utilizan códigos binarios o de dos estados, una ventaja de tener una señal digital es la fácil transmisión de la señal de salida además de mayor fidelidad, fiabilidad y muchas mayor exactitud.



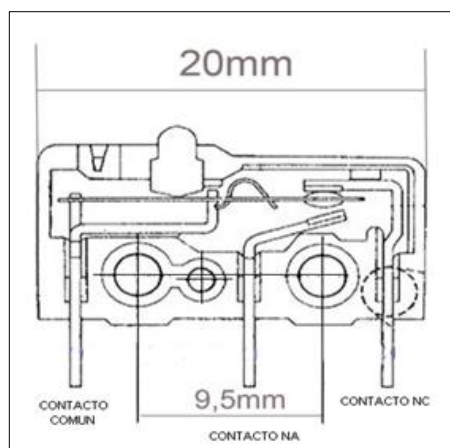
Una desventaja puede ser que no existe un modelo de sensor digital para todas las magnitudes físicas conocidas, o las de mayor interés. (Areny, 2007, págs. 356-365)

#### 2.4.1.1.1. Sensor de Final de Carrera

El final de carrera pertenece al grupo de sensores digitales debido a que la señal que entrega es booleana, es sensor de contacto o también conocido como interruptor de límite, son dispositivos eléctricos, neumáticos o mecánicos situados al final del recorrido de un elemento móvil, como por ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito. Internamente pueden contener interruptores normalmente abiertos (NA), cerrados (NC) o conmutadores dependiendo de la operación que cumplan al ser accionados.

Generalmente estos sensores están compuestos por dos partes: un cuerpo donde se encuentran los contactos y una cabeza que detecta el movimiento. Su uso es muy diverso, empleándose, en general, en todas las máquinas que tengan un movimiento rectilíneo de ida y vuelta o sigan una trayectoria fija, es decir, aquellas que realicen una carrera o recorrido fijo, como por ejemplo ascensores, montacargas, robots, etc. (Areny, 2007, págs. 360-363)

Figura 4 Estructura de un sensor final de carrera

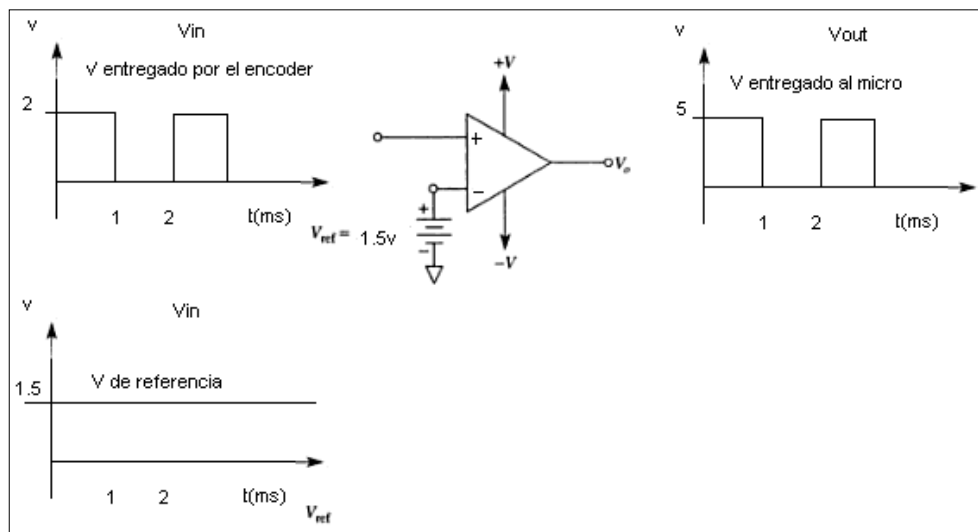


Elaborado por: Jaime Heredia

## 2.5. Encoder Incremental

El encoder es un transductor rotativo, que puede transformar un movimiento angular en una serie de impulsos eléctricos. La señal eléctrica que se obtiene es de forma cuadrada. La señal de salida es generada, por medio de la utilización de un amplificador operacional que funciona en corte y saturación con un voltaje de referencia de 1.5v, voltaje que es obtenido por un divisor de tensión. (Setecindca, 2000)

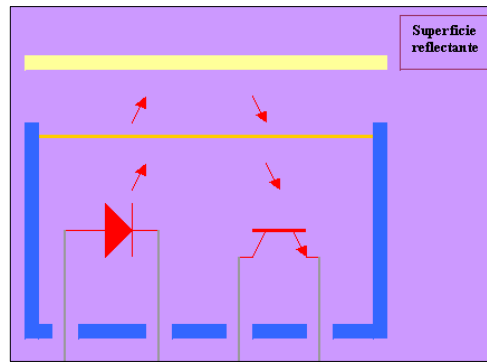
Figura 5 Amplificador Operacional funcionamiento en corte y saturación



Elaborado por: Jaime Heredia

La superficie del encoder consta de franjas claras y oscuras con una distribución alternada, en el momento que se polariza constantemente al diodo emisor la luz generada choca en la superficie reflectante o clara esta reflexión activa al diodo receptor y envía una cantidad de voltaje al amplificador operacional que al estar en corte y saturación produce un voltaje de 1 lógico o cinco voltios durante el tiempo en el que la superficie siga reflejando luz.

Figura 6 Grafica de la superficie reflectante



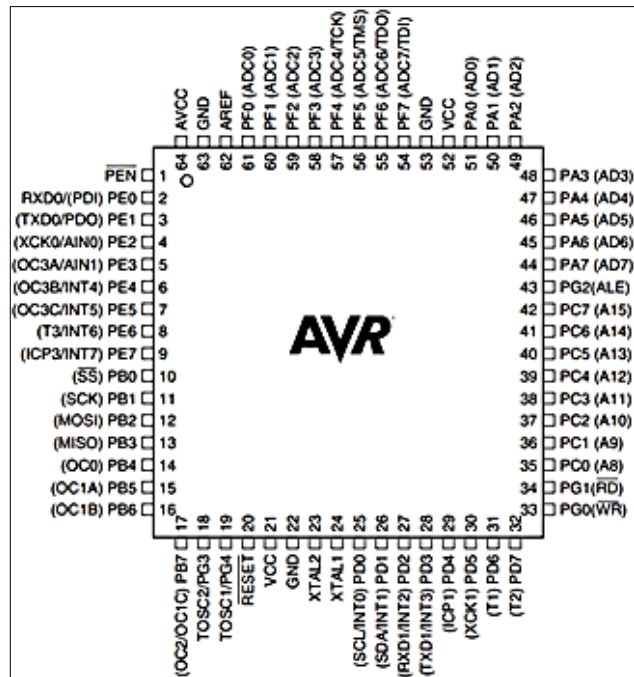
Elaborado por: Jaime Heredia

## 2.6. Microcontroladores

Existen dos ramas en la evolución en curso del microprocesador. Una rama apoya al CPU para la computadora personal y la industria de estaciones de trabajo, donde las principales restricciones son la alta velocidad y gran tamaño de palabra (32 y 64 bits). La otra rama incluye el desarrollo del microprocesador, que es un solo circuito integrado que contiene circuitos y funciones especializados aplicables al diseño de sistemas mecatrónicos; contiene un microprocesador, memoria, puertos de I/O y otros recursos en chip. (Alciatore & Histan, 2008)

Básicamente es una microcomputadora en un solo CI. El programa de control para los robots fue realizado en Bascom y se encuentra en el anexo 1. Ejemplos de Microcontroladores son los PICs de Microchip, el 68HC11 de Motorola, ATMEGA 128 de ATMEL microcontrolador utilizado en cada controladora de los brazos antropomórfico y scara para el envío y recepción de datos la información se encuentra en el anexo 3, ATMEGA8 empleado para la comunicación serial de la tarjeta con la PC, ATMEGA164p es usado en el circuito de cada teach pad siendo una de sus principales funciones el control del GLCD gráfico y el 8096 de Intel. El programa que controla el robot por medio del teach fue realizado en Bascom y se encuentra en el anexo 2.

Figura 7 Diagrama de pines del microcontrolador Atmel Avr 128



Fuente: (Alhammadi, 2006)

### 2.6.1. Interrupciones en un microcontrolador (ISR)

Los microcontroladores AVR con CPU de 8bits cuentan con diferentes interrupciones que son manejadas por hardware, lo que involucra que exista un vector de interrupciones en donde cada posición del vector representa la posición de memoria a donde accede el procesador cuando se genera una interrupción. Estas posiciones varían entre microcontroladores, debido a que cada uno cuentan con diferentes módulos de E/S. La hoja de datos de cada microcontrolador cuenta con la descripción de este vector. Existen varios aspectos que se deben tener en cuenta para el manejo de interrupciones. El primero de ellos es que aparte de habilitar la interrupción que se desea manejar, en este caso la interrupción 0, es necesario activar el habilitador global de interrupciones usando la sentencia macro sei(). Si no se activa el habilitador global de interrupciones no se generará ninguna interrupción. Lo segundo que se debe tener en cuenta es que las interrupciones no reciben ni retornan parámetros, por lo tanto si se desean manejar variables para que estas sean usadas en la lógica principal del programa es necesario usar variables globales. El último aspecto a tener en cuenta es que las interrupciones se

generan en cualquier momento del programa, esto es importante contemplarlo para no tener problemas con la lógica que se esta usando. (uCSystem, 2011)

### 2.6.2 Fuses en un microcontrolador.

Los Fuses en los micros AVR se utilizan para configurar parámetros importantes del sistema mientras que se puede ignorar algunas de las opciones, la mayor parte de las veces, sin duda se necesita tener acceso a algunos de estos valores por ejemplo, los bits de fusible de la fuente de reloj, que le permiten especificar el origen y/o la velocidad de reloj del chip. El proposito de saber programar los byte fuses es obtener el máximo rendimiento del microcontrolador. La familia AVR tiene uno o más Fuse Bytes. Cuántos hay, y lo que hacen, depende del microcontrolador, el ATmega8 tiene dos Fuse Bytes alto y bajo, para un total de 16 bits de configuración, mientras que un ATTINY12 tiene sólo un Fuse Byte. Los fusibles no se borran cuando de la memoria del AVR, por lo que la reprogramación de los fusibles cada vez que se reprograma el dispositivo no es requerida. Dado que los fusibles no son borrados por un borrado de memoria, pueden causar problemas si la configuración seleccionada es incorrecta. Los bits de bloqueo son fusibles muy utiles que se pueden manipular para bloquear el chip. En los microcontrolador ATMEGA128 utilizados en el proyecto fue necesario la habilitacion de los siguientes fuses. (Currás, 2011)

Tabla 2 Fuses habilitados en el Atmega 128

Fuses	Característica
Ext. Crystal/Resonator 16 MHz	Habilitación del fusible externo de 16 MHz.
Brown-out detection level at Vcc=4	Cuando existe una variación de voltaje, el microcontrolador necesitara un voltaje mínimo de 4V para su reseteo.
Serial Program (SPI)	SPI es un estándar para controlar un flujo de bits de forma serial regulado por una señal de reloj.
CKOPT fuse	Con CKOPT sin programar el AVR puede ser utilizado para frecuencias de reloj hasta 8 MHz.

Elaborado por: Jaime Heredia

## 2.7. Visión artificial

La mayoría de los robots industriales que hay instalados actualmente en los procesos productivos, están prácticamente incomunicados con el entorno que les rodea. La necesidad de tener programadas las acciones a efectuar, restringe el ambiente de trabajo a unas condiciones estrictas, al igual que a las pinzas o los materiales que se han de manipular.

Existen multitud de sensores capaces de informar de algunas características del ambiente que envuelve al robot, pero la más completa y la que confiere la máxima adaptabilidad a la máquina, es la visión artificial. (Sánchez, 2002, págs. 257-259)

La importancia de la visión, está confirmada por la gran cantidad de empresas e investigadores que se dedican a mejorar esta técnica. Sin embargo, todavía no se ha implantado la visión en los robots comerciales, dadas las dificultades que existen y que pueden redimirse en las siguientes:

Los sistemas de visión superan, en muchos casos, el coste del sistema de robot industrial. (uCSystem, 2011)

“Exigen extensos desarrollos de software, que potencian el núcleo central de Inteligencia Artificial preciso para que el robot actuara de acuerdo con la información del mundo exterior.” (Sánchez, 2002)

“Hay gran dificultad en el tratamiento de la información visual, debido a factores inherentes al mismo proceso, como son sombras, tipo de iluminación, imágenes unidimensionales y tridimensionales, oclusiones entre objetos, etc.” (Esqueda, 2002). El principal factor que crea confusiones en las ordenes que debe cumplir el robot es la cantidad de luxes, cuando estos son demasiados bajos la cámara hace caso omiso de los elementos a clasificar, a su vez cuando la cantidad de luxes es muy alta existe una confusión en la toma de decisiones debido al brillo que se produce.

“Las aplicaciones más interesantes de visión, dentro de la Robótica, son el reconocimiento y clasificación de objetos, el ensamblado, la soldadura, la

sincronización con otros dispositivos en movimiento y el guiado de robots móviles.”  
(Sánchez, 2002, págs. 283-298)

En el proyecto se ha implementado una cámara Genius Eye 310 con una resolución de 640x480, es compatible con windows y tiene pórtilo serial. La marca y especificaciones se deben configurar en módulo de visión de Labview, la cámara se encargara de buscar un patrón de color para la clasificación de piezas que serán tomadas desde una banda trasportadora.

## **2.8. Motor**

El significado de robótica según el Diccionario de la real academia de la lengua Española es el siguiente: “Máquina destinada a producir movimiento a expensas de otra fuente de energía.” (Real Academia de la Lengua, 2013).

### **2.8.1. Máquinas de corriente continua**

“Las máquinas de corriente continua pueden generar o producir una corriente eléctrica debido a este fenómeno se volvieron máquinas de gran importancia, las más comúnmente utilizadas para este propósito son: generadores o dínamos.” (Mora, 2008, págs. 107-115)

El motor de D.C. es una máquina con un alto grado de flexibilidad para el control de velocidad y del par. La demanda de los motores de D.C. se ha ido reduciendo debido a los bajos costos que tienen los motores en AC. Hoy en día el uso de máquinas c.c. como generador o dinamo está prácticamente obsoleto, esto se debe a que es más fiable generar, transportar y distribuir corriente alterna.

Para el desarrollo del presente proyecto se ha utilizado motores DC brushless o motores sin escobillas, estos motores son aplicados en las siguientes áreas: micros motores para los discos duros de computadoras y en servomotores de accionamiento, activados por interruptores electrónicos.

## **2.8.2. Aspectos constructivos del motor de corriente continua**

El motor de D.C. está constituido por el estator siendo esta la parte fija y el rotor la móvil, cada una de estas partes tiene sus respectivos componentes:

La carcasa o culata.- es una lámina metálica que envuelve al motor, si es una máquina de gran potencia la carcasa es de un material muy resistente como el acero mientras que si es para un motor de baja potencia se utiliza un material un poco más débil como es el hierro forjado.

Los núcleos polares.- son chapas forjadas en acero, donde se ubicara el devanado del inductor o excitación.

Zapata Polar.- es una expansión magnética que se encuentra muy cerca de los polos del motor.

Polo intermedio.- los polos intermedios también denominados polos auxiliares, inter polos se utilizan para mejorar la conmutación, el bobinado de estos polos se conecta en serie con el inducido.

Inducido.- en el inducido es donde se alojan los bobinados, se construye con discos de chapa de acero al silicio

Arrollamiento del inducido.- los arrollamientos son de tipo tambor anteriormente se utilizaba un arrollamiento tipo anillo pero quedo en desuso por el poco aprovechamiento que se tiene del cobre, el tipo tambor dispone ambos lados activos de las espiras en las generatrices externas del apilamiento de rotor. Las bobinas en las maquinas c.c. son cerrados.

Devanado del inductor.- es un conjunto de bobinas que forman el inductor, la capacidad de potencia del motor depende de la cantidad y espesor del alambre que se utilice para realizar en bobinado.

Colector de delgas o conmutador.- que es un conjunto de láminas de cobre, llamadas delgas, aisladas entre sí por una capa de mica y que giran solidariamente con el rotor. Las delgas están conectadas eléctricamente a las bobinas del devanado inducido y



por medio de ellas dicho devanado se puede conectar con el exterior. Cada delga está unida eléctricamente punto de conexión de dos bobinas del devanado inducido, de tal forma que habrá tantas delgas como bobinas simples posea el devanado. (Mora, 2008)

### 2.8.3. Motor Brushless

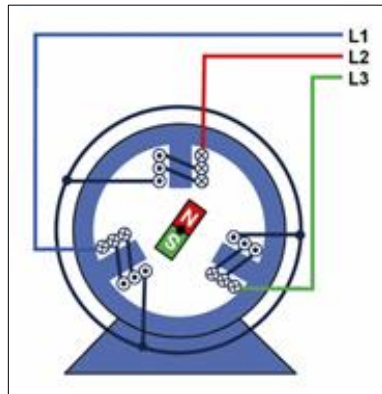
El motor está constituido por imanes permanentes, no poseen escobillas, colector ni delgas.

Figura 8 Diferencia entre motor Brushed y un motor Brushless.



Fuente: (Büch, 2012)

Figura 9 Diagrama Esquemático De Un Motor Brushless



Fuente: (Büch, 2012)

En la realización de este proyecto se ha utilizado los GM8000 motores reductores cilíndricos serie de Pittman que cuentan con engranajes de acero sinterizado, están disponibles en tres longitudes de pila, y cuentan con terminales de salida lateral.

Los motores con caja de reducción Pittman brindan un funcionamiento suave, silencioso y de larga vida. Las armaduras están sesgadas para minimizar rizo de par, incluso a bajas velocidades, y arrollamientos son de resina impregnada de una mayor fiabilidad en aplicaciones de movimiento incremental. (Pittman, 2012)

Las características por las cuales se eligió estos motores es la calidad de sus componentes, como los rodamientos de bolas de alta Fiabilidad y engranajes de alto rendimiento.

Los motores PITTMAN CC poseen una caja de reducción, además son adecuados para robots aficionados. Características de un motor PITTMAN con caja de reducción: tensión, corriente estable, relación de transmisión, tamaño, resolución de encoder, los niveles de ruido, y por supuesto el costo. Son ideales en muchos sentidos para nuestros propósitos, como es las articulaciones de los brazos: Antropomórfico y Scara.

## **2.9. LabVIEW**

### **2.9.1. ¿Qué es LabVIEW?**

LabVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench) es un entorno de desarrollo altamente productivo que los ingenieros y científicos utilizan para la programación gráfica y la integración de hardware sin precedentes, para diseñar y desplegar rápidamente sistemas de medidas y control. En esta plataforma flexible, los ingenieros escalan del diseño a las pruebas y de sistemas pequeños a grandes, al reutilizar IP y perfeccionar sus procesos para alcanzar el rendimiento máximo. (Instruments, 2013).

### **2.9.2. Panel Frontal**

El panel frontal es la interface del usuario con el VI, el lugar donde el usuario puede interactuar con el programa. Normalmente panel frontal es planteado primero, después se diseña el diagrama de bloques para colocar las funciones

que debe realizar el VI con las entradas y salidas del panel frontal además se construye con controles e indicadores, los cuales son las entradas y salidas interactivas del VI, respectivamente. Los controles pueden ser botones de mando, pulsadores, interruptores, diales u otros dispositivos de entrada. Los indicadores son gráficos, leds u otros dispositivos visualizadores. Los controles simulan los dispositivos de entrada y suministran datos al diagrama de bloques del VI. Los indicadores simulan los dispositivos de salida del instrumento y visualizan los datos que el diagrama de bloques adquiere o genera. (Instruments, 2013)

### **2.9.3. Diagrama de bloques**

Los objetos del panel frontal aparecen como terminales en el diagrama de bloques contiene el código fuente gráfico. Adicionalmente, el diagrama de bloques contiene funciones y estructuras incorporadas en las bibliotecas de LabVIEW VI. Los cables conectan cada uno de los nodos en el diagrama de bloques, incluyendo controles e indicadores de terminal, funciones y estructuras.

## **2.10. Matlab**

MATLAB es un sistema interactivo cuyo elemento básico de almacenamiento de información es la matriz, que tiene una característica fundamental no necesita dimensionamiento. Esto le permite resolver varios problemas de computación técnica (especialmente aquellos que tienen formulaciones matriciales y vectoriales) en una fracción de tiempo similar al que se gastaría cuando se escribe un programa en un lenguaje no interactivo como C o FORTRAN. El nombre MATLAB simboliza Matriz Laboratorio o Laboratorio de Matrices. (Esqueda, 2002)

### **2.10.1. Guide**

GUIDE (Interfaz Gráfica de Matlab) es un ambiente de desarrollo que permite crear interfaces gráficas con el usuario, que contengan elementos tales como botones y ventanas de selección, ventanas gráficas, menús, ejes

para graficar, etc. Una vez que se diseña la interfaz gráfica (GUI) que uno desea fijando las características de botones, ventanas, etc. que la conforman, se puede programar dicha interfaz con el editor de archivos.m (Reyes, 2009).

El guide se a utiliza en este proyecto para mostrar en forma de simulación el control teach pad hacia cada uno de los robots.

### **2.10.2. Vrbuild**

El software vrbuild2 (Virtual Reality Build), se encuentra en el tool box de Matlab y se utiliza para crear un entorno de construcción de realidad virtual, se puede crear elementos en tres dimensiones, los cuales pueden ser controlados por el Guide, todos los archivos son registrados y ejecutados por la pantalla script de matlab. El robot antropomórfico y scara fueron simulados en este software como se va a explicar a continuación.

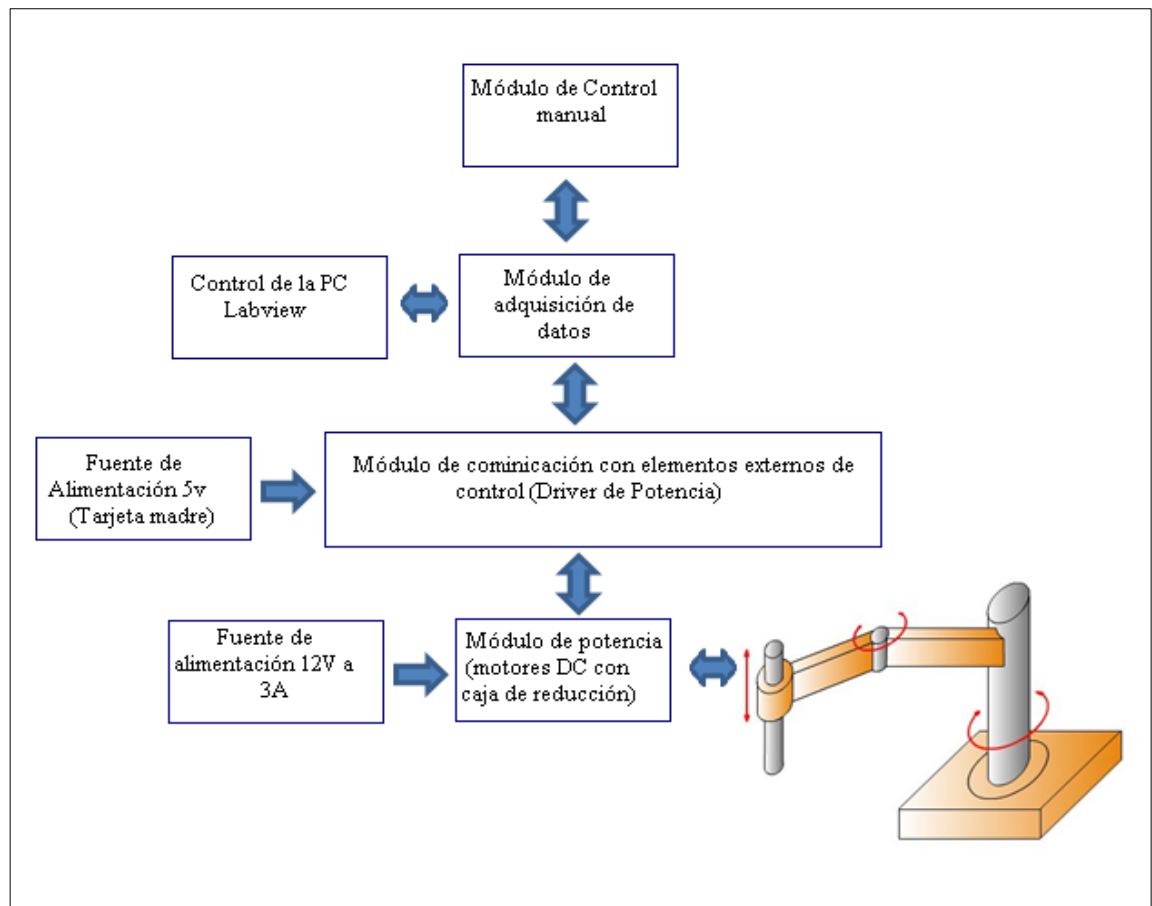
## CAPÍTULO 3

### DESARROLLO DE HARDWARE Y SOFTWARE

#### 3.1. Introducción

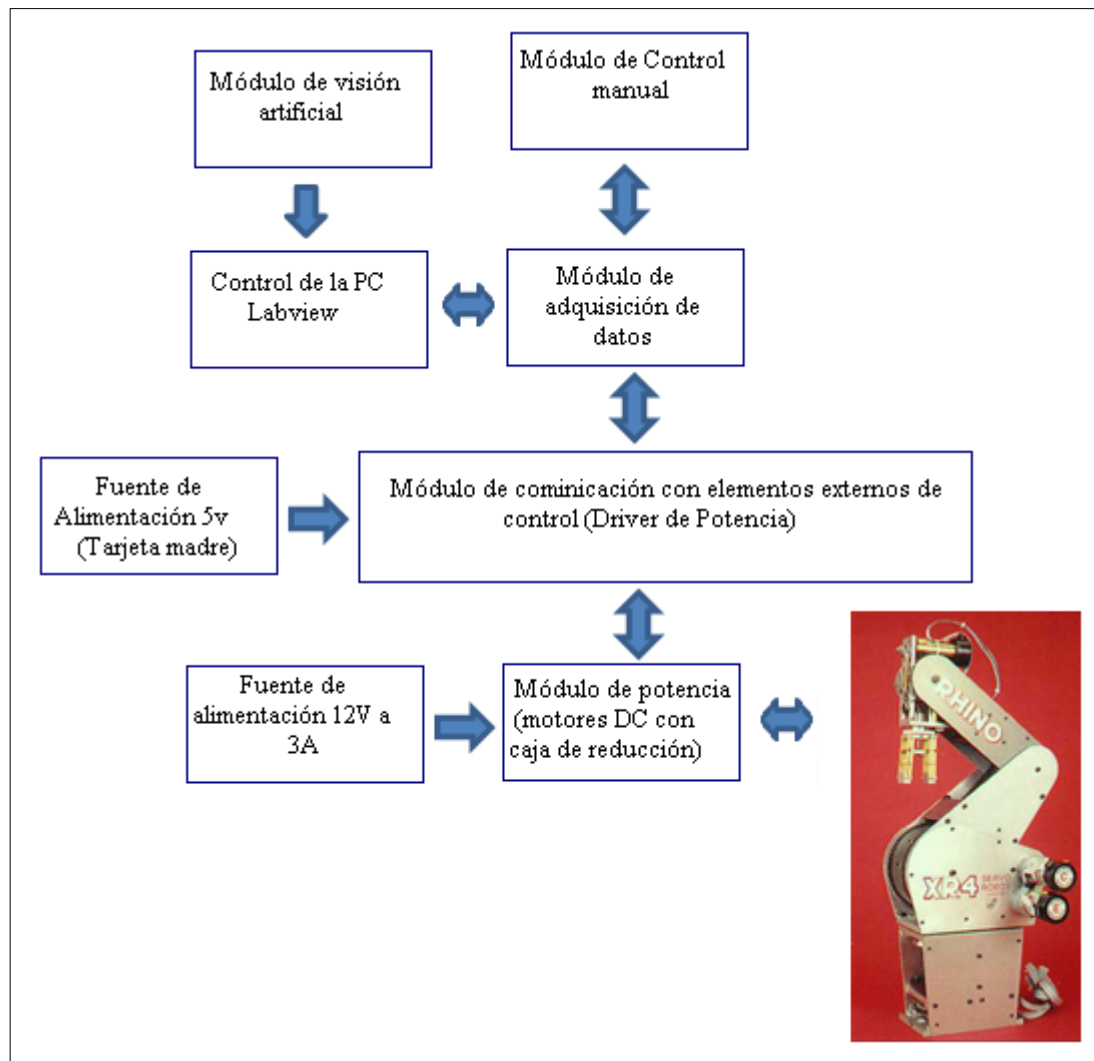
En este capítulo se desarrollará el diseño e implementación del hardware y software, el sistema de control está basado en una tarjeta madre que utiliza un micro controlador ATMEGA AVR 128, ATMEGA8 como interfaz USB, y un teach pad que utiliza el ATMEGA 164P. Además consta de tres tarjetas de potencia las mismas que usan un driver con el CI L298N.

Figura 10 Diagrama de bloques del sistema de control del brazo scara



Elaborado por: Jaime Heredia

Figura 11 Diagrama de bloques del sistema de control del brazo antropomórfico



Elaborado por: Jaime Heredia

Los brazos antropomórfico y scara envían datos lógicos obtenidos por los encoder's que han sido ubicados en cada articulación y el efector, estos datos son receptados por la tarjeta madre que controla la cantidad de grados que se debe mover cada articulación.

El software utilizado para el control es labview y se encarga de controlar cada brazo según los elementos empleados en el panel frontal.

El brazo antropomórfico tiene una cámara web que es controlada por un programa en labview y se encarga de enviar imágenes con un patrón de color específico desarrollando así un programa donde se puede clasificar cada pieza.

Los dos brazos constan de un teach pad que tiene algunas opciones, como un menú con las siguientes opciones, control: Manual y Automático, en la opción manual se tiene la elección de cambiar el sentido horario y anti horario y los grados que se mueve cada articulación además de su propio enter para enviar a cada una de las articulaciones los movimientos para prueba individual.

En el modo automático también se pueden cambiar todos los parámetros y solo se tendría un enter para enviar todos los datos, para ejecutar una secuencia de movimientos.

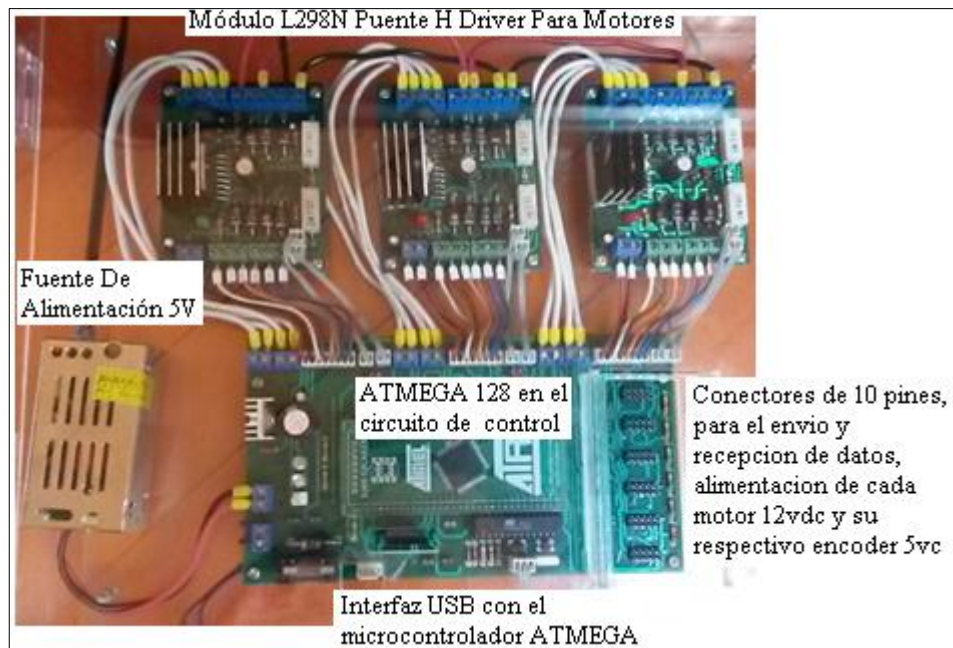
Para enviar el dato de grados que se mueva cada articulación de manera automática y en secuencia, se tiene una prioridad debido a que una específica articulación debe moverse antes que otra para evitar un colapso o choque entre los eslabones. Por ejemplo para regresar a la posición de *reset* en el robot antropomórfico primero se abre el actuador seguido se mueve la muñeca, hombro, codo y finalmente la cintura.

En el software de control, a cada motor se lo ha denominado con dos letras que representan el sentido de giro: horario y anti horario. Cada letra más la cantidad de grados que se desea mover a la articulación son datos enviados al microcontrolador, en el programa se efectúa el algoritmo que compara los datos requeridos con los que recibe del encoder, el cual envía una señal TTL de 1L (uno lógico) y 0L (cero lógico), al llegar al valor deseado por el usuario el motor se detiene y espera una nueva orden.

### **3.2. Hardware**

El hardware es la base física sobre la cual corren las aplicaciones y sistemas informáticos. La adquisición o revalorización del hardware forma parte importante de la planeación y dirección de cualquier sistema computacional. (Rodríguez & Martínez, 2006)

Figura 12 Tarjeta de control



Elaborado por: Jaime Heredia

### 3.2.1. Módulo L298N Puente H Driver Para Motores

El circuito de potencia fue diseñado según algunos parámetros como: la cantidad de corriente (Amperios) y voltaje (Voltios) que utilizaría cada motor para el movimiento de su correspondiente articulación.

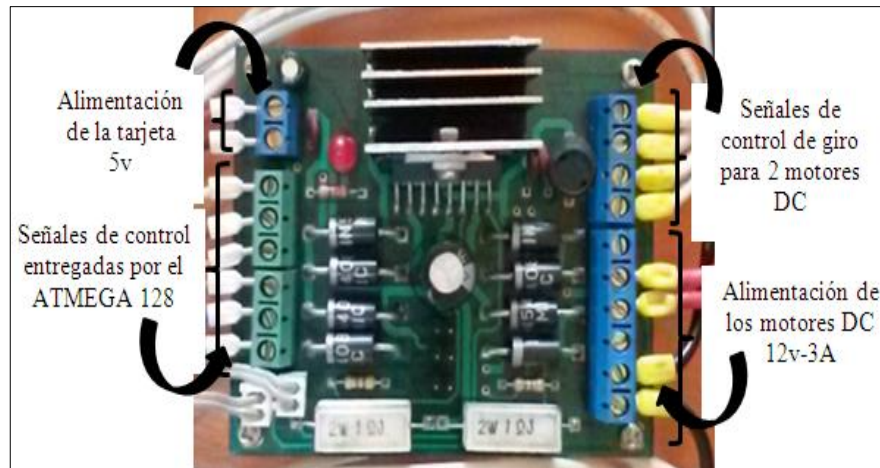
El módulo de control de potencia es basado en el circuito integrado L298N, diseñado para manejar diversos tipos de cargas como motores de corriente continua, motores paso a paso unipolares o bipolares, solenoides, etc.

Consta de 4 amplificadores de potencia que soportan intensidades de corriente de 2 A, con picos de hasta 3 A, controlables independientemente, o como 2 puentes H. En la versión “P” los amplificadores están conectados en paralelo 2 a 2, de modo que se obtiene 2 canales de 3 A, o 1 puente H de 3 A en modo continuo (3,5 A de pico). La capacidad de control de potencia supera los 120 W en total (160 W en intervalos cortos).

Dispone de 2 salidas para medir la corriente de cada puente H, que pueden ser llevadas a un ADC.



Figura 13 Conexión de los pines del driver de potencia



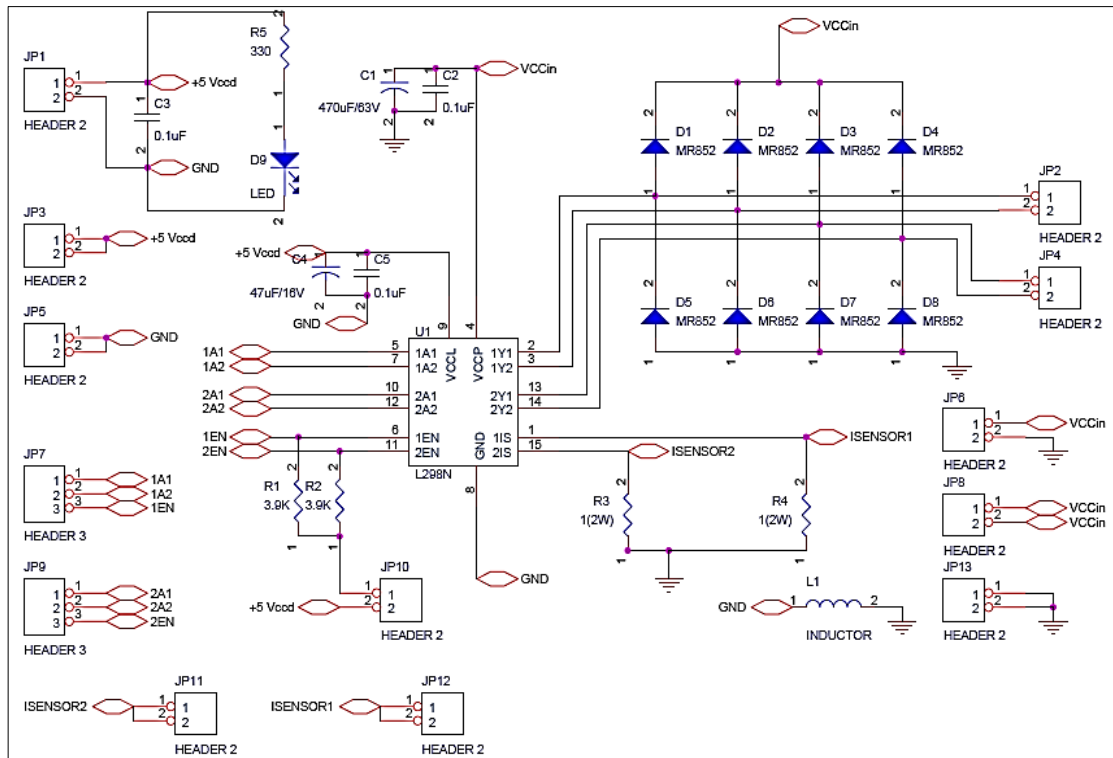
Elaborado por: Jaime Heredia

La separación de las entradas de alimentación de potencia y de lógica, minimiza la transmisión del ruido generado por los motores hacia el sistema de control. 8 diodos schottky de libre circulación suprimen los voltajes inversos generados por cargas inductivas.

### 3.2.1.1. Características Generales Del Módulo De Potencia

- Control de motores, solenoides u otras cargas por encima de 120 W en total (160 W de pico).
- Rango de operación del elemento a controlar de 8 a 46 V.
- Alimentaciones separadas para lógica y potencia necesario para la reducción de transmisión de ruido.
- Conexión a micro controlador mediante 6 señales de control con niveles TTL.
- Tiene un control mediante PWM hasta 25 KHz.
- Posibilidad de medir la corriente de cada motor a través de 2 salidas.
- Alta inmunidad al ruido.
- Protección contra sobre temperatura.

Figura 14 Circuito del módulo de potencia con El C.I. L298N



Elaborado por: Jaime Heredia

### 3.2.1.2. Circuito Integrado L298N

El circuito integrado L298N de alto voltaje y corriente, contiene un puente doble diseñado para aceptar niveles estándar de lógica TTL además soporta cargas inductivas tales como reles, selenoides, motores DC y motores paso a paso.

El circuito integrado L298N posee dos entradas de habilitación para activar ó desactivar el dispositivo de forma independiente de las señales de entradas. Soporta hasta 4A, posee protección de sobre temperatura, baja tensión de saturación.

### 3.2.2. Módulo de control

El módulo de control consta de tres circuitos: la fuente de alimentación, tarjeta con el microcontrolador y tarjeta para la comunicación PC-controladora, que consta de un micro atmega 8 para la conversión de comunicación serial a USB.

El software que controla a cada brazo es Labview, se diseñó programas para que cada pestaña del control tab realice una operación diferente, los datos que se envían a

la tarjeta son de tipo ascii seguido de un enter el cual confirma que una información está siendo enviada, dentro del conjunto de datos que son enviados están los del encoder, la letra que permite el sentido de giro del motor y el enter.

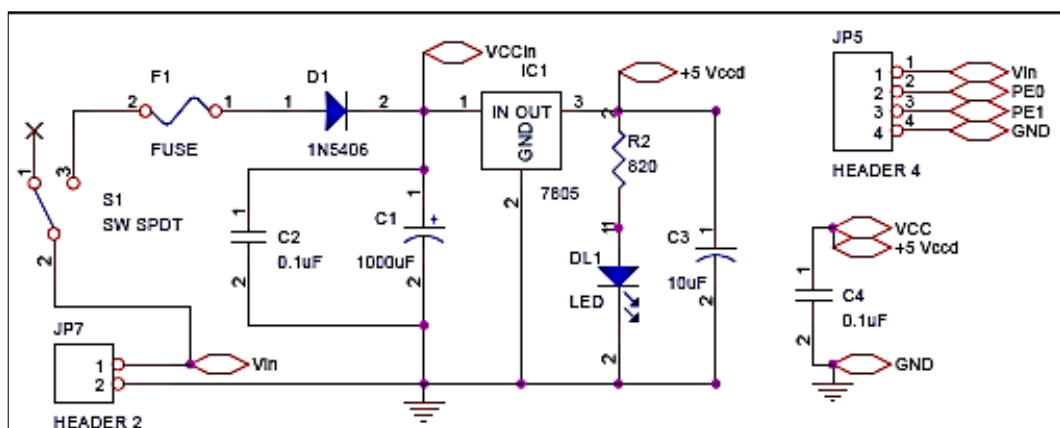
### 3.2.2.1 Fuente de alimentación

La fuente de alimentación consta de un regulador de tensión, la corriente máxima que entrega esta fuente es de 1 Amperio a 5 voltios. Sin embargo, si no se emplea un buen disipador de calor en el regulador de voltaje LM7805, solo se podrá utilizar la mitad de la corriente máxima, debido al excesivo calor generado por este componente.

La salida del transformador pasa a un puente de diodos que se encarga de rectificar la corriente de entrada, y luego un condensador de 1000uF/16V ayuda a eliminar parte del rizado propio de este tipo de rectificador. El 7805, que posee un condensador de 0.1uF a su entrada y otro a su salida, tal como se especifica en su hoja de datos.

Otro condensador de 470uF/16V se encarga de eliminar cualquier pequeño rizado que hubiese quedado, y un LED en serie con una resistencia limitadora se encarga de indicarnos que la fuente se encuentra (o no) encendida.

Figura 15 Diagrama de la fuente de alimentación



Elaborado por: Jaime Heredia

### 3.2.2.2 Distribución de pines del ATMEGA 128 en el circuito de control

El ATMEGA128 tiene un alto rendimiento debido a su bajo consumo de energía, combina 128KB de memoria flash programable, 4KB SRAM, un 4 KB de EEPROM, ocho canales de 10 bits convertidor A / D, y una interfaz de JTAG. El dispositivo trabaja a una frecuencia máxima de 16 MHz y funciona entre 4.5 a 5.5 voltios, según la activación se los respectivos fuses.

Tabla 3: Principales Características del ATMEGA 128

Parámetro	Valor o Cantidad	Unidad/Descripción
Flash	128	Kbytes/ Resistencia: 1.000 ciclos de escritura- borrado.
EEPROM	4	Kbytes/ Resistencia: 100.000 ciclos de escritura-borrado.
SRAM	4	Kbytes
PWM	6	-Resolución programable de 2 a 16 bits
	2	-Resolución 8 bits
Canales Diferenciales	7	
ADC	8	10-bit
Canales Diferenciales	2	Ganancias de (1x,10x,200x)
Memoria Externa Opcional	64	Kbytes
USART	2	Serie programable
In/Out	53	Entradas y salidas programables

Elaborado por: Jaime Heredia

Se utilizó el ATMEGA128 por la cantidad de entradas y salidas que posee, su comunicación vía serial, La distribución de pines se realizó de la siguiente manera:

Tabla 4 Distribución de pines del ATMEGA 128

PIN	Función en el circuito.
1.	PEN: a este pin se lo conecta por medio de una resistencia de 10K a VCC y sirve para habilitar la programación.
2.	PE0: RX recepción desde el puerto serial.
3.	PE1: TX transmisión a el puerto serial.
4.	<p>PE2: Pin utilizado para el <b>ENCODER B</b>, DDRE.0 = 0, el pñrtico E.0 es declarado como entrada, Porte.0 = 1 se habilita la resistencia de pull-up.</p> <p>Sb5 Alias PINE.2 <span style="float: right;">Alias del PINE.2</span></p>
5.	<p>PE3: habilita o deshabilita el enable para el control de giro del motor E, DDRE.3 = 1: Pñrtico PORTE.3 es declarado como SALIDA (OC1C), PORTE.3 = 0: se inicializa con un estado de 0.</p> <p>Pin PORTE.3=0</p>
6.	<p>PE4: habilita o deshabilita el enable para el control de giro del motor F, DDRE.4 = 1: Pñrtico PORTE.4 es declarado como SALIDA (OC1B), PORTE.4 = 0: se inicializa con un estado de 0.</p> <p>Pin PORTE.4=0</p>
7.	<p>PE5: habilita o deshabilita el enable para el control de giro del motor C, DDRE.5 = 1: Pñrtico PORTE.5 es declarado como SALIDA (OC3C), PORTE.5 = 0: se inicializa con un estado de 0.</p> <p>Pin PORTE.5=0</p>
8.	<p>PE6: Pin utilizado para el <b>ENCODER D</b>, DDRE.6 = 0, el pñrtico E.6 es declarado como entrada, Porte.6 = 1 se habilita la resistencia de pull-up.</p> <p>Sa3 Alias PINE.6 <span style="float: right;">Alias del PINE.6</span></p>
9.	<p>PE7: Pin utilizado para el <b>ENCODER C</b>, DDRE.7 = 0, el pñrtico E.7 es declarado como entrada, Porte.7 = 1 se habilita la resistencia de pull-up.</p> <p>Sa4 Alias PINE.7 <span style="float: right;">Alias del PINE.7</span></p>
10.	<p>PB0: Pin utilizado para el <b>ENCODER B</b>, DDRB.0 = 0, el pñrtico B.0 es declarado como entrada, Portb.0 = 1 se habilita la</p>

	resistencia de pull-up.  Swb Alias PINB.0 <span style="float: right;">Alias del PINB.0</span>
11.	PB1: Pin utilizado para el <b>ENCODER C</b> , DDRB.1 = 0, el p�rtico B.1 es declarado como entrada, Portb.1 = 1 <span style="float: right;">se habilita la resistencia de pull-up.</span>  Sb4 Alias PINB.1 <span style="float: right;">Alias del PINB.1</span>
12.	PB2: Pin utilizado para el <b>ENCODER C</b> , DDRB.2 = 0, el p�rtico B.2 es declarado como entrada, Portb.2 = 1 <span style="float: right;">se habilita la resistencia de pull-up.</span>  Swc Alias PINB.2 <span style="float: right;">Alias del PINB.2</span>
13.	PB3: Pin utilizado para el <b>ENCODER D</b> , DDRB.3 = 0, el p�rtico B.3 es declarado como entrada, Portb.3 = 1 <span style="float: right;">se habilita la resistencia de pull-up.</span>  Sb3 Alias PINB.3 <span style="float: right;">Alias del PINB.3</span>
14.	PB4: Pin utilizado para el <b>ENCODER D</b> , DDRB.4 = 0, el p�rtico B.4 es declarado como entrada, Portb.4 = 1 <span style="float: right;">se habilita la resistencia de pull-up.</span>  Swd Alias PINB.4 <span style="float: right;">Alias del PINB.4</span>
15.	PB5: habilita o deshabilita el enable para el control de giro del motor D, DDRB.5 = 1: P�rtico PORTB.5 es declarado como SALIDA (OC3C), PORTB.5 = 0: se inicializa con un estado de 0.  Pin PORTB.5=0
16.	PB6
17.	PB7
18.	PG3
19.	PG4
20.	RESET
21.	VCC voltaje de alimentaci�n de 5v.

22.	GND
23.	XTAL2
24.	XTAL1
25.	<p>PD0: Pin utilizado para el <b>ENCODER F</b>, DDRD.0 = 0, el p�rtico D.0 es declarado como entrada, Portd.0 = 1 se habilita la resistencia de pull-up.</p> <p>Sa1 Alias PIND.0 Alias del PIND.0</p>
26.	<p>PD1: Pin utilizado para el <b>ENCODER E</b>, DDRD.1 = 0, el p�rtico D.1 es declarado como entrada, Portd.1 = 1 se habilita la resistencia de pull-up.</p> <p>Sa2 Alias PIND.1 Alias del PIND.1</p>
27.	PD2
28.	PD3
29.	<p>PD4: Pin utilizado para el <b>ENCODER A</b>, DDRD.4 = 0, el p�rtico D.4 es declarado como entrada, Portd.4 = 1 se habilita la resistencia de pull-up.</p> <p>Swa Alias PIND.4 Alias del PIND.7</p>
30.	<p>PD5: Pin utilizado para el <b>ENCODER A</b>, DDRD.5 = 0, el p�rtico D.5 es declarado como entrada, Portd.5 = 1 se habilita la resistencia de pull-up.</p> <p>Sb6 Alias PIND.5 Alias del PIND.5</p>
31.	<p>PD6: Pin utilizado para el <b>ENCODER A</b>, DDRD.6 = 0, el p�rtico D.6 es declarado como entrada, Portd.6 = 1 se habilita la resistencia de pull-up.</p> <p>Sa6 Alias PIND.6 Alias del PIND.6</p>
32.	<p>PD7: Pin utilizado para el <b>ENCODER B</b>, DDRD.7 = 0, el p�rtico D.7 es declarado como entrada, Portd.7 = 1 se habilita la resistencia de pull-up.</p> <p>Sa5 Alias PIND.7 Alias del PIND.7</p>

33.	PG0(WR)
34.	PG1(RD)
35.	PC0: Ctrlma2 Alias Portc.0, control para la inversión de giro del <b>MOTOR A</b> , DDRC.0 = 1: Pórtico C.0 es declarado como SALIDA, PORTC.0 = 0: El estado inicial del pin C.0=0.
36.	PC1: Ctrlma1 Alias Portc.1, control para la inversión de giro del <b>MOTOR A</b> , DDRC.1 = 1: Pórtico C.1 es declarado como SALIDA, PORTC.1 = 0: El estado inicial del pin C.1=0.
37.	PC2: Ctrlmb2 Alias Portc.2, control para la inversión de giro del <b>MOTOR B</b> , DDRC.2 = 1: Pórtico C.2 es declarado como SALIDA, PORTC.2 = 0: El estado inicial del pin C.2=0.
38.	PC3: Ctrlmb1 Alias Portc.3, control para la inversión de giro del <b>MOTOR B</b> , DDRC.3 = 1: Pórtico C.3 es declarado como SALIDA, PORTC.3 = 0: El estado inicial del pin C.3=0.
39.	PC4: Ctrlmc2 Alias Portc.4, control para la inversión de giro del <b>MOTOR C</b> , DDRC.4 = 1: Pórtico C.4 es declarado como SALIDA, PORTC.4 = 0: El estado inicial del pin C.4=0.
40.	PC5: Ctrlmc1 Alias Portc.5, control para la inversión de giro del <b>MOTOR C</b> , DDRC.5 = 1: Pórtico C.5 es declarado como SALIDA, PORTC.5 = 0: El estado inicial del pin C.5=0.
41.	PC6: Ctrlmd1 Alias Portc.6, control para la inversión de giro del <b>MOTOR D</b> , DDRC.6 = 1: Pórtico C.6 es declarado como SALIDA, PORTC.6 = 0: El estado inicial del pin C.6=0.
42.	PC7: Ctrlmd1 Alias Portc.7, control para la inversión de giro del <b>MOTOR D</b> , DDRC.7 = 1: Pórtico C.7 es declarado como SALIDA, PORTC.7 = 0: El estado inicial del pin C.7=0.
43.	PG2: Led1 Alias Portg.2, enciende el LED cuando el programa empieza.
44.	PA7: Ctrlme2 Alias Porta.7, control para la inversión de giro del <b>MOTOR E</b> , DDRA.7 = 1: Pórtico A.7 es declarado como SALIDA, Porta.7 = 0: El estado inicial del pin A.7=0.
45.	PA6,
46.	PA5: Ctrlmf1 Alias Porta.5, control para la inversión de giro del <b>MOTOR F</b> , DDRA.5 = 1: Pórtico A.5 es declarado como SALIDA, Porta.5 = 0: El estado inicial del pin A.5=0.

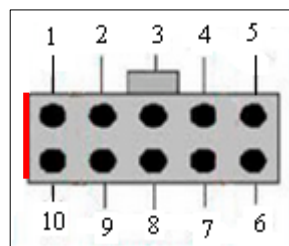


47.	PA4: Ctrlmf1 Alias Porta.4, control para la inversión de giro del <b>MOTOR F</b> , DDRA.4 = 1: Pórtico A.4 es declarado como SALIDA, Porta.4 = 0: El estado inicial del pin A.4=0.
48.	PA3: Pin utilizado para el <b>ENCODER F</b> , DDRA.3 = 0, el pórtico A.3 es declarado como entrada, Porta.3 = 1 se habilita la resistencia de pull-up.  Swf Alias PINA.3 Alias del PINA.3
49.	PA2: Pin utilizado para el <b>ENCODER E</b> , DDRA.2 = 0, el pórtico A.2 es declarado como entrada, Porta.2 = 1 se habilita la resistencia de pull-up.  Swe Alias PINA.2 Alias del PINA.2
50.	PA1: Pin utilizado para el <b>ENCODER F</b> , DDRA.1 = 0, el pórtico A.1 es declarado como entrada, Porta.1 = 1 se habilita la resistencia de pull-up.  Sb1 Alias PINA.1 Alias del PINA.1
51.	PA0: Pin utilizado para el <b>ENCODER E</b> , DDRA.0 = 0, el pórtico A.0 es declarado como entrada, Porta.0 = 1 se habilita la resistencia de pull-up.  Sb2 Alias PINA.0 Alias del PINA.0
52.	VCC voltaje de alimentación 5v.
53.	GND
54.	PF7: Ctrlme1 Alias Portf.7, control para la inversión de giro del <b>MOTOR E</b> , DDRF.7 = 1: Pórtico F.7 es declarado como SALIDA, Portf.7 = 0: El estado inicial del pin F.7=0.
55.	PF6
56.	PF5
57.	PF4
58.	PF3
59.	PF2

60.	PF1
61.	PF0
62.	AREF
63.	GND
64.	AVCC

Elaborado por: Jaime Heredia

Figura 16 Distribución de pines del conector vista desde el encoder



Elaborado por: Jaime Heredia

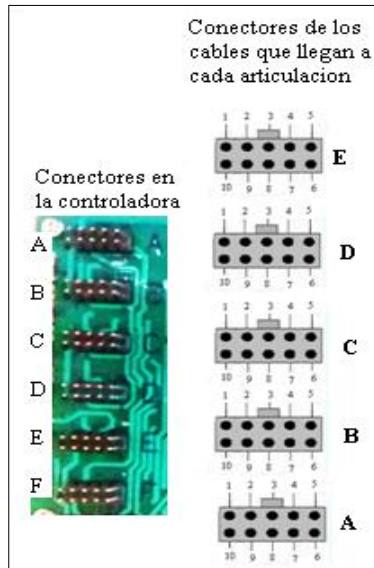
Tabla 5 Función asignada a cada pin

PIN	Función
1	Voltaje Negativo para el motor DC -12v
2	Voltaje Positivo para el motor DC +12v
3	NC
4	Voltaje Positivo para alimentar al encoder +5v
5	Voltaje de referencia para el encoder 0v
6	Señal cuadrada del canal A
7	Señal cuadrada del canal A
8	NC
9	Voltaje Positivo para el motor DC +12v
10	Voltaje Negativo para el motor DC -12v

Elaborado por: Jaime Heredia

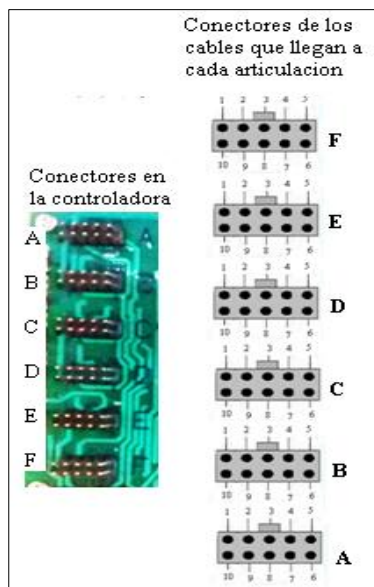
La placa consta de 6 conectores para el similar número de motores que tiene el robot antropomórfico, en el brazo scara se desprecia el conector F ya que existen solo cinco articulaciones, para la conexión de cada conector se debe tomar en cuenta que el conector A es el motor F, así como en el robot scara el conector A es el motor E.

Figura 17 Conexión de cada motor a la tarjeta de control en el brazo Scara según su etiqueta



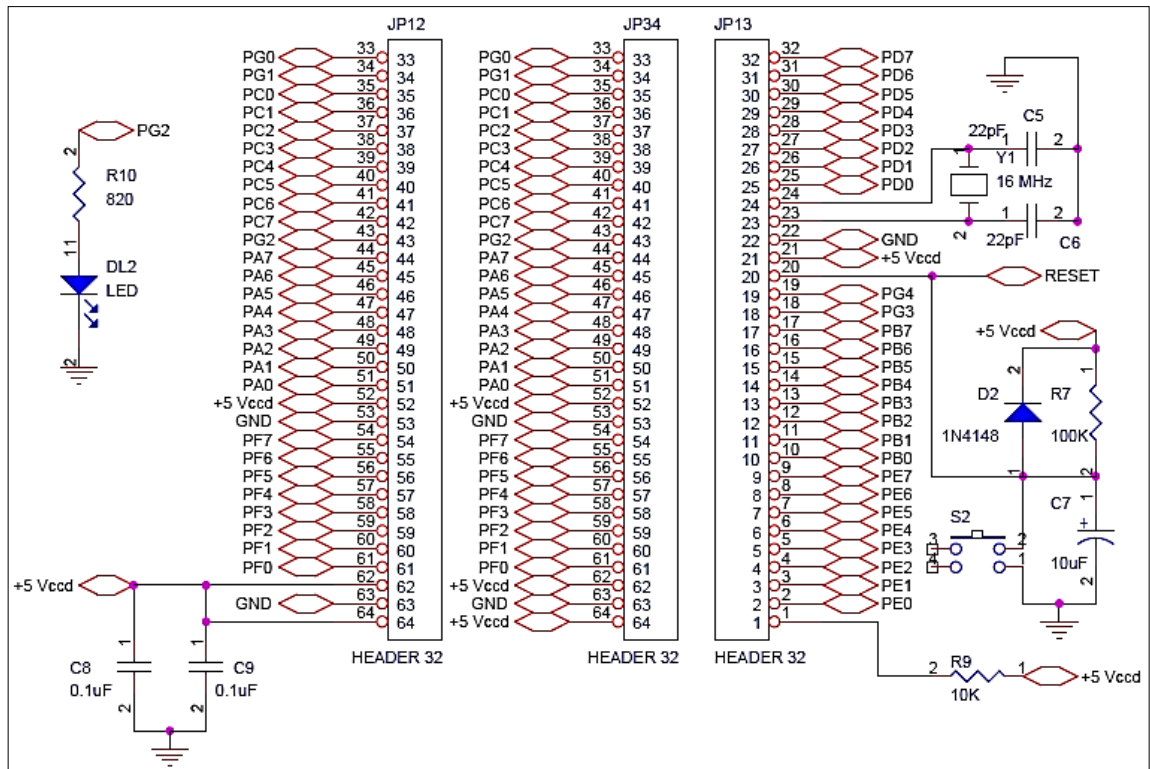
Elaborado por: Jaime Heredia

Figura 18 Conexión de cada motor a la tarjeta de control en el brazo Antropomórfico según su etiqueta



Elaborado por: Jaime Heredia

Figura 19 Conexiones De Pines Del Atmega 128



Elaborado por: Jaime Heredia

### 3.2.2.3 Tarjeta de interfaz USB con el microcontrolador ATMEGA 8

La mayoría de los microcontroladores poseen un puerto de comunicación serial para la transmisión y recepción de datos desde un computador, actualmente cada ordenador posee únicamente puertos USB entonces se requirió de un dispositivo de interfaz. La tarjeta de control que se utiliza en el proyecto envía los datos de forma serial al microcontrolador ATMEGA8, el cual se podría decir que es un convertidor USB-Serial. A través de este microcontrolador puede recibir y enviar datos a un computador de manera serial.

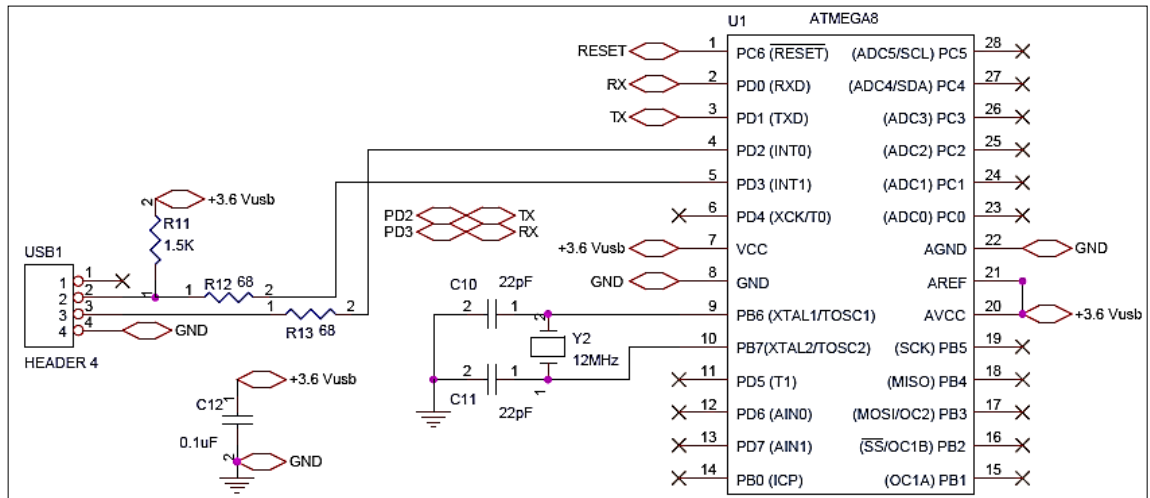
Tabla 6 Características De Microcontrolador ATMEGA8

Parámetro	Valor o Cantidad	Unidad/Descripción
Flash	8	Kbytes/ Resistencia: 1.000 ciclos de escritura- borrado.
EEPROM	512	bytes/ Resistencia: 100.000 ciclos de escritura-borrado.
SRAM	1	Kbytes

PWM	3	
Cristal	16	MHz
ADC	8	6 de 10-bit 2 de 8-bit
USART	1	Serie programable
In/Out	23	Entradas y salidas programables

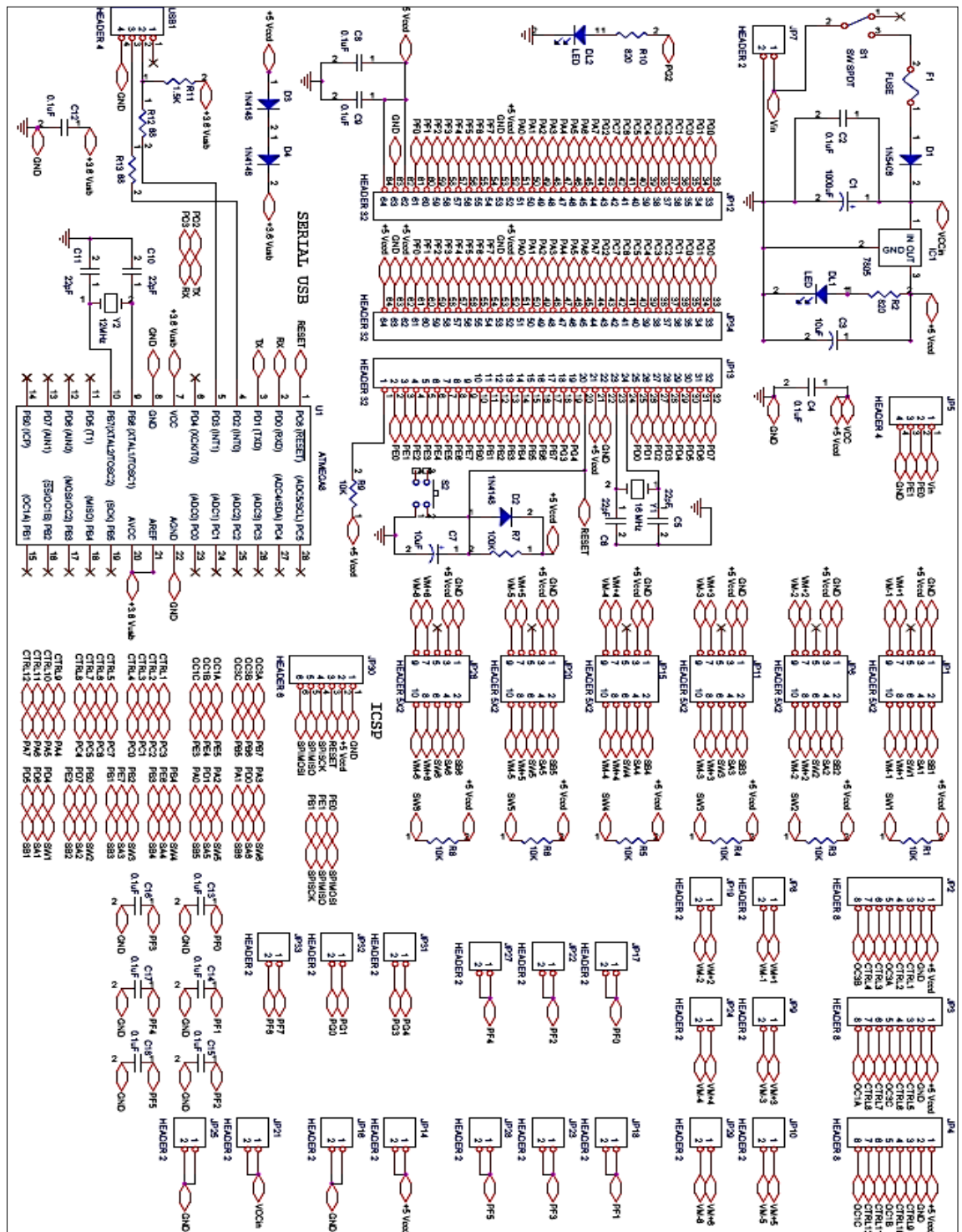
Elaborado por: Jaime Heredia

Figura 20 Interfaz de comunicación entre el microcontrolador ATMEGA128 y la PC



Elaborado por: Jaime Heredia

Figura 21 Circuito del Controlador

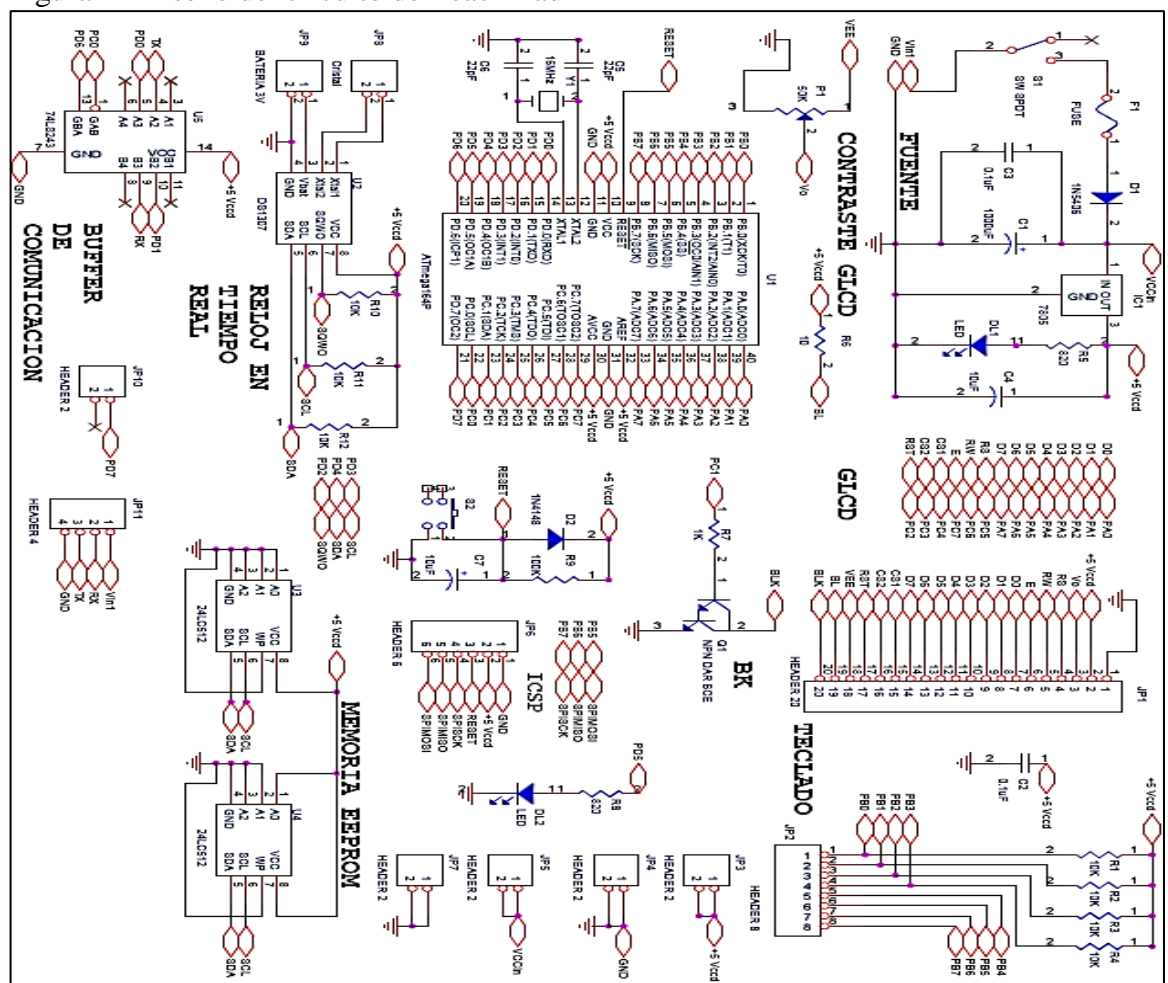


Elaborado por: Jaime Heredia

### 3.2.3 Modulo del Teach Pad

El teach pad es un control que fue diseñado para manipular cada robot de forma manual. Utilizamos un microcontrolador AVR, una fuente de voltaje, un teclado, una pantalla GLCD además de memorias. El funcionamiento del teach pad se basa en un menú donde podemos ubicar cada articulación del robot y hacerla girar según nuestra conveniencia, también tenemos la opción de regresar a la posición inicial pulsando el botón de *reset*. Todas las órdenes enviadas son receptadas por la tarjeta madre la cual ordena el giro de un motor específico.

Figura 22 Diseño del circuito de Teach Pad

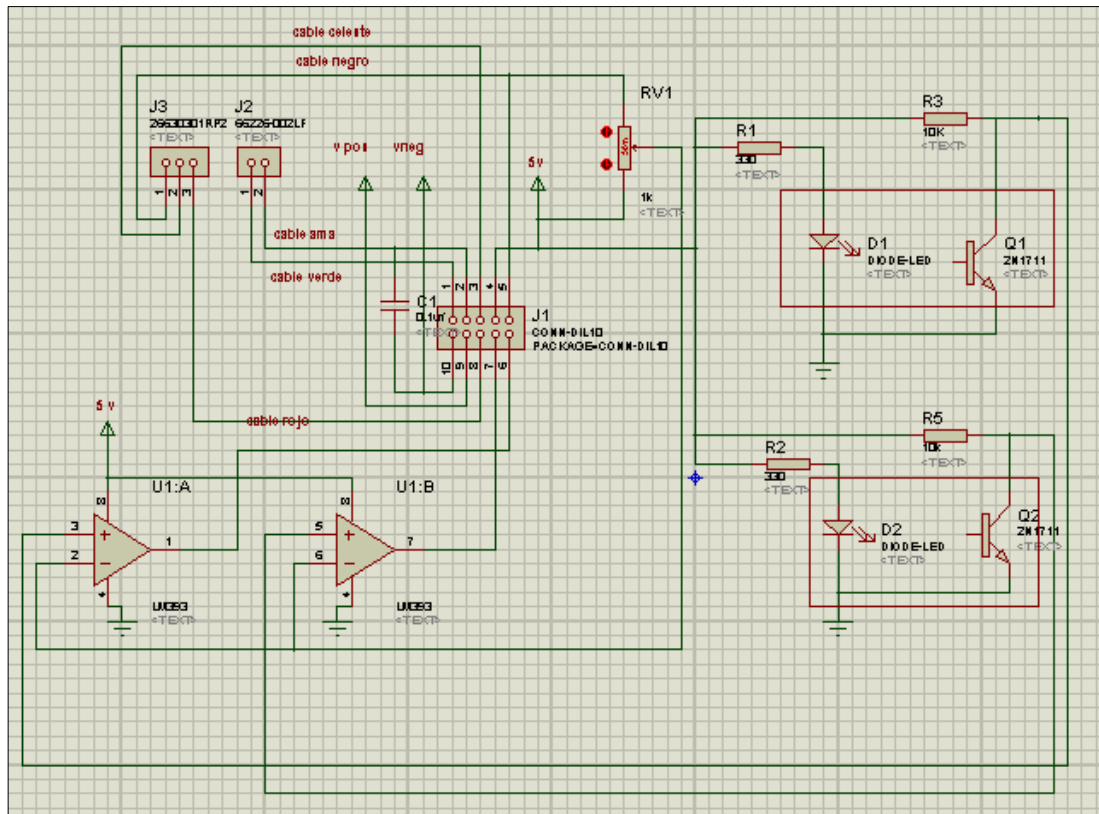


Elaborado por: Jaime Heredia

### 3.2.4 Encoder

El circuito del encoder fue diseñado en el software de Proteus se diseñó según las especificaciones que se dieron anteriormente.

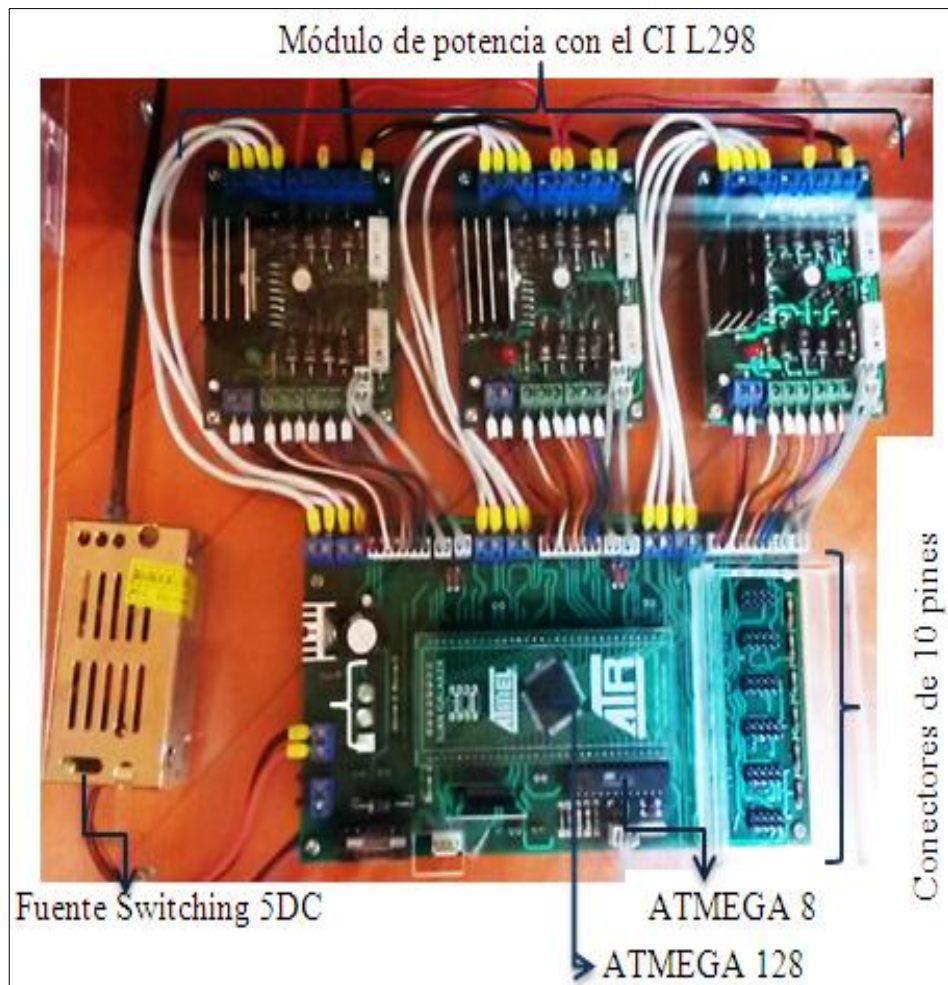
Figura 23 Circuito del Encoder



Elaborado por: Jaime Heredia



Figura 24 Tarjeta de control implementada al robot scara y antropomórfico



Elaborado por: Jaime Heredia

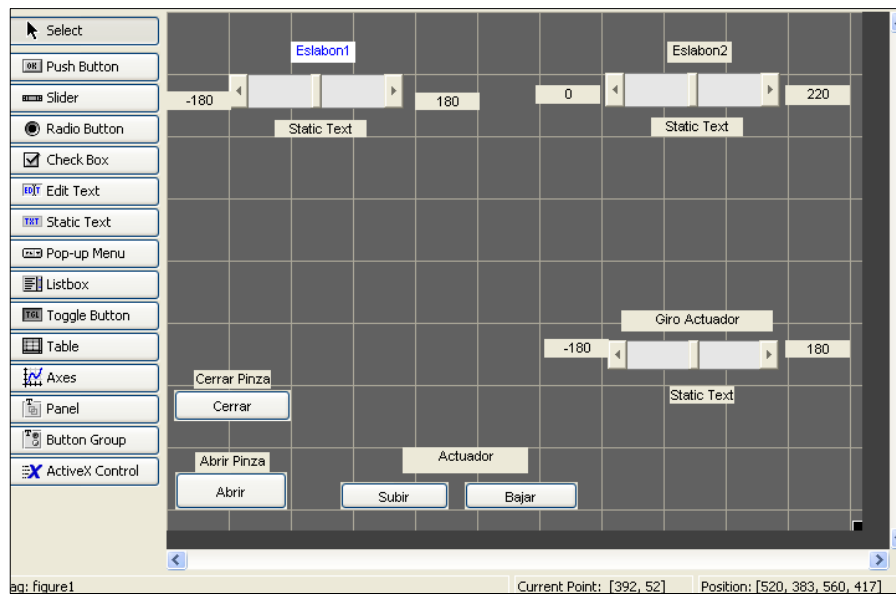
### 3.3. Software

Los software's utilizados para la simulación y el control son: Matlab 2010 y Labview 2012 respectivamente.

#### 3.3.1 Diseño de interfaz gráfica en Guide de Matlab

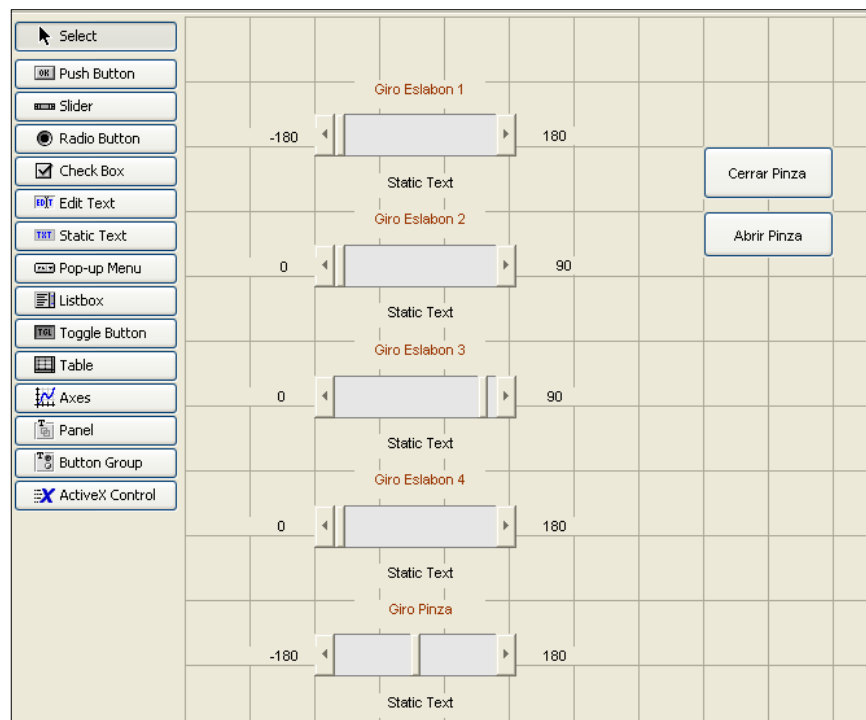
A continuación se va a detallar los elementos que se han utilizado como: strings, label, etc., y como cada uno de estos controlan a los robots diseñados en el software vrbuild.

Figura 25 Diseño en Guide del *teach pad* para el Brazo Scara



Elaborado por: Jaime Heredia

Figura 26 Diseño en Guide del *teach pad* para el Brazo Antropomórfico



Elaborado por: Jaime Heredia

Para la realización de la interfaz gráfica se inició de la siguiente manera:

Al ingresar al icono del Guide que se encuentra al lado derecho del Simulink, se crea un proyecto guide nuevo. Se debe conocer los elementos que guide posee, esto ayuda

a elegir de forma correcta el dispositivo se va a utilizar. En este proyecto se utiliza: slider, static text, push button, chex box.

Tabla 7 Descripción de componentes del Guide

Control	Valor de estilo	Descripción
Check box	'checkbox'	Indica el estado de una opción o atributo
Editable Text	'edit'	Caja para editar texto
Pop-up menu	'popupmenu'	Provee una lista de opciones
List Box	'listbox'	Muestra una lista deslizable
Push Button	'pushbutton'	Invoca un evento inmediatamente
Radio Button	'radio'	Indica una opción que puede ser seleccionada
Toggle Button	'togglebutton'	Solo dos estados, "on" o "off"
Slider	'slider'	Usado para representar un rango de valores
Static Text	'text'	Muestra un string de texto en una caja
Panel button		Agrupar botones como un grupo
Button Group		Permite exclusividad de selección con los radio button

Elaborado por: Jaime Heredia

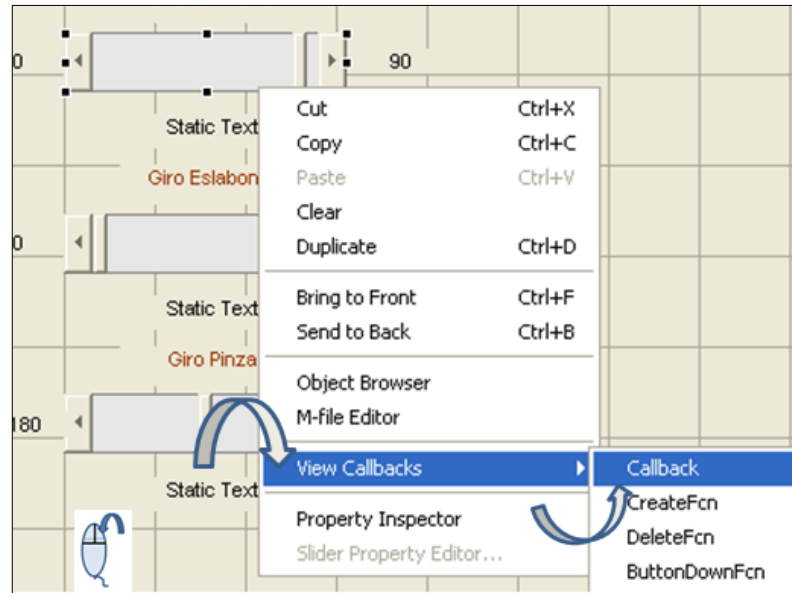
En las propiedades de cada elemento se debe configurar el número de grados que se mueve la articulación a controlar, se debe tomar en cuenta que el nombre que se le ponga a cada elemento es el mismo que se va a declarar en el script de Matlab. Los elementos se encuentran al lado izquierdo de la pantalla de nuevo proyecto, la opción Property Inspector permite ver y personalizar a cada objeto.

Una de las opciones más importantes es el View Callbacks el cual se encuentra al hacer clic derecho en el elemento ubicado en el área de diseño, al ejecutarla abre un archivo .m asociado al diseño y se posiciona en la parte del programa que corresponde a la subrutina que se ejecutará cuando se realice una determinada acción sobre el elemento que estamos editando.

View Callbacks se utiliza para crear una conexión entre la interfaz del usuario con el script de Matlab esto se hace para poder controlar cada articulación. Los valores de las propiedades como: color, valor, posición, string, etc., que se han modificado en los elementos y los valores de las variables transitorias del programa se almacenan en una estructura, los cuales son accedidos mediante un único y mismo identificador

para todos éstos. Tomando el programa listado anteriormente, el identificador se asigna en:

Figura 27 Ubicación de la opción View Callbacks



Elaborado por: Jaime Heredia

`handles.output = hObject`; `handles`, es el identificador a los datos de la aplicación. Esta definición de identificador es salvada con la siguiente instrucción:

1. `guidata(hObject, handles)`; `guidata`, es la sentencia para salvar los datos de la aplicación.

La instrucción `guidata` es la función que guarda las variables y propiedades de los elementos en la estructura de datos de la aplicación, por lo tanto, como regla general, en cada subrutina se debe escribir en la última línea lo siguiente: `guidata(hObject,handles)`;

Esta sentencia garantiza que cualquier cambio o asignación de propiedades o variables quede almacenado. En el momento de llamar a un slider o cualquier otro objeto se crean las siguientes sentencias: `get` y `set`.

La asignación u obtención de valores de los componentes se realiza mediante las sentencias `get` y `set`. Por ejemplo si se requiere que la variable `valor_slider` tenga el valor del Slider se escribe:

```
valor_slider = get(handles.slider1,'Value');
```

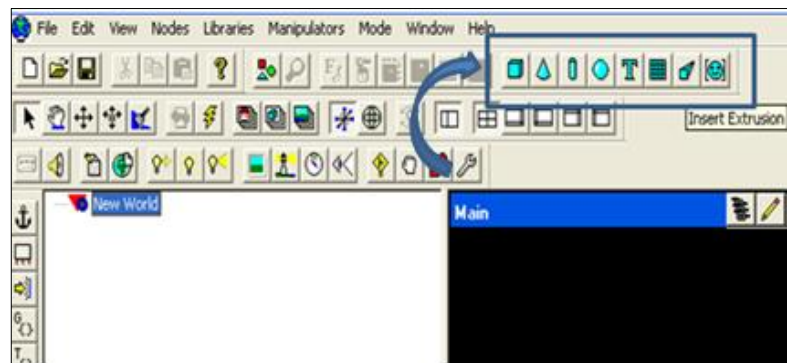
Se debe notar que siempre se obtienen los datos a través de los identificadores handles. Para asignar el valor a la variable valor\_slider al static text etiquetada como text1 se escribe:

```
set(handles.text1,'String', valor_slider);
```

### 3.3.2 Construcción de los brazos Antropomórfico y Scara en 3D utilizando el software vrbuild de Matlab

Se Ingresa al programa vrbuild y se crea un nuevo proyecto. Se debe elegir las formas de los elementos a controlar tomando en cuando algunos aspectos como: la forma, la escala, orientación, color, textura etc.

Figura 28 Formas para realizar la figura

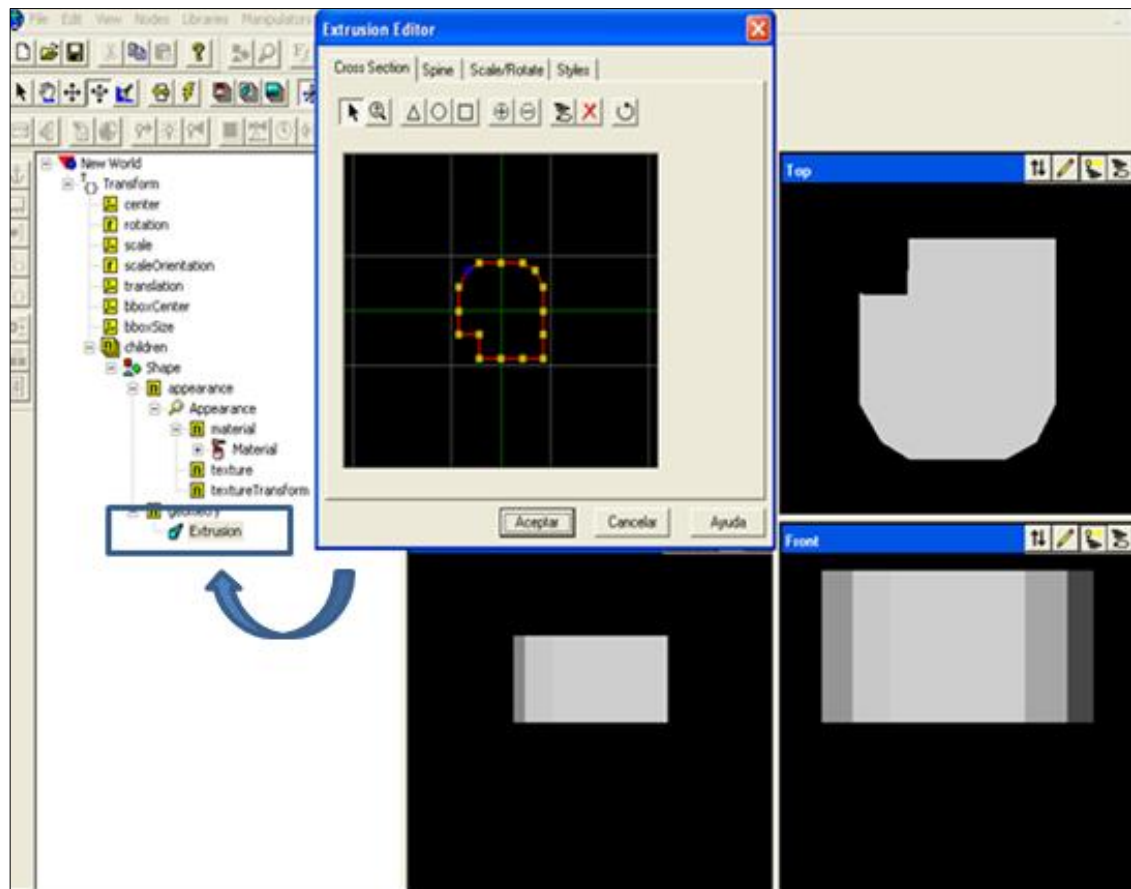


Elaborado por: Jaime Heredia

Los valores para la construcción y ubicación de cada elemento se encuentran en la parte superior derecha, cada figura para su movimiento debe ser independiente a su vez si se desea mover un bloque de formas una deberá estar dentro del lazo de la otra y ser denominada.

Existen formas geométricas básicas que se puede elegir además de una forma llamada Extrusión la cual se puede asemejar una forma inusual que se necesite, a esta figura puede ser modificada según varias opciones como: cortar, girar, rotar y escalar. La figura se encuentra rodeada de puntos que pueden aumentar o ser disminuidos según la necesidad, estos sirven para modificar la figura desde cualquier ángulo.

Figura 29 Extrusión modificada



Elaborado por: Jaime Heredia

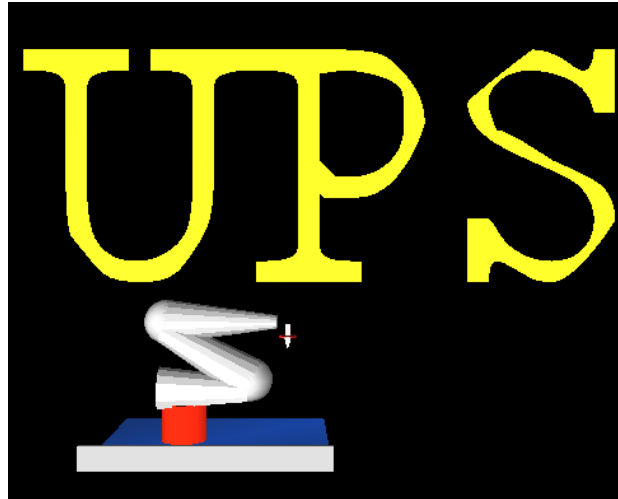
Si se desea que una forma cualquiera se mueva dependiente de otra, se debe crear un grupo incluyendo a las formas que necesitamos a la principal, esto se hace ubicándose en la posición children de la figura principal y pulsando cualquier otra figura en la barra de herramientas. En el ejemplo todas las formas denominadas Grados pertenecen al principal que es Brazo\_3Grados.

La finalidad de hacer un solo bloque es que al llamar a cada uno por el scrip de Matlab podemos tener un control individual pero cuando necesitemos a todos solo basta llamar a Brazo\_3Grados.

Se debe ubicar algunos puntos de luz para poder visualizar de mejor manera la figura cuando ya está culminada, al igual que a las formas, cada punto de luz puede ser ubicado en cualquier lugar del entorno en tres dimensiones, se debe tomar en cuenta que tienen varias opciones como: intensidad, atenuación, color, localización, encender o apagar.

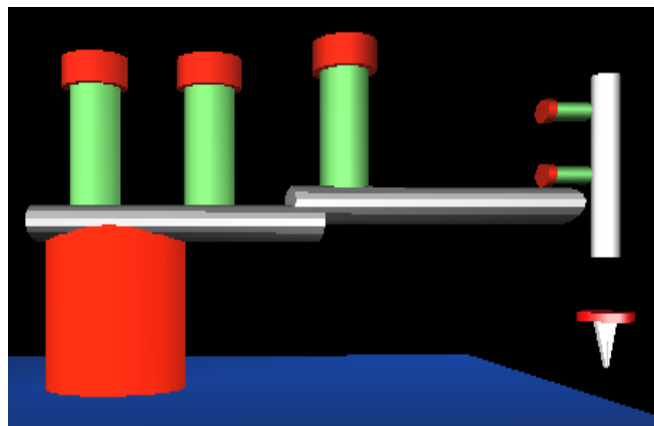
Si se desea ver por diferentes ángulos a la figura ya sea desde vrbild o Matlab se debe añadir al programa Puntos de vista, los cuales permiten la visualización de la figura desde varias posiciones como: frontal, lateral, atrás, abajo o superior.

Figura 30 Diseño final del Brazo Antropomórfico



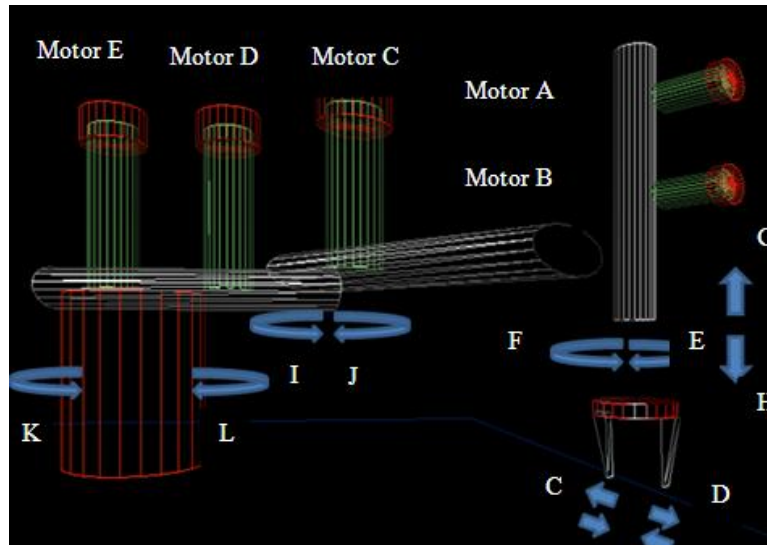
Elaborado por: Jaime Heredia

Figura 31 Diseño final del Brazo Scara



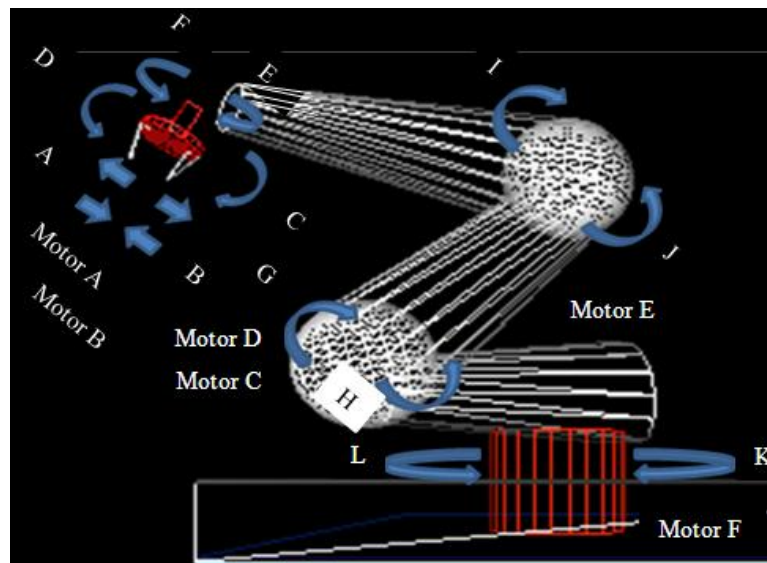
Elaborado por: Jaime Heredia

Figura 32 Movimiento de cada articulación y ubicación de etiquetas en el brazo Scara



Elaborado por: Jaime Heredia

Figura 33 Movimiento de cada articulación y ubicación de etiquetas en el brazo Antropomórfico



Elaborado por: Jaime Heredia

### 3.3.3 Desarrollo del programa en el script de Matlab

La declaración de variables pertenecientes al guide necesarias para crear la interfaz con Matlab se crea al correr el programa. Si se desea incorporar una imagen en el fondo de la pantalla del guide se declara el siguiente comando:



Figura 34 Comando para insertar una imagen

```
[x,map]=imread('scara.jpg','jpg');
```

Elaborado por: Jaime Heredia

Hay que tomar en cuenta que el formato de la foto debe ser jpg . Para abrir el archivo de vrbuild se hace ubica el nombre exacto del archivo .vrl.

Figura 35 Comando para incorporar al archivo de vrbuild

```
world=vrworld('scarbot.vrl','new');  
open(world);  
close(fig);  
fig=vrfigure(world);%cargando Mundo VRML
```

Elaborado por: Jaime Heredia

En el programa de simulación en 3D se mencionó tomar en cuenta cuantas vistas se hicieron de la misma forma estas deben ser llamadas en el script con el mismo nombre que fueron creadas en el archivo vrbuild.

Figura 36 Comando para la incorporación de puntos de vista

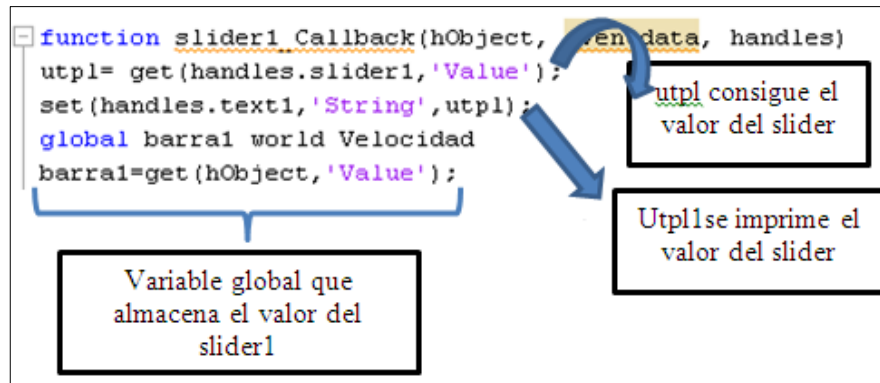
```
set(fig,'Viewpoint','Superior');  
set(fig,'Viewpoint','Atras');  
set(fig,'Viewpoint','Frente');
```

Elaborado por: Jaime Heredia

Se deben declarar variables globales para el control de cada una de las articulaciones los nombres de estas pueden variar en el archivo de este proyecto se encuentran enumeradas y con los siguientes nombres barra1=0; Edit\_cilindro1=0; Antiguo1=0. Ordinariamente, cada función Matlab tiene sus propias variables locales, que están separados de otras funciones. Si la variable global no existe la primera vez que se emite la sentencia global, la variable es inicializada con un valor de 0, si una variable con el mismo nombre que la variable global ya existe en el espacio de trabajo actual, Matlab emite una advertencia y cambia el valor de esa variable para que no coincidan entre sí.

Cuando cada objeto se encuentra en el script después de su respectivo Callbacks se debe conseguir el valor de cada slider que ha sido modificado y ubicarlo en un string, a su vez en valor conseguido es guardado en una variable global llamada barra 1.

Figura 37 Comandos del Callbacks



Elaborado por: Jaime Heredia

Cada slider controla una articulación del robot al ejecutar cada sentencia se debe cargar el archivo vrmf perteneciente al vrbuild, Matlab utiliza el comando vrnode que identifica un nodo específico en el mundo virtual. Si se aplica el método vrnode a un nodo que no existe, el método crea un nodo falso, Si se aplica el método vrnode a un nodo o articulación existente el elemento al que pertenece el nombre en mundo virtual cambia de posición u orientación según sea modificado por el usuario en la interfaz gráfica.

Figura 38 Comando para cargar el archivo del mundo virtual

```

Edit_Eslabon2=vrnode(world, 'grado2');
    
```

Elaborado por: Jaime Heredia

La imagen del vrbuild se encuentra ya cargada además ya se tiene el control de cada elemento, el siguiente comando muestra cómo mover la articulación en forma de rotación, tomando en cuenta en que eje se desea que gire la figura para solucionar este problema ubicamos en la sentencia el nombre de la variable global que contiene la figura seguido del comando .rotation dentro de los corchetes se ubica el vector donde se modifica según el movimiento que se desee.

Figura 39 Comando para configurar el sentido de giro de un elemento en el mundo virtual

```
Variable que controla el rango de movimiento  
  
Edit_cilindro1.rotation=[0 1 0 barra1];  
  
1 0 0 mueve la figura en el eje X.  
0 1 0 mueve la figura en el eje Y.  
0 0 1 mueve la figura en el eje Z.
```

Elaborado por: Jaime Heredia

El proceso es similar para cada articulación del robot tomando en cuenta que en cada una cambia el slider, elemento del vrbuild, y las variables globales.

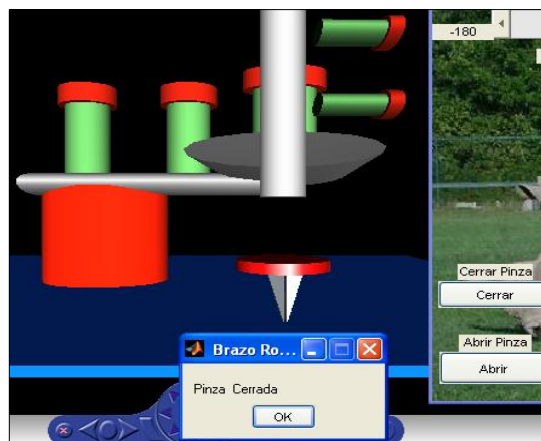
Para simular el movimiento de apertura y cierre del actuador final se propuso que al pulsar un botón se desarrolle un algoritmo donde se encuentra un lazo de programación. Cuando el botón es pulsado para cerrar a la pinza y esta se encuentra cerrada se necesitó del siguiente comando:

Figura 40 Comando para notificar un problema

```
msgbox('Pinza Cerrada',' Brazo Robotico ');
```

Elaborado por: Jaime Heredia

Figura 41 Mensaje de aviso



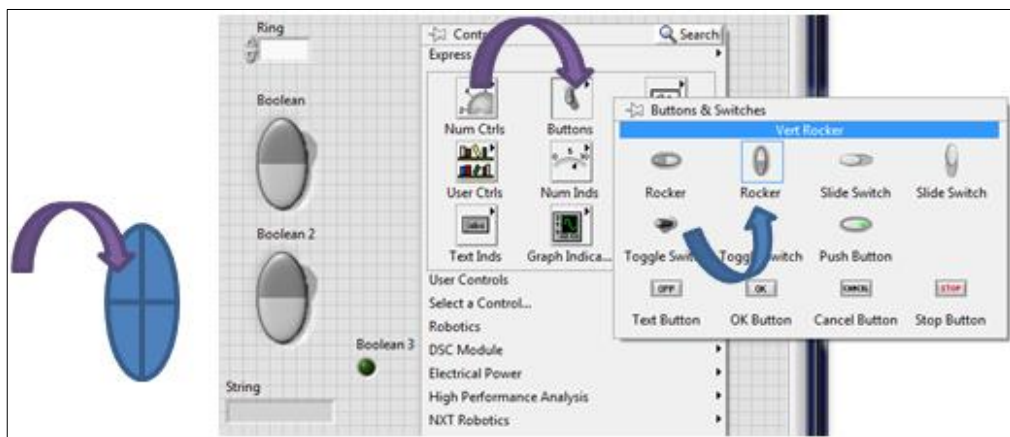
Elaborado por: Jaime Heredia

### 3.3.4 Desarrollo del programa para el control de los brazos antropomórfico y Scara en Labview

El proyecto se basa en tres etapas denominadas control manual, control automático y visión artificial, que se encuentran en una herramienta llamada control Tab, cada una de ellas contiene una programación para cumplir las condiciones que el usuario establezca en el panel frontal.

Se debe iniciar ingresando al software Labview y se crea un nuevo proyecto, se debe tener clara la idea de cómo va a ser diseñado el panel frontal, para el control manual de los robots se va a colocar un ring text para el control de los grados a mover, dos botones en acción mecánica como pulsador para el control del sentido de giro, una palabra (string) para mostrar en letras la palabra horario o antihorario y un led que se enciende cuando el proceso comienza, para hacer en control primero ubicamos en el panel frontal todos los elementos mencionados anteriormente.

Figura 42 Ubicación de Buttons and Switches



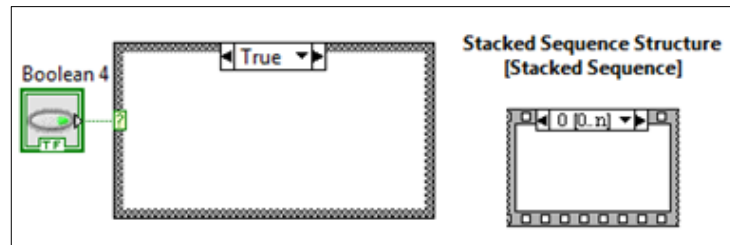
Elaborado por: Jaime Heredia

Como el programa se encuentra se parado por tres categorías se ha elegido un control tab el cual ayudará a la selección del proceso que se necesite, cuando los elementos se encuentran en el panel frontal se puede hacer la programación gráfica en el diagrama de bloques, utilizando algunos elementos como: estructuras, elementos de comparación, lazos de control, comunicación serial, temporizadores, etc.

Las estructuras empleadas fueron el case structure utilizado para desarrollar un programa según las condiciones de verdad o falso, while loop se utilizó para el

control total del programa haciendo que este funcione continuamente además consta de un paro de emergencia y el flat sequence se utiliza para el desarrollo de procesos secuenciales.

Figura 43 Diagramas del case structure y sequence structure



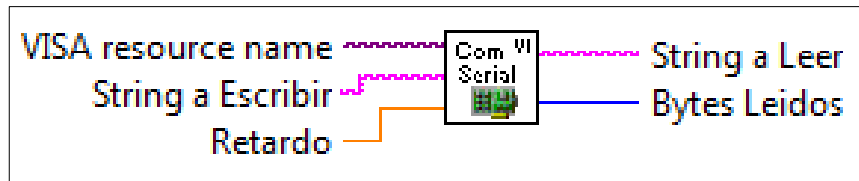
Elaborado por: Jaime Heredia

En el proceso que realiza labview para el control de los brazos robóticos se utilizan temporizadores que entregan un retardo en el desarrollo del proceso el cual evita el colapso en el movimiento de cada brazo.

Para ubicar a cada articulación en una posición específica se necesitó de un dato que controle la posición, este dato es entregado por el encoder de cada motor, el valor entregado depende del número de vueltas que gire cada motor y de la cantidad de zonas negras y blancas que posee cada encoder. Se creó un programa donde el número de vueltas que entrega cada encoder sea proporcional a los ángulos que el usuario desee que se mueva cada articulación, por ejemplo en el motor A su encoder para un movimiento de 90 grados entregó un dato de 1000, entonces para el movimiento de 45 grados el dato debería ser 500, pero hay factores como el torque y la gravedad que hizo que estos números varíen de tal forma que se utilizó una serie de comparadores para escoger el dato ideal al momento de mover una cantidad “x” de grados.

La comunicación entre labview y el microcontrolador es vía serial siendo enviados los datos por medio de un visa de comunicación serial, él envió de datos son de tipo ASCII (American Standard Code for Information Interchange - Código Estándar Estadounidense para el Intercambio de Información), los datos a enviar son la letra que representa el sentido de giro del motor, el número que entrega el algoritmo al desarrollar el programa que representa la cantidad de grados que debe moverse y la tecla enter que se utiliza para no tener confusiones al enviar datos.

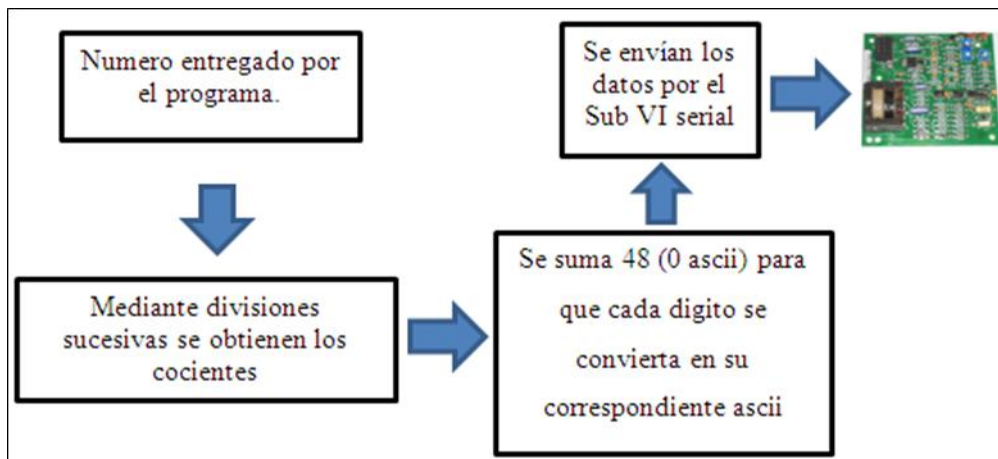
Figura 44 Sub VI para comunicación serial



Elaborado por: Jaime Heredia

El número que entrega el programa es de tipo entero, pero el sub vi de comunicación serial debe enviar el número en código ascii entonces se propuso la idea de dividir el número de cuatro cifras para 1000 y el residuo de la división para 100 y el último para 10.

Figura 45 Diagrama de bloques del proceso de envío de los dígitos a la tarjeta



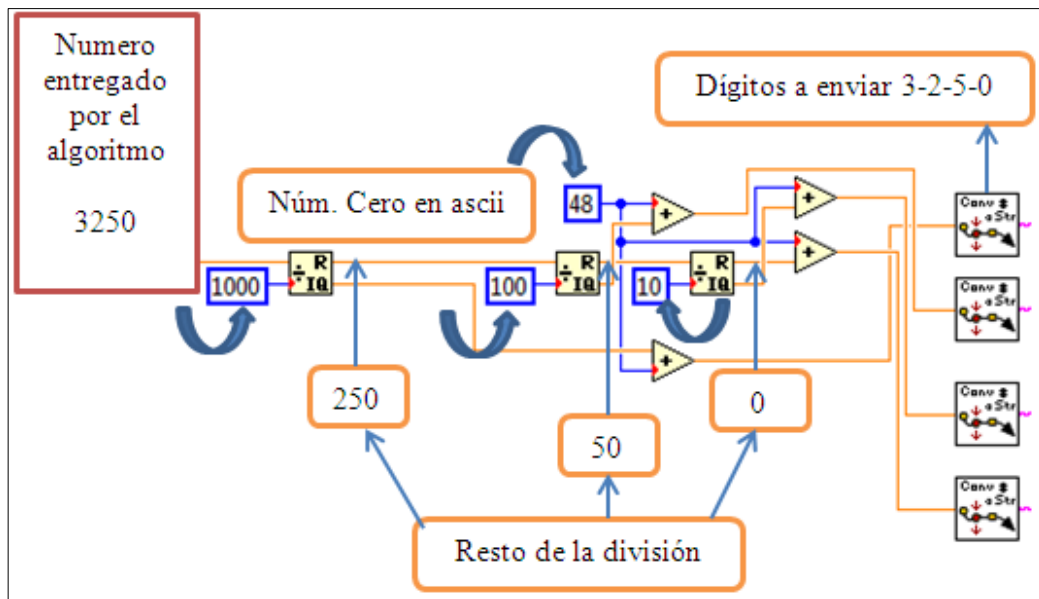
Elaborado por: Jaime Heredia

Tabla 8 Resultado de cocientes

Dividendo	Divisor	Resto	Cociente
3250	1000	250	3
250	100	50	2
50	10	0	5
0	0	0	0

Elaborado por: Jaime Heredia

Figura 46 Diagrama del proceso de envío de datos en labview



Elaborado por: Jaime Heredia

Los cocientes que se obtienen son 3, 2, 5, 0 ahora estos números son dígitos independientes para convertirlos en su correspondiente ascii se sumó el número 48 correspondiente a 0 ascii y se obtuvo los siguientes datos.

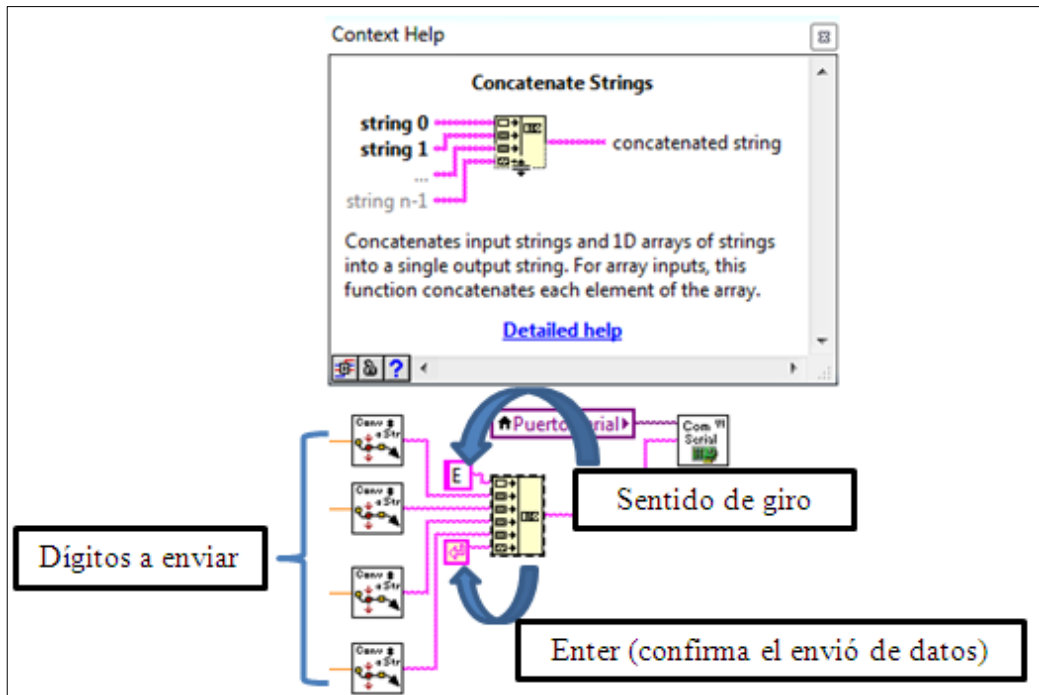
Tabla 9 Tabla de dígitos a enviar

Digito individual	Suma del 0 ascii	Número a enviar (ascii)
3	48	51
2	48	50
5	48	53
0	48	48

Elaborado por: Jaime Heredia

Los datos que se envían añadiendo el enter y la letra de sentido de giro son reunidos por un concatenate strings el cual integra todos los elementos a enviar y lo hace en un solo bloque.

Figura 47 Concatenación de caracteres para el envío



Elaborado por: Jaime Heredia

Tabla 10 Designación de letras para el control de sentido de giro del brazo Antropomórfico

Letra	Sentido	Motor	Articulación a mover
A	Horario	A	Abre el efector
B	Anti Horario	A	Cierra el efector
C	Horario	B	Gira la muñeca
D	Anti Horario	B	Gira la muñeca
E	Horario	C	Mueve la muñeca
F	Anti Horario	C	Mueve la muñeca
G	Horario	D	Giro hombro
H	Anti Horario	D	Giro hombro
I	Anti Horario	E	Gira el codo
J	Horario	E	Gira el codo
K	Horario	F	Gira base
L	Anti horario	F	Gira base

Elaborado por: Jaime Heredia



Tabla 11 Designación de letras para el control de sentido de giro del brazo Scara

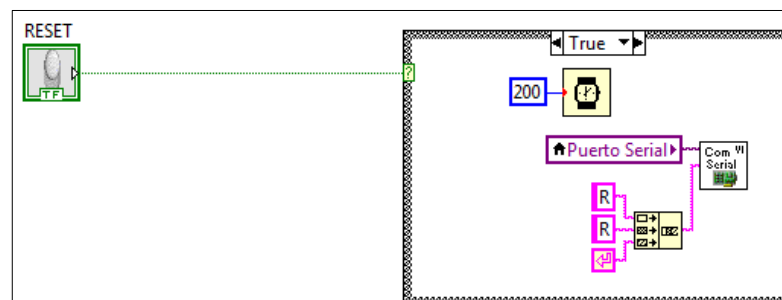
Letra	Sentido	Motor	Articulación a mover
C	Horario	A	Abre actuador
D	Anti Horario	A	Abre actuador
E	Horario	B	Giro muñeca
F	Anti Horario	B	Giro muñeca
G	Horario	C	Sube
H	Anti horario	C	Baja
I	Anti horario	D	Codo
J	horario	D	Codo
K	Anti Horario	E	Hombro
L	horario	E	Hombro

Elaborado por: Jaime Heredia

Finalmente en el control manual se diseñó programas similares para cada articulación, las diferencias son el cambio de letra en el sentido de giro y de factor de multiplicación para mover al efector a una posición deseada.

Cada brazo debe comenzar un proceso desde una posición inicial en el microcontrolador se ha dado prioridad a unas articulaciones para llegar a esta posición, cada robot consta de finales de carrera ubicados en la base de cada articulación al momento de ser pulsados envían una señal al microcontrolador quien detiene esta articulación y envía a mover al siguiente motor y así sucesivamente. En el software de programación existe la opción de reset que el usuario puede elegir para llegar a esta posición.

Figura 48 Posición de reset desde Labview

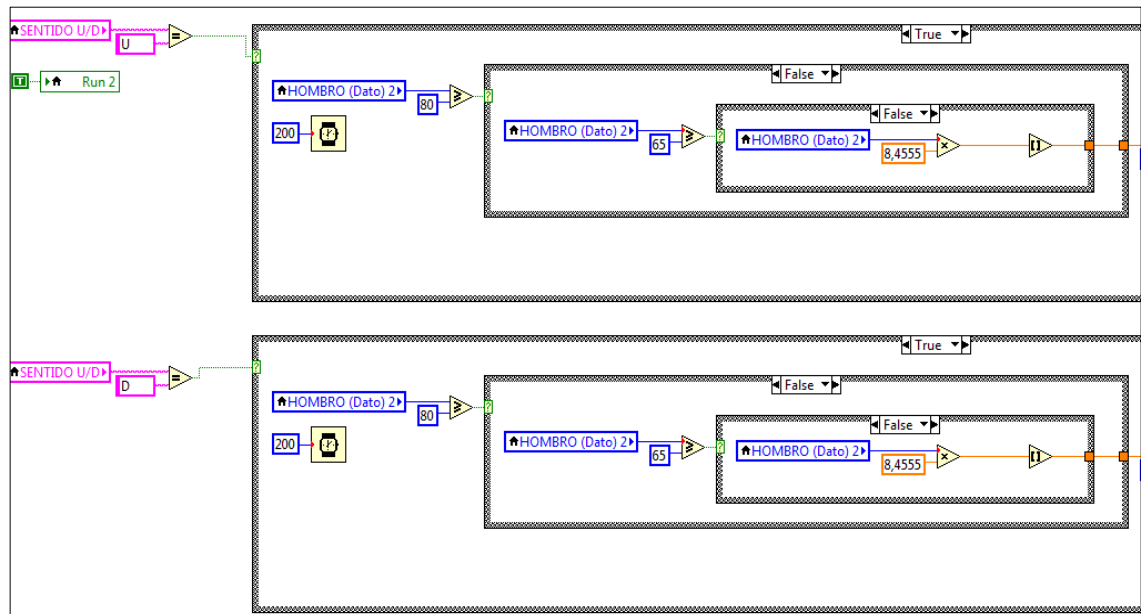


Elaborado por: Jaime Heredia

Para el control del efector se envía un número 80 el cual es constante, esta acción se realiza cuando el botón de control es activado y entrega una acción de verdad.

Para cada articulación se utilizó constantes de multiplicación diferentes para su movimiento, estas dependen de su ubicación en el robot.

Figura 49 Constantes según el rango de giro en la articulación Hombro



Elaborado por: Jaime Heredia

### 3.3.2.1 Visión artificial en Labview

Una de las utilidades que tiene un brazo robótico es la de desechar o clasificar piezas de un color determinado, en la opción banda transportadora que se encuentra en el control tab del robot antropomórfico se incorporó una cámara para buscar un patrón específico de algún color en un área específica y de esta manera desechar o clasificar la pieza.

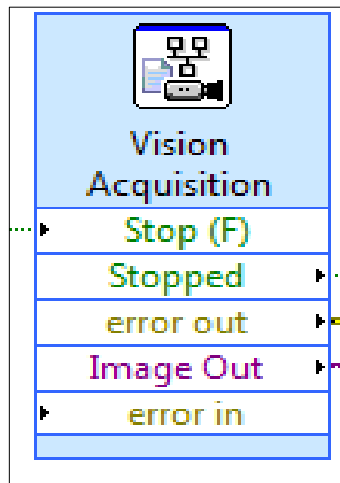
Se utilizó la herramienta llamada visión Acquisition icono que se encuentra en el diagrama de bloques, módulos de visión y movimiento. El objetivo de esta herramienta es tomar una fotografía del área por donde la pieza pasa

El Módulo NI Vision Development es una extensa biblioteca con cientos de algoritmos de procesamiento de imágenes y funciones de visión artificial para

mejorar imágenes, verificar presencia, ubicar características, identificar objetos y medir partes.

A continuación se detallan los pasos para la configuración de la cámara y patrón de color.

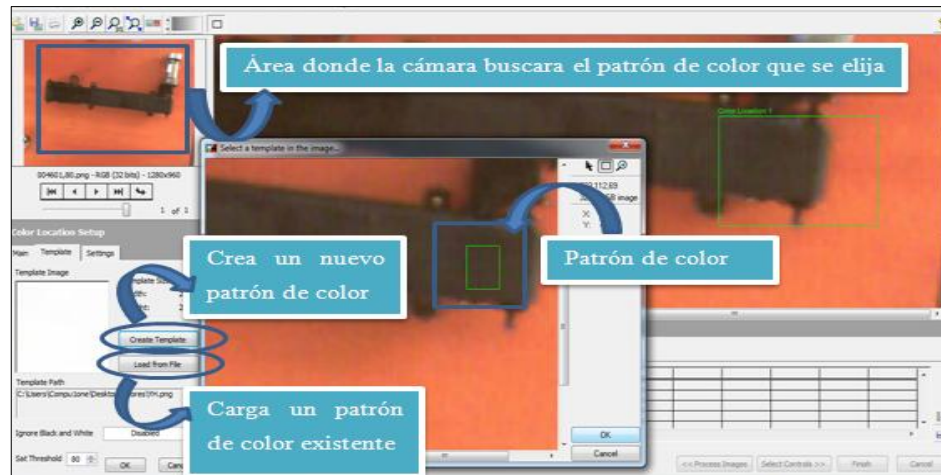
Figura 50 Modulo de Visión Acquisition



Elaborado por: Jaime Heredia

En primer lugar se debe instalar el driver de la cámara que se vaya a utilizar, se elige la cámara que se está utilizando seleccionándola en el panel de la izquierda. Para verificar que la cámara está funcionando se da clic en la flecha que se encuentra en la parte superior de la imagen, en el siguiente paso se localiza el área en donde el patrón de color será extraído para cumplir las condiciones que se especifica en la programación, después se elige el área donde debe encontrarse la pieza para su clasificación, en este caso se eligió el área del extremo derecho de la banda transportadora por donde se supone va a pasar por primera instancia la pieza.

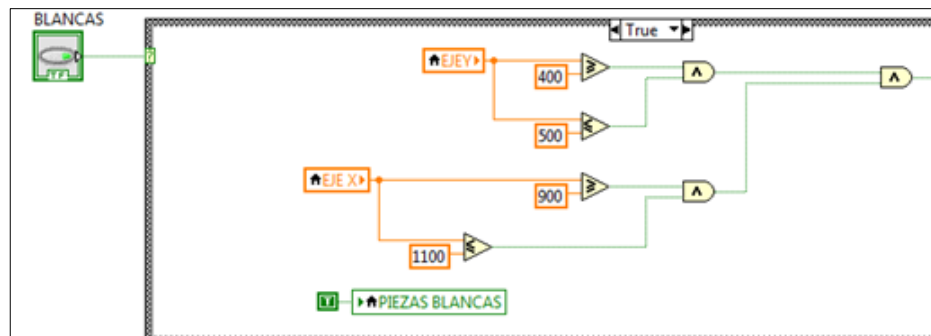
Figura 51 Configuración del patrón de color



Elaborado por: Jaime Heredia

Para elegir el área en donde será buscado el patrón de color se creó una estructura con comparaciones donde se especifica unas coordenadas que determinaran dicha área.

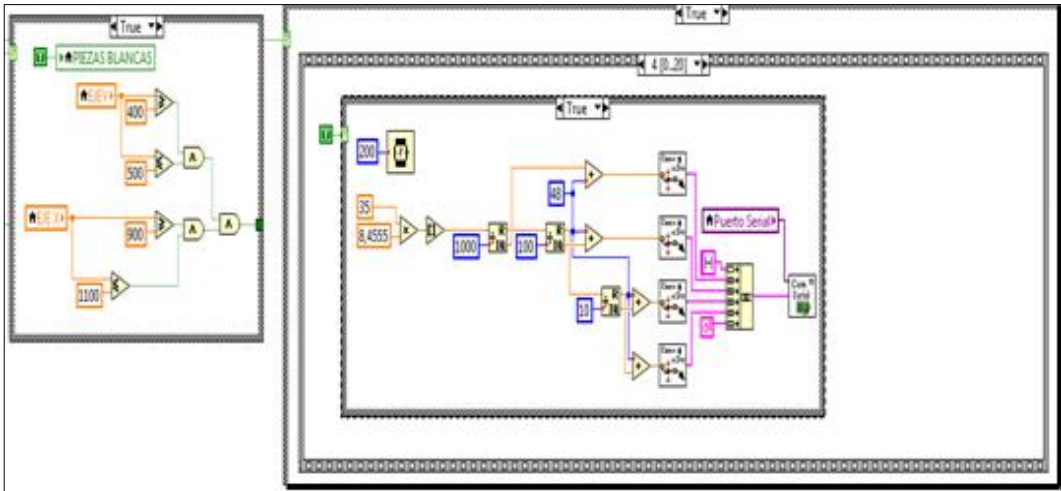
Figura 52 Limitación del área de búsqueda del patrón de color



Elaborado por: Jaime Heredia

Una herramienta muy útil es el módulo de visión assistant se encarga de recibir la imagen adquirida por la cámara y compararla si se encuentra dentro del rango antes mencionado, se crea una figura geométrica en este caso un cuadrado que buscará solo en esa área el patrón de color seleccionado y cuando este coincida dentro del rango de búsqueda se entrega una variable que lleva una condición de verdad a la siguiente estructura que se encarga de mover en tiempos determinados a cada articulación del robot permitiendo así la clasificación de las piezas.

Figura 53 Limitación de rango y programa de control



Elaborado por: Jaime Heredia

## CAPÍTULO 4

### PRUEBAS Y RESULTADOS

En este capítulo se analiza los resultados obtenidos a través de las pruebas realizadas con cada brazo robótico. También los costos de: hardware, diseño de hardware, desarrollo de software y el costo total que alcanzó el proyecto.

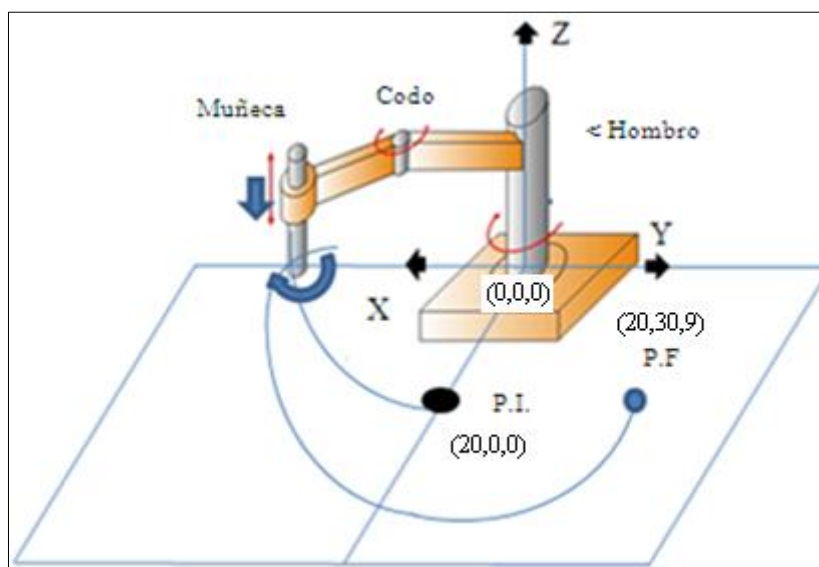
#### 4.1. Análisis del tiempo requerido para realizar una misma trayectoria

En este subcapítulo se analizarán los tiempos que necesitaron cada brazo robótico para cumplir una misma trayectoria en 20 repeticiones.

##### 4.1.1 Análisis del tiempo requerido para realizar una misma trayectoria utilizando el brazo scara

Se programó una trayectoria en el modo automático mediante el software labview 2012, se repitió la acción 20 veces, el propósito de esta prueba es conocer el tiempo promedio en realizar la misma trayectoria obteniendo los siguientes resultados:

Figura 54 Trayectoria realizada por el brazo scara



Elaborado por: Jaime Heredia

La trayectoria que realizó el brazo scara comienza en la posición inicial del robot dado por los finales de carrera ubicados en cada articulación, la articulación del codo

gira 60 grados, en la siguiente posición la articulación de la muñeca gira 90 grados y el efector baja 5 cm finalmente la articulación denominada hombro gira 180 grados.

Tabla 12 Resultados de los tiempos utilizados por el brazo scara para cumplir una misma trayectoria el brazo Scara

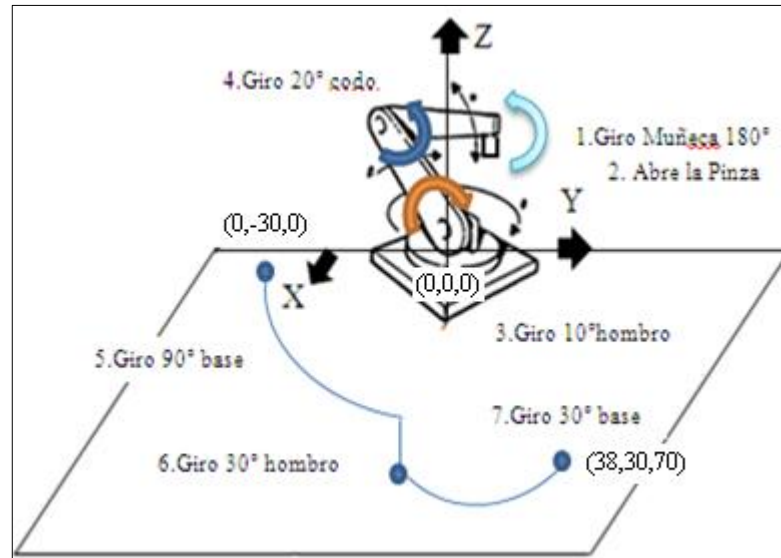
Trayectoria	Tiempo (ms)	Resultados	Error Absoluto
1	10 s 23	Ok	0
2	10 s 20	Ok	0,3
3	10 s 23	Ok	0
4	10 s 21	Ok	0,2
5	10 s 23	Ok	0
6	10 s 20	Ok	0,3
7	10 s 23	Ok	0
8	10 s 21	Ok	0,2
9	10 s 20	Ok	0,3
10	10 s 20	Ok	0,3
11	10 s 26	Ok	0,3
12	10 s 30	Ok	0,7
13	10 s 31	Ok	0,8
14	10 s 23	Ok	0
15	10 s 26	Ok	0,3
16	10 s 25	Ok	0,2
17	10 s 20	Ok	0,3
18	10 s 33	Ok	1
19	10 s 29	Ok	0,6
20	10 s 34	Ok	1.1

Elaborado por: Jaime Heredia

El tiempo promedio requerido por el brazo scara para cumplir una misma trayectoria fue de 10s 24,55 ms la variación de los resultados es de  $\pm 2\%$ .

#### 4.1.2 Análisis del tiempo requerido para realizar una misma trayectoria utilizando el brazo Antropomórfico

Figura 55 Trayectoria realizada por el brazo Antropomórfico



Elaborado por: Jaime Heredia

La trayectoria que cumple para esta prueba el brazo antropomórfico comienza desde su posición inicial, posteriormente el efector abre la pinza, la muñeca gira en un ángulo de  $180^\circ$ , el hombro se desplaza hacia el área de trabajo con un giro de  $10^\circ$ , la posición del codo gira  $20^\circ$  hacia arriba, la base cumple 2 trayectorias una de  $180^\circ$  y otra de  $30^\circ$ , la prioridad en el movimiento de cada articulación es dada para evitar el colapso entre eslabones.



Tabla 13 Resultados y tiempos en que cumple una trayectoria el brazo Antropomórfico

Trayectoria	Tiempo	Resultados	Error
1	16 s 40 ms	Ok	0%
2	16 s 35 ms	Ok	0,5%
3	16 s 50 ms	Ok	1%
4	16 s 40 ms	Ok	0%
5	16 s 45 ms	Ok	0,5%
6	16 s 40 ms	Ok	0%
7	16 s 45 ms	Ok	0,5%
8	16 s 41 ms	Ok	0,1%
9	16 s 39 ms	Ok	0,1%
10	16 s 46 ms	Ok	0,6%
11	16 s 45 ms	Ok	0,5%
12	16 s 40 ms	Ok	0%
13	16 s 50 ms	Ok	1%
14	16 s 33 ms	Ok	0,7%
15	16 s 45 ms	Ok	0,5%
16	16 s 33 ms	Ok	0,7%
17	16 s 40 ms	Ok	0%
18	16 s 41 ms	Ok	0,1%
19	16 s 43 ms	Ok	0,3%
20	16 s 44 ms	Ok	0,4%

Elaborado por: Jaime Heredia

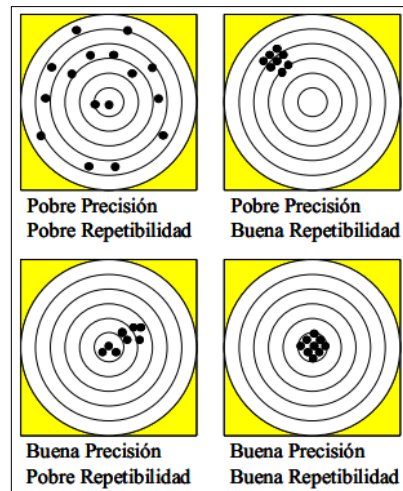
El tiempo promedio requerido por el brazo antropomórfico para cumplir una misma trayectoria fue de 16s 42 ms la variación de los resultados es de  $\pm 2\%$ .

#### 4.2. Precisión y Repetitividad

Precisión es la capacidad para moverse a una posición comandada a una velocidad específica en su área de trabajo establecida. La precisión corresponde a una medida de error es decir que está definida como la diferencia entre el valor medido y el valor comandado

“Cuando un fabricante especifica con qué precisión un manipulador puede regresar a un punto enseñado, está especificando la repetitividad del manipulador”. (Craig, 2006)

Figura 56 Representación gráfica de precisión y repetitividad



Fuente: (Ruiz del Solar & Salazar, 2010)

### 4.3. Exactitud

La exactitud es la capacidad que tiene un robot en este caso el brazo antropomórfico y scara de situar al efector final en un punto específico o señalado dentro del área de trabajo.

La prueba realizada consiste en hacer mover a cada robot desde una misma posición a otra especificada en su propia área de trabajo y comprobar los márgenes de error existentes, la prueba se realizó con 10 datos tomados desde la posición inicial de los robots dada por los finales de carrera hasta la posición  $X=31,3$ ;  $Y=31,5$ ;  $Z=9$  en el brazo antropomórfico y  $X=21,7$ ;  $Y=37,8$ ;  $Z=0$  en el brazo.

Tabla 14 Rangos de valores obtenidos en la prueba de exactitud para el brazo Antropomórfico

Trayectoria	Valor en X=313 (mm)	Valor en Y=315 (mm)	Valor en Z=9 (mm)
1	318	309	9
2	319,5	306,5	9
3	315	309,5	9
4	319,5	306,5	9
5	314	314	9
6	317,5	309,5	9
7	320,5	309	10
8	320,5	309	10
9	315	309,5	9,5
10	314,5	311,5	11
Total	3174	3094	94,5

Elaborado por: Jaime Heredia

Ecuación 5 Formula del valor medio

$$valor\ medio = \frac{(dato1 + dato2 + dato3 + dato\ N)}{N_{datos}}$$

Fuente: (Manuel, 2005)

Ecuación 6 Formula de la desviación

$$Desviacion = dato\ N - valor\ medio.$$

Fuente: (Manuel, 2005)

Ecuación 7 Formula de error absoluto

$$error\ absoluto = \frac{desviación}{N_{datos}}$$

Fuente: (Manuel, 2005)

Ecuación 8 Formula de error relativo

$$error\ relativo(\%) = \frac{error\ absoluto}{valor\ medio} * 100$$

Fuente: (Manuel, 2005)

Tabla 15 Tabulación de datos obtenidos del desplazamiento (mm) en eje X del brazo antropomórfico

Datos obtenidos en el eje X (mm)	Valor medio	Desviación
318	317,4	0,6
319,5	317,4	2,1
315	317,4	2,4
319,5	317,4	2,1
314	317,4	3,4
317,5	317,4	0,1
320,5	317,4	3,1
320,5	317,4	3,1
315	317,4	2,4
314,5	317,4	2,9
Total= 3174		Total= 22,2

Elaborado por: Jaime Heredia

Tabla 16 Tabulación de datos obtenidos del desplazamiento (mm) en el eje Y del brazo antropomórfico

Datos obtenidos en el eje Y(mm)	Valor medio	Desviación
309	309,4	0,4
306,5	309,4	2,9
309,5	309,4	0,1
306,5	309,4	2,9
314	309,4	4,6
309,5	309,4	0,1
309	309,4	0,4
309	309,4	0,4
309,5	309,4	0,1
311,5	309,4	2,1
3094		14

Elaborado por: Jaime Heredia

Tabla 17 Tabulación de datos obtenidos del desplazamiento (mm) en el eje Z del brazo antropomórfico

Datos obtenidos en el eje Z(mm)	Valor medio	Desviación
9	9,45	0,45
9	9,45	0,45
9	9,45	0,45
9	9,45	0,45
9	9,45	0,45
9	9,45	0,45
10	9,45	0,55
10	9,45	0,55
9,5	9,45	0,05
11	9,45	1,55
94,5		5,4

Elaborado por: Jaime Heredia

Tabla 18 Resultado de errores absoluto y relativo en el movimiento del brazo antropomórfico en el eje X

Dato N	Coordenada x	Error relativo	Error absoluto
1	318	0,2	0,6
2	319,5	0,7	2,1
3	315	0,8	2,4
4	319,5	0,7	2,1
5	314	1,1	3,4
6	317,5	0,0	0,1
7	320,5	1,0	3,1
8	320,5	1,0	3,1
9	315	0,8	2,4
10	314,5	0,9	2,9
Total= 10	3174	0,7	2,22

Elaborado por: Jaime Heredia

Tabla 19 Resultado de errores absoluto y relativo en el movimiento del brazo antropomórfico en el eje Y

Dato N	Coordenada y	Error relativo	Error absoluto
1	309	0,1	0,4
2	306,5	0,9	2,9
3	309,5	0,0	0,1
4	306,5	0,9	2,9
5	314	1,5	4,6
6	309,5	0,0	0,1
7	309	0,1	0,4
8	309	0,1	0,4
9	309,5	0,0	0,1
10	311,5	0,7	2,1
Total= 10	3094	0,5	1,4

Elaborado por: Jaime Heredia

Tabla 20 Resultado de errores absoluto y relativo en el movimiento del brazo antropomórfico en el eje Z

Dato N	Coordenada z	Error relativo	Error absoluto
1	9	4,8	0,45
2	9	4,8	0,45
3	9	4,8	0,45
4	9	4,8	0,45
5	9	4,8	0,45
6	9	4,8	0,45
7	10	5,8	0,55
8	10	5,8	0,55
9	9,5	0,5	0,05
10	11	16,4	1,55
Total:	94,5	5,7	0,54

Elaborado por: Jaime Heredia

Tabla 21 Resultados totales de errores absolutos y relativos en la trayectoria cumplida por el brazo antropomórfico

Error total	Coordenadas X	Coordenadas Y	Coordenadas Z
Error Absoluto	2,22	1,4	0,54
Error Relativo	0,7	0,5	5,7

Elaborado por: Jaime Heredia

Tabla 22 Rangos de valores obtenidos en la prueba de exactitud para el brazo Scara

Trayectoria	Valor en X=217 (mm)	Valor en Y=378 (mm)	Valor en Z=0 (mm)
1	217	378	0
2	217	378	0
3	217	378	0
4	215,5	378	0
5	215,5	378	0
6	215,5	378	0
7	216,5	379	0
8	216,5	378,5	0
9	216,5	378,5	0
10	217	378	0

Elaborado por: Jaime Heredia

Tabla 23 Tabulación de datos obtenidos del desplazamiento en (mm) del brazo Scara en el eje X

Datos obtenidos en el eje X(mm)	Valor medio	Desviación
217	216,4	0,6
217	216,4	0,6
217	216,4	0,6
215,5	216,4	0,9
215,5	216,4	0,9
215,5	216,4	0,9
216,5	216,4	0,1
216,5	216,4	0,1
216,5	216,4	0,1
217	216,4	0,6
Total= 2164		Total= 5,4

Elaborado por: Jaime Heredia

Tabla 24 Tabulación de datos obtenidos del desplazamiento en (mm) del brazo Scara en el eje Y

Datos obtenidos en el eje Y(mm)	Valor medio	Desviación
378	378,2	0,2
378	378,2	0,2
378	378,2	0,2
378	378,2	0,2
378	378,2	0,2
378	378,2	0,2
379	378,2	0,8
378,5	378,2	0,3
378,5	378,2	0,3
378	378,2	0,2
Total= 3782		Total= 2,8

Elaborado por: Jaime Heredia

Tabla 25 Resultado de errores absolutos y relativos en el movimiento del brazo Scara en el eje X

Dato N	Coordenada en el eje X	Error relativo	Error absoluto
1	217	0,3	0,6
2	217	0,3	0,6
3	217	0,3	0,6
4	215,5	0,4	0,9
5	215,5	0,4	0,9
6	215,5	0,4	0,9
7	216,5	0,0	0,1
8	216,5	0,0	0,1
9	216,5	0,0	0,1
10	217	0,3	0,6
Total= 10	2164	0,2	0,54

Elaborado por: Jaime Heredia

Tabla 26 Resultado de errores absoluto y relativo en el movimiento del brazo Scara en el eje Y

Dato N	Coordenada en el eje Y	Error relativo	Error absoluto
1	378	0,1	0,2
2	378	0,1	0,2
3	378	0,1	0,2
4	378	0,1	0,2
5	378	0,1	0,2
6	378	0,1	0,2
7	379	0,2	0,8
8	378,5	0,1	0,3
9	378,5	0,1	0,3
10	378	0,1	0,2
Total= 10	3782	0,1	0,28

Elaborado por: Jaime Heredia

Tabla 27 Resultados totales de errores absolutos y relativos en la trayectoria cumplida por el brazo scara

Error total	Coordenadas X	Coordenadas Y	Coordenadas Z
Error Absoluto	0,54	0,28	0
Error Relativo	0,2	0,1	0

Elaborado por: Jaime Heredia

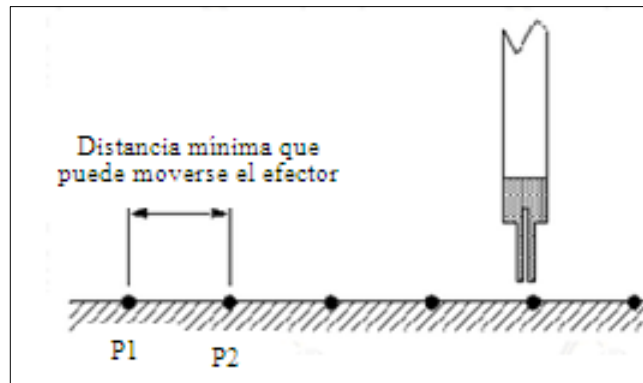
#### 4.4 Resolución Espacial

“La resolución espacial, se entiende como la menor variación posible en el posicionamiento del efector final.” (Ollero, 2001)

La resolución espacial es la distancia entre dos puntos adyacentes. Siendo estos puntos P1 y P2.



Figura 57 Representación gráfica de la resolución escalar



Elaborado por: Jaime Heredia

Tabla 28 Resolución Espacial del brazo Scara y Antropomórfico encontradas según pruebas realizadas

Brazo Robótico	Resolución Escalar (mm)
Scara	1
Antropomórfico	1,5

Elaborado por: Jaime Heredia

#### 4.5 Análisis de costos

En este subcapítulo se analiza los costos de: hardware, diseño de hardware, desarrollo de software y el costo total que alcanzó el proyecto de investigación.

##### 4.5.1 Costo de desarrollo de Hardware

Los elementos que fueron entregados por la Universidad Politécnica Salesiana constan con un valor de cero en la tabla de costos de hardware, a continuación se detalla el precio unitario de cada elemento.

Tabla 29 Costos de hardware

Cantidad	Descripción	Precio.Unit. \$	Precio.Total. \$
100	Resistencias de ¼ de watts	0,03	3
8	Resistencias de 2 de watts	0,054	1
5	Potenciómetros	0,75	3,75
20	Condensador Cerámico diferentes capacidades	0,15	3
20	Condensador Electrolítico diferentes capacidad	0,6	12
60	Sócalos diferentes medidas	0,4	24
2	Regulador de voltaje	0,55	1,10
2	Puente de diodos	0,45	0,90
64	Diodos rectificadores a 3 A	0,6	38,4
11	Amplificadores operacionales	0,88	9,68
3	Microcontroladores Atmega 128	26,5	79,5
2	Microcontroladores Atmega 8	4,5	9
2	Microcontroladores Atmega 164	7,5	15
8	L298N	4,04	32,32
8	Cristal de 16 MGHZ	1,5	12
2	Interruptores	0,20	0,40
3	Sócalo	0,59	1,78
1	GLCD	24	48
1	Fuente Swiching	45	90
4	Cajas de acrílico	45	180
2	Muebles porta robot	150	300
20	Diodos emisor de luz	0,20	4
2	PC para programas especializados	700	2.520,00
5	Bornera	0,4	2
4	Motores Pittman con encoder	0	0
6	Motores Brevel con encoder	0	0
10	Elaboración de Tarjetas	2,66	266,36
1	Subscripcion Revistas IEEE	114,5	114,5
2	Tarjetas para investigación	2143,25	4.296,50
2	Tarjetas de Prueba	159	319,48
1	Brazo Antropomorfico	0	0
1	Laptop	1500	1500
1	Caja con herramientas	200	200
1	Brazo Scara	0	0
<b>TOTAL</b>			<b>\$ 10087,67</b>

Elaborado por: Jaime Heredia

#### 4.5.2 Costo de desarrollo del software

Los software's utilizados para el desarrollo de este proyecto son: Labview, Matlab, vrbuid2, bascom. Todos los software's constan con un valor de cero por que fueron entregados por la Universidad Politécnica Salesiana.

## CONCLUSIONES

- Se analizó el estado del arte de los brazos antropomórfico y scara, fueron necesarios algunos arreglos mecánicos y electrónicos tanto en los motores como en sus respectivos encoders, las cadenas que forman parte de cada brazo debieron ser calibradas para mejorar la precisión de los brazos.
- Se diseñaron 2 tarjetas de control y 6 de potencia para el control de cada brazo. Cada placa fue diseñada para que sus pistas ocupen las dos caras de la baquelita con esto se optimizó el tamaño de la controladora.
- Se diseñaron los programas de control en los software's Matlab y Labview, además se simuló los movimientos y limitaciones de cada robot en el software vrbuild que es un constructor de objetos en 3D este programa es controlado por una interfaz con el usuario desarrollada en Guide.
- Los fuses de configuración del microcontrolador deben ser activados y desactivados según la utilidad de estos en el proyecto, en el atmega 128 de la tarjeta de control se activó el fuse de cristal externo de 16MHZ, el fuse Brown-out detection level y el fuse serial program. El fuse CKOPT por defecto viene activado en el microcontrolador este fuse no permite una velocidad de transmisión mayor a 8 MHz sea con cristal externo o interno y crea un conflicto en el envío de datos, al desactivar dicho fuse se logró la transmisión de datos a la velocidad requerida de 16MHz.
- El área de trabajo para el modo de visión artificial debió ser modificado en el aspecto de la iluminación, cuando el patrón de color de la pieza a ser clasificada es de color claro y el entorno supera los 283 luxes se produce reflejos de luz que provocaron varias confusiones al dispositivo receptor en este caso la cámara y como consecuencia un erróneo envío de datos, de igual manera cuando la luz del ambiente es menor a 78 luxes no hay recepción de datos aun cuando la pieza esté en el lugar de clasificación, la cantidad de luxes en la que funciona actualmente el modulo sin ningún inconveniente es de 168,7 luxes.
- En las pruebas de exactitud, los errores absolutos y relativos en el brazo scara son casi cero, mientras que en el brazo antropomórfico tienen un porcentaje

mayor, este valor de inexactitud viene dado por la parte mecánica del brazo antropomórfico ya que consta de un mayor número de cadenas y engranajes para la tracción, además al tener mayor longitud sus cadenas se produce una des calibración leve que afecta al posicionamiento del brazo antropomórfico.

- Al realizar el envío de datos por medio del teach pad se concluyó que el robot puede llegar a una posición de forma más exacta, ya que los datos enviados no son escalados, de esta manera se obtiene una mayor exactitud en los movimientos deseados, además de una reducción en el espacio escalar de cada robot.
- El brazo robótico antropomórfico cumplió una misma trayectoria llevando en su efector final diferentes pesos, con un peso de 10gr demora 13,11s y con 1Kgr tardo en cumplir la trayectoria 17,3s, este es el rango de pesos que puede llevar el robot, se puede notar un pequeño incremento en la cantidad de tiempo pero, no hay ninguna afectación en la exactitud del robot para llegar al punto de destino, cuando el peso excede su valor máximo se produce una sobre corriente la cual está siendo censada por el pin análogo digital del micro el cual hace que el proceso se pare por el alto consumo de corriente.

## RECOMENDACIONES

- Se recomienda acoplar a la tarjeta de control otros dispositivos programables como la FPGA, MyDAC, arduino, etc, el uso de estos dispositivos hacen que el algoritmo de control pueda ser más eficiente que el presentado en este proyecto.
- No cambiar la distribución de pines de la tarjeta de control, tampoco los niveles de voltaje utilizados para el movimiento de los motores y alimentación del encoder además se debe tener en cuenta que al hacer mantenimiento o al retirar un motor, las cadenas que conforman el robot pierden su posición original alterando el correcto funcionamiento del brazo robótico.

## LISTA DE REFERENCIAS

- Alciatore, D., & Histan, M. (2008). *Introducción a la Mecatrónica y los Sistemas de Medición*. México D.F.: McGraw-Hill Interamerica.
- Alhammadi, K. (2006). *Applying Wide Field of View Retroreflector Technology to Free Space Optical Robotic Communications*. North Carolina: CRIM.
- Areny, R. P. (2007). *Sensores y Acondicionadores de Señal*. México D.F.: Alfaomega.
- Bravo, C. A. (23 de junio de 2000). *Disclab*. Recuperado el 26 de septiembre de 2013, de Disclab: [http://www.disclab.ua.es/robolab/EJS4/PRR\\_Suficiencia\\_Intro\\_2.html](http://www.disclab.ua.es/robolab/EJS4/PRR_Suficiencia_Intro_2.html)
- Büch, R. (2012). *Brushless Motors and Controllers*. Norderstedt: Books on Demand GmbH.
- Craig, J. J. (2006). *Robótica*.
- Currás, J. F. (10 de octubre de 2011). *Hardware libre*. Recuperado el 20 de octubre de 2013, de Los fuses en los AVR's: <https://sites.google.com/a/ghandalf.org/hardware-libre/arduino/los-fuses-en-los-avr>
- Cursomicros. (4 de marzo de 2009). *Programación de Microcontroladores*. Recuperado el 27 de septiembre de 2013, de Programación de Microcontroladores: <http://www.cursomicros.com/>
- Esqueda, J. (20 de noviembre de 2002). *Proton*. Recuperado el 15 de octubre de 2013, de Proton: <http://proton.ucting.udg.mx/~cheko/pdf/matv.pdf>
- Esquembre, D. F. (5 de agosto de 2007). *Robolab*. Recuperado el 20 de septiembre de 2013, de Robolab: [http://www.aurova.ua.es/robolab/EJS2/RRR\\_Intro\\_3.html](http://www.aurova.ua.es/robolab/EJS2/RRR_Intro_3.html)

- Instruments, N. (4 de julio de 2013). *National Instruments*. Recuperado el 25 de agosto de 2013, de National Instruments: <http://www.ni.com/labview/why/esa/>
- Manuel, V. (2005). *Muestreo Estadístico. Diseño Y Aplicaciones*. Santiago de Chile: Universitaria.
- Mora, J. F. (2008). *Máquinas Eléctricas*. Madrid: MonoComp, S.A.
- Muñoz, M. V. (31 de julio de 2009). *Espol*. Recuperado el 10 de julio de 2013, de Espol: <http://www.dspace.espol.edu.ec/bitstream/123456789/5241/4/Precalculo%20de%20Villena%20-%202004%20-%20Coordenadas%20Polares.pdf>
- Ollero, A. (2001). *Robótica: manipuladores y robots móviles*. Barcelona: GyERSA.
- Pittman. (29 de noviembre de 2012). *METEK*. Recuperado el 10 de septiembre de 2013, de METEK: <http://www.pittman-motors.com/Brushless-DC-Motors/High-Performance-Slotless-Brushless-DC-Servo-Motor.aspx>
- Real Academia de la Lengua, R. A. (1 de enero de 2013). *Diccionario de la lengua española*. Recuperado el 3 de junio de 2013, de Diccionario de la lengua española: <http://www.rae.es/>
- Reyes, L. (20 de noviembre de 2009). *Física*. Recuperado el 25 de julio de 2013, de Física: [http://fisica.udea.edu.co/~lab-gicm/Curso%20FPGA\\_2011/2012\\_Interfaz%20grafica%20en%20Guide%20Matlab.pdf](http://fisica.udea.edu.co/~lab-gicm/Curso%20FPGA_2011/2012_Interfaz%20grafica%20en%20Guide%20Matlab.pdf)
- Rodríguez, n., & Martínez, W. (2006). *Planificación Y Evaluación de Proyectos Informáticos*. San Jose: Universidad Estatal a distancia.
- Ruiz del Solar, J., & Salazar, R. (1 de febrero de 2010). *Robotica Universidad de Chile*. Recuperado el 22 de octubre de 2013, de Robotica Universidad de Chile: <http://robotica.li2.uchile.cl/EL63G/capitulo1.pdf>

Sánchez, J. A. (2002). *Avances en Robótica y Visión por computador*. La Mancha: COMPOBELL.

Setecindca. (6 de febrero de 2000). *Setecindca*. Recuperado el 28 de agosto de 2013, de Setecindca: <http://www.setecindca.com/descargas/encoders/encoder-eltra.pdf>

Steiner, E. (2003). *Matemáticas para las ciencias aplicadas*. New York: REVERTE, S.A.

Tomás, F. (9 de mayo de 2010). *Laplace*. Recuperado el 22 de julio de 2013, de Laplace: [http://laplace.us.es/wiki/index.php/Coordenadas\\_cartesianas.\\_Definici%C3%B3n](http://laplace.us.es/wiki/index.php/Coordenadas_cartesianas._Definici%C3%B3n)

uCSystem. (10 de enero de 2011). *uCSystem*. Recuperado el 15 de agosto de 2013, de uCSystem - Sistemas Basados en Microcontrolador: [http://ucsystem.blogspot.com/2011\\_12\\_01\\_archive.html](http://ucsystem.blogspot.com/2011_12_01_archive.html)

Zambrana, J. C. (2 de julio de 2009). *Universidad Carlos III de Madrid*. Recuperado el 13 de septiembre de 2013, de Universidad Carlos III de Madrid: [http://iearobotics.com/alberto/lib/exe/fetch.php?media=theses:tfg\\_carlos\\_rod\\_zam.pdf](http://iearobotics.com/alberto/lib/exe/fetch.php?media=theses:tfg_carlos_rod_zam.pdf)



## ANEXOS

### Anexo 1 Programa del Microcontrolador para el control de los motores.

```
'-----
'name          : CtrBrazoScara2.bas
'copyright     : (c) 2013, MCS Electronics
'purpose       : Brazo
'micro         : Mega128
'-----

'$sim
$regfile = "m128def.dat"           'Micro a Utilizar ATMEGA128
'$regfile = "m32can.dat"          ' specify the used micro
$crystal = 16000000                'frecuencia del cristal 16 Mhz
$baud = 9600                        'use baud TECH
$baud1 = 9600                       'use baud USB
'$lib "mcsbyte.lbx"                ' for smaller code
'$baud = 19200                      'Baud rate com.serial hardware
'$hwstack = 32                      'default use 32 for the hardware stack
'$swstack = 10                      'default use 10 for the SW stack
'$framesize = 40                    'default use 40 for the frame space

Config Com1 = Dummy , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Config Com2 = Dummy , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0

'try the second hardware UART
Open "com2:" For Binary As #1

***** INTERRUPCION INT0 *****
Config Int7 = Rising
On Int7 Int7_int                    'Initialise the INT0 Interrupt
Config Int6 = Rising
On Int6 Int6_int                    'Initialise the INT0 Interrupt
Config Int1 = Rising
On Int1 Int1_int                    'Initialise the INT0 Interrupt
Config Int0 = Rising
On Int0 Int0_int                    'Initialise the INT0 Interrupt
***** Converso ADC *****

Dim A As Word
Dim S As Single
Dim Valor As String * 10
***** INTERRUPCION de serial *****
Dim D1 As String * 20
Dim Control As Byte
On Urxc Rec_isr                      'Define subrutina de Interrupcion
On Urxcl Rec_isr1                    'Define subrutina de Interrupcion
*****
'*          LED1          *
*****

Ddrg.2 = 1
Portg.2 = 0
Led1 Alias Portg.2

***** MOTOR F *****
Ddra.4 = 1                            'Portico A.4 como SALIDA
Porta.4 = 0                            'Estado inicial del pin A.4=0
Ddra.5 = 1                            'Portico A.5 como SALIDA
Porta.5 = 0                            'Estado inicial del pin A.5=0
```

```

Ddre.4 = 1          'Portico E.4 como SALIDA (OC1B)
Porte.4 = 0          'Estado inicial del pin E.4=0
Ctrlmf1 Alias Porta.4
Ctrlmf2 Alias Porta.5
Pwmf Alias Porte.4
***** MOTOR E *****
'Ddra.6 = 1          'Portico A.6 como SALIDA
'Porta.6 = 0          'Estado inicial del pin A.6=0
'Ddrf.7 = 1          'Portico A.7 como SALIDA
'Portf.7 = 0          'Estado inicial del pin A.7=0
'Ddra.7 = 1          'Portico A.7 como SALIDA
'Porta.7 = 0          'Estado inicial del pin A.7=0
'Ddre.3 = 1          'Portico E.3 como SALIDA (OC1C)
'Porte.3 = 0          'Estado inicial del pin E.3=0
'Ctrlme1 Alias Porta.6
'Ctrlme1 Alias Portf.7
'Ctrlme2 Alias Porta.7
'Pwme Alias Porte.3
***** MOTOR D *****
'Ddrc.7 = 1          'Portico C.7 como SALIDA
'Portc.7 = 0          'Estado inicial del pin C.7=0
'Ddrc.6 = 1          'Portico C.6 como SALIDA
'Portc.6 = 0          'Estado inicial del pin C.6=0
'Ddrb.5 = 1          'Portico B.5 como SALIDA (OC3C)
'Portb.5 = 0          'Estado inicial del pin B.5=0
'Ctrlmd1 Alias Portc.7
'Ctrlmd2 Alias Portc.6
'Pwmd Alias Portb.5
***** MOTOR C *****
'Ddrc.5 = 1          'Portico C.5 como SALIDA
'Portc.5 = 0          'Estado inicial del pin C.5=0
'Ddrc.4 = 1          'Portico C.4 como SALIDA
'Portc.4 = 0          'Estado inicial del pin C.4=0
'Ddre.5 = 1          'Portico E.5 como SALIDA (OC1A)
'Porte.5 = 0          'Estado inicial del pin E.5=0
'Ctrlmc1 Alias Portc.5
'Ctrlmc2 Alias Portc.4
'Pwmc Alias Porte.5
***** MOTOR B *****
'Ddrc.3 = 1          'Portico C.3 como SALIDA
'Portc.3 = 0          'Estado inicial del pin C.3=0
'Ddrc.2 = 1          'Portico C.2 como SALIDA
'Portc.2 = 0          'Estado inicial del pin C.2=0
'Ddrb.7 = 1          'Portico B.7 como SALIDA (OC3A)
'Portb.7 = 0          'Estado inicial del pin B.7=0
'Ctrlmb1 Alias Portc.3
'Ctrlmb2 Alias Portc.2
'Pwmb Alias Portb.7
***** MOTOR A *****
'Ddrc.1 = 1          'Portico C.1 como SALIDA
'Portc.1 = 0          'Estado inicial del pin C.1=0
'Ddrc.0 = 1          'Portico C.0 como SALIDA
'Portc.0 = 0          'Estado inicial del pin C.0=0
'Ddrb.6 = 1          'Portico B.6 como SALIDA (OC3B)
'Portb.6 = 0          'Estado inicial del pin B.6=0
'Ctrlma1 Alias Portc.1
'Ctrlma2 Alias Portc.0
'Pwma Alias Portb.6
***** ENCODER F *****

```

Ddra.3 = 0	'Portico A.3 como entrada
Porta.3 = 1	'Habilita resistencia de pull-up A.3
Ddrd.0 = 0	'Portico D.0 como entrada
Portd.0 = 1	'Habilita resistencia de pull-up D.0
Ddra.1 = 0	'Portico A.1 como entrada
Porta.1 = 1	'Habilita resistencia de pull-up A.1
Swf Alias Pina.3	'Alias del PINA.3
Sa1 Alias Pind.0	'Alias del PIND.0
Sb1 Alias Pina.1	'Alias del PINA.1

\*\*\*\*\* ENCODER E \*\*\*\*\*

Ddra.2 = 0	'Portico A.2 como entrada
Porta.2 = 1	'Habilita resistencia de pull-up A.2
Ddrd.1 = 0	'Portico D.1 como entrada
Portd.1 = 1	'Habilita resistencia de pull-up D.1
Ddra.0 = 0	'Portico A.0 como entrada
Porta.0 = 1	'Habilita resistencia de pull-up A.0
Swe Alias Pina.2	'Alias del PINA.2
Sa2 Alias Pind.1	'Alias del PIND.1
Sb2 Alias Pina.0	'Alias del PINA.0

\*\*\*\*\* ENCODER D \*\*\*\*\*

Ddrb.4 = 0	'Portico B.4 como entrada
Portb.4 = 1	'Habilita resistencia de pull-up B.4
Ddre.6 = 0	'Portico E.6 como entrada
Porte.6 = 1	'Habilita resistencia de pull-up E.6
Ddrb.3 = 0	'Portico B.3 como entrada
Portb.3 = 1	'Habilita resistencia de pull-up B.3
Swd Alias Pinb.4	'Alias del PINB.4
Sa3 Alias Pine.6	'Alias del PINE.6
Sb3 Alias Pinb.3	'Alias del PINB.3

\*\*\*\*\* ENCODER C \*\*\*\*\*

Ddrb.2 = 0	'Portico B.2 como entrada
Portb.2 = 1	'Habilita resistencia de pull-up B.2
Ddre.7 = 0	'Portico E.7 como entrada
Porte.7 = 1	'Habilita resistencia de pull-up E.7
Ddrb.1 = 0	'Portico B.1 como entrada
Portb.1 = 1	'Habilita resistencia de pull-up B.1
Swc Alias Pinb.2	'Alias del PINB.2
Sa4 Alias Pine.7	'Alias del PINE.7
Sb4 Alias Pinb.1	'Alias del PINB.1

\*\*\*\*\* ENCODER B \*\*\*\*\*

Ddrb.0 = 0	'Portico B.0 como entrada
Portb.0 = 1	'Habilita resistencia de pull-up B.0
Ddrd.7 = 0	'Portico D.7 como entrada
Portd.7 = 1	'Habilita resistencia de pull-up D.7
Ddre.2 = 0	'Portico E.2 como entrada
Porte.2 = 1	'Habilita resistencia de pull-up E.2
Swb Alias Pinb.0	'Alias del PINB.0
Sa5 Alias Pind.7	'Alias del PIND.7
Sb5 Alias Pine.2	'Alias del PINE.2

\*\*\*\*\* ENCODER A \*\*\*\*\*

Ddrd.4 = 0	'Portico D.4 como entrada
Portd.4 = 1	'Habilita resistencia de pull-up D.4
Ddrd.6 = 0	'Portico D.6 como entrada
Portd.6 = 1	'Habilita resistencia de pull-up D.6
Ddrd.5 = 0	'Portico D.5 como entrada
Portd.5 = 1	'Habilita resistencia de pull-up D.5
Swa Alias Pind.4	'Alias del PIND.4
Sa6 Alias Pind.6	'Alias del PIND.6

Sb6 Alias Pind.5

'Alias del PIND.5

Const Retardo = 500

Const Retardo1 = 2200

Const Rebote = 60

Const Filat = 4

Const Filad = 7

'Variables

Dim X As Byte , Y As Byte

Dim B As Byte

Dim I As Word

Dim Flag1 As Bit

Dim Flag2 As Bit

Dim Flag3 As Bit

Dim Flag4 As Bit

Dim Flag5 As Bit

Dim Flag6 As Bit

Dim Flag7 As Bit

Dim Flag8 As Bit

Dim Flag9 As Bit

Dim Flag10 As Bit

Dim Flag11 As Bit

Dim Flag12 As Bit

Dim Flag13 As Bit

Dim Flag14 As Bit

Dim Flag15 As Bit

Dim Flag16 As Bit

Dim Flag17 As Bit

Dim Flag18 As Bit

Dim Flag19 As Bit

Dim Flag20 As Bit

Dim Flag21 As Bit

Dim Flag22 As Bit

Dim Bandera As Bit

Dim Cuenta As Byte

Dim Datox As Byte

Dim Datoy As Byte

Dim Datoz As Byte

Dim Datox1 As Word

Dim Datoy1 As Word

Dim Datoz1 As Word

Dim Tiempo As Byte

Dim Bandera2 As Bit

Dim Bandera3 As Bit

Dim Contador\_a As Word

Dim Senluz As Byte

Dim Contmenu As Byte

Dim Forma1 As Byte , Forma2 As Byte , Forma3 As Byte , Forma4 As Byte

Dim Forma5 As Byte , Forma6 As Byte , Forma7 As Byte , Forma8 As Byte

Dim Forma9 As Byte , Forma10 As Byte , Forma11 As Byte , Forma12 As Byte

Dim Columna As Byte

Dim Sensorir As Byte

Dim Sensorsonido As Byte

Dim Menu(4) As String \* 25

Dim Temporal1 As String \* 1

Dim Temporal2 As String \* 1

Dim Temporal3 As String \* 4

Dim Serial1 As String \* 10

Dim Sentido As String \* 3

```

Wait 1
Enable Interrupts
Enable Urx
Enable Urx1
Enable Int0
Enable Int1
Enable Int6
Enable Int7
Datox = 0
Datoy = 0
Datoz = 0
Datox1 = 0
Datoy1 = 0
Datoz1 = 0
Flag1 = 0
Flag2 = 0
Contador_a = 0
Tiempo = 0
  Do
    B = Inkey()
    Loop Until B = 0
***** Animación Inicial *****
  For X = 0 To 5
    Led1 = 1
    Waitms 200
    Led1 = 0
    Waitms 200
  Next X

  Do
    B = Inkey()
    Loop Until B = 0
Bandera = 0
Do
If Flag11 = 1 Then
Flag11 = 0
  If Bandera = 0 Then
    While Swa = 1
      Ctrlma1 = 1
      Ctrlma2 = 0
      Pwma = 1
      Waitms 200
      Ctrlma1 = 0
      Ctrlma2 = 0
      Pwma = 0
      Waitms 200
    Wend
    Ctrlma1 = 0
    Ctrlma2 = 0
    Pwma = 0
    Contador_a = 0
    Datox1 = 0
    Bandera = 1
  End If
End If

If Flag12 = 1 Then
Flag12 = 0
  If Bandera = 1 Then

```

```

    Bandera = 0
    Ctrlma1 = 0
    Ctrlma2 = 1
    Pwma = 1
    For I = 1 To 8
        Ctrlma1 = 0
        Ctrlma2 = 1
        Pwma = 1
        Waitms 50
        Ctrlma1 = 0
        Ctrlma2 = 0
        Pwma = 0
        Waitms 50
    Next I
    Ctrlma1 = 0
    Ctrlma2 = 0
    Pwma = 0
End If
End If
If Flag1 = 1 Then
    Flag1 = 0
    Contador_a = 0
    Ctrlmb1 = 1
    Ctrlmb2 = 0
    Pwmb = 1
    Bandera2 = 0
    While Datox1 >= Contador_a
        If Sa5 = 1 Then
            If Bandera2 = 0 Then
                Contador_a = Contador_a + 1
                Bandera2 = 1
            End If
        Else
            Bandera2 = 0
        End If
    Wend
    Ctrlmb1 = 0
    Ctrlmb2 = 0
    Pwmb = 0
    Contador_a = 0
    Datox1 = 0
End If

If Flag2 = 1 Then
    Flag2 = 0
    Contador_a = 0
    Ctrlmb1 = 0
    Ctrlmb2 = 1
    Pwmb = 1
    Bandera3 = 0
    While Datox1 >= Contador_a
        If Sa5 = 1 Then
            If Bandera3 = 0 Then
                Contador_a = Contador_a + 1
                Bandera3 = 1
            End If
        Else
            Bandera3 = 0
        End If
    End If

```

```

Wend
Ctrlmb1 = 0
Ctrlmb2 = 0
Pwmb = 0
Contador_a = 0
Datox1 = 0
End If

If Flag3 = 1 Then
Flag3 = 0
Contador_a = 0
While Datox1 >= Contador_a
Ctrlmc1 = 1
Ctrlmc2 = 0
Pwmc = 1
Wend
Ctrlmc1 = 0
Ctrlmc2 = 0
Pwmc = 0
Contador_a = 0
Datox1 = 0
End If

If Flag4 = 1 Then
Flag4 = 0
Contador_a = 0
While Datox1 >= Contador_a
Ctrlmc1 = 0
Ctrlmc2 = 1
Pwmc = 1
Wend
Ctrlmc1 = 0
Ctrlmc2 = 0
Pwmc = 0
Contador_a = 0
Datox1 = 0
End If

If Flag5 = 1 Then
Flag5 = 0
Contador_a = 0
While Datox1 >= Contador_a
Ctrlmd1 = 1
Ctrlmd2 = 0
Pwmd = 1
Wend
Ctrlmd1 = 0
Ctrlmd2 = 0
Pwmd = 0
Contador_a = 0
Datox1 = 0
End If

If Flag6 = 1 Then
Flag6 = 0
Contador_a = 0
While Datox1 >= Contador_a
Ctrlmd1 = 0
Ctrlmd2 = 1
Pwmd = 1

```

```

Wend
Ctrlmd1 = 0
Ctrlmd2 = 0
Pwmd = 0
Contador_a = 0
Datox1 = 0
End If
If Flag7 = 1 Then
Flag7 = 0
Contador_a = 0
If Swe = 1 Then
Ctrlme1 = 1
Ctrlme2 = 0
Pwme = 1
End If
While Datox1 >= Contador_a
If Swe = 0 Then
Ctrlme1 = 0
Ctrlme2 = 0
Pwme = 0
Datox1 = 65535
End If
Wend
Ctrlme1 = 0
Ctrlme2 = 0
Pwme = 0
Contador_a = 0
Datox1 = 0
End If

If Flag8 = 1 Then
Flag8 = 0
Contador_a = 0
While Datox1 >= Contador_a
Ctrlme1 = 0
Ctrlme2 = 1
Pwme = 1
Wend
Ctrlme1 = 0
Ctrlme2 = 0
Pwme = 0
Contador_a = 0
Datox1 = 0
End If
If Flag9 = 1 Then
Flag9 = 0
Contador_a = 0
While Datox1 >= Contador_a
Ctrlmf1 = 1
Ctrlmf2 = 0
Pwmf = 1
Wend
Ctrlmf1 = 0
Ctrlmf2 = 0
Pwmf = 0
Contador_a = 0
Datox1 = 0
End If

```



```

If Flag10 = 1 Then
  Flag10 = 0
  Contador_a = 0
  While Datox1 >= Contador_a
    Ctrlmf1 = 0
    Ctrlmf2 = 1
    Pwmf = 1
  Wend
  Ctrlmf1 = 0
  Ctrlmf2 = 0
  Pwmf = 0
  Contador_a = 0
  Datox1 = 0
End If
If Flag13 = 1 Then
  Flag13 = 0
  Gosub Resetm
End If
Loop

Do
***** Control de Motor Efector Abre *****
If Flag1 = 1 Then
  Contador_a = 0
  Print # 1 , "C"
  Print "C" ;
  Flag1 = 0
  Ctrlmb1 = 1
  Ctrlmb2 = 0
  Pwmb = 1
  If Swb = 0 Then
    Waitms 50
  Else
    Waitms 200
  End If
  Pwmb = 0
  Ctrlmb1 = 0
  Ctrlmb2 = 0
  Print # 1 , Str(contador_a) ; Sentido
' Disable Int0
End If
***** Control de Motor Efector Cierra*****
If Flag2 = 1 Then
  Contador_a = 0
  Print # 1 , "D"
  Print "D" ;
  Flag2 = 0
  Ctrlmb1 = 0
  Ctrlmb2 = 1
  Pwmb = 1
  If Swb = 0 Then
    Waitms 50
  Else
    Waitms 200
  End If
  Pwmb = 0
  Ctrlmb1 = 0
  Ctrlmb2 = 0
  Print # 1 , Str(contador_a) ; Sentido

```

```

End If
***** Control de Motor Efector Sube *****
If Flag3 = 1 Then
  Contador_a = 0
  Print # 1 , "E"
  Print "E" ;                               'Chr(13);
  Flag3 = 0
  Ctrlmc1 = 1
  Ctrlmc2 = 0
  Pwmc = 1
  If Swc = 0 Then
    Waitms 50
  Else
    Waitms 200
  End If
  Pwmc = 0
  Ctrlmc1 = 0
  Ctrlmc2 = 0
  Print # 1 , Str(contador_a) ; Sentido
End If
***** Control de Motor Efector Baja *****
If Flag4 = 1 Then
  Contador_a = 0
  Print # 1 , "F"
  Flag4 = 0
  Ctrlmc1 = 0
  Ctrlmc2 = 1
  Pwmc = 1
  If Swc = 0 Then
    Waitms 50
  Else
    Waitms 200
  End If
  Pwmc = 0
  Ctrlmc1 = 0
  Ctrlmc2 = 0
  Print # 1 , Str(contador_a) ; Sentido
End If

***** Control de Motor Muñeca Horario *****
If Flag5 = 1 Then
  Contador_a = 0
  Print # 1 , "G"
  Flag5 = 0
  Ctrlmd1 = 1
  Ctrlmd2 = 0
  Pwmd = 1
  If Swd = 0 Then
    Waitms 50
  Else
    Waitms 200
  End If
  Pwmd = 0
  Ctrlmd1 = 0
  Ctrlmd2 = 0
  Print # 1 , Str(contador_a) ; Sentido
End If
***** Control de Motor Muñeca AntiHorario *****
If Flag6 = 1 Then

```

```

Contador_a = 0
Print # 1 , "H"          '???'
Flag6 = 0
Ctrlmd1 = 0
Ctrlmd2 = 1
Pwmd = 1
If Swd = 0 Then
    Waitms 50
Else
    Waitms 200
End If
Pwmd = 0
Ctrlmd1 = 0
Ctrlmd2 = 0
Print # 1 , Str(contador_a) ; Sentido
End If
***** Control de Motor Codo Horario *****
If Flag7 = 1 Then
    Contador_a = 0
    Print # 1 , "I"
    Flag7 = 0
    Ctrlme1 = 0
    Ctrlme2 = 1
    Pwme = 1
    If Swe = 0 Then
        Waitms 50
    Else
        Waitms 200
    End If
    Pwme = 0
    Ctrlme1 = 0
    Ctrlme2 = 0
    Print # 1 , Str(contador_a) ; Sentido
End If
***** Control de Motor Codo AntiHorario *****
If Flag8 = 1 Then
    Contador_a = 0
    Print # 1 , "J"
    Flag8 = 0
    Ctrlme1 = 1
    Ctrlme2 = 0
    Pwme = 1
    If Swe = 0 Then
        Waitms 50
    Else
        Waitms 200
    End If
    Pwme = 0
    Ctrlme1 = 0
    Ctrlme2 = 0
    Print # 1 , Str(contador_a) ; Sentido
End If
***** Control de Motor Hombro AntiHorario *****
If Flag9 = 1 Then
    Contador_a = 0
    Print # 1 , "K"
    Flag9 = 0
    Ctrlmf1 = 0
    Ctrlmf2 = 1

```

```

Pwmf = 1
If Swf = 0 Then
    Waitms 50
Else
    Waitms 200
End If
Pwmf = 0
Ctrlmf1 = 0
Ctrlmf2 = 0
Print # 1 , Str(contador_a) ; Sentido
End If
***** Control de Motor Hombro Horario *****
If Flag10 = 1 Then
    Contador_a = 0
    Print # 1 , "L"
    Flag10 = 0
    Ctrlmf1 = 1
    Ctrlmf2 = 0
    Pwmf = 1
    If Swf = 0 Then
        Waitms 50
    Else
        Waitms 200
    End If
    Pwmf = 0
    Ctrlmf1 = 0
    Ctrlmf2 = 0
    Print # 1 , Str(contador_a) ; Sentido
End If
*****
Rec_isr:
Led1 = 1                                'TECH
    Disable Urxc1
    Disable Urxc
    Input Serial1 Noecho
    Temporal1 = Mid(serial1 , 1 , 1)
    Select Case Temporal1
        Case "C"
            Flag1 = 1
            Print "J"
        Case "D"
            Flag2 = 1
            Print "I"
        Case "E"
            Flag3 = 1
            Print "H"
        Case "F"
            Flag4 = 1
            Print "G"
        Case "G"
            Flag5 = 1
            Print "F"
        Case "H"
            Flag6 = 1
            Print "E"
        Case "I"
            Flag7 = 1
            Print "D"
        Case "J"

```

```

        Flag8 = 1
    '    Print "C"
    Case "K"
        Flag9 = 1
    '    Print "B"
    Case "L"
        Flag10 = 1
    '    Print "A"
    '    Case "B"
    '        Flag11 = 1
    '        Print "K"
    '    Case "A"
    '        Flag12 = 1
    '        Print "L"
    End Select

    Do
        B = Inkey()
    Loop Until B = 0
    '    Wait 1
    Led1 = 0
    Enable Urxc
    Enable Urxc1
    Return
    *****
Rec_isr1:                                'USB
    Disable Urxc1
    Disable Urxc
    Input # 1 , Serial1 Noecho
    Temporal1 = Mid(serial1 , 1 , 1)
    '    Temporal2 = Mid(serial1 , 2 , 1)
    '    Temporal3 = Mid(serial1 , 2 , 4)
    '    Datox1 = ASC(temporal2)
    Datox1 = Val(temporal3)
    '    Datox1 = Datox * 1

    Select Case Temporal1
    Case "C"
        Flag1 = 1
    '    Datox1 = Datox * 1
    Case "D"
        Flag2 = 1
    '    Datox1 = Datox * 1
    Case "E"
        Flag3 = 1
    '    Datox1 = Datox * 1
    Case "F"
        Flag4 = 1
    '    Datox1 = Datox * 1
    Case "G"
        Flag5 = 1
    '    Datox1 = Datox * 1
    Case "H"
        Flag6 = 1
    '    Datox1 = Datox * 1
    Case "I"
        Flag7 = 1
    '    Datox1 = Datox * 1
    Case "J"

```

```

    Flag8 = 1
'   Datox1 = Datox * 1
    Case "K"                                'AH BASE
      Flag9 = 1
'   Datox1 = Datox * 1
    Case "L"                                'H BASE
      Flag10 = 1
'   Datox1 = Datox * 1
    Case "A"
      Flag11 = 1
'   Datox1 = Datox * 1
    Case "B"
      Flag12 = 1
'   Datox1 = Datox * 1
    Case "R"
      Flag13 = 1
End Select

```

```

Do
  B = Inkey(#1)
Loop Until B = 0

```

```

Enable Urxc
Enable Urxc1
Return

```

Resetm:

```

While Swa = 1
  Ctrlma1 = 1
  Ctrlma2 = 0
  Pwma = 1
Wend
Ctrlma1 = 0
Ctrlma2 = 0
Pwma = 0

```

```

-----
While Swc = 1
  Ctrlmc1 = 1
  Ctrlmc2 = 0
  Pwmc = 1
Wend
Ctrlmc1 = 0
Ctrlmc2 = 0
Pwmc = 0

```

```

-----
While Swd = 1
  Ctrlmd1 = 1
  Ctrlmd2 = 0
  Pwmd = 1
Wend
Ctrlmd1 = 0
Ctrlmd2 = 0
Pwmd = 0

```

```

-----
While Swb = 1
  Ctrlmb1 = 1
  Ctrlmb2 = 0
  Pwmb = 1
Wend

```

```

Ctrlmb1 = 0
Ctrlmb2 = 0
Pwmb = 0
'-----
While Swf = 1
  Ctrlmf1 = 1
  Ctrlmf2 = 0
  Pwmf = 1
Wend
Ctrlmf1 = 0
Ctrlmf2 = 0
Pwmf = 0
'-----
While Swe = 1
  Ctrlme1 = 1
  Ctrlme2 = 0
  Pwme = 1
Wend
Ctrlme1 = 0
Ctrlme2 = 0
Pwme = 0
Return

*****
Int7_int:
  Disable Int7
  Contador_a = Contador_a + 1
  If Sb4 = 0 Then
    Sentido = "AH"
  Else
    Sentido = "H"
  End If
  Enable Int7
Return
'=====
Int6_int:
  Disable Int6
  Contador_a = Contador_a + 1
  If Sb3 = 0 Then
    Sentido = "AH"
  Else
    Sentido = "H"
  End If
  Enable Int6
Return
'=====
Int1_int:
  Disable Int1
  Contador_a = Contador_a + 1
  If Sb2 = 0 Then
    Sentido = "AH"
  Else
    Sentido = "H"
  End If
  Enable Int1
Return
'=====
Int0_int:
  Disable Int0

```

```

Contador_a = Contador_a + 1
If Sb1 = 0 Then
  Sentido = "AH"
Else
  Sentido = "H"
End If
Enable Int0
Return
*****
End

```

## Anexo 2 Programa realizado en Bascom para el control de los robots mediante un teach pad

```

'-----
'name          : Tech5.bas
'copyright     : (c) 2013, MCS Electronics
'purpose       : Control Brazo
'micro         : Mega644PA
'-----

$sim
$regfile = "m644def.dat"          ' specify the used micro
$crystal = 8000000                ' used crystal frequency
$crystal = 16000000              ' used crystal frequency
$baud = 9600                      ' use baud rate
$hwstack = 32                    ' default use 32 for the hardware stack
$swstack = 10                    ' default use 10 for the SW stack
$framesize = 40                  ' default use 40 for the frame space
$lib "mcsbyte.lbx"

$initmicro

Ddrc.1 = 1                        'Luz de Fondo (BK)
Portc.1 = 1                       'BK=1

***** Declaración de Funciones y Subrutinas *****
*****
Declare Sub Writebyte512a(address As Word , Value As Byte)
Declare Function Readbyte512a(address As Word) As Byte

Declare Sub Writebyte512b(address As Word , Value As Byte)
Declare Function Readbyte512b(address As Word) As Byte

Dim B As Byte , Adr1 As Word , Bb As Byte
Dim Hdr As Byte , L As Byte , H As Byte

***** INTERRUPCION de serial *****
Dim D1 As String * 3
Dim Control As Byte

On Urxr Rec_isr                   'Define subrutina de Interrupcion Serial

*****

'Subrutinas para el control del GLCD
$lib "glcdKS108.lib"
'Configuracion de los pines del GLCD

```



Config Graphlcd = 128 \* 64sed , Dataport = Porta , Controlport = Portc , Ce = 3 , Ce2 = 4 , Cd = 5 ,  
Rd = 6 , Reset = 2 , Enable = 7

'El dataport es el portico donde esta conectado las lineas de Datos del GLCD

'El controlport es el portico donde estan los pines utilizados para el control del GLCD

'CE=CS1 Chip select 1

'CE2=CS2 Chip select 2

'CD=Data/instruction (RS)

'RD=Read(R/W)

'RESET = reset

'ENABLE= Chip Enable

\*\*\*\*\* DS1307 RTC Y MEMORIAS 24LC512 \*\*\*\*\*

Config Sda = Portd.4

'Configura I2C pin Datos PORTD.4

Config Scl = Portd.3

'Configura I2C pin Reloj PORTD.3

'Direccion del DS1307

Const Ds1307w = &HD0

'Constante escritura DS1307

Const Ds1307r = &HD1

'Constante lectura DS1307

'address of 24LC512(A0=0 A1=0 A2=0)

Const M24lc512aw = &HA0

'Direccion de Memoria1

Const M24lc512ar = &HA1

'address of 24LC512(A0=1 A1=0 A2=0)

Const M24lc512bw = &HA2

'Direccion de Memoria2

Const M24lc512br = &HA3

'Declara variables necesarias para el DS1307

Dim Dias As Byte

Dim Segu As Byte

Dim Minu As Byte

Dim Hora As Byte

Dim Diam As Byte

Dim Mes As Byte

Dim Anio As Byte

Dim Dias1 As String \* 3

Dim Mes1 As String \* 3

Dim Texto1 As String \* 5 , Texto2 As String \* 2 , Texto3 As String \* 2

Dim Texto4 As String \* 30 , Texto5 As String \* 15

Dim Texto6 As String \* 20 , Texto7 As String \* 20 , Texto8 As String \* 2

Dim Texto9 As String \* 10 , Texto10 As String \* 1

Dim Texto11 As String \* 10 , Texto12 As String \* 10

Dim Temporal1 As String \* 1

Dim Serial1 As String \* 10

'Const Retardo = 500

'Const Retardo1 = 2200

'Const Rebote = 60

Const Filat = 4

Const Filad = 7

\*\*\*\*\* DEFINICIONES INICIALES PARA CONFIGURAR EL TECLADO \*\*\*\*\*

Ddrb.0 = 0

Portb.0 = 1

Ddrb.1 = 0

Portb.1 = 1

Ddrb.2 = 0

Portb.2 = 1  
Ddrb.3 = 0  
Portb.3 = 1

Ddrb.4 = 1  
Portb.4 = 1  
Ddrb.5 = 1  
Portb.5 = 1  
Ddrb.6 = 1  
Portb.6 = 1  
Ddrb.7 = 1  
Portb.7 = 1

F1 Alias Portb.4 'nombres para los pines de las filas  
F2 Alias Portb.5  
F3 Alias Portb.6  
F4 Alias Portb.7

C1 Alias Pinb.0 'nombres para los pines de las columnas  
C2 Alias Pinb.1  
C3 Alias Pinb.2  
C4 Alias Pinb.3

\*\*\*\*\* DEFINICIONES INICIALES PARA 74LS243 \*\*\*\*\*

'Ddrc.0 = 1  
'Portc.0 = 1  
'Ddrd.6 = 1  
'Portd.6 = 1  
'Gab Alias Portc.0  
'Gba Alias Portd.6

\*\*\*\*\*

\*\*\*\*\*

\* LED1 \*

\*\*\*\*\*

Ddrd.5 = 1  
Portd.5 = 0  
Led1 Alias Portd.5

\*\*\*\*\*

'Variables  
Dim I As Byte  
Dim J As Byte  
Dim X As Word  
Dim X1 As Word  
Dim X2 As Word  
Dim Y As Byte , Z As Byte  
Dim W As Word  
Dim Z1 As Byte  
Dim Dir1 As Word  
Dim Dir2 As Word  
Dim Dir3 As Word  
Dim Flag1 As Bit  
Dim Flag2 As Bit  
Dim Flag3 As Bit  
Dim Flag4 As Bit  
Dim Flag5 As Bit  
Dim Flag6 As Bit  
Dim Flag7 As Bit

Dim Flag22 As Bit  
Dim Flag24 As Bit  
Dim Flag25 As Bit  
Dim Cuenta As Byte

Dim Tecla As Byte  
Dim Contador As Word  
Dim Contadorb As Word  
Dim Contador1 As Word  
Dim Contador2 As Word  
Dim Contador3 As Word  
Dim Contador4 As Word  
Dim Contador5 As Word  
Dim Numerop As Word  
Dim Numerop1 As Word  
Dim Pantalla As Word  
Dim Tecla1 As Byte  
Dim Clientes As Word  
Dim Clientesb As Word  
Dim Calcular As Single  
Dim Total As Single  
Dim Total1 As Single  
Dim B0 As Byte  
Dim B1 As Byte  
Dim Keyread As Byte

Dim Senluz As Byte  
Dim Contmenu As Byte  
Dim Forma1 As Byte , Forma2 As Byte , Forma3 As Byte , Forma4 As Byte  
Dim Forma5 As Byte , Forma6 As Byte , Forma7 As Byte , Forma8 As Byte  
Dim Forma9 As Byte , Forma10 As Byte , Forma11 As Byte

Dim Forma51 As Byte , Forma61 As Byte , Forma71 As Byte , Forma81 As Byte  
Dim Forma91 As Byte , Forma101 As Byte

Dim Forma52 As Byte , Forma62 As Byte , Forma72 As Byte , Forma82 As Byte  
Dim Forma92 As Byte , Forma102 As Byte

Dim Forma53 As Byte , Forma63 As Byte , Forma73 As Byte , Forma83 As Byte  
Dim Forma93 As Byte , Forma103 As Byte

Dim Columna As Byte  
Dim Fila As Byte  
Dim Sensorir As Byte  
Dim Sensorsonido As Byte  
Dim Menu(4) As String \* 25  
Dim Temporal As String \* 1  
Dim Cliente As String \* 60  
Dim Producto As String \* 40  
Dim Precio(100) As String \* 6

Dim Valor1 As String \* 10  
Dim Valor11 As Word  
Dim Valor2 As String \* 10  
Dim Valor22 As Word  
Dim Valor3 As String \* 10  
Dim Valor33 As Word  
Dim Valor4 As String \* 10

Dim Valor44 As Word  
Dim Valor5 As String \* 10  
Dim Valor55 As Word  
Dim Valor6 As String \* 10  
Dim Valor66 As Word

Dim Elegir As Byte  
Dim Lista1 As Byte  
Dim Resto As Word  
Dim Pantallas As Word  
Dim Pantallab As Word  
Dim Pantallasc As Word  
Dim Pantallasd As Word  
Dim Producto1 As String \* 30  
Dim Producto2 As String \* 30  
Dim Tipo1 As Byte , Tipo2 As Byte , Tipo3 As Byte , Tipo4 As Byte  
Dim Mover As Byte  
Dim Cantidad As Byte  
Dim Items(100) As Byte  
Dim Revisar(100) As Byte  
Dim Registro As Word  
Dim Registro1 As Word  
Dim Clave1 As String \* 6

Dim Bandera10 As Bit  
Dim Bandera11 As Bit  
Dim Bandera12 As Bit  
Dim Cuenta1 As Byte  
Dim Clave1(12) As String \* 1  
Dim Clavea1(5) As String \* 1  
Dim Clavea2(80) As String \* 1  
Dim Clavea3(20) As Byte

Const Retardo = 3  
Dim Band29 As Bit                   'Tiempo en seg.  
Dim Sentido1 As Byte  
Dim Sentido2 As Byte  
Dim Sentido3 As Byte  
Dim Sentido4 As Byte  
Dim Sentido5 As Byte  
Dim Sentido6 As Byte  
'Especifica el tipo de letra a utilizar  
'SetFont Font8x8

Contmenu = 1                   'Posicion inicial del menu  
Flag1 = 0  
Flag2 = 0  
Flag3 = 0  
Flag24 = 0  
Flag25 = 0  
J = 0  
Contador = 0  
Numerop = 0  
Bandera12 = 0  
'+++++-----  
'Dias = 7                           'dia de la semana(D=1,L=2,M=3,M=4,J=5,V=6,S=7)  
'Diam = 16  
'Mes = 4  
'Anio = 11  
'Gosub Setdate

```

'Segu = 0
'Minu = 50
'Hora = 11
'Gosub Settime
'++++++
'Retardo para inicializar GLCD

'Readeeprom X , 2
'Clientesb = X
'Clientes = X - 1
'Readeeprom X , 8
'Lista1 = X - 1
'Pantallas = X \ 3
'Resto = X Mod Pantallas
'Pantallasc = Pantallas - 1
'Pantallasd = Pantallas
'If Resto <> 0 Then
' Pantallasc = Pantallas
' Pantallasd = Pantallas + 1
'End If
'Pantalla = 0
'Gosub Traerprecio
'Gosub Limpiarprecios
Setfont Font8x8
"Lcdat 1 , 1 , Resto ; "a"
"Wait 2
'Readeeprom X , 6
'Registro = X

Waitms 500
Cls
***** Animación Inicial *****
For X = 0 To 5
  Led1 = 1
  Waitms 200
  Led1 = 0
  Waitms 200
Next X

Do
  B = Inkey()
Loop Until B = 0

Enable Interrupts
Enable Urxc

'-----
Gosub Vergraficos
'-----
Gosub Vercaratula
'-----

Cls

'Do
' Gosub Lista
'Loop

Do
  If Bandera12 = 0 Then

```

```

    Gosub Vercaratula
Else
    Gosub Getdatetime
    Setfont Font16x16
    Lcdat 1 , 25 , "RELOJ"
    Lcdat Filat , 1 , Texto4
    Setfont Font8x8
    Lcdat Filad , 6 , Texto5 ; ""
End If

Tecla = 255
Gosub Barrido
Gosub Rebote
Select Case Tecla
    Case 1
    Case 2
    Case 3
    Case 4
    Case 7
    Case 10
        Gosub Clavemaster
    Case 11
        Toggle Bandera12
    Cls
    Case 12
    Case 13
    Case 14
End Select
Loop

***** Subrutina Leer Teclas *****
Clavemaster:
    Bandera10 = 1
    Cuenta = 1
    Cuental = 0
    Cls
    Box(1 , 1) -(126 , 62) , 1
    Lcdat 3 , 5 , " INGRESE CLAVE "

    While Bandera10 = 1
        Tecla = 255
        Gosub Barrido
        Gosub Rebote
        Select Case Tecla
            If Cuenta < 5 And Tecla < 10 Then
                Select Case Cuenta
                    Case 1
                        Lcdat 6 , 50 , "*"
                        Clave1 = ""
                        Clave1 = Str(tecla)
                    Case 2
                        Lcdat 6 , 60 , "*"
                        Clave1 = Clave1 + Str(tecla)
                    Case 3
                        Lcdat 6 , 70 , "*"
                        Clave1 = Clave1 + Str(tecla)
                    Case 4
                        Lcdat 6 , 80 , "*"
                        Clave1 = Clave1 + Str(tecla)

```

```

        End Select
        Cuenta = Cuenta + 1
    End If
    Case 15
        If Cuenta = 5 Then
            'Gosub Manual
            Gosub Lista
            Bandera10 = 0
        Else
            Lcdat 6 , 30 , "Clave debe "
            Lcdat 7 , 5 , "tener 4 Digitos"
            Wait 3
            Lcdat 6 , 30 , "      "
            Lcdat 7 , 5 , "      "
            Cuenta = 1
        End If
    Case 14
        Cuenta = 1                'Borra dato mal ingresado
        Lcdat 6 , 30 , "      "
        Lcdat 7 , 5 , "      "
    End Select
Wend
Return
*****

Vercaratula:
    Setfont Font8x6
    Box(1 , 1 )-(126 , 62) , 1
    Lcdat 2 , 20 , "UNIV.POLITECNICA"
    Lcdat 3 , 20 , " " ; Chr(34) ; "SALESIANA" ; Chr(34) ; " "
    Lcdat 5 , 20 , " BRAZO      "
    Lcdat 6 , 20 , " ANTROFOMORFICO "
    Lcdat 7 , 20 , " XR-3      "
Return
*****

Vermenu:
' Cls
    Setfont Font8x8
    Box(1 , 1 )-(126 , 62) , 1
    Lcdat 2 , 10 , "MENU PRINCIPAL"
    Lcdat 4 , 10 , "B= MANUAL      "
    Lcdat 5 , 10 , "C= AUTOMATICO "
    Lcdat 7 , 10 , "*= SALIR      "
' Lcdat 7 , 20 , " XR-3      "
Return
*****

Manual:
    Cls
    Setfont Font8x6
' Box(1 , 1 )-(126 , 62) , 1
    Lcdat 1 , 1 , "MOTOR F" , Forma5
    Lcdat 2 , 1 , "MOTOR C" , Forma6
    Lcdat 3 , 1 , "MOTOR E" , Forma7
    Lcdat 4 , 1 , "MOTOR D" , Forma8
    Lcdat 5 , 1 , "MOTOR B" , Forma9
    Lcdat 6 , 1 , "MOTOR A" , Forma10
    Lcdat 8 , 1 , "RESET " , Forma11
    Lcdat 8 , 80 , "*= SALIR"
Return
*****

```

```

Ingresaclave:
Box(1, 1) -(126, 62), 1
Lcdat 3, 5, " INGRESE CLAVE "
  Cantidad = 1
  Flag22 = 1
  While Flag22 = 1
    Tecla = 255
    Gosub Barrido
    Gosub Rebote
    Select Case Tecla
      If Cantidad < 4 And Tecla < 10 Then
        Select Case Cantidad
          Case 1
            Lcdat 6, 50, "*"
            Clave1 = ""
            Clave1 = Str(tecla)
          Case 2
            Lcdat 6, 60, "*"
            Clave1 = Clave1 + Str(tecla)
          Case 3
            Lcdat 6, 70, "*"
            Clave1 = Clave1 + Str(tecla)
        End Select
        Cantidad = Cantidad + 1
      End If
    Case 15
      If Cantidad = 4 Then
        Flag22 = 0
      Else
        Lcdat 6, 30, "Clave debe"
        Lcdat 7, 5, "tener 3 Digitos"
        Wait 3
        Lcdat 6, 30, "      "
        Lcdat 7, 5, "      "
        Cantidad = 1
      End If
    Case 14
      Cantidad = 1
      Lcdat 6, 30, "      "
      Lcdat 7, 5, "      "
      'Borra dato mal ingresado
    End Select
  Wend
  Cls
Return
*****

Vergraficos:
  Cls
  Showpic 0, 0, Caratula1
  Wait 3
  Showpic 0, 0, Caratula2
  Wait 3
  Cls
Return
*****

=====
=====

Lista:
  Sentido1 = 0
  Sentido2 = 0

```



```
Sentido3 = 0
Sentido4 = 0
Sentido5 = 0
```

```
Flag4 = 1
Forma5 = 1
Forma6 = 0
Forma7 = 0
Forma8 = 0
Forma9 = 0
Forma10 = 0
Forma11 = 0
```

```
Gosub Resetformas
Columna = 1
Fila = 0
Gosub Manual
Gosub Sentidohah
Gosub Vueltas
Gosub Enters
```

```
While Flag4 = 1
  Tecla = 255
  Gosub Barrido
  Gosub Rebote
  Select Case Tecla
    Case 1
      If Fila = 1 Then
        Select Case Columna
          Case 1
            Toggle Sentido1
          Case 2
            Toggle Sentido2
          Case 3
            Toggle Sentido3
          Case 4
            Toggle Sentido4
          Case 5
            Toggle Sentido5
          Case 6
            Toggle Sentido6
        End Select
        Gosub Sentidohah
      End If
    Case 2
      If Fila = 2 Then
        Select Case Columna
          Case 1
            Valor11 = Valor11 + 1
            If Valor11 > 810 Then
              Valor11 = 0
            End If
          Case 2
            Valor22 = Valor22 + 1
            If Valor22 > 1570 Then
              Valor22 = 0
            End If
          Case 3
            Valor33 = Valor33 + 1
```

```

    If Valor33 > 400 Then
        Valor33 = 0
    End If
Case 4
    Valor44 = Valor44 + 1
    If Valor44 > 1200 Then
        Valor44 = 0
    End If
Case 5
    Valor55 = Valor55 + 1
    If Valor55 > 2000 Then
        Valor55 = 0
    End If
Case 6
    Valor66 = Valor66 + 1
    If Valor66 > 10 Then
        Valor66 = 0
    End If
End Select
Gosub Vueltas
End If
Case 3
    If Fila = 2 Then
        Select Case Columna
            Case 1
                Valor11 = Valor11 - 1
                If Valor11 = 65535 Then
                    Valor11 = 810
                End If
            Case 2
                Valor22 = Valor22 - 1
                If Valor22 = 65535 Then
                    Valor22 = 1570
                End If
            Case 3
                Valor33 = Valor33 - 1
                If Valor33 = 65535 Then
                    Valor33 = 400
                End If
            Case 4
                Valor44 = Valor44 - 1
                If Valor44 = 65535 Then
                    Valor44 = 1200
                End If
            Case 5
                Valor55 = Valor55 - 1
                If Valor55 = 65535 Then
                    Valor55 = 2000
                End If
            Case 6
                Valor66 = Valor66 - 1
                If Valor66 = 65535 Then
                    Valor66 = 10
                End If
        End Select
        Gosub Vueltas
    End If
Case 4
    Fila = Fila + 1

```

```
If Fila > 3 Then
  Fila = 1
End If
```

```
Gosub Resetformas
```

```
Select Case Fila
```

```
Case 1
```

```
  Select Case Columna
```

```
    Case 1
```

```
      Forma51 = 1
```

```
      Forma52 = 0
```

```
      Forma53 = 0
```

```
    Case 2
```

```
      Forma61 = 1
```

```
      Forma62 = 0
```

```
      Forma63 = 0
```

```
    Case 3
```

```
      Forma71 = 1
```

```
      Forma72 = 0
```

```
      Forma73 = 0
```

```
    Case 4
```

```
      Forma81 = 1
```

```
      Forma82 = 0
```

```
      Forma83 = 0
```

```
    Case 5
```

```
      Forma91 = 1
```

```
      Forma92 = 0
```

```
      Forma93 = 0
```

```
    Case 6
```

```
      Forma101 = 1
```

```
      Forma102 = 0
```

```
      Forma103 = 0
```

```
  End Select
```

```
Case 2
```

```
  Select Case Columna
```

```
    Case 1
```

```
      Forma51 = 0
```

```
      Forma52 = 1
```

```
      Forma53 = 0
```

```
    Case 2
```

```
      Forma61 = 0
```

```
      Forma62 = 1
```

```
      Forma63 = 0
```

```
    Case 3
```

```
      Forma71 = 0
```

```
      Forma72 = 1
```

```
      Forma73 = 0
```

```
    Case 4
```

```
      Forma81 = 0
```

```
      Forma82 = 1
```

```
      Forma83 = 0
```

```
    Case 5
```

```
      Forma91 = 0
```

```
      Forma92 = 1
```

```
      Forma93 = 0
```

```
    Case 6
```

```
      Forma101 = 0
```

```
      Forma102 = 1
```

```

        Forma103 = 0
    End Select
Case 3
    Select Case Columna
        Case 1
            Forma51 = 0
            Forma52 = 0
            Forma53 = 1
        Case 2
            Forma61 = 0
            Forma62 = 0
            Forma63 = 1
        Case 3
            Forma71 = 0
            Forma72 = 0
            Forma73 = 1
        Case 4
            Forma81 = 0
            Forma82 = 0
            Forma83 = 1
        Case 5
            Forma91 = 0
            Forma92 = 0
            Forma93 = 1
        Case 6
            Forma101 = 0
            Forma102 = 0
            Forma103 = 1
    End Select
End Select
Gosub Sentidohah
Gosub Vueltas
Gosub Enters
Case 5
    Columna = Columna + 1
    If Columna = 8 Then
        Columna = 1
    End If
    Select Case Columna
        Case 1
            Forma5 = 1
            Forma6 = 0
            Forma7 = 0
            Forma8 = 0
            Forma9 = 0
            Forma10 = 0
            Forma11 = 0
        Case 2
            Forma5 = 0
            Forma6 = 1
            Forma7 = 0
            Forma8 = 0
            Forma9 = 0
            Forma10 = 0
            Forma11 = 0
        Case 3
            Forma5 = 0
            Forma6 = 0
            Forma7 = 1

```

```

    Forma8 = 0
    Forma9 = 0
    Forma10 = 0
    Forma11 = 0
Case 4
    Forma5 = 0
    Forma6 = 0
    Forma7 = 0
    Forma8 = 1
    Forma9 = 0
    Forma10 = 0
    Forma11 = 0
Case 5
    Forma5 = 0
    Forma6 = 0
    Forma7 = 0
    Forma8 = 0
    Forma9 = 1
    Forma10 = 0
    Forma11 = 0
Case 6
    Forma5 = 0
    Forma6 = 0
    Forma7 = 0
    Forma8 = 0
    Forma9 = 0
    Forma10 = 1
    Forma11 = 0
Case 7
    Forma5 = 0
    Forma6 = 0
    Forma7 = 0
    Forma8 = 0
    Forma9 = 0
    Forma10 = 0
    Forma11 = 1
End Select
Gosub Resetformas
Fila = 0
Gosub Manual
Gosub Sentidohah
Gosub Vueltas
Gosub Enters
Case 14
    Flag4 = 0
    Cls
Case 15
    If Fila = 3 Or Columna = 7 Then
        Select Case Columna
            Case 1
                Print "1"
            Case 2
                Print "2"
            Case 3
                Print "3"
            Case 4
                Print "4"
            Case 5
                Print "5"

```

```

        Case 6
            Print "6"
        Case 7
            Print "7"
    End Select
End If
End Select
Wend
Return

```

```

=====
=====
Sentidohah:

```

```

    Setfont Font8x8
    If Sentido1 = 0 Then
        Lcdat 1 , 50 , " H" , Forma51
    Else
        Lcdat 1 , 50 , "AH" , Forma51
    End If
    If Sentido2 = 0 Then
        Lcdat 2 , 50 , " H" , Forma61
    Else
        Lcdat 2 , 50 , "AH" , Forma61
    End If
    If Sentido3 = 0 Then
        Lcdat 3 , 50 , " H" , Forma71
    Else
        Lcdat 3 , 50 , "AH" , Forma71
    End If
    If Sentido4 = 0 Then
        Lcdat 4 , 50 , " H" , Forma81
    Else
        Lcdat 4 , 50 , "AH" , Forma81
    End If
    If Sentido5 = 0 Then
        Lcdat 5 , 50 , " H" , Forma91
    Else
        Lcdat 5 , 50 , "AH" , Forma91
    End If
    If Sentido6 = 0 Then
        Lcdat 6 , 50 , " H" , Forma101
    Else
        Lcdat 6 , 50 , "AH" , Forma101
    End If

```

```

Return

```

```

=====
=====
Vueltas:

```

```

    Texto1 = Str(valor11)
    Valor1 = Format(texto1 , "0000" )
    Texto1 = Str(valor22)
    Valor2 = Format(texto1 , "0000" )
    Texto1 = Str(valor33)
    Valor3 = Format(texto1 , "0000" )
    Texto1 = Str(valor44)
    Valor4 = Format(texto1 , "0000" )
    Texto1 = Str(valor55)
    Valor5 = Format(texto1 , "0000" )
    Texto1 = Str(valor66)
    Valor6 = Format(texto1 , "0000" )

```

```

Setfont Font8x8
If Sentido1 = 0 Then
  Lcdat 1 , 75 , Valor1 , Forma52
Else
  Lcdat 1 , 75 , Valor1 , Forma52
End If
If Sentido2 = 0 Then
  Lcdat 2 , 75 , Valor2 , Forma62
Else
  Lcdat 2 , 75 , Valor2 , Forma62
End If
If Sentido3 = 0 Then
  Lcdat 3 , 75 , Valor3 , Forma72
Else
  Lcdat 3 , 75 , Valor3 , Forma72
End If
If Sentido4 = 0 Then
  Lcdat 4 , 75 , Valor4 , Forma82
Else
  Lcdat 4 , 75 , Valor4 , Forma82
End If
If Sentido5 = 0 Then
  Lcdat 5 , 75 , Valor5 , Forma92
Else
  Lcdat 5 , 75 , Valor5 , Forma92
End If
If Sentido6 = 0 Then
  Lcdat 6 , 75 , Valor6 , Forma102
Else
  Lcdat 6 , 75 , Valor6 , Forma102
End If

```

Return

'=====

Enters:

```

Setfont Font8x6
If Sentido1 = 0 Then
  Lcdat 1 , 115 , "E1" , Forma53
Else
  Lcdat 1 , 115 , "E1" , Forma53
End If
If Sentido2 = 0 Then
  Lcdat 2 , 115 , "E2" , Forma63
Else
  Lcdat 2 , 115 , "E2" , Forma63
End If
If Sentido3 = 0 Then
  Lcdat 3 , 115 , "E3" , Forma73
Else
  Lcdat 3 , 115 , "E3" , Forma73
End If
If Sentido4 = 0 Then
  Lcdat 4 , 115 , "E4" , Forma83
Else
  Lcdat 4 , 115 , "E4" , Forma83
End If
If Sentido5 = 0 Then
  Lcdat 5 , 115 , "E5" , Forma93

```

```

Else
  Lcdat 5 , 115 , "E5" , Forma93
End If
If Sentido6 = 0 Then
  Lcdat 6 , 115 , "E6" , Forma103
Else
  Lcdat 6 , 115 , "E6" , Forma103
End If

```

```
Return
```

```

=====
=====

```

```
Resetformas:
```

```

Forma51 = 0
Forma61 = 0
Forma71 = 0
Forma81 = 0
Forma91 = 0
Forma101 = 0

```

```

Forma52 = 0
Forma62 = 0
Forma72 = 0
Forma82 = 0
Forma92 = 0
Forma102 = 0

```

```

Forma53 = 0
Forma63 = 0
Forma73 = 0
Forma83 = 0
Forma93 = 0
Forma103 = 0

```

```
Return
```

```

=====
=====

```

```
***** Subrutina de Barrido de Teclas *****
```

```
Barrido:
```

```

F1 = 0
If C1 = 0 Then : Tecla = 1 : Return : End If
If C2 = 0 Then : Tecla = 2 : Return : End If
If C3 = 0 Then : Tecla = 3 : Return : End If
If C4 = 0 Then : Tecla = 10 : Return : End If
  F1 = 1 : F2 = 0
If C1 = 0 Then : Tecla = 4 : Return : End If
If C2 = 0 Then : Tecla = 5 : Return : End If
If C3 = 0 Then : Tecla = 6 : Return : End If
If C4 = 0 Then : Tecla = 11 : Return : End If
  F2 = 1 : F3 = 0
If C1 = 0 Then : Tecla = 7 : Return : End If
If C2 = 0 Then : Tecla = 8 : Return : End If
If C3 = 0 Then : Tecla = 9 : Return : End If
If C4 = 0 Then : Tecla = 12 : Return : End If
  F3 = 1 : F4 = 0
If C1 = 0 Then : Tecla = 14 : Return : End If
If C2 = 0 Then : Tecla = 0 : Return : End If
If C3 = 0 Then : Tecla = 15 : Return : End If
If C4 = 0 Then : Tecla = 13 : Return : End If
  F4 = 1
  Waitms 1

```



```

Return
***** Subrutina de Rebote de Teclas *****
Rebote:
  While C1 = 0 : Wend
  While C2 = 0 : Wend
  While C3 = 0 : Wend
  While C4 = 0 : Wend
  Waitms 1
Return
*****

```

```

***** Subrutina Cambiar el Tiempo *****

```

```

Reloj:
  Flag3 = 1
  Forma5 = 1
  Forma6 = 0
  Forma7 = 0
  Forma8 = 0
  Forma9 = 0
  Forma10 = 0
  Forma11 = 0
  Columna = 1
  Texto1 = Str(hora)
  Texto2 = Str(minu)
  Texto3 = Str(segu)
  Texto6 = Format(texto1 , "00" )
  Texto7 = Format(texto2 , "00" )
  Texto8 = Format(texto3 , "00" )
  Setfont Font16x16
  Lcdat Filat , 1 , Texto6 , Forma5
  Lcdat Filat , 33 , ":"
  Lcdat Filat , 49 , Texto7 , Forma6
  Lcdat Filat , 81 , ":"
  Lcdat Filat , 97 , Texto8 , Forma7

  Setfont Font8x8
  Texto1 = Str(diam)
  Texto2 = Str(anio)
  Texto9 = Format(texto1 , "00" )
  Texto10 = Format(texto2 , "00" )
  Lcdat Filad , 6 , Dias1 , Forma8
  Lcdat Filad , 30 , "-"
  Lcdat Filad , 38 , Texto9 , Forma9
  Lcdat Filad , 54 , "/"
  Lcdat Filad , 62 , Mes1 , Forma10
  Lcdat Filad , 86 , "/20"
  Lcdat Filad , 110 , Texto10 , Forma11

```

```

While Flag3 = 1
  Tecla = 255
  Gosub Barrido
  Gosub Rebote
  Select Case Tecla
  Case 1
    Select Case Columna
    Case 1
      Setfont Font16x16
      Hora = Hora + 1
      If Hora = 24 Then

```

```

    Hora = 0
End If
Texto1 = Str(hora)
Texto6 = Format(texto1 , "00" )
Lcdat Filat , 1 , Texto6 , Forma5
Case 2
Setfont Font16x16
Minu = Minu + 1
If Minu = 60 Then
    Minu = 0
End If
Texto2 = Str(minu)
Texto7 = Format(texto2 , "00" )
Lcdat Filat , 49 , Texto7 , Forma6
Case 3
Setfont Font16x16
Segu = Segu + 1
If Segu = 60 Then
    Segu = 0
End If
Texto3 = Str(segu)
Texto8 = Format(texto3 , "00" )
Lcdat Filat , 97 , Texto8 , Forma7

Case 4
Setfont Font8x8
Dias = Dias + 1
If Dias = 8 Then
    Dias = 1
End If
Select Case Dias
Case 1
    Dias1 = "Dom"           'Mostrar Día De La Semana
Case 2
    Dias1 = "Lun"          'Mostrar Día De La Semana
Case 3
    Dias1 = "Mar"          'Mostrar Día De La Semana
Case 4
    Dias1 = "Mie"          'Mostrar Día De La Semana
Case 5
    Dias1 = "Jue"          'Mostrar Día De La Semana
Case 6
    Dias1 = "Vie"          'Mostrar Día De La Semana
Case 7
    Dias1 = "Sab"          'Mostrar Día De La Semana
End Select
Lcdat Filad , 6 , Dias1 , Forma8
Case 5
Setfont Font8x8
Diam = Diam + 1
If Diam = 32 Then
    Diam = 1
End If
Texto3 = Str(diam)
Texto7 = Format(texto3 , "00" )
Lcdat Filad , 38 , Texto7 , Forma9
Case 6
Setfont Font8x8
Mes = Mes + 1

```

```

If Mes = 13 Then
  Mes = 1
End If
Select Case Mes
  Case 1
    Mes1 = "Ene"          'Mostrar el mes
  Case 2
    Mes1 = "Feb"         'Mostrar el mes
  Case 3
    Mes1 = "Mar"         'Mostrar el mes
  Case 4
    Mes1 = "Abr"         'Mostrar el mes
  Case 5
    Mes1 = "May"         'Mostrar el mes
  Case 6
    Mes1 = "Jun"         'Mostrar el mes
  Case 7
    Mes1 = "Jul"         'Mostrar el mes
  Case 8
    Mes1 = "Ago"         'Mostrar el mes
  Case 9
    Mes1 = "Sep"         'Mostrar el mes
  Case 10
    Mes1 = "Oct"         'Mostrar el mes
  Case 11
    Mes1 = "Nov"         'Mostrar el mes
  Case 12
    Mes1 = "Dic"         'Mostrar el mes
End Select
Lcdat Filad , 62 , Mes1 , Forma10
Case 7
  Setfont Font8x8
  Anio = Anio + 1
  If Anio = 100 Then
    Anio = 0
  End If
  Texto6 = Str(anio)
  Texto8 = Format(texto6 , "00" )
  Lcdat Filad , 110 , Texto8 , Forma11
End Select
Case 2
  Select Case Columna
  Case 1
    Setfont Font16x16
    Hora = Hora - 1
    If Hora = 255 Then
      Hora = 23
    End If
    Texto1 = Str(hora)
    Texto6 = Format(texto1 , "00" )
    Lcdat Filat , 1 , Texto6 , Forma5
  Case 2
    Setfont Font16x16
    Minu = Minu - 1
    If Minu = 255 Then
      Minu = 59
    End If
    Texto2 = Str(minu)
    Texto7 = Format(texto2 , "00" )

```

```

    Lcdat Filat , 49 , Texto7 , Forma6
Case 3
    Setfont Font16x16
    Segu = Segu - 1
    If Segu = 255 Then
        Segu = 59
    End If
    Texto3 = Str(segu)
    Texto8 = Format(texto3 , "00" )
    Lcdat Filat , 97 , Texto8 , Forma7
Case 4
    Setfont Font8x8
    Dias = Dias - 1
    If Dias = 0 Then
        Dias = 7
    End If
    Select Case Dias
        Case 1
            Dias1 = "Dom"           'Mostrar Día De La Semana
        Case 2
            Dias1 = "Lun"          'Mostrar Día De La Semana
        Case 3
            Dias1 = "Mar"          'Mostrar Día De La Semana
        Case 4
            Dias1 = "Mie"          'Mostrar Día De La Semana
        Case 5
            Dias1 = "Jue"          'Mostrar Día De La Semana
        Case 6
            Dias1 = "Vie"          'Mostrar Día De La Semana
        Case 7
            Dias1 = "Sab"          'Mostrar Día De La Semana
    End Select
    Lcdat Filad , 6 , Dias1 , Forma8
Case 5
    Setfont Font8x8
    Diam = Diam - 1
    If Diam = 0 Then
        Diam = 31
    End If
    Texto3 = Str(diam)
    Texto7 = Format(texto3 , "00" )
    Lcdat Filad , 38 , Texto7 , Forma9
Case 6
    Setfont Font8x8
    Mes = Mes - 1
    If Mes = 0 Then
        Mes = 12
    End If
    Select Case Mes
        Case 1
            Mes1 = "Ene"           'Mostrar el mes
        Case 2
            Mes1 = "Feb"          'Mostrar el mes
        Case 3
            Mes1 = "Mar"          'Mostrar el mes
        Case 4
            Mes1 = "Abr"          'Mostrar el mes
        Case 5
            Mes1 = "May"          'Mostrar el mes

```

```

Case 6
  Mes1 = "Jun"           'Mostrar el mes
Case 7
  Mes1 = "Jul"          'Mostrar el mes
Case 8
  Mes1 = "Ago"          'Mostrar el mes
Case 9
  Mes1 = "Sep"          'Mostrar el mes
Case 10
  Mes1 = "Oct"          'Mostrar el mes
Case 11
  Mes1 = "Nov"          'Mostrar el mes
Case 12
  Mes1 = "Dic"          'Mostrar el mes
End Select
Lcdat Filad , 62 , Mes1 , Forma10
Case 7
  Setfont Font8x8
  Anio = Anio - 1
  If Anio = 255 Then
    Anio = 99
  End If
  Texto6 = Str(anio)
  Texto8 = Format(texto6 , "00" )
  Lcdat Filad , 110 , Texto8 , Forma11
End Select
Case 3
  Columna = Columna + 1
  If Columna = 8 Then
    Columna = 1
  End If
  Select Case Columna
  Case 1
    Forma5 = 1
    Forma6 = 0
    Forma7 = 0
    Forma8 = 0
    Forma9 = 0
    Forma10 = 0
    Forma11 = 0
  Case 2
    Forma5 = 0
    Forma6 = 1
    Forma7 = 0
    Forma8 = 0
    Forma9 = 0
    Forma10 = 0
    Forma11 = 0
  Case 3
    Forma5 = 0
    Forma6 = 0
    Forma7 = 1
    Forma8 = 0
    Forma9 = 0
    Forma10 = 0
    Forma11 = 0
  Case 4
    Forma5 = 0
    Forma6 = 0

```

Forma7 = 0  
Forma8 = 1  
Forma9 = 0  
Forma10 = 0  
Forma11 = 0

Case 5

Forma5 = 0  
Forma6 = 0  
Forma7 = 0  
Forma8 = 0  
Forma9 = 1  
Forma10 = 0  
Forma11 = 0

Case 6

Forma5 = 0  
Forma6 = 0  
Forma7 = 0  
Forma8 = 0  
Forma9 = 0  
Forma10 = 1  
Forma11 = 0

Case 7

Forma5 = 0  
Forma6 = 0  
Forma7 = 0  
Forma8 = 0  
Forma9 = 0  
Forma10 = 0  
Forma11 = 1

End Select

SetFont Font16x16

Texto1 = Str(hora)

Texto2 = Str(minu)

Texto3 = Str(segu)

Texto6 = Format(texto1 , "00" )

Texto7 = Format(texto2 , "00" )

Texto8 = Format(texto3 , "00" )

Lcdat Filat , 1 , Texto6 , Forma5

Lcdat Filat , 33 , ":"

Lcdat Filat , 49 , Texto7 , Forma6

Lcdat Filat , 81 , ":"

Lcdat Filat , 97 , Texto8 , Forma7

SetFont Font8x8

Texto1 = Str(diam)

Texto2 = Str(ano)

Texto9 = Format(texto1 , "00" )

Texto10 = Format(texto2 , "00" )

Lcdat Filad , 6 , Dias1 , Forma8

Lcdat Filad , 30 , "-"

Lcdat Filad , 38 , Texto9 , Forma9

Lcdat Filad , 54 , "/"

Lcdat Filad , 62 , Mes1 , Forma10

Lcdat Filad , 86 , "/20"

Lcdat Filad , 110 , Texto10 , Forma11

Case 10

Gosub Setdate

Gosub Settime

```

    Flag3 = 0
  End Select
Wend

```

```
Return
```

```
***** Subrutina Leer el Tiempo *****
```

```

Getdatetime:
  I2cstart          'Genera inicio de I2C
  I2cwbyte Ds1307w 'Envia Constante escritura del DS1307
  I2cwbyte 0        'Direccion Inicial de memoria del DS1307
  I2cstart          'Genera inicio de I2C
  I2cwbyte Ds1307r 'Envia Constante lectura del DS1307
  I2crbyte Segu , Ack 'Lee los segundos
  I2crbyte Minu , Ack 'Lee los minutos
  I2crbyte Hora , Ack 'Lee la hora
  I2crbyte Dias , Ack 'Lee el dia de la
semana(D=1,L=2,M=3,M=4,J=5,V=6,S=7)
  I2crbyte Diam , Ack 'Lee dia del mes
  I2crbyte Mes , Ack 'Lee el mes del año
  I2crbyte Anio , Nack 'Lee el año
  I2cstop
  Segu = Makedec(segu)
  Minu = Makedec(minu)
  Hora = Makedec(hora)
  Dias = Makedec(dias)
  Diam = Makedec(diam)
  Mes = Makedec(mes)
  Anio = Makedec(anio)

```

```
Select Case Dias
```

```

  Case 1
    Dias1 = "Dom"          'Mostrar Día De La Semana
  Case 2
    Dias1 = "Lun"         'Mostrar Día De La Semana
  Case 3
    Dias1 = "Mar"         'Mostrar Día De La Semana
  Case 4
    Dias1 = "Mie"         'Mostrar Día De La Semana
  Case 5
    Dias1 = "Jue"         'Mostrar Día De La Semana
  Case 6
    Dias1 = "Vie"         'Mostrar Día De La Semana
  Case 7
    Dias1 = "Sab"         'Mostrar Día De La Semana
End Select

```

```
Select Case Mes
```

```

  Case 1
    Mes1 = "Ene"          'Mostrar el mes
  Case 2
    Mes1 = "Feb"          'Mostrar el mes
  Case 3
    Mes1 = "Mar"          'Mostrar el mes
  Case 4
    Mes1 = "Abr"          'Mostrar el mes
  Case 5
    Mes1 = "May"          'Mostrar el mes
  Case 6

```

```

    Mes1 = "Jun"                'Mostrar el mes
Case 7
    Mes1 = "Jul"               'Mostrar el mes
Case 8
    Mes1 = "Ago"              'Mostrar el mes
Case 9
    Mes1 = "Sep"              'Mostrar el mes
Case 10
    Mes1 = "Oct"              'Mostrar el mes
Case 11
    Mes1 = "Nov"              'Mostrar el mes
Case 12
    Mes1 = "Dic"              'Mostrar el mes
End Select

Texto1 = Str(hora)
Texto2 = Str(minu)
Texto3 = Str(segu)
Texto4 = Format(texto1 , "00" ) + ":" + Format(texto2 , "00" ) + ":" + Format(texto3 , "00" )
Texto1 = Str(anio)
Texto2 = Str(diam)
Texto5 = Dias1 + "-" + Format(texto2 , "00" ) + "/" + Mes1 + "/"20" + Format(texto1 , "00" )
Return
***** Subrutina Leer el Tiempo *****
Getdatetime1:
    I2cstart                    'Genera inicio de I2C
    I2cwbyte Ds1307w            'Envia Constante escritura del DS1307
    I2cwbyte 0                  'Direccion Inicial de memoria del DS1307
    I2cstart                    'Genera inicio de I2C
    I2cwbyte Ds1307r            'Envia Constante lectura del DS1307
    I2crbyte Segu , Ack         'Lee los segundos
    I2crbyte Minu , Ack         'Lee los minutos
    I2crbyte Hora , Ack         'Lee la hora
    I2crbyte Dias , Ack        'Lee el dia de la
semana(D=1,L=2,M=3,M=4,J=5,V=6,S=7)
    I2crbyte Diam , Ack         'Lee dia del mes
    I2crbyte Mes , Ack         'Lee el mes del año
    I2crbyte Anio , Nack        'Lee el año
    I2cstop
    Segu = Makedec(segu)
    Minu = Makedec(minu)
    Hora = Makedec(hora)
    Dias = Makedec(dias)
    Diam = Makedec(diam)
    Mes = Makedec(mes)
    Anio = Makedec(anio)

    Texto1 = Str(hora)
    Texto2 = Str(minu)
    Texto3 = Str(segu)
    Texto11 = Format(texto1 , "00" ) + Format(texto2 , "00" ) + Format(texto3 , "00" )
    Texto1 = Str(anio)
    Texto2 = Str(mes)
    Texto3 = Str(diam)
    Texto12 = Format(texto1 , "00" ) + Format(texto2 , "00" ) + Format(texto3 , "00" )
Return
***** Subrutina Grabar Fecha *****
Setdate:
    Dias = Makebcd(dias)

```



```

Diam = Makebcd(diam)
Mes = Makebcd(mes)
Anio = Makebcd(anio)
I2cstart                                'Genera inicio de I2C
I2cwbyte Ds1307w                        'Envia Constante escritura del DS1307
I2cwbyte 3                              'Direccion del dia del mes del DS1307
I2cwbyte Dias                           'Nuevo dia de la semana
I2cwbyte Diam                            'Nuevo dia del mes
I2cwbyte Mes                             'Nuevo mes del año
I2cwbyte Anio                            'Nuevo Año
I2cstop
Return

```

\*\*\*\*\* Subrutina Grabar Hora \*\*\*\*\*

```

Settime:
  Segu = Makebcd(segu)
  Minu = Makebcd(minu)
  Hora = Makebcd(hora)
I2cstart                                'Genera inicio de I2C
I2cwbyte Ds1307w                        'Envia Constante escritura del DS1307
I2cwbyte 0                              'Direccion de segundos del DS1307
I2cwbyte Segu                            'Nuevo Segundo
I2cwbyte Minu                             'Nuevo Minuto
I2cwbyte Hora                            'Nueva Hora
I2cstop
Return

```

\*\*\*\*\*  
\*\*\*\*\*

```

Rec_isr:
Led1 = 1
  Disable Urx
  Input Serial1 Noecho
  Temporal1 = Mid(serial1 , 1 , 1)
  Select Case Temporal1
    Case "A"
      ' Lcdat 7 , 50 , "a"
      ' Print "c"
      ' Input Texto9 Noecho
      ' X = Val(texto9)
      ' Writeeprom X , 2
      ' Readeeprom X , 2
      ' Clientes = X - 1
      ' Flag1 = 1
    Case "B"
      ' Lcdat 7 , 70 , "b"
      ' Input Texto9 Noecho
      ' Print "c"
      ' X = Val(texto9)
      ' Writeeprom X , 8
      ' Flag2 = 1
    Case "C"
      ' Flag24 = 1
    Case "D"
      ' Flag25 = 1
  End Select
Do
  B = Inkey()
Loop Until B = 0

```

```

    Enable Urxc
'   Wait 1
    Led1 = 0
Return
*****
_init_micro:
    Ddrc.0 = 1
    Portc.0 = 1
    Ddrd.6 = 1
    Portd.6 = 1
'   Gab Alias Portc.0
'   Gba Alias Portd.6
Return

End

$include "font8x8.font"
$include "font8x6.font"
$include "font8x6b.font"
$include "font16x16.font"

Caratula1:
'include the picture data
$bgf "caratula.bgf"
Caratula2:
'include the picture data
$bgf "picmicro.bgf"

```



### **Anexo 3** Datasheet del Micro controlador ATMEGA 128

#### Features

- High-performance, Low-power Atmel®AVR®8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 128Kbytes of In-System Self-programmable Flash program memory
  - 4Kbytes EEPROM
  - 4Kbytes Internal SRAM
  - Write/Erase cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits

#### In-System Programming by On-chip Boot Program

#### True Read-While-Write Operation

- Up to 64Kbytes Optional External Memory Space
- Programming Lock for Software Security
- SPI Interface for In-System Programming
- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix acquisition
  - Up to 64 sense channels
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes

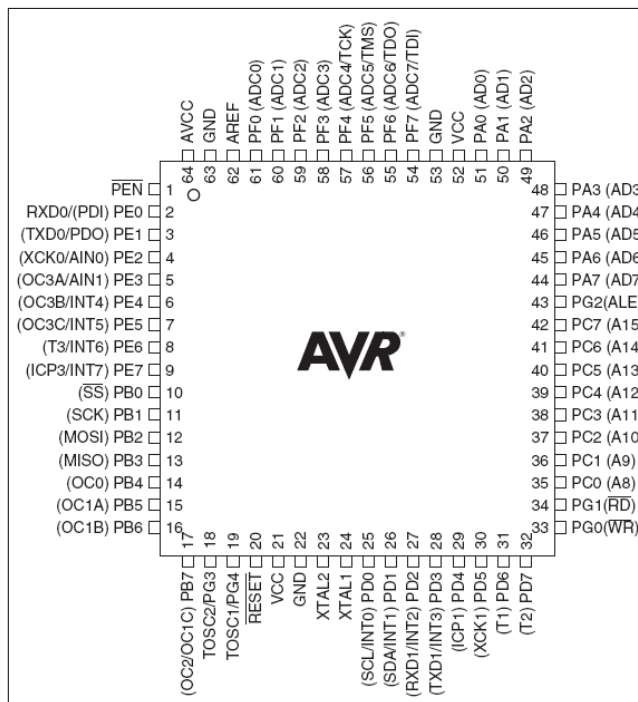
- Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
- Real Time Counter with Separate Oscillator
- Two 8-bit PWM Channels
- 6 PWM Channels with Programmable Resolution from 2 to 16 Bits
- Output Compare Modulator
- 8-channel, 10-bit ADC
- 8 Single-ended Channels
- 7 Differential Channels
- 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial Interface
- Dual Programmable Serial USARTs
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with On-chip Oscillator
- On-chip Analog Comparator
- Special Microcontroller Features
- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- Software Selectable Clock Frequency
- ATmega103 Compatibility Mode Selected by a Fuse
- Global Pull-up Disable
- I/O and Packages
- 53 Programmable I/O Lines
- 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
- 2.7 - 5.5V ATmega128L
- 4.5 - 5.5V ATmega128
- Speed Grades
- 0 - 8MHz ATmega128L

## Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow \text{\$FF} - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow \text{\$00} - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (\text{\$FF} - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow \text{\$FF}$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
<b>BRANCH INSTRUCTIONS</b>					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow \text{STACK}$	None	4
RETI		Interrupt Return	$PC \leftarrow \text{STACK}$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1 / 2

			$k + 1$		
BRPL	k	Branch if Plus	if $(N = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if $(H = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if $(H = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if $(T = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if $(T = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if $(V = 1)$ then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if $(V = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2

## Distribución de pines del ATMEGA 128



## Anexo 4 Especificaciones del driver de potencia

### Valores Máximos Soportados En El Módulo

Parámetro	Valor	Unidades
Voltaje máximo $V_{pot}$	46	V
Imáx por driver	3	A
Voltaje máximo, lógica	7	V
Disipación de potencia	25	W

Elaborado por: Jaime Heredia

### Valores De Operación

Parámetro	Valor	Unidades
V. alimentación de potencia.	8-30	V
Corriente por dirver	2	A
V lógica	5	V
V nivel lógico "0"	-0.3 -1.5	V
V nivel lógico "1"	2.3 a 5	V

Elaborado por: Jaime Heredia

### Descripción de señales.

Señal	Descripción
IN1	TTL. Controla el nivel de la salida de potencia M1
IN2	TTL. Controla el nivel de la salida de potencia M2
IN3	TTL. Controla el nivel de la salida de potencia M3
IN4	TTL. Controla el nivel de la salida de potencia M4
E1	Enable para las salidas M1 y M2. "1"= Enable. "0" = alta impedancia.
E2	Enable para las salidas M3 y M4. "1"= Enable. "0" = alta impedancia.
M1	Salida del amplificador de potencia 1.
M2	Salida del amplificador de potencia 2.
M3	Salida del amplificador de potencia 3.
M4	Salida del amplificador de potencia 4.
+5V	Entrada de alimentación para la lógica.
GND	GND de la lógica
+Vpot	Tensión de alimentación de las cargas. 8-46 V. Recomendado: 8-30 V.
GNDpot	GND de la alimentación de potencia.
TP1 (Isen A)	Sensado de corriente del puente A (salidas M1 y M2). Valor: Imotor x 0,5
TP1 (Isen B)	Sensado de corriente del puente B (salidas M3 y M4). Valor: Imotor x 0,5

Elaborado por: Jaime Heredia

### Descripción De Los Conectores

Conector	Descripción
J1	Alimentación para la lógica (también disponible en J3). MOLEX de 2 vías.
J3	Líneas de control y alimentación para la lógica. MOLEX de 8 vías.
J2	Alimentación de potencia ( +Vpot y GNDpot). Bornas con tornillo.



J4	+Vpot. Bornas con tornillo.
J6	GNDpot. Bornas con tornillo.
J5	Salidas de potencia M1, M2, M3 y M4. Bornas con tornillo.
TP1	Salida para sensado de corriente del puente A. TP1 (voltios) = 0,5 x IA ; I en amperios
TP2	Salida para sensado de corriente del puente B. TP2 (voltios) = 0,5 x IB ; I en amperios
J7	Polarización de las líneas E1 y E2. Permite polarizar estas líneas en "UP" o en "DOWN".
J8,J9,J10,J11,J12	Puentes que permiten utilizar en paralelo los canales 1 con 4 y 2 con 3. Instalados en versión "P".

Elaborado por: Jaime Heredia

### Precauciones

- No se debe invertir la polaridad de la alimentación, ni el voltaje de entrada lógica TTL y tampoco el voltaje de alta corriente (potencia). Esto causaría daños graves en el módulo, como el fallo del micro controlador.
- Es recomendable incorporar un fusible desde la fuente de alimentación de potencia.
- No cortocircuitar las salidas entre sí, esto quiere decir que no se debe hacer puentes si no se ocupa todas las salidas del driver de potencia.
- Desactivar la alimentación siempre antes de realizar cualquier cableado o modificación del mismo.
- Si se instalan los puentes para la utilización como un único puente H, éste se controla mediante 3 líneas solamente: IN3, IN4 y E2. En este caso NUNCA dejar conectadas las otras 3 entradas de control, pues quedarían conectadas en paralelo a las anteriores y podrían dañar las salidas del micro controlador.
- Si se va a utilizar el módulo prolongadamente con tensiones superiores a 30 V y corrientes próximas a las máximas permitidas, es recomendable sustituir el disipador por otro de mayor capacidad.
- La caja donde se instale el módulo deberá tener rejillas de ventilación.

