

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.**

## **FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

"Diseño de Brazo Robótico y Diseño de Brazo Empuja Frascos  
Controlados por LabVIEW y dsPICs"

### **TESIS DE GRADO**

Previa a la obtención del Título de:

### **INGENIERO EN ELECTRICIDAD**

Presentado por:

Gabriela Fernanda Sánchez Espinoza

Especialización: Electrónica y Automatización Industrial

Carlos Alberto Alvarado Chóez

Especialización: Electrónica y Telecomunicaciones

Guayaquil - Ecuador

Año 2008

# **AGRADECIMIENTO**

Agradecemos a Dios, a nuestros padres y hermanos que nos han brindado su apoyo a lo largo de nuestra carrera.

# DEDICATORIA

A nuestra querida familia y amigos.

# TRIBUNAL DE GRADUACIÓN



---

Ing. Holger Cevallos  
Presidente del Tribunal



---

Ing. Carlos Valdivieso  
Director de Tesis



---

Ing. Alberto Larco G.  
Miembro Principal

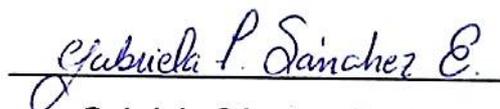


---

Ing. Carlos Salazar L.  
Miembro Principal

# DECLARACIÓN EXPRESA

La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral.

  
Gabriela Sánchez Espinoza

  
Carlos Alvarado Chóez

# RESUMEN

Basados en los conocimientos adquiridos en el Tópico de graduación “Plataforma para el Desarrollo de Proyectos con dsPICs y Visión Robótica con LabVIEW” dirigido por el Ing. Carlos Valdivieso, catedrático de la Facultad de Ingeniería Eléctrica y Computación, se desarrolló esta tesis, que como principal objetivo tuvo la construcción de un Brazo Robótico y un Empuja Frascos con ayuda de dsPICs y LabVIEW.

En el capítulo 1 encontrará una breve introducción hacia los brazos robóticos, su clasificación, tipos y aplicaciones en el Ecuador, así como de las perspectivas futuras que tiene este campo en nuestro país.

En el capítulo 2 se muestran las características morfológicas de los brazos Robóticos, además de algunos ejemplos de brazos de utilización industrial y el impacto de los mismos en nuestro país.

El capítulo 3 está basado en la construcción de un brazo Robótico y un brazo Empuja Frascos con propósito educativo que se utilizarán para nuestra facultad. En su desarrollo se han tomado en consideración ciertas especificaciones y condiciones a cumplir para el adecuado control y funcionamiento de la plataforma

El capítulo 4 es una introducción acerca de las herramientas tecnológicas a utilizarse en el desarrollo del dispositivo controlador del brazo robótico y del empuja frascos.

En el capítulo 5 se incluyen algunos ejemplos realizados por otros grupos de nuestro Tópico. Recordando que el brazo robótico y empuja frascos trabajarán en procesos de visión con LabVIEW en donde intervinieron 18 personas. También en este capítulo encontrará algunas posibles mejoras para futuros brazos robóticos de similares características.

Como en todo proceso se necesitan realizar pruebas para verificar la funcionalidad de los brazos y así conocer los límites de operación de los mismos, es por eso que el capítulo 6 muestra las pruebas que se realizaron, los datos que se obtuvieron de estas y sus respectivos análisis.

# INDICE GENERAL

RESUMEN

INDICE GENERAL

ABREVIATURAS

INDICE DE FIGURAS

INDICE DE TABLAS

|  |    |
|--|----|
| CAPITULO 1 .....   | 22 |
| 1. BRAZOS ROBOTICOS .....  | 22 |
| 1.1 INTRODUCCION GENERAL .....   | 22 |
| 1.2 CLASIFICACION DE LOS BRAZOS ROBÓTICOS .....  | 24 |
| 1.3 TIPOS DE ROBOTS .....  | 25 |
| 1.4 APLICACIONES CON BRAZOS ROBOTICOS .....  | 25 |
| 1.5 USO DE LOS BRAZOS ROBOTICOS EN EL ECUADOR .....  | 26 |
| 1.5.1 Husky.- Un robot mejorado para preformas .....                                       | 27 |
| 1.5.2 Robots para soldadura por puntos: ofrecen máxima precisión y alta<br>velocidad ..... | 28 |
| 1.5.3 RAFTER 300S Robots Soldadores.....   | 29 |
| 1.6 PERSPECTIVAS FUTURAS.....  | 31 |

|  |    |
|--|----|
| CAPITULO 2.....  | 33 |
| 2. MORFOLOGÍA DE LOS BRAZOS ROBÓTICOS Y SUS<br>APLICACIONES INDUSTRIALES ..... | 33 |
| 2.1 COMPONENTES MANIPULADOR Y CONTROLADOR .....                                | 33 |
| 2.1.1 Componente Manipulador .....   | 33 |
| 2.1.2 Componente Controlador .....   | 35 |
| 2.2 DISPOSITIVOS DE ENTRADA Y SALIDA DE DATOS.....                             | 36 |
| 2.3 CARACTERÍSTICAS Y CONFIGURACIONES MORFOLOGICAS.....                        | 38 |
| 2.3.1 Características De Los Brazos Robóticos.....                             | 38 |
| 2.3.2 Tipos De Configuraciones Morfológicas .....                              | 39 |
| 2.4 BRAZOS ROBOTICOS DE UTILIZACION INDUSTRIAL .....                           | 40 |
| 2.4.1 M-6iB.....   | 40 |
| 2.4.2 Robot Júpiter .....  | 41 |
| 2.4.3 Robot industrial MoveMaster MITSUBISHI .....                             | 42 |
| 2.4.4 Robot industrial tipo PUMA 500.....                                      | 43 |
| 2.5 IMPACTO DE LOS BRAZOS ROBÓTICOS EN LA INDUSTRIA.....                       | 44 |
| CAPITULO 3.....  | 46 |
| 3. DISEÑO Y CONTRUCCIÓN DE BRAZO ROBÓTICO Y BRAZOS<br>EMPUJA FRASCOS .....     | 46 |
| 3.1 CARACTERÍSTICAS PRINCIPALES.....   | 46 |
| 3.2 CONDICIONES QUE DEBEN CUMPLIR .....  | 49 |
| 3.2.1 Características y Condiciones del Brazo Robótico.....                    | 50 |
| 3.2.2 Condiciones del Brazo Empuja Frascos .....                               | 51 |

|                 |   |     |
|-----------------|---|-----|
| 3.3             | MATERIALES QUE SE PUEDEN USAR.....                                  | 52  |
| 3.4             | INSTRUCCIONES GENERALES.....  | 53  |
| 3.5             | HARDWARE Y MECÁNICA.....  | 54  |
| 3.5.1           | Ensamblaje del Brazo Robótico.....                                  | 54  |
| 3.5.2           | Construcción de Brazo Empuja frascos .....                          | 55  |
| 3.5.3           | Diseño de la Tarjeta Controladora.....                              | 63  |
| 3.6             | FIRMWARE Y PROGRAMACIÓN.....  | 75  |
| 3.6.1           | Lenguaje de Programación.....                                       | 75  |
| 3.6.2           | Compilador a usar .....   | 76  |
| 3.6.3           | Dispositivo de programación a usarse en la tarjeta dsPIC30F4012 ... | 77  |
| 3.6.4           | Tiempo de ciclo de instrucción.....                                 | 78  |
| CAPITULO 4..... |   | 81  |
| 4.              | LABVIEW Y DSPICS .....  | 81  |
| 4.1.            | GENERALIDADES DE LABVIEW.....                                       | 81  |
| 4.1.1           | LabVIEW .....   | 81  |
| 4.1.2           | Instrumentos virtuales.....   | 83  |
| 4.1.3           | Instrumentos Virtuales de bajo nivel (SubVIs) .....                 | 84  |
| 4.2.            | GENERALIDADES DE DSPICS. ....                                       | 85  |
| 4.2.1           | Familias dsPIC para el control de motores .....                     | 86  |
| 4.3.            | ADQUISICIÓN DE DATOS CON LABVIEW.....                               | 87  |
| 4.4.            | DSPICS CONTROLANDO EL BRAZO ROBÓTICO.....                           | 105 |
| 4.5.            | INTERFAZ ENTRE LABVIEW Y DSPICS. ....                               | 116 |

|                 |  |     |
|-----------------|--|-----|
| 4.4.1           | Programación del dsPIC30F4012 para recepción y envío de parámetros .....           | 116 |
| 4.6.            | EJEMPLOS DE APLICACIONES.....  | 122 |
| CAPITULO 5..... |  | 130 |
| 5.              | POSIBLES APLICACIONES DE LOS BRAZOS ROBÓTICOS EN LA PLATAFORMA DE DESARROLLO ..... | 130 |
| 5.1.            | OBJETIVOS DE LA APLICACIÓN DEL BRAZO ROBÓTICO DISEÑADO                             | 130 |
| 5.2.            | ÁREAS DE USO .....   | 131 |
| 5.3.            | FORMAS DE USO .....  | 132 |
| 5.4.            | VISIÓN CON LABVIEW APLICADO AL PROCESO.....  | 133 |
| 5.4.1.          | Inspección de botellas con tapa.....   | 133 |
| 5.4.2.          | Inspección de turbiedades en el líquido de las botellas. ....                      | 138 |
| 5.4.3.          | Inspección del nivel de líquido en cada botella. ....                              | 145 |
| 5.5.            | ACTUALIZACIONES Y MEJORAS PARA BRAZO ROBÓTICO .....                                | 150 |
| 5.6.            | ANÁLISIS DE COSTOS.....  | 152 |
| CAPITULO 6..... |  | 155 |
| 6.              | PRUEBAS REALIZADAS.....  | 155 |
| 6.1             | PRUEBAS REALIZADAS .....   | 155 |
| 6.1.1.          | Pruebas de ubicación .....   | 156 |
| 6.1.2.          | Pruebas de peso .....  | 157 |
| 6.1.3.          | Pruebas de frecuencia PWM.....   | 157 |
| 6.2             | DATOS OBTENIDOS .....  | 158 |

|                                 |     |
|---------------------------------|-----|
| 6.3 ANÁLISIS DE RESULTADOS..... | 165 |
| CONCLUSIONES Y RECOMENDACIONES  |     |
| ANEXOS                          |     |
| BIBLIOGRAFIA                    |     |

# INDICE DE FIGURAS

|   |    |
|---|----|
| FIGURA 1. Robot Husky .....                                     | 27 |
| FIGURA 2. Robots para soldadura por puntos: M-900iA/260L .....  | 28 |
| FIGURA 3. Robots Automotrices .....                             | 29 |
| FIGURA 4. Pórtico de soldadura Rafter para vigas .....          | 29 |
| FIGURA 5. Pórtico de corte Handy, 2 Plasmas y 2 Oxicortes ..... | 30 |
| FIGURA 6. Articulaciones del Robot.....                         | 34 |
| FIGURA 7. Partes del manipulador.....                           | 35 |
| FIGURA 8. Sistema controlador.....                              | 37 |
| FIGURA 9. Comunicación de un Robot .....                        | 38 |
| FIGURA 10. Robot para manipulación de construcción modular..... | 40 |
| FIGURA 11. Júpiter.....   | 41 |
| FIGURA 12. Movemaster.....                                      | 42 |
| FIGURA 13. Puma 500 .....                                       | 43 |
| FIGURA 14. Brazo Robótico RA-01C .....                          | 54 |

|  |    |
|--|----|
| FIGURA 15. Vista Superior Futaba S3003 .....                   | 56 |
| FIGURA 16. Vista lateral Futaba S3003 .....                    | 56 |
| FIGURA 17. Tornillos para Motor.....                           | 56 |
| FIGURA 18. Soporte MDF .....                                   | 57 |
| FIGURA 19. U plástica para soporte al riel metálico.....       | 58 |
| FIGURA 20. Accesorios de Servomotor .....                      | 58 |
| FIGURA 21. Cuerpo del Brazo Empuja Frascos.....                | 59 |
| FIGURA 22. Unión de Brazo con acople de Servo .....            | 60 |
| FIGURA 23. Unión de acople con el Servomotor.....              | 60 |
| FIGURA 24. Acople Servo y MDF.....                             | 61 |
| FIGURA 25. Ajuste de Servomotor y MDF .....                    | 61 |
| FIGURA 26. Ajuste de MDF al Riel de Aluminio .....             | 62 |
| FIGURA 27. Ajuste MDF al Riel de Aluminio vista posterior..... | 62 |
| FIGURA 28. Vista General de los Brazos .....                   | 63 |
| FIGURA 29. Esquemático de la tarjeta dsPIC30F4012 .....        | 68 |
| FIGURA 30. Circuito impreso de la tarjeta dsPIC30F4012.....    | 69 |

|  |    |
|--|----|
| FIGURA 31. Vista superior de la tarjeta dsPIC30F4012 .....   | 69 |
| FIGURA 32. Disposición de Señales en los espadines .....   | 70 |
| FIGURA 33. Espadines para conexión ICSP .....  | 73 |
| FIGURA 34. Conexión del MAX232.....  | 73 |
| FIGURA 35. Tarjeta dsPIC30F4012 .....  | 74 |
| FIGURA 36. Área de trabajo del programa Mikrobasic.....  | 76 |
| FIGURA 37. Programador PICKit 2.....   | 77 |
| FIGURA 38. Cuadro de dialogo para la configuración de bits del registro de configuración del dsPIC30F4012..... | 79 |
| FIGURA 39. Ejemplo de Diagrama de Bloques .....  | 83 |
| FIGURA 40. Ejemplo de Panel Frontal .....  | 84 |
| FIGURA 41. Ejemplo de utilización de SubVIs .....  | 85 |
| FIGURA 42. VI's para brazo robótico y brazo empuja frascos.....  | 87 |
| FIGURA 43. VI Iniciar Puerto.....  | 88 |
| FIGURA 44. Diagrama de Flujo de VI Iniciar puerto .....  | 89 |
| FIGURA 45. VI Cerrar Puerto .....  | 91 |

|   |     |
|---|-----|
| FIGURA 46. Estructura Cerrar Puerto .....                               | 91  |
| FIGURA 47. VI Configurar Empuja frascos.....                            | 93  |
| FIGURA 48. VI Configurar Velocidad de Servos.....                       | 93  |
| FIGURA 49. Diagrama de Flujo configuración de velocidad de Servos ..... | 94  |
| FIGURA 50. VI Configuración Por Defecto .....                           | 95  |
| FIGURA 51. Estructura Configuración por defecto .....                   | 95  |
| FIGURA 52. VI Mover Servo.....  | 96  |
| FIGURA 53. Diagrama de Flujo para Mover Servos .....                    | 97  |
| FIGURA 54. VI Mover Servo I.....  | 98  |
| FIGURA 55. VI Mover Servo 2.....  | 98  |
| FIGURA 56. VI Mover Servo 3.....  | 99  |
| FIGURA 57. VI Mover Servo 4.....  | 99  |
| FIGURA 58. VI Mover Servo 5.....  | 100 |
| FIGURA 59. VI Empuja Frascos .....                                      | 100 |
| FIGURA 60. Estructura de Empuja Frascos .....                           | 101 |

|  |     |
|--|-----|
| FIGURA 61. VI Enviar Secuencia .....   | 101 |
| FIGURA 62. Archivo secuencia de movimiento .....   | 102 |
| FIGURA 63. Diagrama de Flujo enviar secuencia .....  | 104 |
| FIGURA 64. Diagrama de Flujo para manejo de servos.....  | 110 |
| FIGURA 65. Diagrama de flujo programación empuja frascos .....   | 114 |
| FIGURA 66. PWM por software para brazo empuja frascos.....   | 116 |
| FIGURA 67. Diagrama de programación para recepción y envío de datos en<br>la tarjeta dsPIC30F4012..... | 119 |
| FIGURA 68. Panel Frontal de la Aplicación .....  | 122 |
| FIGURA 69. Diagrama de Bloque de la Aplicación.....  | 123 |
| FIGURA 70. Estructura Inicial de la Aplicación .....   | 126 |
| FIGURA 71. Estructura para configurar y mover brazo empuja frascos .....                               | 127 |
| FIGURA 72. Estructura para mover servos .....  | 127 |
| FIGURA 73. Estructura para agregar posiciones al archivo .....   | 128 |
| FIGURA 74. Estructura para abrir archivo y ejecutar secuencia de<br>movimientos .....                  | 128 |
| FIGURA 75. Estructura para finalizar la aplicación .....   | 129 |

|   |     |
|---|-----|
| FIGURA 76. Vista Superior de la plataforma de trabajo ..... | 139 |
| FIGURA 77. Comparación de Color.....                        | 140 |
| FIGURA 78. Análisis Central.....                            | 142 |
| FIGURA 79. Análisis superior .....                          | 143 |
| FIGURA 80. Análisis Inferior.....                           | 144 |
| FIGURA 81. Filtro de luminiscencia .....                    | 146 |
| FIGURA 82. Detección de nivel .....                         | 147 |
| FIGURA 83. Cálculo de Nivel.....                            | 149 |
| FIGURA 84. Prueba a cero Grados .....                       | 158 |
| FIGURA 85. Prueba a 90 Grados .....                         | 159 |
| FIGURA 86. Prueba a 180 Grados .....                        | 159 |
| FIGURA 87. Brazo Posición Inicial .....                     | 160 |
| FIGURA 88. Brazo Posición intermedio.....                   | 161 |
| FIGURA 89. Brazo Posición Final.....                        | 161 |
| FIGURA 90. 1(oz.) .....                                     | 162 |
| FIGURA 91. 3 (Oz.) .....                                    | 163 |

|  |     |
|--|-----|
| FIGURA 92. 4 (Oz.) .....   | 163 |
| FIGURA 93. 7 (Oz.) .....   | 164 |
| FIGURA 94. Estructura de VI de Conversión de grados a Posición en<br>decimal ..... | 167 |

## INDICE DE TABLAS

|   |     |
|---|-----|
| Tabla I. Servos a usarse .....  | 55  |
| Tabla II. Descripción de parámetros .....   | 121 |
| Tabla III. Resultados de la Prueba de Posición en los servomotores del<br>Brazo Robótico..... | 160 |
| Tabla IV. Resultados de Pruebas de Posición Brazo Empuja Frascos.....                         | 162 |
| Tabla V. Resultados de la Prueba de Peso Brazo Robótico.....                                  | 164 |
| Tabla VI. Resultados de la prueba de Frecuencia PWM .....                                     | 165 |

## ABREVIATURAS

|        |   |
|--------|---|
| AFRI   | Asociación Francesa de Robótica Industrial                              |
| Bit    | Es la unidad más pequeña de información que utiliza un ordenador        |
| Byte   | Conjunto de 8 bits  |
| cm.    | Centímetros   |
| CMOS   | Semiconductor complementario del óxido de metal                         |
| DB9    | Conector para puerto serie  |
| DsPIC  | Microcontrolador de Microchip con soporte para procesamiento de señales |
| EIA    | Asociación de Industrias Electrónicas                                   |
| Hz.    | Hertzios  |
| ICSP   | Programación serial en circuito   |
| IFR    | Federación Internacional de Robótica                                    |
| MAX232 | Integrado de conversión de TTL a RS232                                  |
| NI     | Nacional Instruments  |
| oz.    | Onzas   |
| PIC    | Microcontrolador de Microchips  |
| PWM    | Modulación de ancho de pulso  |
| RS232  | Estándar de comunicación serial   |
| Rx     | Recepción   |
| SCARA  | Brazo robótico articulado de respuesta selectiva                        |
| SubVI  | Instrumento Virtual de bajo nivel                                       |
| TTL    | Lógica transistor-transistor  |
| Tx     | Transmisión   |
| USB    | Bus Serial Universal  |
| V      | Voltio  |
| VI     | Instrumento Virtual   |

# CAPITULO 1

## 1. BRAZOS ROBOTICOS

### 1.1 INTRODUCCION GENERAL

Con el afán de contribuir con la universidad en materia de conocimientos en robótica, se ha implementado una plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW.

En esta tesis se propone el diseño de un brazo robótico y un brazo empuja frascos que serán controlados por LabVIEW y por un microcontrolador avanzado que es el dsPIC, ubicados en una plataforma con bandas transportadoras simulando procesos de embotellado para propósitos didácticos en la Escuela Superior Politécnica del Litoral (ESPOL) [Anexo A]. El brazo robótico y el brazo empuja frascos podrán ser utilizados para la manipulación de frascos. La función del brazo robótico va a ser la traslación del frasco de un lugar a otro de la plataforma y el empuja frascos para

expulsar los frascos que no cumplan los requerimientos así como detenerlos para su análisis. Tanto el brazo robótico como el brazo empuja frascos podrán ser utilizados para aplicaciones de visión robótica con LabVIEW o con otras aplicaciones.

El brazo humano es un diseño asombroso pues nos permite colocar la mano casi en cualquier lugar donde la necesitemos. Los brazos robóticos imitan la habilidad del brazo humano de ser flexibles, precisos y de estar listos para realizar acciones en una amplia variedad de acciones, por consiguiente son capaces de realizar tareas en las que se encuentra en riesgo el hombre o el acceso a ellas es difícil.

La necesidad cada vez mayor de aumentar la productividad y conseguir productos acabados de una calidad uniforme llevó a la industria a que gire cada vez más hacia una automatización.

La mayoría de los robots industriales son simplemente brazos robóticos que tienen unas pinzas o herramientas en el extremo. El brazo mecánico es el mecanismo de articulación fundamental de cualquier robot industrial o robot autónomo.

## 1.2 CLASIFICACION DE LOS BRAZOS ROBÓTICOS

La maquinaria para la automatización rígida dio paso al robot con el desarrollo de controladores rápidos, basados en el microprocesador, así como el empleo de motores en bucle cerrado, que permiten establecer con exactitud la posición real de los elementos del robot y establecer el error con la posición deseada. Esta evolución ha dado origen a una serie de tipos de robots, que se citan a continuación [1]:

- Manipuladores
- Robots de repetición y aprendizaje
- Robots por control por computador
- Robots inteligentes
- Micro-Robots

Figuras ejemplo de esta división lo podrá encontrar en el ANEXO F-2.

Nuestro brazo robótico según la división AFRI [2] e IFR [3] es de tipo C, manipulador programable de secuencia continua, esta división se puede observar en el ANEXO F-1.

### **1.3 TIPOS DE ROBOTS**

Hay diferentes tipos de robots clasificados según el medio en el que se pueden transportar:

- Terrestres (vehículos, robots con patas, manipuladores industriales) [4]
- Aéreos (dirigibles) [5]
- Acuáticos (nadadores, submarinos) [6]

Nuestros brazos robóticos son de tipo Terrestre debido a su clasificación de Manipulador Industrial.

Algunos ejemplos gráficos de robots clasificados según su tipo lo podemos encontrar en el ANEXO F-3

### **1.4 APLICACIONES CON BRAZOS ROBOTICOS**

Debido a la versatilidad que tienen los brazos robóticos, pueden ser usados en casi todas partes, solo con ciertas modificaciones en su forma y comportamiento [7].

Carga y Descarga de máquinas

Soldadura por puntos

Soldadura por arco continua

Recubrimiento con espray

Operaciones de proceso

Taladro y acanalado (fresa)

Rectificado, pulido y cepillado

Remachado

Corte por chorro de agua

Corte por láser

Laboratorios

Agricultura

Espacio [8] [9]

Educación

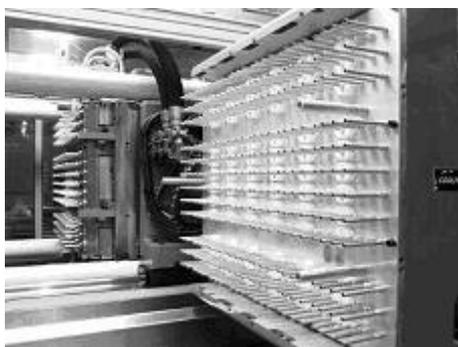
Ejemplos de aplicaciones con brazos robóticos se pueden observar en el ANEXO F-4

## **1.5 USO DE LOS BRAZOS ROBOTICOS EN EL ECUADOR**

El uso de los robots en el Ecuador ha ido en aumento desde hace algunos años es por eso que nombraremos algunos como ejemplo.

### 1.5.1 Husky.- Un robot mejorado para preformas

**FIGURA 1. Robot Husky**



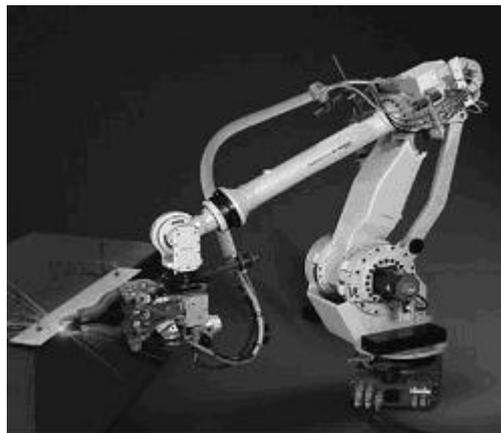
Fuente: <http://www.interempresas.net/Plastico/FeriaVirtual/ResenyaProducto.asp?R=23748>

Husky ha actualizado su robot CoolPik [figura 1] en sus sistemas para preformas HyPET con un sistema de manipulación más fiable basado en tecnología de vacío. La transferencia mediante vacío ofrece diversas ventajas, como más fiabilidad, menor mantenimiento y puesta en marcha más sencilla [10].

Este es un Robot manipulado por PLCs utilizados en Amcor PET Packaging para hacer botellas con plástico alimenticio.

### 1.5.2 Robots para soldadura por puntos: ofrecen máxima precisión y alta velocidad

**FIGURA 2. Robots para soldadura por puntos: M-900iA/260L**



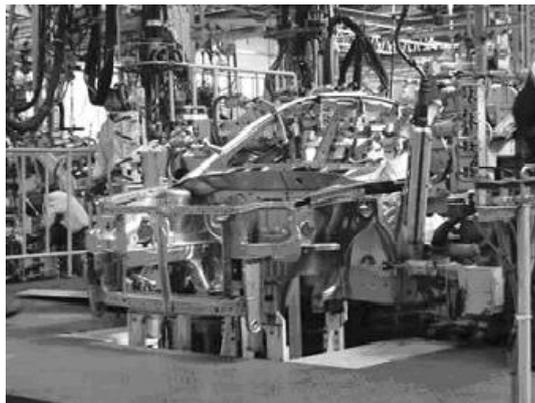
**Fuente: <http://www.interempresas.net/Robotica/FeriaVirtual/ResenyaProducto.asp?R=22246>**

La serie M-900iA de Fanuc Robotics [figura 2] ha sido diseñada para dar solución a aquellas aplicaciones que requieren máxima precisión, alta carga, alta velocidad y máxima fiabilidad en la soldadura por puntos [11]. Con una construcción modular de seis ejes y servo controlado eléctricamente, el robot ha sido diseñado para cubrir las demandas de la fabricación y los sistemas de procesos. La serie M-900iA se controla con el último

controlador R-J3iB. El cliente puede usar las funciones de control de robot más recientes. Admite una carga máxima de 260 Kg., con un alcance máximo de 3,100 mm y una repetitividad de  $\pm 0,3$ .

Este es el tipo de Brazos que usa Aymesa en la producción de automóviles. [Figura 3]

**FIGURA 3. Robots Automotrices**

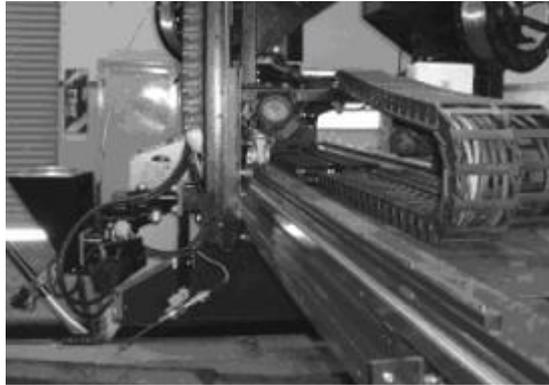


**Fuente: FANUC Robotics Ibérica**

### **1.5.3 RAFTER 300S Robots Soldadores**

Otro ejemplo sería Kubiec y Fedimetal [12], empresas que usan robots soldadores como el Rafter 300s [figura 4,5] [31]:

**FIGURA 4. Pórtico de soldadura Rafter para vigas**



Fuente: <http://www.walmaring.com.ar/>

**FIGURA 5. Pórtico de corte Handy, 2 Plasmas y 2 Oxicortes**



Fuente: <http://www.walmaring.com.ar/>

## 1.6 PERSPECTIVAS FUTURAS

En el Ecuador se tiene muchas esperanzas en la tecnología sobretodo en la automatización de procesos, es por eso que en la actualidad se profundiza en el estudio de la misma.

No es novedad que los robots se han venido introduciendo en las industrias del Ecuador desde hace algunos años. Esto se ha debido a dos razones principales, primero con el transcurso del tiempo los robots han incrementado su potencial tecnológico en gran manera y segundo la interfaz de los robots es más amigable con el usuario, y son más fáciles de conectar con otro software que en sus inicios, por ende más sencillos de instalar.

En el Ecuador se espera que el mercado de la robótica tenga una expansión tal que ya no solo las super industrias tengan acceso a esa tecnología, las cuales han sido clientes predilectos en los últimos años, sino que también alcance a las medianas y pequeñas industrias y por que no también a las microempresas.

La robótica es una tecnología con futuro y para el futuro. Si continúan las tendencias actuales, y si algunos de los estudios

de investigación en el laboratorio actualmente en curso se convierten finalmente en una tecnología factible, los robots del futuro serán unidades móviles con uno o más brazos, capacidades de sensores múltiples y con la misma potencia de procesamiento de datos y de cálculo que las grandes computadoras actuales. Serán capaces de responder a órdenes dadas con voz humana. Así mismo serán capaces de recibir instrucciones generales y traducirlas, con el uso de la inteligencia artificial en un conjunto específico de acciones requeridas para llevarlas a cabo. Podrán ver, oír, palpar, aplicar una fuerza media con precisión a un objeto y desplazarse por sus propios medios [13] [14].

En resumen, los futuros robots tendrían muchos de los atributos de los seres humanos. Sin embargo, como otras tecnologías, hay peligros potenciales implicados y deben establecerse salvaguardas para no permitir su mal uso. El paso del presente al futuro exigirá mucho trabajo de ingeniería mecánica, ingeniería electrónica, informática, ingeniería industrial, tecnología de materiales, ingenierías de sistemas de fabricación y ciencias sociales.

# CAPITULO 2

## 2. MORFOLOGÍA DE LOS BRAZOS ROBÓTICOS Y SUS APLICACIONES INDUSTRIALES

### 2.1 COMPONENTES MANIPULADOR Y CONTROLADOR

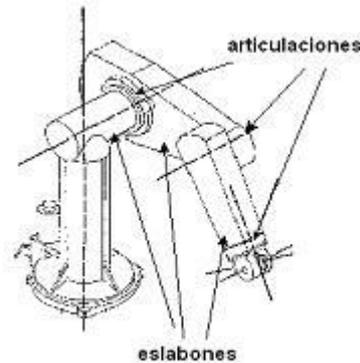
El brazo robótico esta compuesto principalmente por dos elementos, el componente manipulador y el componente controlador, el primero de características netamente mecánicas y el segundo electrónico [15].

#### 2.1.1 Componente Manipulador

Mecánicamente, es el componente principal. Está formado por una serie de elementos estructurales sólidos o eslabones unidos mediante articulaciones [figura 6] que

permiten un movimiento relativo entre dos eslabones consecutivos [16].

**FIGURA 6. Articulaciones del Robot**

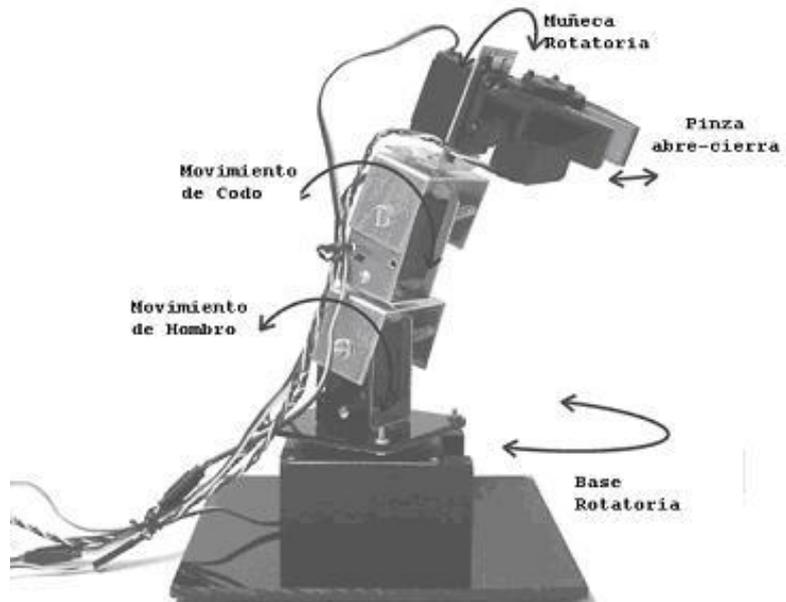


**Fuente:**[http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm)

Las partes que conforman el manipulador [figura 7] reciben, entre otros, los nombres de: cuerpo, brazo, muñeca y actuador final (o elemento terminal). A este último se le conoce habitualmente como aprehensor, garra, pinza.

El actuador final (pinza) es un dispositivo que se une a la muñeca del brazo del robot con la finalidad de activarlo para la realización de una tarea específica. La razón por la que existen distintos tipos de elementos terminales es precisamente por las funciones que realizan [17].

**FIGURA 7. Partes del manipulador**



**Fuente: RA-01 Robotic Arm & Controller, Manual & User's Guide, [www.imagesco.com](http://www.imagesco.com)**

El Brazo Robótico utilizado en esta tesis utiliza como elemento terminal una pinza manipulada por un servo motor.

### **2.1.2 Componente Controlador**

Como su nombre indica, es el que regula cada uno de los movimientos del manipulador, las acciones, cálculos y procesamiento de la información. El controlador recibe y envía señales a otras máquinas-herramientas (por medio de

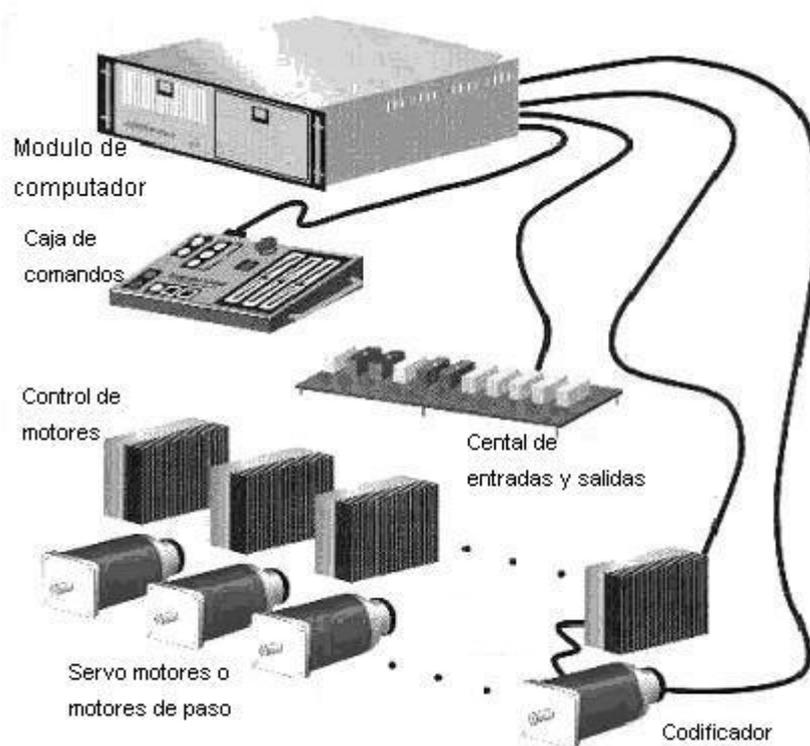
señales de entrada/salida) para manipulación del brazo robótico y del empuja frascos.

## **2.2 DISPOSITIVOS DE ENTRADA Y SALIDA DE DATOS**

Como ya mencionamos el robot está formado por un manipulador y un controlador, ahora estos se comunican entre sí y con el medio externo por medio de dispositivos, conocidos como dispositivos de entrada y salida.

Lo más comunes son: teclado, monitor y caja de comandos.

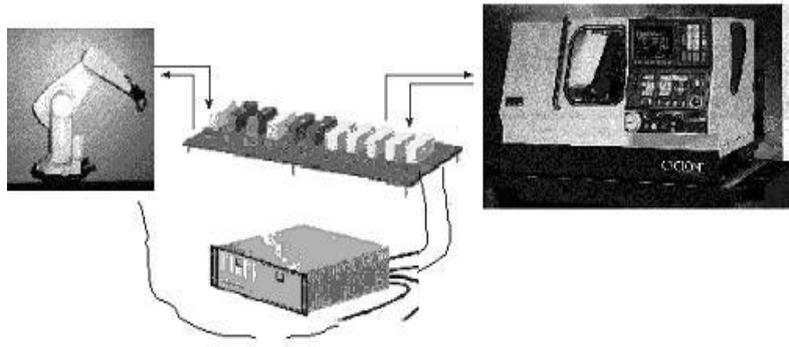
En la figura 8 se tiene un controlador (módulo de computadora) que envía señales a los motores de cada uno de los ejes del robot y la caja de comandos la cual sirve para enseñarle las posiciones al manipulador del robot.

**FIGURA 8. Sistema controlador**

Fuente:[http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm)

Las señales de entrada y salida se obtienen mediante el módulo electrónico instalado en el controlador del robot el cual le permite tener comunicación con el computador.

**FIGURA 9. Comunicación de un Robot**



Fuente:[http://cfievalladolid2.net/tecnocyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecnocyr_01/robotica/sistema/morfologia.htm)

## 2.3 CARACTERÍSTICAS Y CONFIGURACIONES MORFOLOGICAS

### 2.3.1 Características De Los Brazos Robóticos

A continuación se describen las características más relevantes propias de los robots [15].

- Grados de libertad.
- Espacio de trabajo.
- Precisión de los movimientos.
- Capacidad de carga.
- Velocidad.
- Tipo de actuadores
- Programabilidad

### 2.3.2 Tipos De Configuraciones Morfológicas

La estructura del manipulador y la relación entre sus elementos proporcionan una configuración mecánica, que da origen al establecimiento de los parámetros que hay que conocer para definir la posición y orientación del elemento terminal.

Fundamentalmente, existen cuatro estructuras clásicas en los manipuladores, que se relacionan con los correspondientes modelos de coordenadas en el espacio y que se citan a continuación:

- cartesiana
- cilíndrica
- esférica
- de brazo articulado,

Como se mencionó en el párrafo anterior estas son estructuras clásicas pero podemos mencionar también una que no es clásica, como lo es:

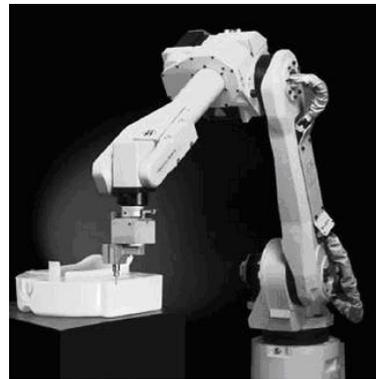
SCARA.- *Selective Compliance Assembly Robot Arm* por sus siglas en inglés, Brazo robótico articulado de respuesta selectiva.

En el ANEXO F-5 se encontrará figuras ejemplos de algunas de las configuraciones morfológicas de los brazos robóticos

## 2.4 BRAZOS ROBOTICOS DE UTILIZACION INDUSTRIAL

### 2.4.1 M-6iB

FIGURA 10. Robot para manipulación de construcción modular



Fuente: <http://www.interempresas.net/Robotica/FeriaVirtual/ResenyaProducto.asp?R=22225>

El M-6iB [Figura 10] es un robot de seis ejes, de construcción modular, servo controlado eléctricamente y

está diseñado para una gran variedad de aplicaciones industriales [18].

Dotado de 6 ejes, admite una carga máxima de 6 Kg. y un alcance máximo de 951 mm, con una repetibilidad de  $\pm 0,08$ .

#### 2.4.2 Robot Júpiter

FIGURA 11. Júpiter

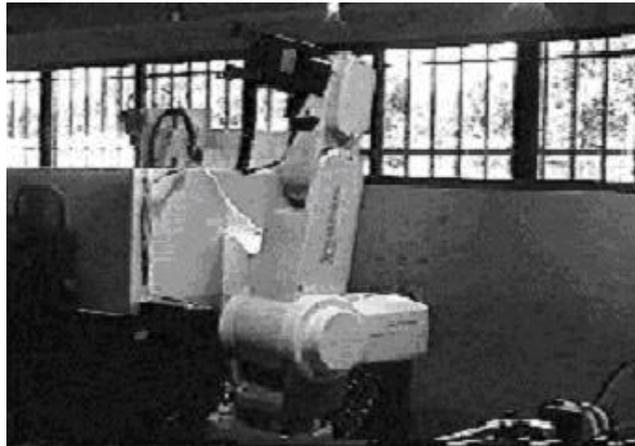


Fuente: [//www.chi.itesm.mx/profesional/laboratorios/equipo\\_cim.html](http://www.chi.itesm.mx/profesional/laboratorios/equipo_cim.html)

Este robot Júpiter [figura 11] tiene una configuración tipo SCARA, cuenta con 4 grados de libertad, su impulsión es electro-mecánica y se usa principalmente para ensamble e inspección [19].

### 2.4.3 Robot industrial MoveMaster MITSUBISHI

FIGURA 12. Movemaster

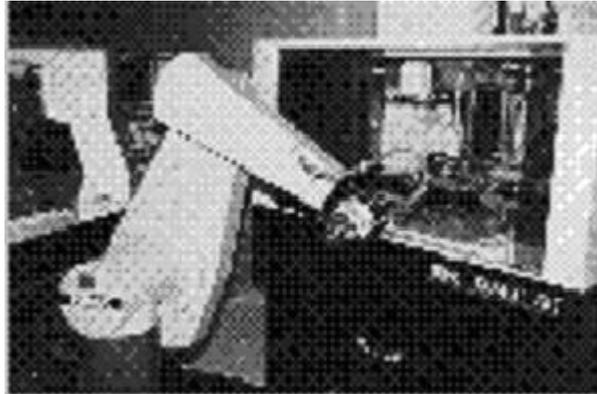


Fuente://[www.chi.itesm.mx/profesional/laboratorios/equipo\\_cim.html](http://www.chi.itesm.mx/profesional/laboratorios/equipo_cim.html)

El robot Mitsubishi [figura 12] es de tipo brazo articulado y tiene 5 articulaciones. El principal uso de este robot es para ensamble y carga/descarga de piezas a la estación de medición [19].

#### 2.4.4 Robot industrial tipo PUMA 500

FIGURA 13. Puma 500



Fuente://[www.chi.itesm.mx/profesional/laboratorios /  
equipo\\_cim.html](http://www.chi.itesm.mx/profesional/laboratorios/equipo_cim.html)

Este robot PUMA 500 [figura 13] tiene 6 grados de libertad, es de tipo brazo articulado y su impulsión es electro-mecánica. Se utiliza para la carga y descarga de las máquinas de control numérico [19].

## 2.5 IMPACTO DE LOS BRAZOS ROBÓTICOS EN LA INDUSTRIA

Considerando que el uso de los brazos robóticos en la industria se va incrementando a la vez de que el mundo empresarial va avanzando en lo que a tecnología se refiere y que a su vez la industria siempre estará tratando de disminuir costos y aumentar la calidad de su producto, es aquí donde los brazos robóticos juegan un papel muy importante, ya que ayudan a crear productos manufacturados de mayor calidad y menor costo, sin embargo también provocan la pérdida de empleos no calificados, especialmente en cadenas de montaje industriales, lo que en parte les ayuda a disminuir los gastos en la industria.

Claro está que se pueden crear empleos en sectores de soporte lógico y desarrollo de sensores, en la instalación y mantenimiento de los mismos, además de la conversión de las antiguas fábricas y diseño de fábricas nuevas, pero al mismo tiempo estas nuevas áreas de empleo implican mayores niveles de capacidad y formación.

Y si consideramos nuestra actual sociedad en la que el mayor porcentaje de la población no accede a un nivel medio o superior

de educación, las industrias se enfrentan a la tarea de formar trabajadores que pierden su empleo debido a la automatización y enseñarles nuevas capacidades para que puedan tener un puesto de trabajo, a un costo de mano de obra relativamente bajo, o simplemente enfrentarse a un costo mayor al contratar personal mas calificado, que pueden sacar mayor provecho a la nueva tecnología logrando así un mayor desarrollo de la empresa.

## **CAPITULO 3**

### **3. DISEÑO Y CONTRUCCIÓN DE BRAZO ROBÓTICO Y BRAZOS EMPUJA FRASCOS**

El brazo robótico que se ensambló fue adquirido en la página oficial de Images Co. [20] por un precio aproximado de 500 dólares en el que se incluía los materiales para el ensamblaje y además de una tarjeta controladora para el brazo robótico. El brazo empuja frascos fue diseñado y construido por nuestro grupo de trabajo además de que se diseñó una tarjeta controladora mucho mejor para controlar el brazo robótico y el brazo empuja frascos.

#### **3.1 CARACTERÍSTICAS PRINCIPALES**

Basados en las características morfológicas descritas en el capítulo 2 el movimiento de un brazo robótico debe de tener al menos las siguientes características:

- Movilidad
- Gobernabilidad
- Autonomía
- Polivalencia

**La movilidad** puede ser de dos tipos: traslacional, cuando el robot se mueve de un sitio para otro y articulada o rotacional, cuando gira su cuerpo.

El brazo cumple con la movilidad articulada, la que gira sobre su propio eje hasta  $\pm 90$  grados y tiene un radio de alcance de hasta 7 pulgadas (17.78 centímetros).

El brazo empuja frascos tiene también movilidad articulada, este sólo podrá realizar movimientos verticales sobre su propio eje con un radio fijo de 5 pulgadas (12.7 centímetros), el ángulo que puede alcanzar es máximo 180 grados.

**La gobernabilidad** depende del programa a través del cual se le comunican las instrucciones, bien sea directamente en el robot o por medio de un ordenador exterior.

El Brazo Robótico es manejado mediante una tarjeta la cual será la interfaz entre el robot y el computador, el tipo de comunicación es por el puerto serial, en el ordenador tendremos un instrumento virtual que será el encargado de enviar una serie de comandos como velocidad, número de servo a mover, señal de activación del brazo empuja frascos y su respectivo límite, además del parámetro que indicará las nuevas posiciones del brazo robótico.

**La autonomía** depende de los captadores o sensores que lleve incorporados para que pueda conocer su entorno y así posicionarse correctamente.

En este caso el nivel de autonomía del brazo robótico y del brazo empuja frascos es nulo, debido a que este no lleva ningún tipo de sensor o algo que le indique lo que sucede a su alrededor, este puede llegar a ser un grave déficit del brazo, pero debido a que se lo diseñó netamente académico y de bajo costo no cumple con este requisito.

**La polivalencia** depende de la capacidad de adecuación del robot para realizar tareas diferentes.

Debido a la constitución física de los brazos construidos: el brazo robótico que está conformado por 5 servos, 4 que constituyen el cuerpo y uno extra que acciona la tenaza, y el brazo empuja frascos que en realidad es una palanca dirigida por un solo servo.

Podemos decir que no pueden cumplir otra función, en el caso del brazo robótico el de agarrar, mover o girar la botella y en el caso del brazo empuja frascos realizará la función de empujar o de detener una botella a la vez.

La construcción del brazo empuja frascos es una tarea que puede ser tan difícil o complicada como se quiera, sin embargo se ha tratado de construir uno sin grandes pretensiones, con un funcionamiento básico que cumpla con cierto requerimiento, sin muchos atavíos y que funcione bien.

### **3.2 CONDICIONES QUE DEBEN CUMPLIR**

A continuación se mostrarán las condiciones que cumple el brazo robótico y que el brazo empuja frascos debe cumplir para los procesos en la plataforma de desarrollo de procesos con dsPICs y visión robótica con LabVIEW.

### 3.2.1 Características y Condiciones del Brazo Robótico

- El brazo dispone de un mecanismo de giro en torno a su eje, de no más de 180°.
- El brazo está conformado por servos con un rango de movimiento de no más de 180 °. En función de hombro, codo y muñeca.
- Tiene una pinza, que puede abrirse y cerrarse, para coger objetos. Con una abertura de máximo 32mm o 1.25”.
- Puede levantar un peso de hasta 3.0 oz.
- Tiene una altura de 10 pulgadas.
- El brazo robótico está acoplado a una base sólida, la cual deberá poder acoplarse fácilmente a la plataforma, que construirá otro grupo de estudiantes, y que también será controlado por circuitería desarrollada con dsPICs.
- Gira piezas en la cinta transportadora.
- Es capaz de retirar piezas de la cinta transportadora anteriormente mencionada y soltarlas en un recipiente.
- El brazo puede ser manejado desde una interfaz programada en cualquier lenguaje de programación.

Para este Tópico el lenguaje seleccionado fue LabVIEW.

- Y el programa de LabVIEW deberá poder acoplarse a cualquier otro instrumento virtual.
- El brazo una vez realizada la acción puede regresar a su posición original o neutra.

### **3.2.2 Condiciones del Brazo Empuja Frascos**

- El brazo empuja frascos debe tener una base con un rango de movimiento de máximo 180 °.
- El cuerpo del brazo empuja frascos no necesitará conformarse por partes, podrá ser un solo cuerpo.
- Debe tener dos extensiones, la una cóncava que tenga la forma de la botella y otra recta perpendicular al brazo y paralela al frasco.
- Debe ser capaz de empujar usando la parte cóncava y de un solo movimiento expulsar el frasco de la banda transportadora.
- Debe poder detener un frasco, usando la extensión recta, para su análisis justo en frente de la cámara.

- El brazo empuja frascos también será controlado por un instrumento virtual desde LabVIEW.

### **3.3 MATERIALES QUE SE PUEDEN USAR**

Es muy importante el uso de materiales livianos para la construcción de un brazo mecánico, ya que todo peso adicional redundará en una complejidad mecánica y económica, debido a que obligará a utilizar motores de mayores potencias.

El cuerpo de los robots puede ser de metal, plástico, madera u otro material, un buen material es el aluminio, este es fácil de conseguir, relativamente económico y extremadamente liviano en comparación con su dureza.

Con respecto a los motores que se utilizan pueden ser motores DC con sus respectivos reductores, o servo motores que tienen como ventaja de que no son necesarios frenos y son fáciles de direccionar.

El brazo robótico adquirido está formado por piezas de plástico, aluminio, madera, con tornillos de plástico y metálicos, además de

servo motores, formando así una estructura lo suficientemente robusta para soportar el peso que levantará en las aplicaciones de la plataforma de desarrollo de proyectos localizada en el laboratorio de Microcontroladores.

El brazo empuja frascos será construido con aluminio, una base de MDF, un soporte plástico, tornillos metálicos y un servomotor.

### **3.4 INSTRUCCIONES GENERALES**

Esta sección trata acerca de los pasos a seguir para cumplir con el objetivo de ensamblar el brazo robótico, diseñar y construir el brazo empuja frascos, además de la tarjeta controladora de los mismos.

- Definir los objetivos generales de los brazos.
- Establecer características y funciones.
- Seleccionar en el mercado el brazo robótico más factible.
- Buscar materiales para el brazo empuja frascos.
- Ensamblar el brazo robótico adquirido.
- Construir el brazo empuja frascos.
- Diseñar sistema controlador.

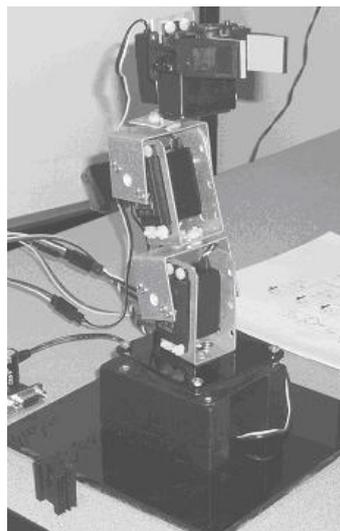
- Diseñar la interfaz con la computadora.
- Diseñar los instrumentos virtuales para las aplicaciones con visión.
- Implantarlo en la plataforma.
- Realizar pruebas.

### 3.5 HARDWARE Y MECÁNICA

#### 3.5.1 Ensamblaje del Brazo Robótico

El ensamblaje del brazo robótico RA-01C se lo puede encontrar en la página oficial de Images Co. [20].

**FIGURA 14. Brazo Robótico RA-01C**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

El brazo robótico utiliza 5 servomotores HI-TEC:

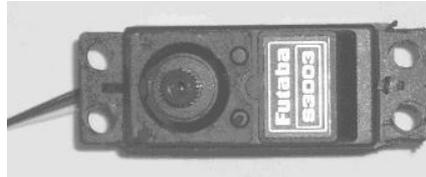
**Tabla I. Servos a usarse**

| <b>Servomotor</b> | <b>Cantidad</b> | <b>Especificación<br/>Técnica</b> |
|-------------------|-----------------|-----------------------------------|
| HS-645MG          | 1               | Anexo B-1                         |
| HS-422            | 2               | Anexo B-2                         |
| HS-322 HD         | 2               | Anexo B-3                         |

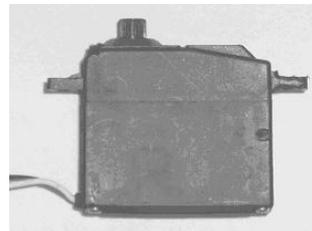
### **3.5.2 Construcción de Brazo Empuja frascos**

La construcción del brazo empuja frascos es sencilla, ya que por la operación que realizará no necesita que el cuerpo sea muy elaborado.

En primer lugar, usaremos un servomotor Futaba S-3003, como se puede observar en la figura 15 y 16.

**FIGURA 15. Vista Superior Futaba S3003**

Fuente: Gabriela Sánchez, laboratorio de Microcontroladores

**FIGURA 16. Vista lateral Futaba S3003**

Fuente: Gabriela Sánchez, laboratorio de Microcontroladores

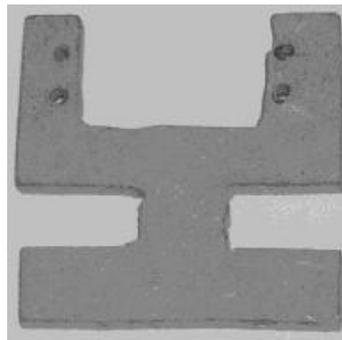
Se utilizarán también 4 pernos cabeza de tornillo de 4x1 ½” para ajustar el motor a la pieza de MDF. Como se muestra en la figura 17.

**FIGURA 17. Tornillos para Motor**

Fuente: Gabriela Sánchez, laboratorio de Microcontroladores

Una pieza de MDF de 5x5cm, como la que se muestra en la figura 18, que será utilizada para sostener el servomotor y ajustar a la barra metálica que se encuentra sujeta a la cámara de iluminación usada en la plataforma de desarrollo para los procesos de visión con LabVIEW, quedando situado el brazo empuja frascos sobre la banda transportadora.

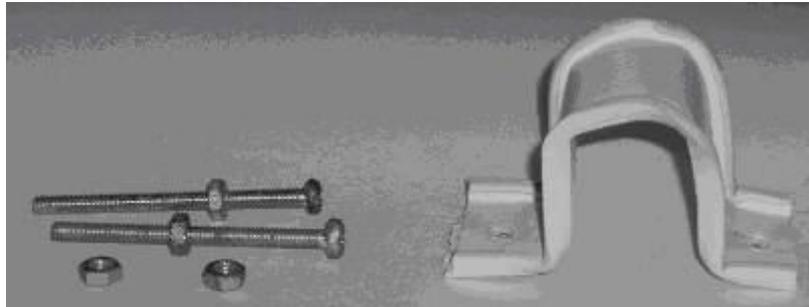
**FIGURA 18. Soporte MDF**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

Se necesitará una pieza plástica en forma de U para poder sujetar la pieza de la figura 18 a la barra metálica que sostendrá al brazo empuja frascos, 2 pernos de 4x2" cabeza de tornillo y doble tuerca.

**FIGURA 19. U plástica para soporte al riel metálico**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

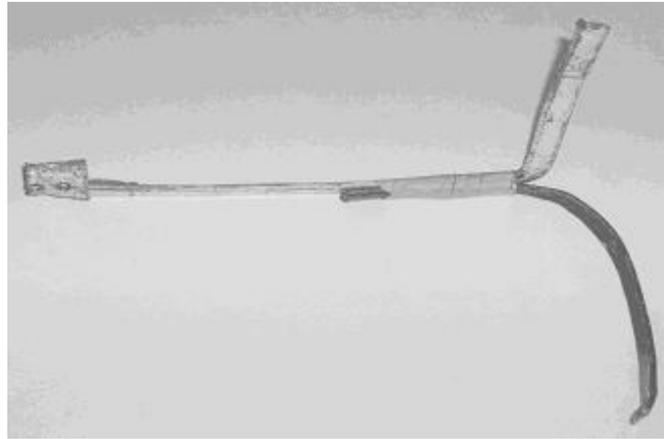
Otras de las piezas claves son una pieza plástica pequeña que servirá como acople, un tornillo tripa de pato 6x $\frac{1}{2}$ " , que unirá esta pieza al rotor del servomotor, y 2 pernos cabeza de tornillo de 4x $\frac{3}{4}$ " con tuerca. Como se muestra en la figura 20.

**FIGURA 20. Accesorios de Servomotor**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

**FIGURA 21. Cuerpo del Brazo Empuja Frascos**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

La figura 21 nos muestra como es el brazo empuja frascos, es una lámina de aluminio con dos extensiones en los extremos la una paralela al frasco a tratar y la otra horizontal con una forma cóncava que coincide con la forma del mismo.

Una vez que tenemos todos los materiales empezamos a unir las piezas de la siguiente manera.

Primero la lámina de aluminio se ajusta al acople por medio de los pernos como se muestra en la figura 22.

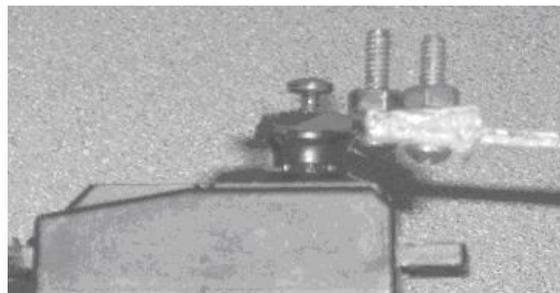
**FIGURA 22. Unión de Brazo con acople de Servo**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

A continuación el acople se lo une al servomotor con el tornillo tripa de pato, figura 23.

**FIGURA 23. Unión de acople con el Servomotor**

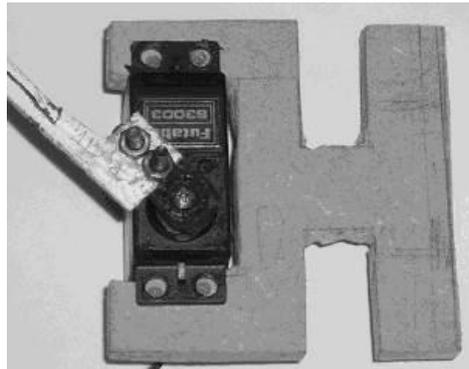


**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

A estas alturas el brazo empuja frascos está construido en su totalidad, ahora lo necesario será acoplarlo al soporte de

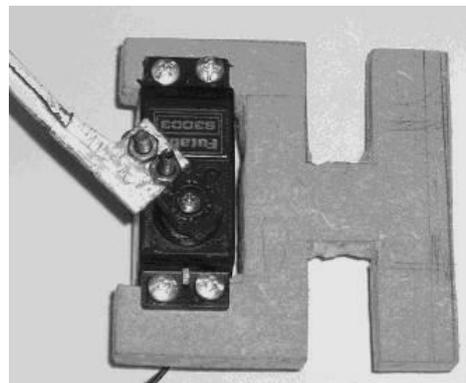
MDF como se ve en la figura 24, y luego sujetarlo con los pernos de 4x1 ½” como la figura 25.

**FIGURA 24. Acople Servo y MDF**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

**FIGURA 25. Ajuste de Servomotor y MDF**

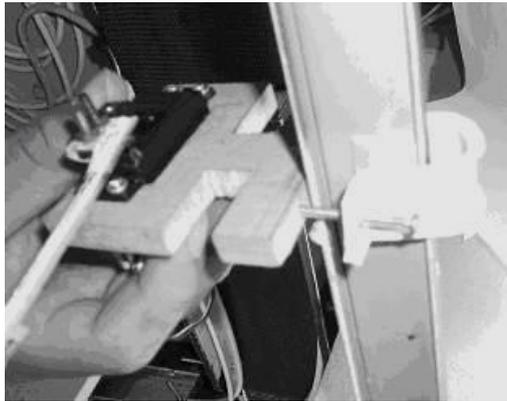


**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

Por último se colocará el soporte en la viga de aluminio colocada en la cámara de iluminación, la unión se coloca

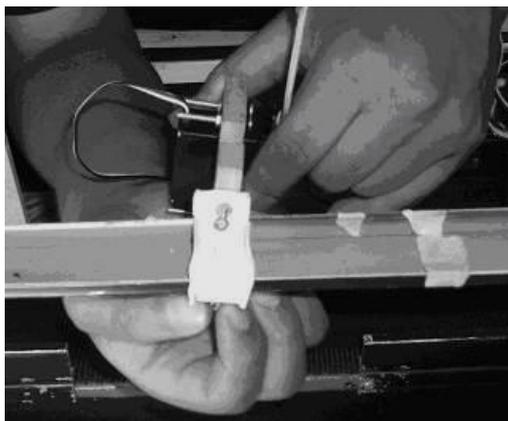
con los pernos de 4x2" con la dos tuercas, de esta manera se asegura de que no se aflojen las piezas y sus juntas. Como en la figura 26 y 27.

**FIGURA 26. Ajuste de MDF al Riel de Aluminio**



**Fuente: Gabriela Sánchez y Carlos Alvarado, laboratorio de Microcontroladores**

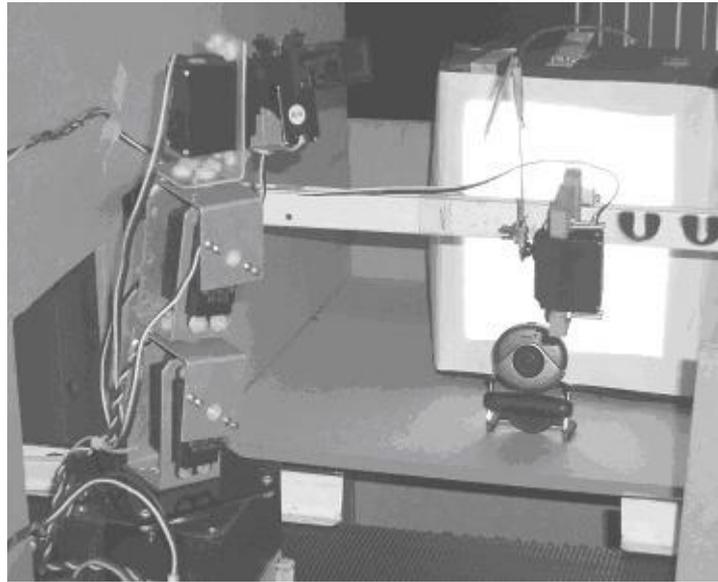
**FIGURA 27. Ajuste MDF al Riel de Aluminio vista posterior**



**Fuente: Gabriela Sánchez y Carlos Alvarado, laboratorio de Microcontroladores**

Una vez finalizado el ensamblaje tanto del brazo robótico como del brazo empuja frascos, una vista general de ellos en la plataforma sería como la figura 28.

**FIGURA 28. Vista General de los Brazos**



**Fuente: Gabriela Sánchez, laboratorio de Microcontroladores**

### **3.5.3 Diseño de la Tarjeta Controladora**

Esta tarjeta se la usará para controlar el brazo robótico y el brazo empuja frascos. Sirve de interfaz entre los brazos y el instrumento virtual de LabVIEW.

Cabe mencionar que el Kit del brazo robótico RA-01C incluía una tarjeta controladora construida con un PIC16F873, la cual se decidió cambiar por una tarjeta con un dsPIC30F4012. [24]

El dsPIC30F4012 tiene la posibilidad de introducir más código de instrucción ejecutándose a mayor velocidad. También ha sido fabricada para el manejo de motores por control de ancho de pulso a una mayor resolución, dándonos así una mayor precisión en el manejo de nuestros servomotores para el brazo empuja frascos y con mayor razón para el brazo robótico. También tiene la posibilidad de manejo de sensores, de comunicación y de programación del microcontrolador en circuito.

### **3.5.3.1 Materiales de la Tarjeta dsPIC30F4012**

Los materiales electrónicos utilizados en la elaboración de la tarjeta controladora con dsPIC30F4012 fueron:

- Un dsPIC30F4012 (IC1)
- Un MAX-232, Convertidor a RS-232. (IC2)
- Un Cristal 10Mhz (Q1).
- Led indicador (LED1)
- capacitores 22pF (C1, C2)
- capacitores electrolítico 1uF
- (C5,C6,C7,C8,C9)
- 1 capacitor electrolítico de 10uF (C4)
- 1 capacitor fijo de 0.1uF (C3)
- 1 resistencia 1K ohm, 1/4w (R1)
- 1 resistencia 4.7k ohm, 1/4w (R2)
- 1 jack para alimentación de 5v (X1)
- 1 conector serial hembra (X2)
- borneras tornillo (X3,X4,X5)
- Espadines para conectar servo motores y para puentes e ICSP(SV1-SV10, JP1, JP2)
- Puentes para seleccionar programación
- Botonera (S1)

### 3.5.3.2 Características de la tarjeta

Las características principales de la tarjeta diseñada con dsPIC30F4012 son las siguientes:

- Comunicación Serial con la PC.
- 6 Salidas PWM de motor: 3 independientes y 3 complementarias.
- Salidas PWM usando la salida de comparación o se pueden usar como entradas de interrupción externas.
- Programación Serial en Circuito (ICSP).
- 6 Entradas A/D (Analógico/Digital) para futuras pruebas con sensores o para interrupción por cambio de estado.
- Una entrada por interrupción externa o como salida digital.
- Comunicación serial alternativo o también puede ser usado como entrada o salida digital, también para interrupción por cambio de estado.
- Un botón para reinicializar la tarjeta.

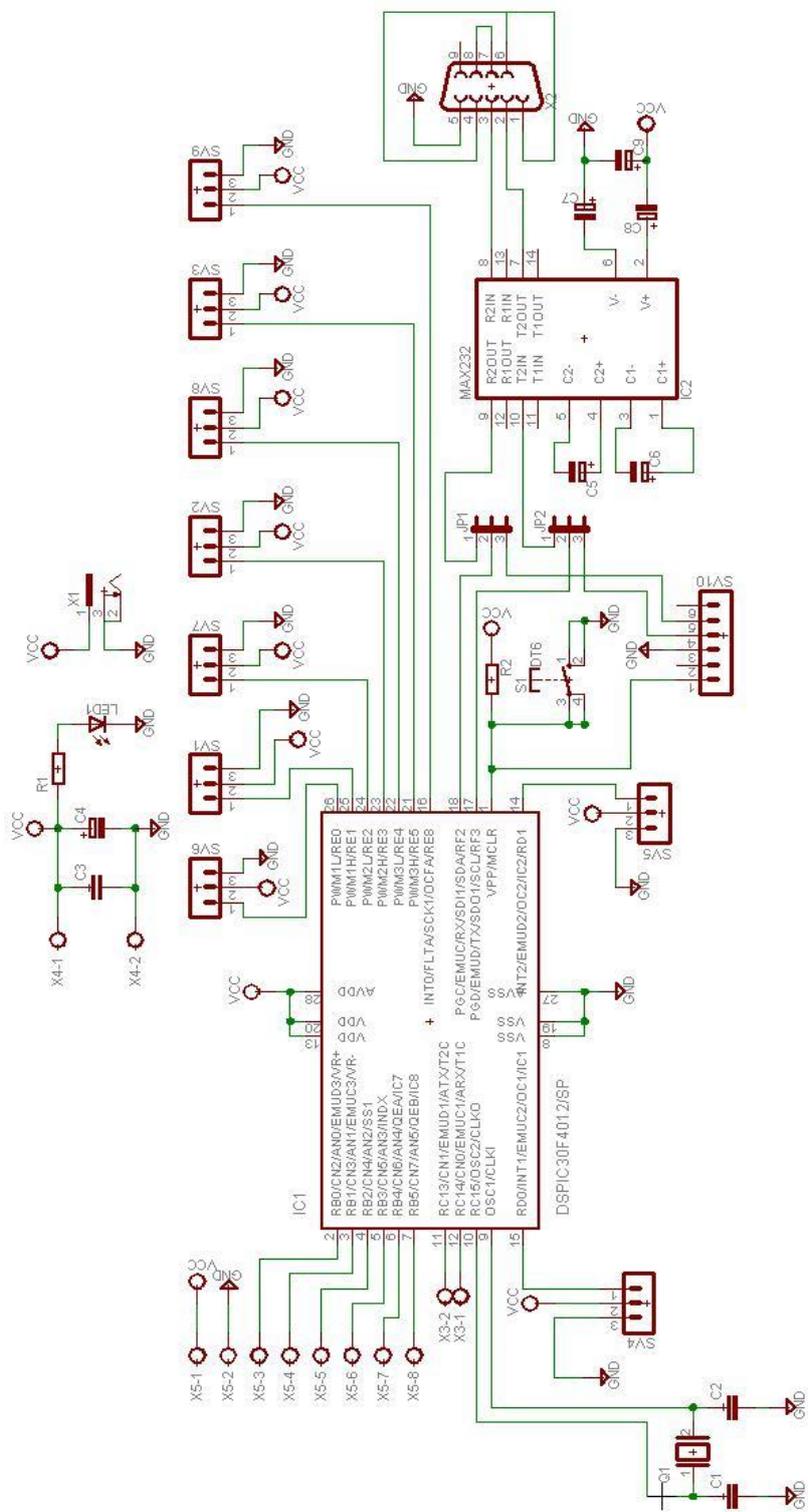
- Alimentación de 5 Voltios.

### **3.5.3.3 Esquemático y Placa de la tarjeta**

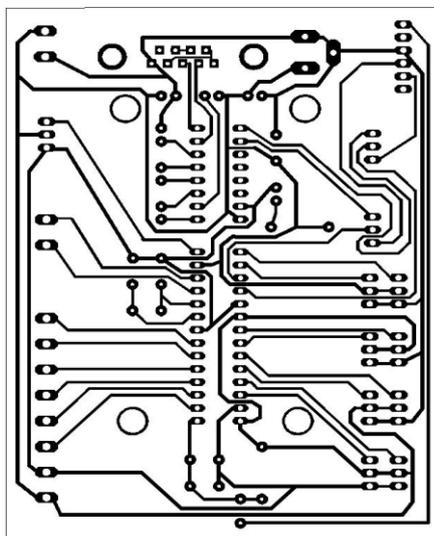
El diseño de la tarjeta dsPIC30F4012 se la realizó con el programa EAGLE Versión 4.16 para Windows. Esta aplicación nos ayudó a realizar el esquemático y el circuito impreso de nuestra tarjeta.

En las figuras 29, 30 y 31 podemos apreciar el circuito esquemático el diagrama de circuito impreso y de la vista superior de la tarjeta respectivamente.

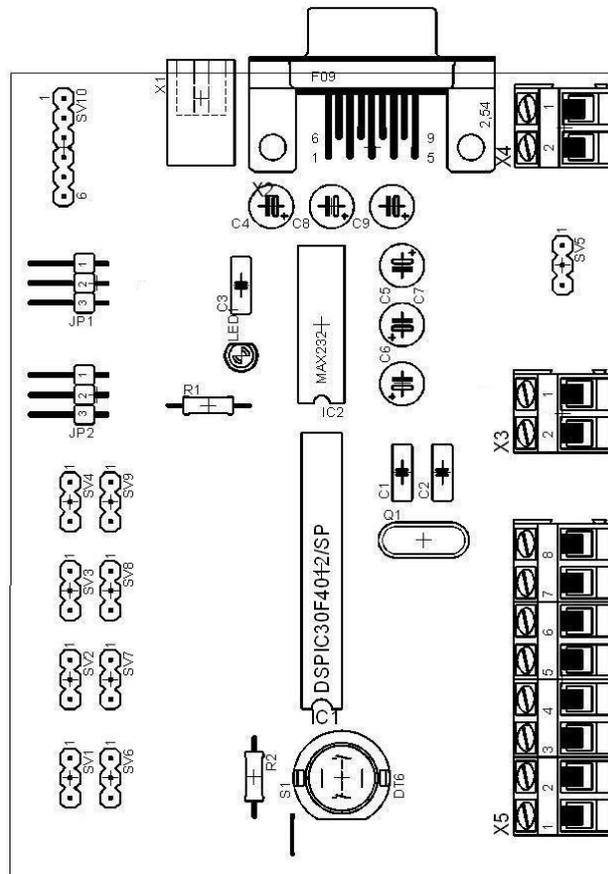
FIGURA 29. Esquemático de la tarjeta dsPIC30F4012



**FIGURA 30. Circuito impreso de la tarjeta dsPIC30F4012**

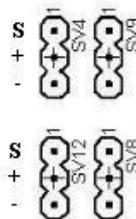


**FIGURA 31. Vista superior de la tarjeta dsPIC30F4012**



Los espadines SV1 al SV9 tienen la disposición para conectar los servomotores. La figura 32 muestra como se dispone el PIN S que estará conectado a uno de los pines del dsPIC para la señal PWM u otra señal además de las conexiones de alimentación de +V y GND.

**Figura 32 Disposición de Señales en los espadines**



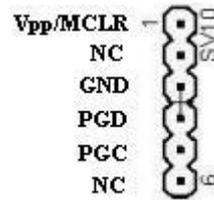
Los 5 servomotores del brazo robótico: Servo1, Servo 2, Servo 3, Servo 4, Servo 5, van a ser conectados en SV1, SV2, SV3, SV4, SV5 respectivamente. El brazo empuja frascos se lo conectará en SV6. Si se puede apreciar el diagrama esquemático vamos a usar las tres salidas PWM de lógica positiva PWM1H, PWM2H, PWM3H para los tres primeros servomotores del brazo y los otros dos restantes se los conecta en OC1 y OC2 que van a operar en modo PWM. SV7 corresponde al pin 26 como salida digital la cual va a salir una señal PWM pero realizada por software.

Las borneras X5 se pueden usar para el uso de sensores ya que están conectadas a los pines analógicos/digitales en las borneras X5-3 a X5-8. X5-1 corresponde a +5V y X5-2 a GND para hacer pruebas con estos pines, principalmente cuando se quieran usar los pines para usar interrupciones por cambio de estado.

Se puede dar alimentación de voltaje de +5V ya sea en la Bornera X4 ó en el conector JACK X1. En X4-1 son los +5V y X4-2 es GND.

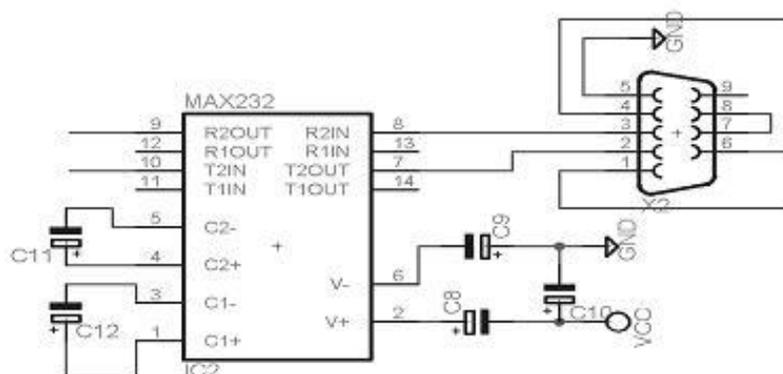
La Bornera X3 se lo usa si en algún momento se puede usar comunicación serial alternativa (AUART) en vez de usar el puerto de comunicación serial por el conector DB9. X3-2 Es para la recepción (ARx) y el X3-1 son para la transmisión (ATx).

SV10 se lo usa para la programación del dsPIC30F4012 en circuito con un programador que soporte ICSP. Los espadines JP1 y JP2 tienen la función de seleccionar la programación por ICSP o la comunicación serial por el conector DB9 con la ayuda de puentes de color azul. Si vemos la figura de la vista superior de la tarjeta, para seleccionar ICSP se sitúan los dos puentes que conecten los pines 2 y 3 de JP1 y JP2 y para la comunicación serial se sitúan los puentes con los pines 1 y 2 de JP1 y JP2. La conexión correspondiente ICSP a cada espadín se muestra en la figura 33.

**FIGURA 33. Espadines para conexión ICSP**

El MAX232 es utilizado en nuestra tarjeta para la comunicación serial. Su propósito es cambiar los niveles de voltaje TTL a los del estándar RS232 y viceversa. En RS232 un cero lógico representa en voltaje un valor positivo de entre +3V y +12V, y un uno lógico representa en voltaje un valor negativo de entre -3V y -12V. El MAX232 se conectará a la computadora por el conector DB9. [30]

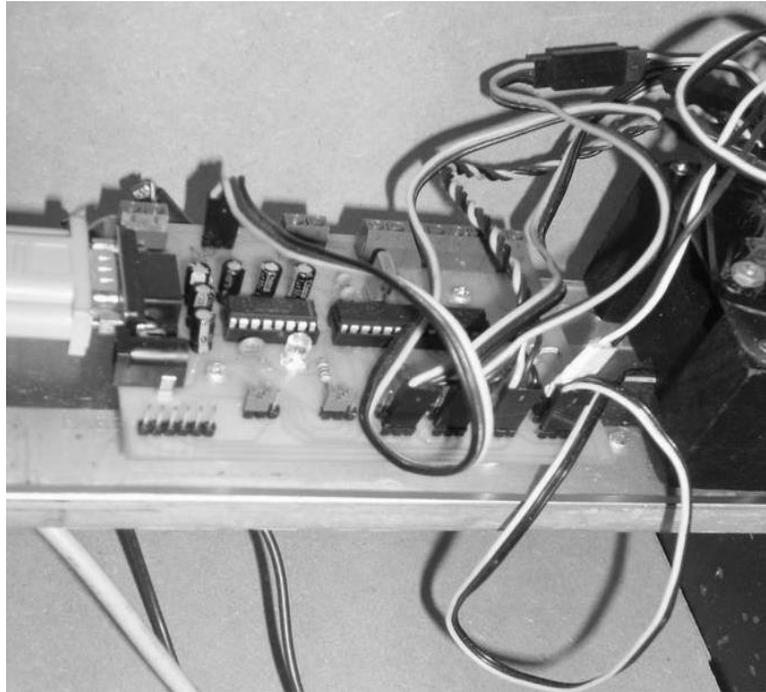
**FIGURA 34. Conexión del MAX232**



La conexión del MAX232 se hace con los pines 18 (Rx) y 17 (Tx) del dsPIC30F4012 con los pines 9 (R2OUT) y 10 (T2IN) del MAX232 si lo seleccionamos con los puentes JP1 y JP2.

La tarjeta dsPIC30F4012 colocada en la base junto al brazo robótico conectado así como el brazo empuja frascos, se lo puede apreciar en la figura 35, así como en el ANEXO C podemos ver el diseño de placa en tamaño real.

**FIGURA 35. Tarjeta dsPIC30F4012**



Fuente: Carlos Alvarado, Laboratorio De Microcontroladores

### **3.6 FIRMWARE Y PROGRAMACIÓN**

En esta sección detallamos el lenguaje de programación que vamos a usar, el programa compilador que se usó para programación del firmware, el dispositivo programador del dsPIC30F4012 y de la velocidad a la que va operar el programa que se realizó.

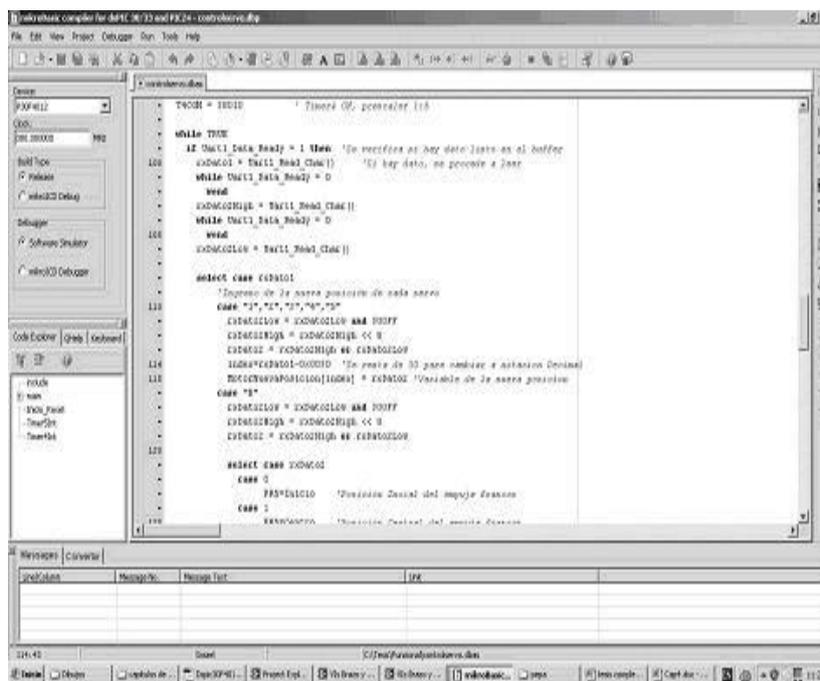
#### **3.6.1 Lenguaje de Programación**

El firmware esta programado en el lenguaje de programación llamado BASIC. Un lenguaje profesional y de alto nivel que nos ayuda mucho en la programación y nos simplifica algunos problemas en la sintaxis ya que en lenguaje ASSEMBLER la programación del firmware tendría más líneas de código que en BASIC y seria más complejo su depuración y mantenimiento.

### **3.6.2 Compilador a usar**

El software que vamos a usar para la programación en BASIC del firmware es el compilador Mikrobasic para dsPIC 30/33 Versión 5.0. Este compilador es muy útil ya que fue diseñado para la programación de dsPICs. [Figura 36]

**FIGURA 36. Área de trabajo del programa Mikrobasic**

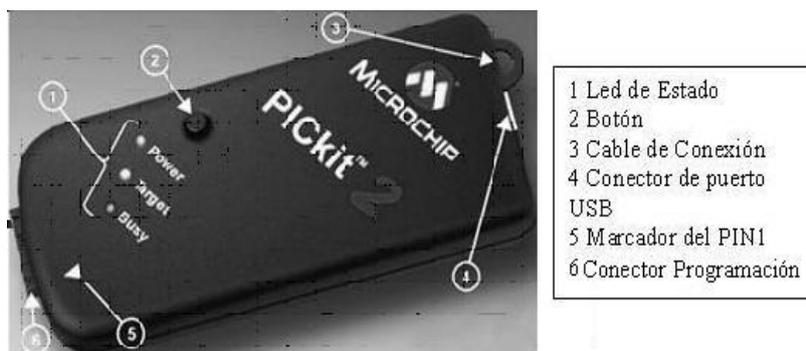


Fuente: Carlos Alvarado, Laboratorio Microcontroladores

### 3.6.3 Dispositivo de programación a usarse en la tarjeta dsPIC30F4012

Para este componente utilizamos la característica de los microcontroladores de Microchip que permite reprogramarlo en circuito por medio de la interfase de programación ICSP y aprovechamos el programador de Microchip PICkit™ 2 [Figura 38] como herramienta de desarrollo de bajo costo [21].

FIGURA 37. Programador PICkit 2



Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/51553E.pdf>

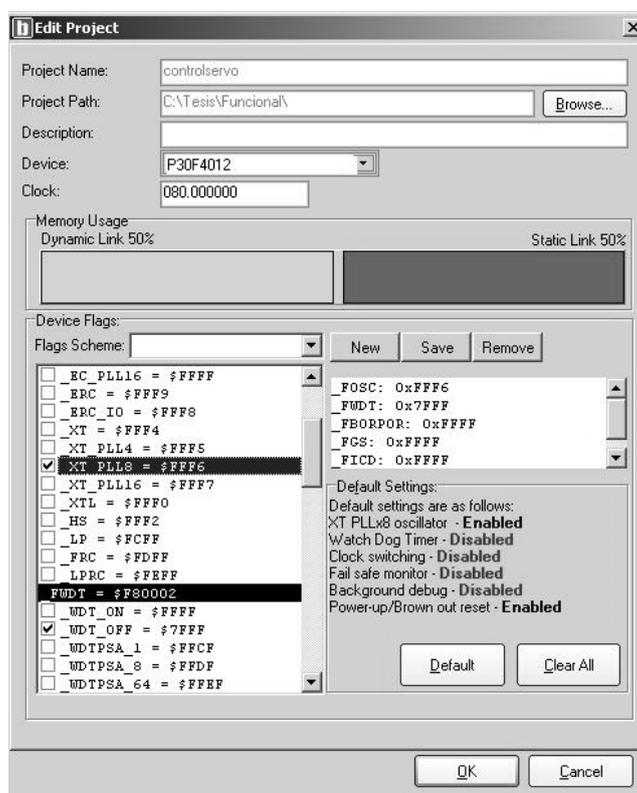
### 3.6.4 Tiempo de ciclo de instrucción

El firmware va a operar a una Frecuencia Focs=80Mhz, Para esto necesitamos un Cristal de 10Mhz y se selecciona el multiplicador de 8 para obtener los 80MHz. Cada ciclo de instrucción va demorar 4 veces el tiempo de oscilación (12.5ns), entonces el ciclo de instrucción sería 50ns. El tiempo en que se ejecuta cada instrucción estará asociado a código en ASSEMBLER y no en BASIC.

El compilador Mikrobasic tiene la opción de mostrarnos la traducción en ASSEMBLER de un programa hecho en el lenguaje BASIC.

Para escoger el multiplicador de 8 se debe hacer cambios en el registro de configuración del dsPIC30F4012. Para ello al hacer un nuevo proyecto en mikroBasic nos presenta un cuadro de diálogo que nos muestra que dsPIC vamos a usar, la velocidad de reloj y las opciones de operación que nos ofrece mediante el registro de configuración del dsPIC. En el menú *Project* se da clic en *Edit project*. En la ventana *Edit Project* se selecciona **XT\_PLL8** en el cuadro DEVICE FLAGS [figura 38]

**FIGURA 38. Cuadro de dialogo para la configuración de bits del registro de configuración del dsPIC30F4012**



Para poder ver el equivalente en Asembler de nuestro programa se selecciona del menú **View** la opción **View as Assembly**. Con esto podemos ver y analizar su respectivo código en Assembler.

# CAPITULO 4

## 4. LABVIEW Y DSPICS

### 4.1. GENERALIDADES DE LABVIEW

LabVIEW 8.5 es el programa que vamos usar para enviar los comandos y ejecutar acciones que harán movimientos específicos en el brazo robótico y en el brazo empuja frascos. LabVIEW es una poderosa herramienta para la adquisición de datos en procesos industriales. La programación gráfica nos ayuda de manera fácil en la realización de nuestra interfaz para el brazo robótico y el brazo empuja frascos.

#### 4.1.1 LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un entorno de desarrollo basado en

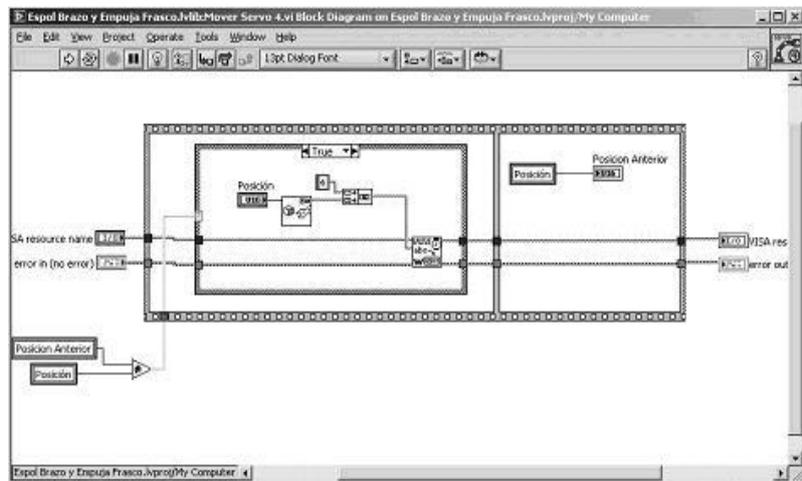
programación gráfica. Utiliza símbolos gráficos en lugar de lenguaje textual para describir acciones de programación. Está totalmente integrado para la comunicación con hardware y tarjetas de adquisición de datos.

Los sistemas tradicionales de automatización y medida consisten en instrumentos específicos para tareas específicas. Normalmente se está obligado a diseñar el sistema desde cero y ello conlleva poseer un buen conocimiento de programación de ordenadores. Se puede decir que en los sistemas tradicionales el hardware define el sistema. Todo esto cambia usando el concepto de instrumentos basados en ordenador o instrumentos virtuales. De este modo se pueden diseñar sistemas de automatización y medida de bajo costo. La programación gráfica con LabVIEW permite a los no programadores un método fácil para implementar aplicaciones complejas de comprobación, medida y automatización [22].

#### 4.1.2 Instrumentos virtuales

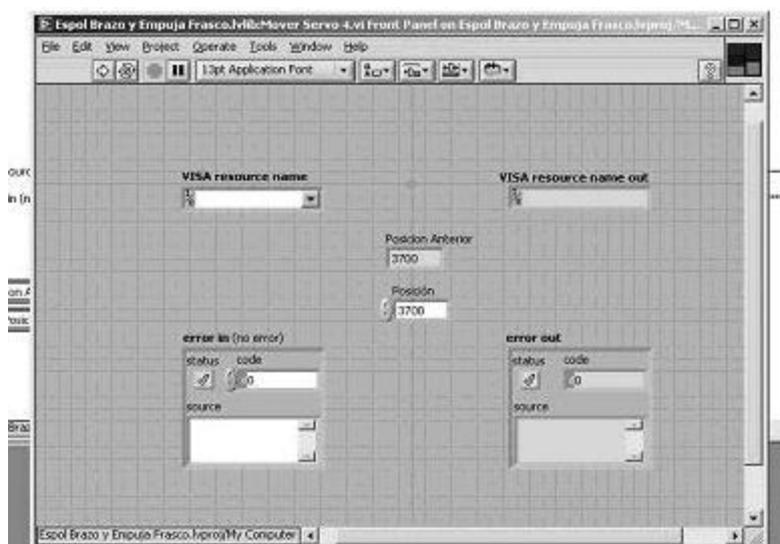
Los ficheros generados con LabVIEW se llaman Instrumentos Virtuales, VIs. Cada VI se compone de dos partes principales: el **panel frontal** [Figura 39] o interfase con el usuario y el **diagrama de bloques** [Figura 40] o código fuente y cada VI tiene su **icono y conector**

FIGURA 39. Ejemplo de Diagrama de Bloques



Fuente: Carlos Alvarado, Laboratorio de Microcontroladores

**FIGURA 40. Ejemplo de Panel Frontal**



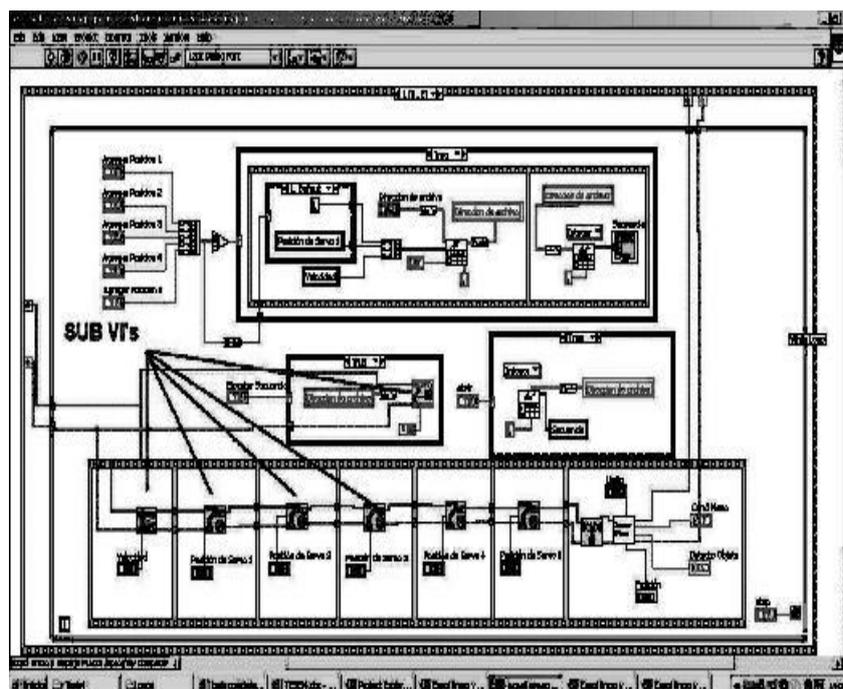
**Fuente: Carlos Alvarado, Laboratorio de Microcontroladores**

### 4.1.3 Instrumentos Virtuales de bajo nivel (SubVIs)

Los SubVIs sirven para la reutilización de código de bloques de programa y construcción de jerarquías. Normalmente, al diseñar una aplicación en LabVIEW, se comienza desde el VI más general definiendo las entradas y salidas de la aplicación. Después se crean SubVIs que realizan tareas más sencillas dentro del VI general. Este método de diseño es una de las ventajas de LabVIEW. Se pueden diseñar fácilmente aplicaciones complejas utilizando una estructura jerárquica y usando elementos comunes varias veces dentro

de la aplicación. El uso de SubVIs permite realizar aplicaciones fáciles de comprender, depurar y mantener.

FIGURA 41. Ejemplo de utilización de SubVIs



Fuente: Carlos Alvarado, Laboratorio de Microcontroladores

## 4.2. GENERALIDADES DE DSPICs.

Microchip Technology Inc. Uno de los fabricantes en los últimos años en la comercialización de los microcontroladores de 8 bits, ha querido poner en manos de sus clientes nuevos componentes llamados dsPICs, que les permita introducirse en las aplicaciones contemporáneas del proceso digital de señales de forma sencilla

basada en arquitecturas y repertorios de instrucciones conocidos [23].

Microchip ha dividido los diferentes modelos de la serie dsPIC en tres familias caracterizadas por su área de aplicación, nombradas a continuación:

- Familia de dsPIC de Propósito General
- Familia de dsPIC de Control de Sensores
- Familia de dsPIC de Control de Motores

#### **4.2.1 Familias dsPIC para el control de motores**

La familia de los dsPIC30F [Anexo D] reúne siete modelos para soportar aplicaciones de control de motores tanto de corriente continua, de inducción monofásica y trifásica, pudiendo también ser utilizados en sistemas de alimentación ininterrumpidas (UPS), fuentes de alimentación conmutadas, inversores. Siendo adecuado en aplicaciones donde el algoritmo para el control de motores es de elevada complejidad [25].

### 4.3. ADQUISICIÓN DE DATOS CON LABVIEW.

Basándonos en como controlará el dsPIC30F4012 al brazo robótico y al brazo empuja frascos procedemos a realizar los instrumentos virtuales desde LabVIEW. Los instrumentos Virtuales que nosotros vamos a realizar van a ser las que vemos en la figura 42:

**FIGURA 42. VI's para brazo robótico y brazo empuja frascos**



#### 4.3.1 VI's de Comunicación

Los VI's de Comunicación se encargarán de Abrir la comulación serial, de cerrar la comunicación serial, y de limpiar el buffer del puerto. Son 2 VI's de comunicación:

- Iniciar Puerto.
- Cerrar Puerto.

#### 4.3.1.1 Iniciar Puerto

Sirve para abrir el puerto para la comunicación, el tiempo de espera de recepción de datos, la tasa de transmisión, el control de flujo, Paridad, número de bits de datos, bits de parada. También si al inicializar se necesita hacer una configuración por defecto al brazo robótico y al brazo empuja frascos.

FIGURA 43. VI Iniciar Puerto

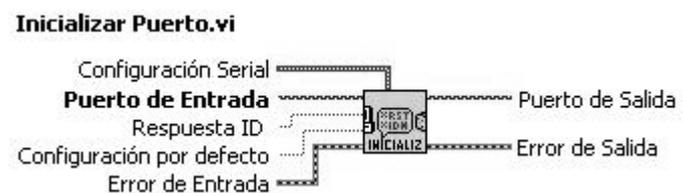
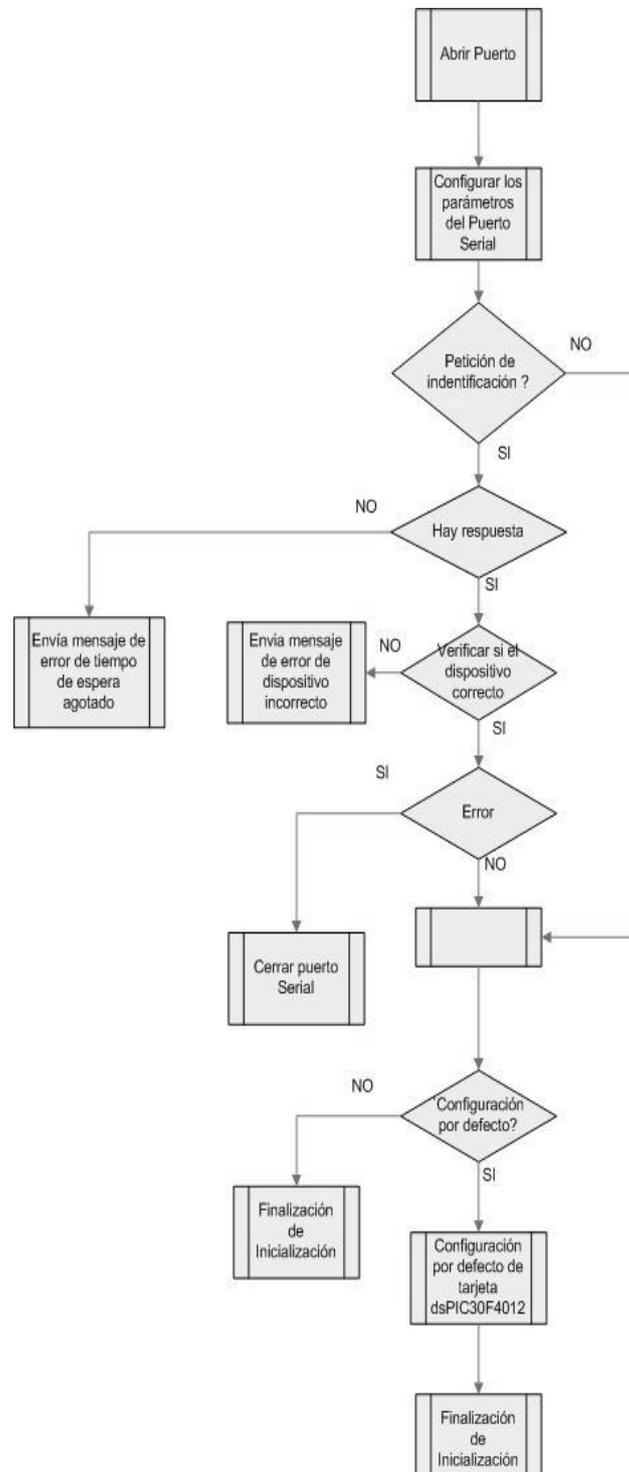


FIGURA 44. Diagrama de Flujo de VI Iniciar puerto



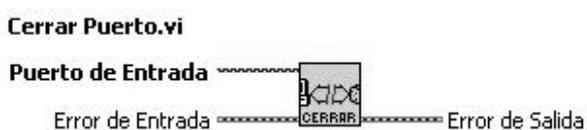
La lógica utilizada en el VI Iniciar puerto fue la siguiente:

- Se selecciona el puerto serial, se configura el mismo a una tasa de transmisión y recepción de 9600 bps, sin bit de paridad, sin control de flujo y con un bit de parada.
- Si se escoge la acción de identificación de dispositivo, enviamos una petición para identificar si es la tarjeta dsPIC30F4012. Si hay un problema de comunicación, o me envía información diferente de "200" enviara mensajes de error. Si hay error se procede a cerrar el puerto.
- Si no hay error en la comunicación y el dispositivo se identifica como el correcto se procederá a la configuración de la tarjeta para la inicialización de las posiciones de los servomotores del brazo robótico y del brazo empuja frascos. Esta configuración inicial o por defecto se la puede escoger como o no.

### 4.3.1.2 Cerrar Puerto

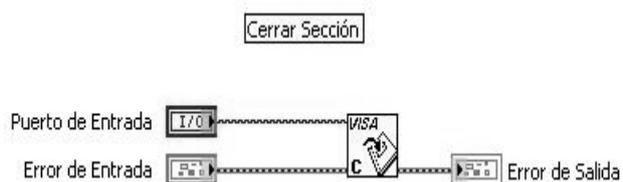
Sirve simplemente para cerrar la comunicación en el puerto serial.

**FIGURA 45. VI Cerrar Puerto**



Para cerrar la comunicación después de una aplicación con los brazos, se procederá a usar este VI ingresando el puerto que se abrió al iniciar una aplicación.

**FIGURA 46. Estructura Cerrar Puerto**



### **4.3.2 VI's de Configuración**

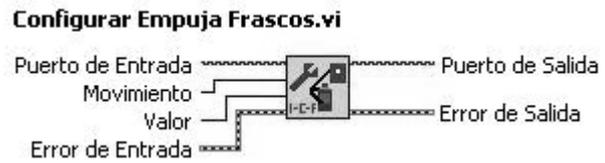
Los VI's de configuración son tres y nos sirven para configurar la velocidad de los servomotores, la posición inicial, central, final del servomotor del brazo empuja frascos. También para resetear el brazo robótico y el brazo empuja frascos con los valores de configuración por defecto similares al prender la tarjeta. Los VIs son:

- Configurar Empuja Frascos.
- Configurar velocidad servos.
- Configuración por defecto.

#### **4.3.2.1 Configurar Empuja frascos**

Nos ayuda a configurar tres posiciones; la posición inicial, central y final del empuja frascos, mediante el conector movimiento del VI, siendo estas posiciones diferentes a las que están por defecto.

**FIGURA 47. VI Configurar Empuja frascos**



#### 4.3.2.2 Configurar velocidad de servos

Nos configura un valor de retardo para el movimiento a la posición deseada. Si mayor es el valor, más demorará en ubicarse a la posición deseada.

Se envía una trama de 3 bytes indicando la acción de velocidad y el valor de la misma.

**FIGURA 48. VI Configurar Velocidad de Servos**



**FIGURA 49. Diagrama de Flujo configuración de velocidad de Servos**



Pensando que este VI será utilizado en un lazo cerrado se lo diseñó de tal manera que envíe una velocidad si y solo si existe un variación con respecto a la anterior.

#### **4.3.2.3 Configuración por defecto**

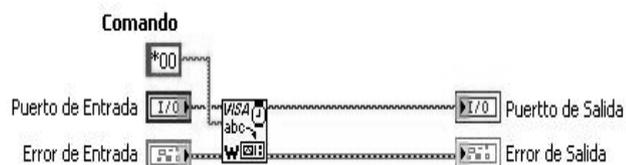
Manda un comando de 3 bytes para el reseteo con los valores predeterminados de velocidad y posiciones del brazo robótico y brazo empuja frascos al prender la tarjeta.

**FIGURA 50. VI Configuración Por Defecto**

**Configuración Por Defecto.vi**



**FIGURA 51. Estructura Configuración por defecto**



Como se observa en la figura 51 la estructura de configuración por defecto se envía el carácter asterisco (\*) para pedir la configuración por defecto, según la tabla II, PAG 98.

### 4.3.3 VI's de Acción

Los VI's de Acción son los encargados de enviar los comandos para el movimiento de los servomotores tanto para el brazo robótico como para el brazo empuja frascos, ejecuta movimientos independientes como ejecutar secuencias de movimientos para el brazo robótico.

Los Instrumentos virtuales son los siguientes:

- Mover Servos
- Mover Servo 1
- Mover Servo 2
- Mover Servo 3
- Mover Servo 4
- Mover Servo 5
- Empuja Frascos
- Enviar Secuencia

#### 4.3.3.1 Mover Servos

Este VI mueve los 5 servomotores en el cual se puede seleccionar que servo se requiere mover y a que posición.

FIGURA 52. VI Mover Servo

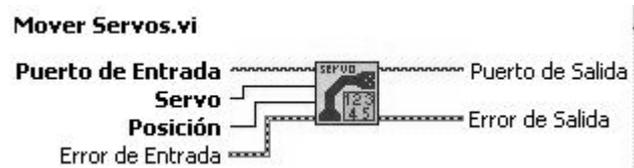
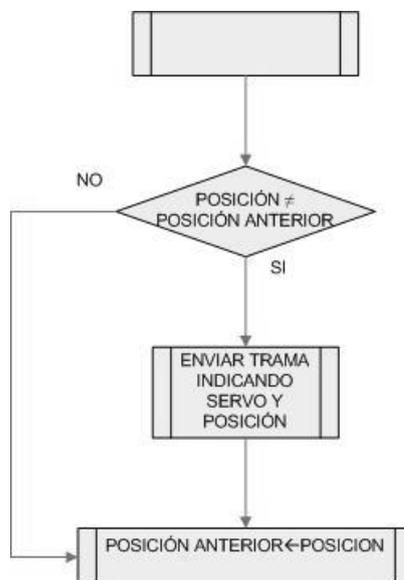


FIGURA 53. Diagrama de Flujo para Mover Servos

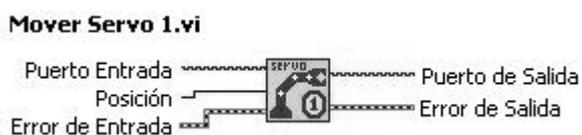


Se llevará el mismo formato de operación, se enviará una trama vía serial si y solo si existe una variación entre la posición anterior y la nueva, podemos utilizar como formato guía este diagrama de flujo para los VI's de mover servo 1, mover servo 2, etc, ya que operan de manera similar.

### 4.3.3.2 Mover Servo 1

Envía el comando a la tarjeta para que mueva el servomotor 1 ó de la base a la posición deseada.

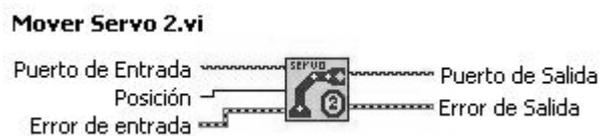
FIGURA 54. VI Mover Servo I



### 4.3.3.3 Mover Servo 2

Envía el comando a la tarjeta para que mueva el servomotor 2 a la posición deseada.

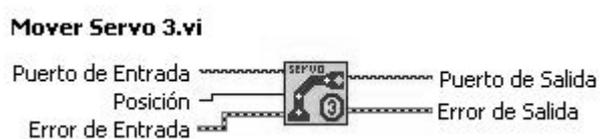
FIGURA 55. VI Mover Servo 2



#### 4.3.3.4 Mover Servo 3

Envía el comando a la tarjeta para que mueva el servomotor 3 a la posición deseada.

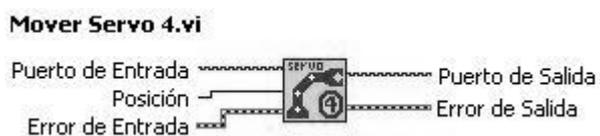
FIGURA 56. VI Mover Servo 3



#### 4.3.3.5 Mover Servo 4

Envía el comando a la tarjeta para que mueva el servomotor 4 ó de la muñeca a la posición deseada.

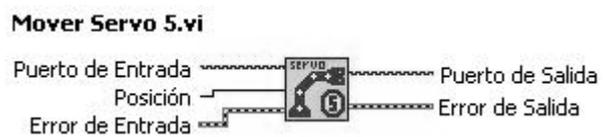
FIGURA 57. VI Mover Servo 4



#### 4.3.3.6 Mover Servo 5

Envía el comando a la tarjeta para que mueva el servomotor 5 ó la mano a la posición deseada.

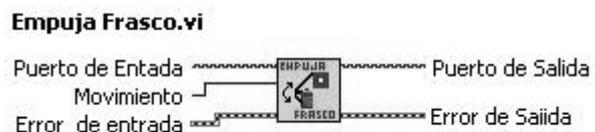
**FIGURA 58. VI Mover Servo 5**



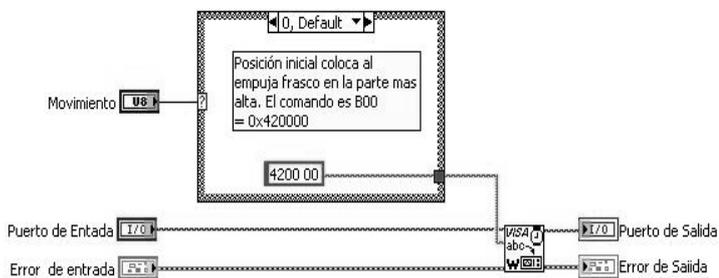
#### 4.3.3.7 Empuja Frascos

Este VI controla el brazo empuja frascos con la posibilidad de ubicarse en las 3 posiciones requeridas en los procesos de la plataforma de proyectos con dsPICs y visión robótica, ya sea para la posición inicial, posición central ó posición para detener frasco y posición final.

**FIGURA 59. VI Empuja Frascos**



**FIGURA 60. Estructura de Empuja Frascos**

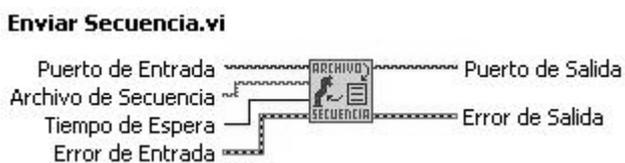


Se utiliza un proceso de selección según el caso, si es 0 se enviará la trama correspondiente a posición inicial, si es 1 a central o 2 a final.

#### 4.3.3.8 Enviar Secuencia.

Esta secuencia sirve para mover al brazo robótico mediante secuencia de movimientos que se leen en un archivo.

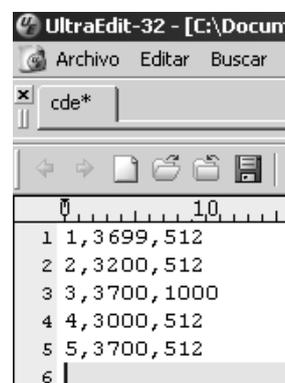
**FIGURA 61. VI Enviar Secuencia**



Este VI recibe el puerto de entrada, el archivo donde se encuentra la secuencia de posiciones y su respectiva velocidad, el tiempo de espera de posicionamiento del servomotor. Este tiempo se lo ingresa para no provocar un error de lectura ya que este VI espera un reconocimiento que enviará la tarjeta indicando que terminó el cambio de ciclo de trabajo de la señal PWM requerida para el posicionamiento del servomotor. El puerto de salida es un indicador del puerto usado con sus características originales una de ellas sería la tasa de transmisión.

El archivo de secuencia debe tener el siguiente formato:

**FIGURA 62. Archivo secuencia de movimiento**



**Motor1, Posición1, Velocidad1**

**Motor2, Posición2, Velocidad2**

.....

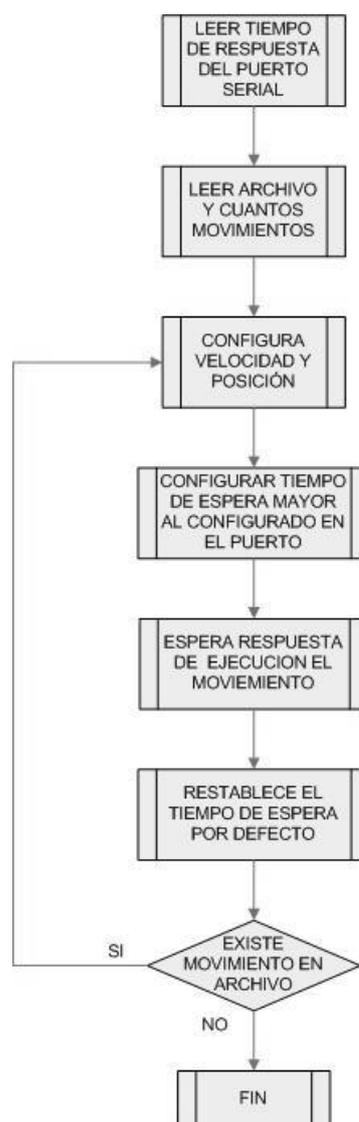
.....

En la figura 63 encontraremos el diagrama de flujo que explica el envío de la secuencia de movimiento, la explicación del mismo la damos a continuación.

- Primero se lee el archivo y observa cuantos movimientos tendrá el brazo robótico.
- Se envía velocidad, servo a mover y su respectiva nueva posición.
- Se cambia el tiempo de espera del puerto, a la cual el usuario estime conveniente. El tiempo de espera por defecto es 10s.
- Espera la respuesta de ejecución del movimiento.
- Restablece el tiempo de espera por defecto.
- Verifica si existe una nueva línea de movimientos, si la respuesta es afirmativa

regresa al lazo, si no termina la ejecución del VI.

**FIGURA 63. Diagrama de Flujo enviar secuencia**



#### 4.4. dsPICs CONTROLANDO EL BRAZO ROBÓTICO.

El dsPIC30F4012 controla la posición de los servomotores del brazo robótico y el empuja frascos mediante modulación PWM. El dsPIC se encargará de generar las 6 señales PWM, cinco para el brazo robótico y uno para el brazo empuja frascos. El periodo del tren de pulsos de la señal PWM de todos los servos es de 20ms y sus posiciones dependerán del ancho del pulso en alto (Ciclo de Trabajo).

En ésta y en la siguiente sección encontrará partes del código fuente. Todo el código completo lo puede ver en el anexo E.



#### 4.4.1 Controlando al brazo Robótico

Los tres primeros servos están conectados al módulo de control PWM del dsPIC. El cual tiene sus propios temporizadores.

```
'configurar la señal PWM a 50Hz de los PWM para los
tres primeros servos
Pwm_Mc_Init(800,1,%01110000,0)
PTCON.3=1
```

Se selecciona el prescalador de 1:16 ingresando un 1 en el bit 3 del registro PTCON del módulo de control de motor PWM dando lugar a que los 800Hz que se ingresaron en la función se divida en 16 lo que nos da 50Hz que es lo deseado [26].

Para mover los tres primeros servos se uso la función ***Pwm\_Mc\_Set\_Duty(duty,canal)*** donde *duty* es un valor decimal de 16 bits que cambia el ciclo de trabajo y *canal* se selecciona cual de los 3 canales PWM de Motor. **MotorPosicion[i]** es una variable que contiene la posición actual del servomotor *i* tal que *i* puede ser los servomotores 1, 2, 3, 4, 5. El canal estará asociado con los tres primeros servomotores para los tres canales PWM Motor.

```
Pwm_Mc_Set_Duty(MotorPosicion[i],i)
```

En los dos servomotores restantes  $i = 4, 5$  se uso los pines OC1 y OC2 respectivamente que forman parte de los módulos comparadores de salida (OUTPUT COMPARE) los cuales se los puede seleccionar en modo PWM para nuestro propósito. Para generar la señal PWM, estos módulos necesitan usar los temporizadores 2 y 3 del microcontrolador a diferencia del módulo de control PWM que es independiente. En nuestra programación, se selecciono el temporizador timer2 para OC1 y el temporizador timer3 en OC2. Para iniciar la señal PWM se usa ***Pwm\_Init (Frecuencia, Canal, Prescalador, Temporizador)*** donde el canal es 1 para OC1 y 2 para OC2.

```
'configurar la señal PWM a 50Hz de los PWM para los 'dos
servos restantes a la vez que se inicializa la 'posición de
los servos.
Pwm_Init(0, 1, 8, 2)    ' OC1 / prescalador de 8 / timer2
PR2=49999
Pwm_Init(0, 2, 8, 3)    ' OC2 / prescalador de 8 / timer3
PR3=49999
```

Para que cada salida OC1 y OC2 tenga una frecuencia de 50Hz o 20ms de período de la señal PWM, se necesita poner un valor de 16bits en los Registros *PR2* y *PR3* del

temporizador y donde la señal PWM estará gobernada por la fórmula siguiente [27].

$$PWM \text{ Period} = [PRy + 1] \bullet Tcy \bullet (TMRy \text{ Prescale})$$

$$PRy = \frac{PWM \text{ Period}}{Tcy \bullet (TMRy \text{ Prescale})} - 1, \text{ donde}$$

$$Tcy = \frac{4}{Fosc} ; y = 2, 3$$

$$PRy = \frac{50Hz}{(4/8 * 10^7 Hz) * 8} - 1$$

$$PRy = 49999$$

Nos da un valor de 49999 para PR2 y PR3 ya que los dos van a trabajar a 50Hz o 20ms de periodo. Aunque se lo puede ingresar en el campo *frecuencia* de la función *Pwm\_Init* nosotros lo hicimos modificando los Registros *PRy* donde  $y = 2, 3$

Para mover los servomotores de la muñeca y de la pinza, se cambia también el ciclo de trabajo mediante la función ***Pwm\_Set\_Duty(Duty, Canal)*** a cada salida OC1 y OC2 para los servomotores 4 y 5 respectivamente.

```

Pwm_Set_Duty(MotorPosicion[i],1)
Pwm_Set_Duty(MotorPosicion[i],2)

```

Dentro del firmware tenemos el manejo del posicionamiento de los servomotores así también de la velocidad a la que se van a posicionar los cinco servomotores del brazo robótico esta rutina es de mucha importancia, ya que los servomotores se moverán con mayor suavidad y estabilidad.

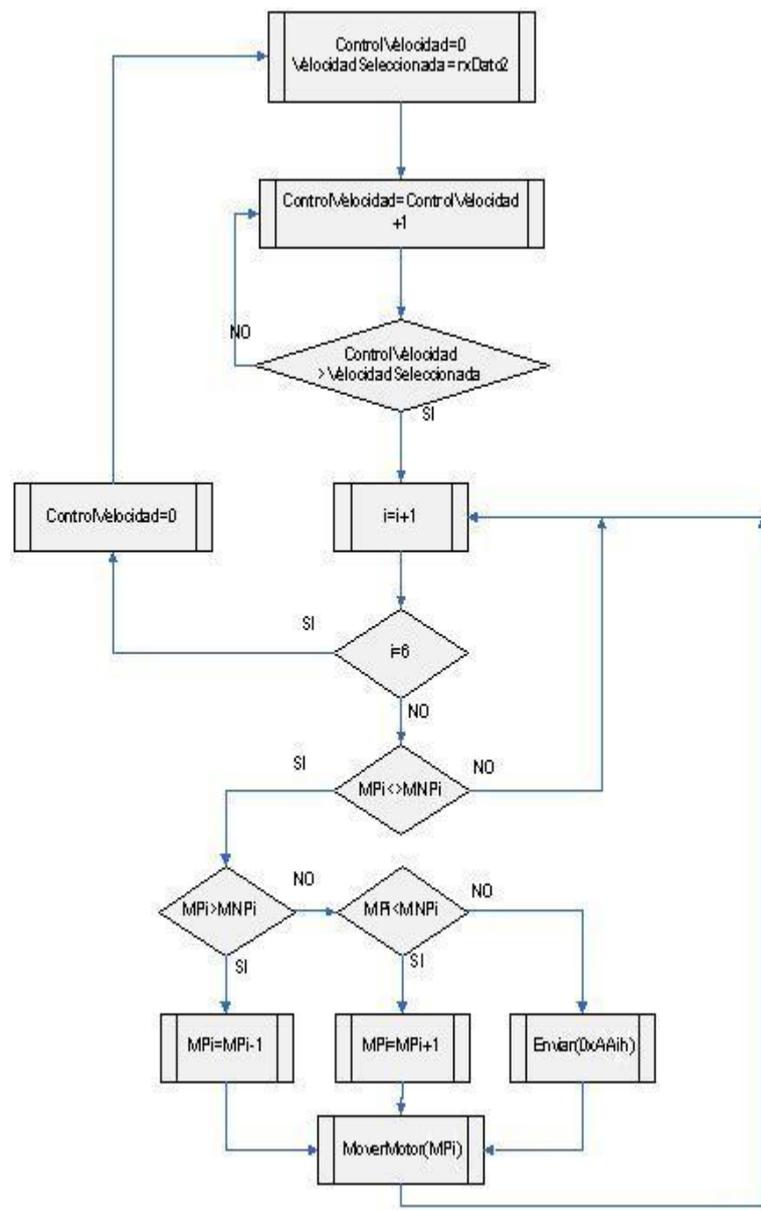
```

ControlVelocidad = ControlVelocidad + 1
  if ControlVelocidad >= VelocidadSeleccionada then
    for i = 1 to 5
      if MotorPosicion[i] <> MotorNuevaPosicion[i] then
'Verificamos si la nueva posición es menor a la actual
        if MotorPosicion[i] > MotorNuevaPosicion[i] then
'Si es así, se disminuye la posición del servo a mover
          MotorPosicion[i] = MotorPosicion[i] - 1
        end if
'Verificamos si nueva posición es mayor a la actual
        if MotorPosicion[i] < MotorNuevaPosicion[i] then
'Si es así se procede a aumentar la posición del servo a mover
          MotorPosicion[i] = MotorPosicion[i] + 1
        end if
        if MotorPosicion[i] = MotorNuevaPosicion[i] then
          Uart1_Write_Char($AA)
          Uart1_Write_Char(i)
        end if
'Se procede a la actualización del PWM para ejecutar la acción
        select case i
          case 1,2,3
            Pwm_Mc_Set_Duty(MotorPosicion[i],i)
          case 4
            Pwm_Set_Duty(MotorPosicion[i],1)
          case 5
            Pwm_Set_Duty(MotorPosicion[i],2)
        end select
      end if
    next i
    ControlVelocidad = 0
'Se setea el control del retardo
  end if

```

El diagrama de flujo correspondiente al manejo de los servomotores del brazo robótico y el control de velocidad de posicionamiento puede ver en la figura 64.

**FIGURA 64. Diagrama de Flujo para manejo de servos**



Para el manejo de los servos se realiza la siguiente secuencia de operaciones:

- Inicializar en cero la variable *controldevelocidad*
- La velocidad seleccionada en el panel frontal de LabVIEW se receipta en la variable *rxDato2*, y a la vez esta se almacena en la variable *VelocidadSeleccionada*
- Se incrementa en 1 el valor de *Controldevelocidad* para dar un retardo hasta llegar a la *VelocidadSeleccionada*
- Se implementa un lazo FOR para verificar cambios de posición en el motor *i*.
- Si la nueva posición del motor *MNP<sub>i</sub>* es diferente a la posición anterior *MP<sub>i</sub>* entonces se realiza la operación de movimiento controlando si es movimiento positivo o negativo.
- Si la nueva posición no era diferente se procede a comparar las posiciones del siguiente motor  $i=i+1$  hasta recorrer los 5 motores.
- Si la posición anterior *MP<sub>i</sub>* llega a ser igual a la nueva posición *MNP<sub>i</sub>* retorna una señal de reconocimiento, indicando que ya llegó a la posición seleccionada.
- Se restablece el *ControlVelocidad* en cero.

- El procedimiento representado por este diagrama de flujo se encuentra dentro de un ciclo infinito.

#### 4.4.2 Controlando el Brazo Empuja Frascos

Para generar la onda PWM para el servo motor del brazo empuja frascos se lo hizo por software con la ayuda de los temporizadores TMR4 y TMR5. Primero se inicializan los registros correspondientes al temporizador TMR4.

```

IFS1 = IFS1 and $FF9F
'Limpiar banderas TMR4IF y TMR5IF
IEC1 = IEC1 or $0060
'Habilitar interrupción por TMR4 y TMR5
T4CON = $8010
'Timer4 ON, prescalador 1 as 8

```

Se usan dos procedimientos que mediante interrupciones por desbordamiento de TMR4 y TMR5, harán poner en estado alto o bajo el pin0 en el puerto E, haciendo que forme una onda cuadrada [28] [29]

```

'Timer5Int y Timer4Int: Procedimientos para la generación
del PWM aplicado al 'Servo del empuja frascos.
sub procedure Timer5Int org $40
'Interrupción por TMR5 / Vector de Interrupción 40h
PORTE.0=0
' Mandar a nivel bajo la Señal
T5CON = $0010
' Desactivar el temporizador TMR5
IFS1 = IFS1 and $FFBF
' Limpiar la bandera TMR5IF
end sub
sub procedure Timer4Int org $3E

```

```

'Interrupción por TMR4 / Vector de Interrupción 3Eh
PORTE.0=1
' Mandar a nivel alto la señal
T5CON = $8010
' Habilitar el Temporizador TMR5
IFS1 = IFS1 and $FFDF      ' Limpiar la bandera TMR4IF
end sub

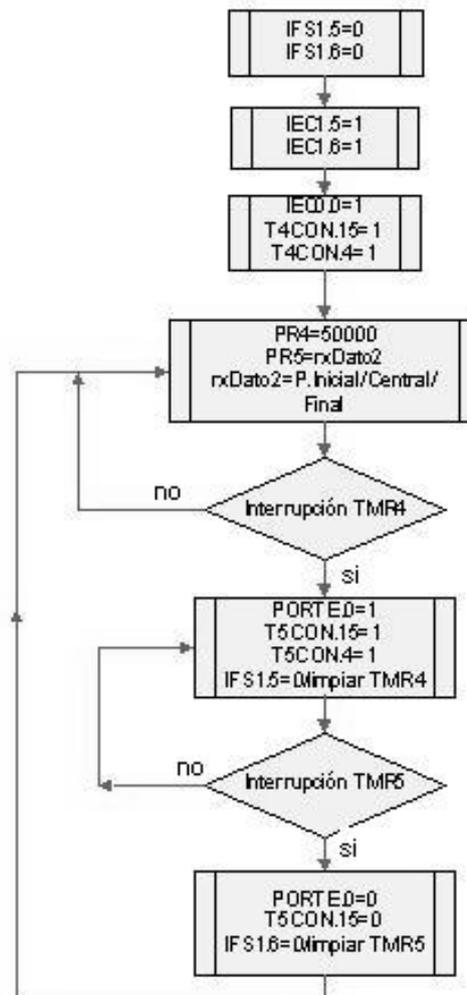
```

Los tiempos en estado bajo y alto del pin0 en el puerto E se van a cambiar mediante los registros PR5 y PR4. Se ingresa un valor constante de PR4=50000 para que cambie a estado alto por interrupción cada 20ms lo que resulta una señal periódica de 50Hz. El valor PR5 se lo usa para hacer el pin PORTE.0 a estado bajo luego de haberse puesto en estado alto producto de PR4. Este valor de PR5 va a variar el ciclo de trabajo para nuestro propósito y poder controlar la posición del servo para el brazo empuja frascos.

El brazo empuja frascos va a tener 3 posiciones para las aplicaciones de la plataforma, la posición de inicio, central y final las cuales van a estar gobernadas por el registro PR5 para poder ejecutar estas posiciones.

El diagrama de flujo que corresponde a la programación del empuja frascos lo puede ver en la figura 65.

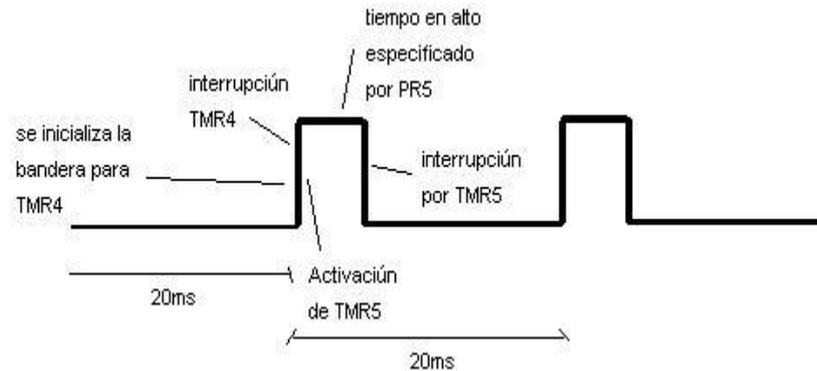
FIGURA 65. Diagrama de flujo programación empuja frascos



El procedimiento para generar el PWM del empuja frascos se puede describir de la siguiente manera [Figura 66]:

- Se inicializan los temporizadores
- Se limpian las banderas de desborde de los temporizadores
- Se habilitan las interrupciones por desborde de los temporizadores
- Se activa el temporizador 4
- Para poder obtener un valor de 20ms de temporización (periodo del PWM) se ingresó en el registro PR4 el valor de 50000, que enviará un estado alto al puerto asociado con el PWM cuando el temporizador llegue al valor de 50000.
- El valor del temporizador 5 estará dado de acuerdo a la posición que se indique en el panel frontal de LabVIEW, que pueden ser la posición inicial, central o final.
- Si ocurre la interrupción por TMR4 se habilitará el TMR5 y se restablece la bandera de TMR4
- Una vez que haya transcurrido el tiempo indicado para TMR5 este enviará un estado bajo al puerto asociado con el PWM.
- Se limpia el TMR5 hasta que se activado nuevamente por la interrupción TMR4.

**FIGURA 66. PWM por software para brazo empuja frascos**



#### 4.5. INTERFAZ ENTRE LABVIEW Y dsPICS.

##### 4.4.1 Programación del dsPIC30F4012 para recepción y envío de parámetros

Se procede a realizar una rutina de comunicación entre LabVIEW y el dsPIC30F4012 para poder ejecutar algunas acciones por medio de comandos de 3 bytes en la cual el byte mas significativo nos indica que acción deseamos realizar y los 2 bytes restantes el parámetro que deseamos de dicha acción. La tasa de transferencia es de 9600bps.

```

Uart1_Init(9600)
while TRUE
  if Uart1_Data_Ready = 1 then
    'Se verifica si hay dato
    'listo en el buffer
    rxDatol = Uart1_Read_Char()
    'Si hay dato, se procede a leer
  
```

```

while Uart1_Data_Ready = 0
  wend
rxDato2High = Uart1_Read_Char()
while Uart1_Data_Ready = 0
  wend
rxDato2Low = Uart1_Read_Char()
select case rxDato1
  'Ingreso de la nueva posición de cada servo
  case "1","2","3","4","5"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    rxDato2 = rxDato2High or rxDato2Low
    index=rxDato1-0x0030
  'Se resta de 30 para cambia a notación Decimal
  MotorNuevaPosicion[index] = rxDato2
  'Variable de la nueva posición
  case "B"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    rxDato2 = rxDato2High or rxDato2Low
    select case rxDato2
      case 0
        PR5=Inicio
  'Posición Inicial del empuja frascos
      case 1
        PR5=Centro
  'Posición Central del empuja frascos
      case 2
        PR5=Fin
  'Posición Final del empuja frascos
    end select
  'Ingreso de la velocidad de los servos del brazo robótico
  case "v"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
  VelocidadSeleccionada = rxDato2High or rxDato2Low
  'Ingreso de la Posición Inicial del empuja frascos
  case "I"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    Inicio = rxDato2High or rxDato2Low
  'Ingreso de la Posición Final del empuja frascos
  case "F"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    Fin = rxDato2High or rxDato2Low
  'Ingreso de la Posición Central del empuja frascos
  case "C"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    Centro = rxDato2High or rxDato2Low
  'Identificación del dispositivo
  case "N"
    Uart1_Write_Text("200")
  'Identificación de comando para reseteo del sistema
  case "*"
    Inicio_Reset()
  end select
end if

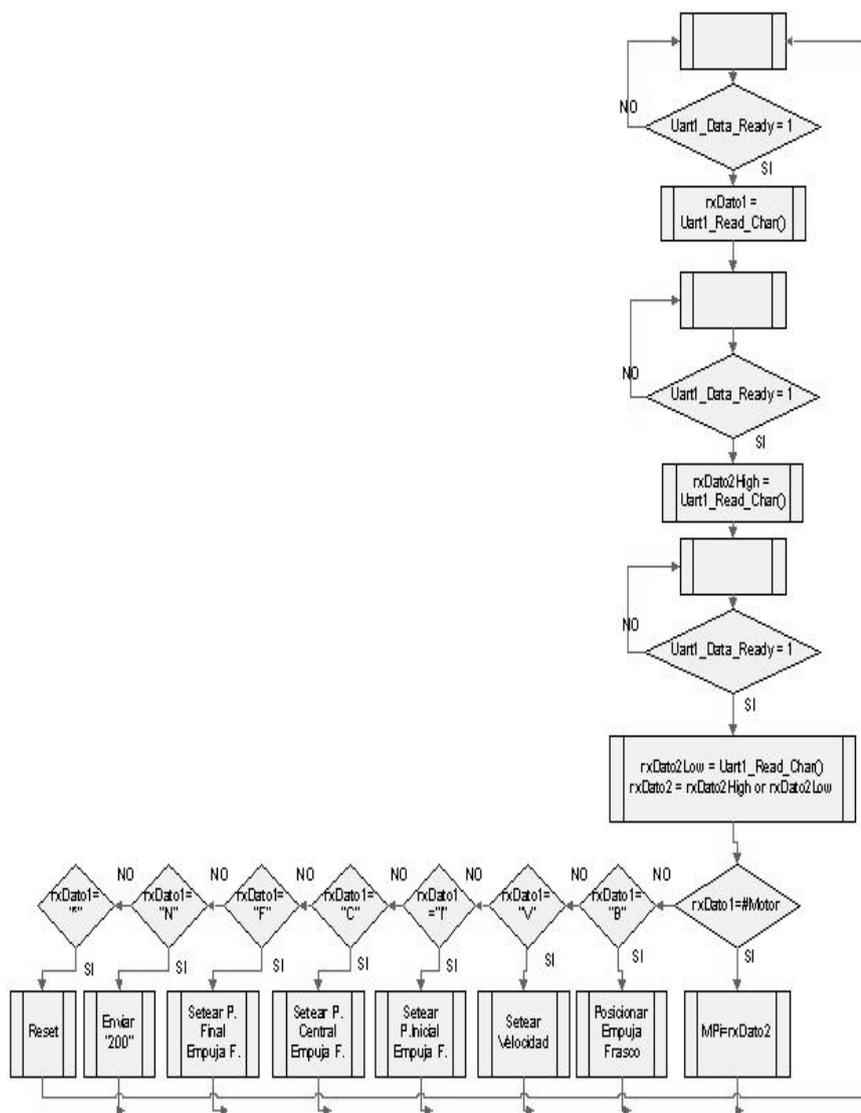
```

Como podrá apreciar en el código, para seleccionar que función quiero realizar se procede a separar cada tarea mediante la estructura ***select case***.

Cada instrumento virtual realizado en LabVIEW que hemos mencionado en la sección 4.3, va mandar los comandos para que el dsPIC30F4012 los interprete según nuestro programa.

El diagrama de flujo que corresponde a esta programación lo puede ver en la figura 67.

**FIGURA 67. Diagrama de programación para recepción y envío de datos en la tarjeta dsPIC30F4012**



La lógica para recepción y envío de datos se describe a continuación:

- Primer lugar se verifica se existe datos en la memoria intermedia de recepción, siendo este un dato de 8 bits en una lazo infinito hasta que reciba una respuesta afirmativa de existencia de datos.
- Si la respuesta es afirmativa se almacena ese dato en *rxDato1*.
- Para *rxDato2* se realiza el mismo procedimiento por dos veces consecutivas ya que éste esta compuesta de 16 bits, los cuales se separan en 2 variables *rxDatoHigh* y *rxDatoLow*, las cuales se unirán en *rxDato2* una vez receptados.
- Se analizará que acción específica *rxDato1* en la tarjeta ya sea para configuración o acciones en los brazos. *rxDato2* será un parámetro de operación sobre *rxDato1*.

Más información sobre estos parámetros se puede encontrar en la tabla II.

Tabla II. Descripción de parámetros

| <b>rxDato1<br/>ASCII</b> | <b>Descripción</b>   |
|--------------------------|--|
| 1                        | Mover servomotor 1 a la posición rxDato2 (0h-FFFFh)                                      |
| 2                        | Mover servomotor 2 a la posición rxDato2 (0h-FFFFh)                                      |
| 3                        | Mover servomotor 3 a la posición rxDato2 (0h-FFFFh)                                      |
| 4                        | Mover servomotor 4 a la posición rxDato2 (0h-FFFFh)                                      |
| 5                        | Mover servomotor 5 a la posición rxDato2 (0h-FFFFh)                                      |
| B                        | Mover brazo empuja frascos a la posición rxDato2 (0=inicial, 1=central, 2=final)         |
| V                        | Configura la velocidad de los 5 servomotores según el parámetro rxDato2 (0h-FFFFh)       |
| I                        | Configura posición inicial del brazo empuja frascos especificada por rxDato2             |
| F                        | Configura posición final del brazo empuja frascos especificada por rxDato2               |
| C                        | Configura posición central del brazo empuja frascos especificada por rxDato2             |
| N                        | Envío de identificación del dispositivo, rxdato2 puede ser cualquier valor               |
| *                        | Selecciona configuración por defecto para lo brazos, rxDato2 puede tomar cualquier valor |

## 4.6. EJEMPLOS DE APLICACIONES

El siguiente programa nos sirve para manipular el brazo robótico, seleccionando la posición de cada servomotor del brazo y la velocidad de los servos. Este programa tiene la opción de guardar las posiciones en un archivo para poder armar una secuencia de movimientos y a la vez ejecutarlo.

A continuación se muestra el panel frontal [figura 68] y el diagrama de bloque [figura 69] de la programación de la aplicación.

**FIGURA 68. Panel Frontal de la Aplicación**

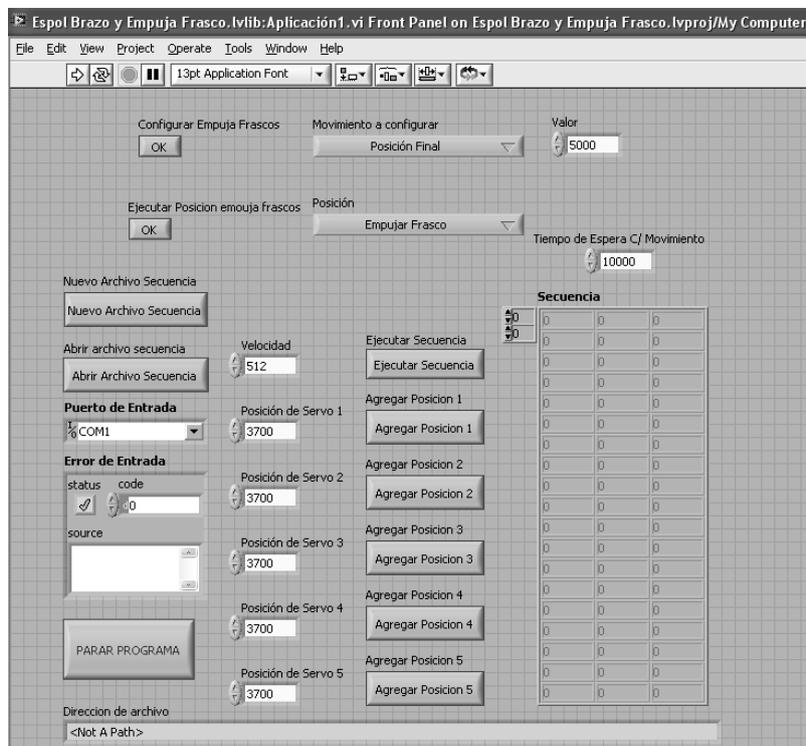
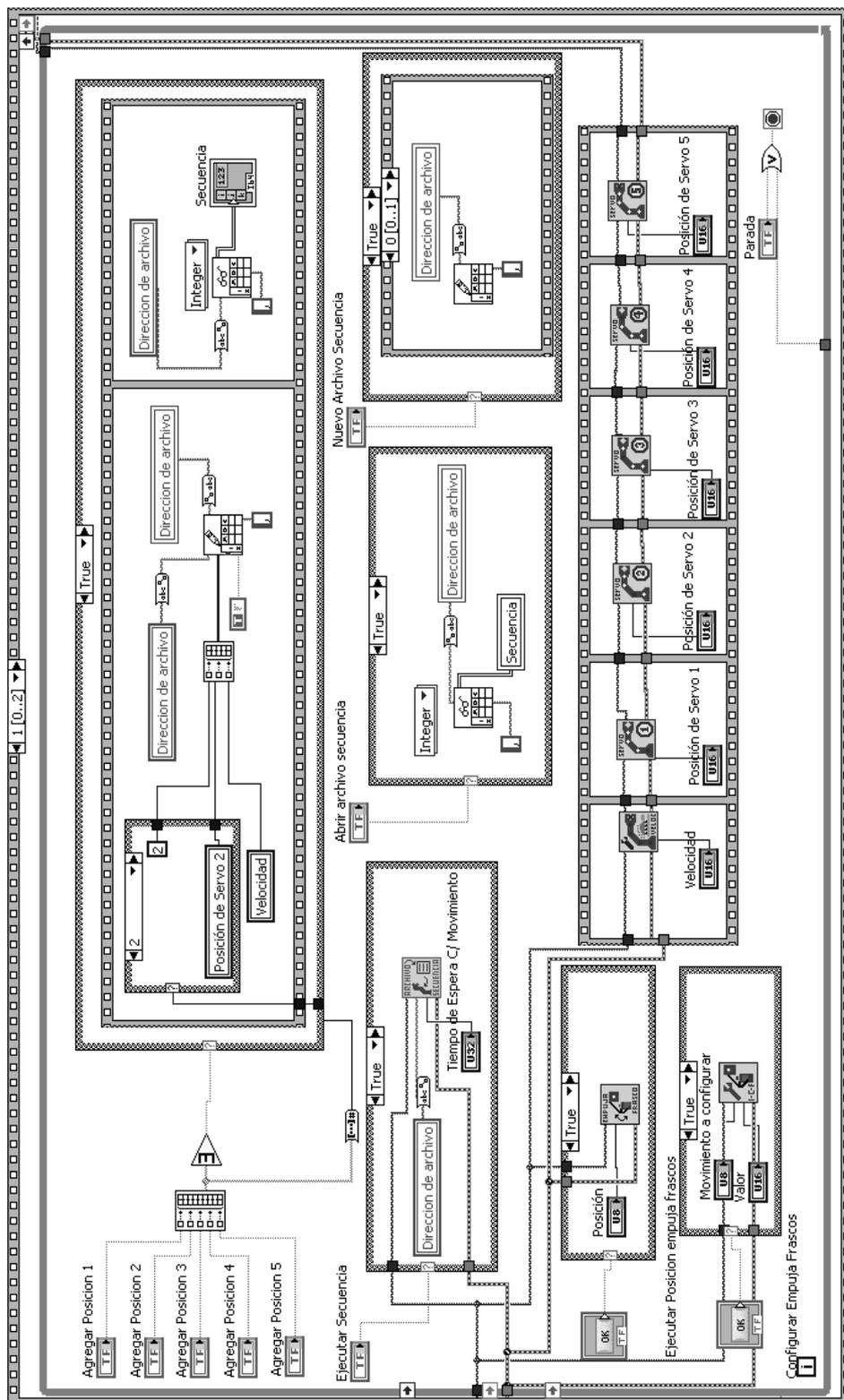


FIGURA 69. Diagrama de Bloque de la Aplicación



En el panel frontal [figura 68] podemos observar las opciones para operar el brazo robótico y el brazo empuja frascos.

En este ejemplo de la aplicación supongamos que se requiere mover el brazo empuja frascos a la posición central para detener el frasco, se deberá seguir los siguientes pasos:

- Configurar el movimiento, seleccionando posición central, y el valor de posición este valor se lo ha obtenido mediante pruebas.
- Presionar OK en la opción “configurar empuja frascos”.
- En el menú posición seleccionar detener frasco
- Presionar OK en la opción “ejecutar posición empuja frascos”.

De esta manera podemos apreciar el movimiento del brazo empuja frascos.

La segunda parte del ejemplo será mover el brazo robótico a una posición determinada en el área de operación.

Para observar la ejecución de este ejemplo basta con variar los controles de posición de cada servo, estos se moverán a la velocidad indicada en el control velocidad.

El tercer ejemplo será ejecutar una secuencia de movimientos una vez almacenadas en un archivo.

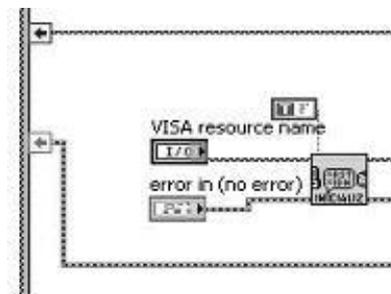
- Dar clic en el botón *Nuevo Archivo Secuencia*, se desplegará en pantalla la ventana *guardar como*.
- Una vez guardado se procede a escoger las posiciones requeridas por el usuario para los diferentes servos.
- Una vez que los servos del brazo han alcanzado una posición requerida se procederá a guardar las posiciones en el archivo mediante los botones de *agregar posiciones* correspondientes a cada servo.
- La secuencia que se va creando se puede observar en la ventana de secuencia ubicada al lado derecho del panel frontal.
- Una vez que la secuencia esta lista, se procederá a ejecutar dando clic en el botón *Ejecutar Secuencia*.
- El usuario tiene la opción de abrir una secuencia ya creada y dar clic en ejecutar secuencia para ver la acción.

Para finalizar la ejecución del VI se deberá dar clic en el botón *Para Programa* que se encuentra en la parte inferior izquierda del panel frontal.

Un breve análisis del diagrama de bloques se describe a continuación.

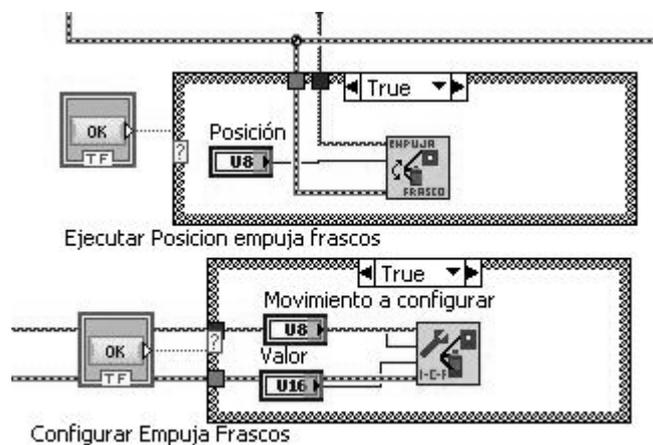
En este diseño primero se inicializa la comunicación del puerto serial y damos un reseteo a la tarjeta dsPIC30F4012 para dar las posiciones por defecto del brazo robótico.

**FIGURA 70. Estructura Inicial de la Aplicación**



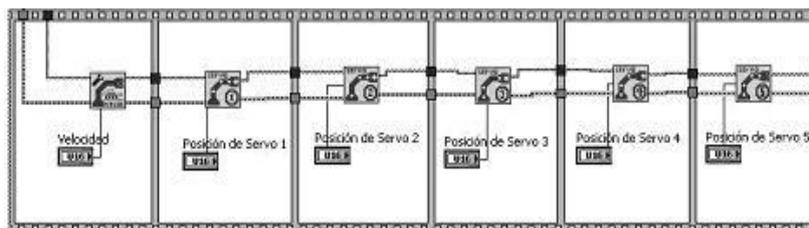
En la figura 71 se observa la estructura del ejemplo usando el VI de configuración y movimiento del brazo empuja frascos, se usó una estructura de verdadero o falso para ejecutar estas dos acciones una vez ingresados los valores de movimiento y configuración que el usuario requiera.

**FIGURA 71. Estructura para configurar y mover brazo empuja frascos**



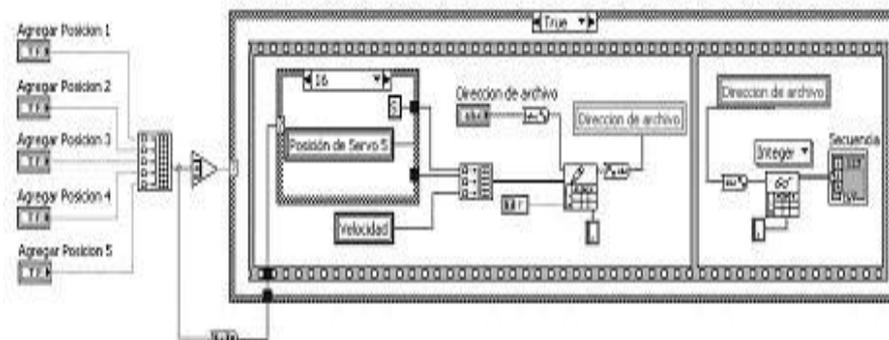
En esta estructura de la figura 72 va a ejecutar la configuración de velocidad, y las nuevas posiciones de los servos cada vez que haya un cambio en la variable asociadas a los VI's.

**FIGURA 72. Estructura para mover servos**



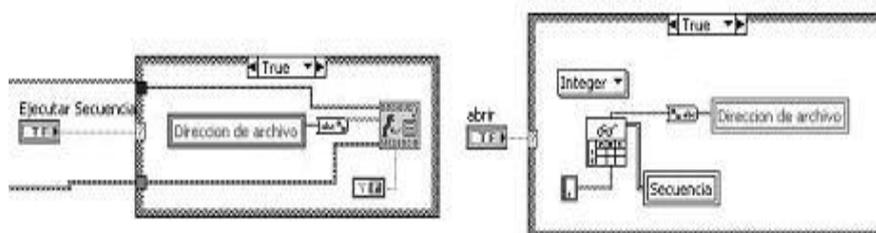
En la figura 73 se muestra como se ingresa las posiciones en un archivo y a la vez que lo muestra en el panel frontal.

**FIGURA 73. Estructura para agregar posiciones al archivo**

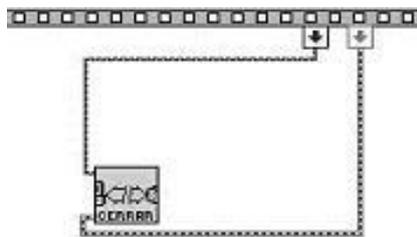


En la figura 74 muestra la forma como se abre el archivo de secuencia, y también el procedimiento para ejecutar la secuencia de movimientos de los servomotores del brazo robótico

**FIGURA 74. Estructura para abrir archivo y ejecutar secuencia de movimientos**



Al presionar el botón PARO en el panel frontal se sale de toda la estructura de lazo cerrado para cerrar la comunicación y terminar el programa. En la figura 75 se muestra el instrumento virtual para cerrar la comunicación.

**FIGURA 75. Estructura para finalizar la aplicación**

# **CAPITULO 5**

## **5. POSIBLES APLICACIONES DE LOS BRAZOS ROBÓTICOS EN LA PLATAFORMA DE DESARROLLO**

En esta tesis nos referimos a un Brazo Robótico y a un Brazo empuja Frascos ideados para operar en la plataforma de desarrollo para proyectos con dsPICs y visión robótica con LabVIEW en la cual se simularán procesos de embotellado.

### **5.1. OBJETIVOS DE LA APLICACIÓN DEL BRAZO ROBÓTICO DISEÑADO**

Los objetivos de nuestros brazos robóticos son:

- Sacar de la plataforma las botellas que no cumplan con los requisitos, este objetivo es tanto para el empuja frascos como para el brazo robótico.

- Rotar la botella, en ciertas ocasiones para verificar la etiqueta.
- Trasladar las botellas de un lugar de la plataforma a otro. (brazo robótico)
- Si se pusiese una cámara en la pinza del brazo, tuviese una vista más cercana y más eficiente del punto a tratar. Como por ejemplo observar si el frasco tiene o no tiene tapa.
- Detener en un punto específico la botella, (brazo empuja frascos).
- Ayudar en el análisis y estudio del comportamiento de un brazo robótico en un modelo a escala de una planta de procesamiento de botellas.

## **5.2. ÁREAS DE USO**

Las áreas en las que nuestros brazos están siendo usados son:

- Inspección de botellas con tapa.
- Inspección de botellas con etiqueta.
- Inspección de turbiedades en el líquido de las botellas.
- Inspección del nivel de líquido en cada botella.
- Filtro de tamaño de botellas.
- Inspección de botellas de diferente color.

### 5.3. FORMAS DE USO

Los brazos robóticos tienen muchas formas de uso, pero principalmente lo aplicamos en el proceso de embotellado, revisando ciertas características de las botellas tratadas.

Nuestro brazo empuja frascos puede usarse en:

- Empujando los frascos hacia un envase de desecho.
- Empujando el frasco de una banda a otra.

El brazo robótico también tiene diferentes formas de usarse que aunque algunas no son aplicadas en la plataforma diseñada por otro grupo del Tópico de graduación puede ser realizadas con tan solo ciertos cambios de la siguiente manera:

- Tomando el frasco y colocándolo en un envase de desecho.
- Tomando el frasco y colocándolo en una posición diferente en la plataforma.
- Rotando el frasco.
- Colocando tapa.

- Colocando etiqueta, cambiando la pinza actual por otro accesorio que permita adherir la etiqueta y presionarla contra la botella.
- Soldadura de punto, cambiando la pinza por una punta soldadora.

#### **5.4. VISIÓN CON LABVIEW APLICADO AL PROCESO**

La visión es una de las nuevas aplicaciones que trae LabVIEW incorporado, es de mucha ayuda cuando se requiere trabajar con adquisición de imágenes en los procesos.

En este literal se ampliará la explicación de algunos ejemplos del uso del brazo robótico y del empuja frascos utilizando la visión que ofrece LabVIEW y en el área que se la está aplicando.

##### **5.4.1. Inspección de botellas con tapa.**

Uno de los problemas en el proceso de embotellado sobretodo en la línea de control de calidad es la presencia de tapas en las botellas. Esto obliga a la incorporación de un puesto de inspección que detecte las anomalías del frasco.

Esta inspección debe realizarse de tal manera que los resultados no se vean alterados por las perturbaciones que se dan por las múltiples interferencias que existen en un ambiente industrial.

Para la realización de este puesto de inspección se propone un sistema de inspección automática utilizando técnicas de Visión Robótica que ofrece el módulo National Instruments Vision Development compatible con Cámaras USB.

El objetivo consiste en desarrollar una aplicación en lenguaje G utilizando LabVIEW, de tal manera que se adquieran imágenes digitales del frasco, estas a su vez son procesadas para la detección de frascos sin tapa.

De forma general esta aplicación consiste en un sistema de control estructurado en pequeña escala para corrección de errores en una envasadora, la cual nos da una idea real de su funcionamiento dentro del campo industrial.

Este sistema brinda la capacidad de inspección para detectar los frascos que no tienen tapa. Los frascos se

movilizan por las bandas transportadoras, y el sistema podría funcionar en conjunto con otros sistemas de inspección. Dependiendo del resultado, el frasco avanza o se desecha.

El sistema de visión de detección de frascos sin tapa y con tapa está compuesto por:

**Una banda transportadora.** Encargada de transportar lo envases que van a ser examinados.

**Sistema de iluminación.** Constituido por una cámara, que elimina la influencia de fuentes de iluminación que se encuentren en el exterior de la misma, y lámparas que brindan una iluminación constante y necesaria en el interior de la cámara para poder captar los detalles de las tapas de los frascos que tengan tapa o los detalles de los cuellos de los frascos que no tengan tapa.

**Brazo empuja frascos,** que se encarga de retirar los frascos de la línea en el momento en que se determine que no tienen tapa. Además cumple una función adicional, que

es la de detener las botellas en la posición exacta para que puedan ser examinadas.

Además se utiliza el brazo robótico, transformándolo en un brazo cámara robótico, el cual se utiliza para examinar los envases desde diferentes ángulos, en nuestro caso tenemos en cuenta dos vistas: Vista lateral y vista superior.

El programa desarrollado en LabVIEW controla el funcionamiento de la aplicación y en conjunto con las partes antes mencionadas constituyen un sistema de detección de frascos con tapa y frascos sin tapa.

En esta aplicación el usuario sigue una serie de instrucciones para poner en marcha el sistema.

El usuario tiene que seleccionar las cuatro plantillas a utilizarse: Vista lateral y vista superior del frasco con tapa, y vista lateral y superior del frasco sin tapa. Además que se tienen las siguientes opciones: Seleccionar plantillas existentes o crear nuevas.

Una vez seleccionadas las plantillas, empieza el proceso de detección. La banda transporta los frascos y en el momento en que se detecte que un envase entra a la cámara de iluminación se realizan los siguientes pasos:

El brazo cámara robótico captura una imagen digital tanto de la vista frontal como de la vista superior del frasco examinado.

La aplicación determinará si el frasco tiene o no tapa. Si el frasco tiene tapa, este sigue en la línea. Caso contrario, la palanca empuja frascos lo retira de línea.

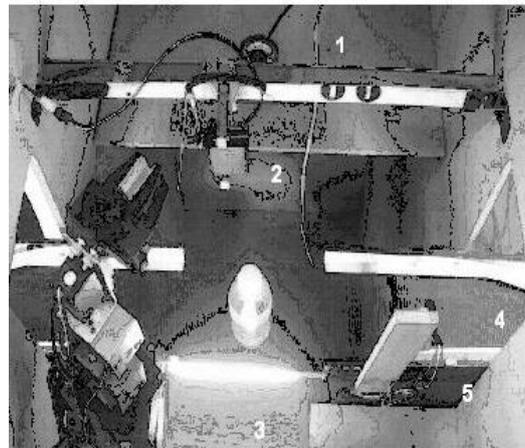
Este proceso se diseñó de tal manera que una vez finalizado el proceso con una botella, lo siga realizando infinitamente con las botellas siguientes, hasta que el sistema sea detenido por el usuario, y la banda transportadora se haya detenido.

#### **5.4.2. Inspección de turbiedades en el líquido de las botellas.**

Este proceso para su operación utilizará elementos presentes en el ejemplo anterior, tales como la banda transportadora, la cámara de iluminación, el brazo robótico, brazo empuja frascos y una cámara fija para adquisición de imágenes.

Si analizáramos un proceso completo, una de las fases más importantes sería observar el líquido de las botellas buscando basura, turbiedades u objetos extraños al producto. Este análisis puede hacerse al inicio para verificar que los frascos estén libres de objetos extraños antes de verter el contenido en ellos, como para la comprobación de que el contenido vertido sea el esperado.

El sistema a diseñar podrá ser capaz de identificar partículas en suspensión mayor a 2mm. Partes del proceso se puede observar en la figura 76.

**FIGURA 76. Vista Superior de la plataforma de trabajo**

1. Cámara USB
2. Empuja Frascos
3. Iluminación de fondo
4. Banda transportadora
5. Sensores

**Fuente: Grupo 2, Plataforma para proyectos con dsPICs y Visión Robótica con LabVIEW, Lab. de Microcontroladores**

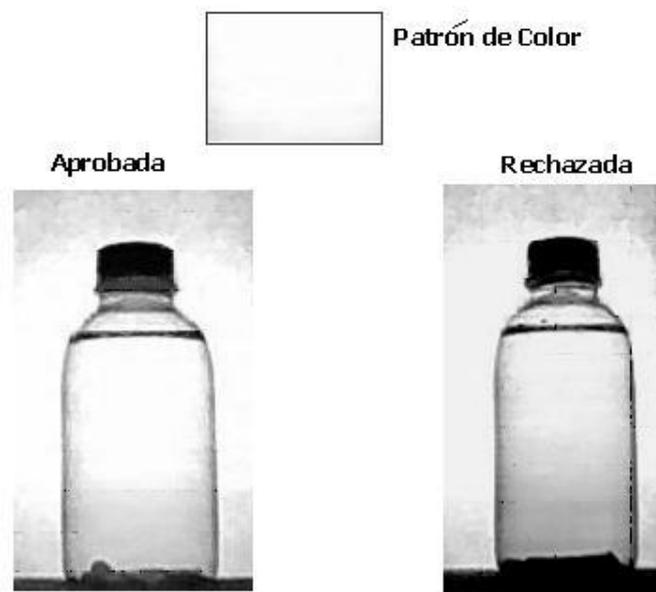
Debido a que este tipo de análisis depende mucho del color del líquido tratado y del color de la botella utilizada, se realiza lo que se denomina determinación de áreas de interés, para poder realizar un análisis tal que ningún factor pueda ser evadido y por ende se pueda dar una mala detección.

Primero una vez que el sistema ha adquirido una imagen de la botella a tratar, se convierte la misma a escala de grises, y seguidamente se busca el patrón de la tapa de la imagen capturada, para en base a esto establecer las proporciones de la imagen.

Se divide la imagen en tres rectángulos de trabajo, alto, medio y bajo.

Como se había mencionado el color es algo muy importante es por eso que se hace la comparación de color con una imagen previamente almacenada de una botella en perfectas condiciones y se establece el número de coincidencias. Figura 77

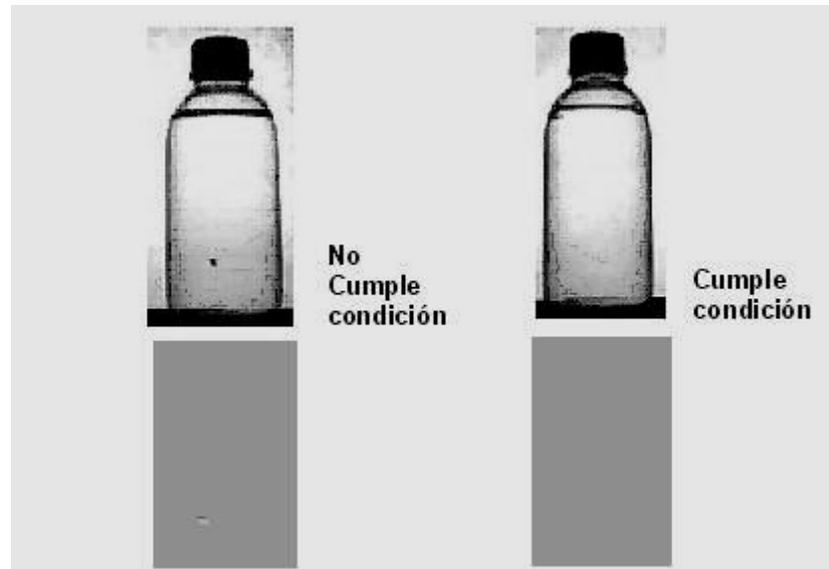
**FIGURA 77. Comparación de Color**



**Fuente: Grupo 2, proyecto 9, Plataforma para el desarrollo proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores.**

Para obtener el área de análisis central se deberá buscar el patrón de la tapa en el rectángulo de trabajo, para poder determinar los límites verticales del contenido del frasco, además de los límites horizontales del mismo. En base a los límites verticales y horizontales se establece el área de análisis central.

El análisis central consiste en convertir la imagen de escala de grises a binaria, luego se remueven los bordes del frasco y se eliminan las partículas pequeñas que pueden ser debido al ruido al obtener la imagen, por último se buscan partículas restantes en el área central. El resultado del análisis se puede observar en la figura 78.

**FIGURA 78. Análisis Central**

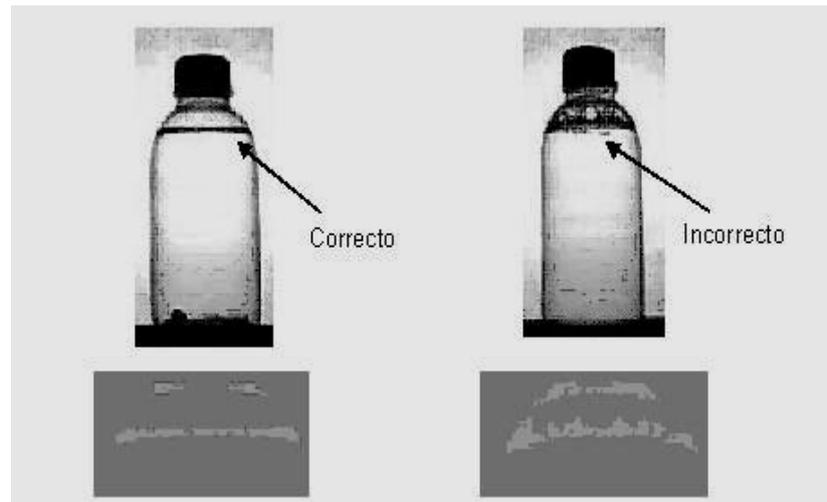
**Fuente: Grupo 2, proyecto 9, Plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores**

Para el análisis superior se recorta la imagen con las coordenadas del rectángulo alto, con la imagen ya convertida a escala de grises se transforma la misma a binario, se remueven los bordes de la imagen, y se eliminan las partículas pequeñas debido al ruido.

Se calcula el área total de la imagen y se compara con un umbral calculado anteriormente con una imagen de ejemplo adquirida de una botella en excelente estado. De esta

manera se determinan si existen impurezas en el área superior. Como se observa en la figura 79.

**FIGURA 79. Análisis superior**

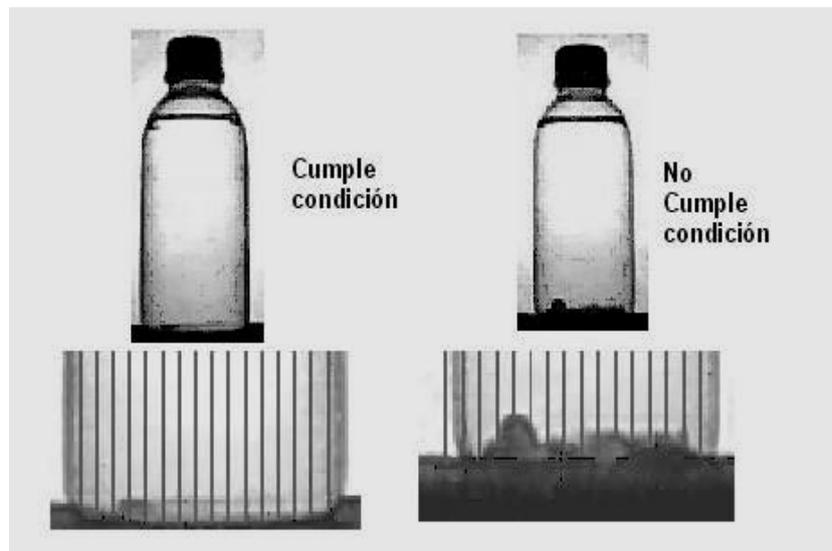


**Fuente: Grupo 2, proyecto 9, Plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores**

Por último se realiza el análisis inferior para esto se recorta la imagen con las coordenadas del rectángulo bajo, una vez convertida la imagen a escala de grises, se detectan los bordes a través de líneas verticales perpendiculares al fondo del frasco, el contador de líneas verticales debe ser igual a una resolución con la que se esta trabajando, además de que se determina la mayor diferencia vertical entre los diferentes puntos hallados.

Este dato obtenido se compara con un umbral al inicio del programa especificado, para determinar la existencia de impurezas en la parte inferior. Como lo muestra la Figura 80.

**FIGURA 80. Análisis Inferior**



**Fuente: Grupo 2, proyecto 9, Plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores**

Una vez concluido todo los análisis, si y solo si la botella cumple con cada una de las condiciones establecidas, esta podrá continuar de caso contrario será expulsada por el brazo empuja frascos.

### **5.4.3. Inspección del nivel de líquido en cada botella.**

Como se aclaró en el ejemplo anterior, para este proceso se utilizan ciertos objetos que se usaron en la inspección de frascos con tapa, como por ejemplo la banda transportadora, la cámara de iluminación, por supuesto el brazo empuja frascos, y además una cámara fija para adquisición de imágenes.

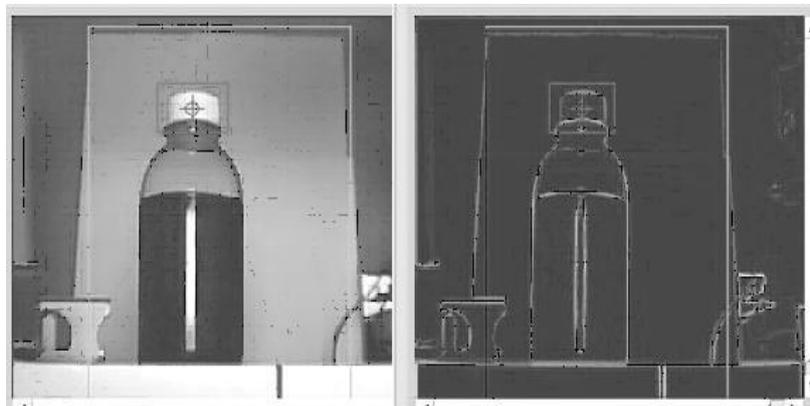
El objetivo del proyecto es la detección de niveles de llenado de envases que serán llevados por la banda transportadora.

Se plantea un rango de niveles permitidos para los líquidos de los envases que se van a monitorear, es decir los envases que tengan líquidos dentro de ese rango se mantendrán en la banda transportadora, caso contrario serán rechazados mediante el brazo empuja frascos.

En primer lugar se reserva memoria para las “imágenes” que se van a usar. La primera es la plantilla de la tapa del envase y la imagen de video que se va a capturar mediante una cámara Web, luego se hará una búsqueda de dicha

plantilla dentro de la imagen de video capturada, ambas imágenes son trabajadas luego de haberles pasado un filtro de Luminiscencia, el cual le extrae un plano de colores a la imagen, esto se hace para obtener un mayor puntaje en la detección. Como se puede observar en la figura 81.

**FIGURA 81. Filtro de luminiscencia**



**Fuente: Grupo 8, Plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores**

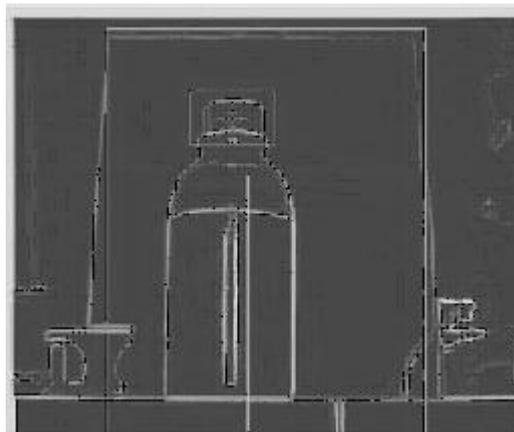
En la captura de vídeo se crea una región de interés (ROI), de tal manera que la plantilla sea buscada sólo dentro de esa región.

La plantilla puede ser guardada previamente en la computadora o puede ser creada en el instante cuando se procede a medir el nivel de llenado de los envases.

Si la imagen de la plantilla se encuentra guardada en la computadora donde se está trabajando, la imagen es extraída de manera dinámica mediante el uso del VI llamado "Current VI"

Para una mejor detección de los bordes se usa un filtro de detección de bordes, que clarifica y resalta los bordes de la imagen en análisis. Figura 82.

**FIGURA 82. Detección de nivel**



**Fuente: Grupo 8, Plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores**

Con las coordenadas obtenidas luego de la búsqueda de la plantilla se procede a trazar una línea vertical desde la parte inferior de la tapa hasta aproximadamente unos 10 píxeles por debajo del límite inferior del envase, con el fin de reconocer u obtener el nivel del líquido contenido en él.

El resultado de esta operación es una medida de la altura del nivel del líquido dada en píxeles, de tal manera que hay que buscar un mecanismo donde se pueda convertir los píxeles en alguna medida del mundo real.

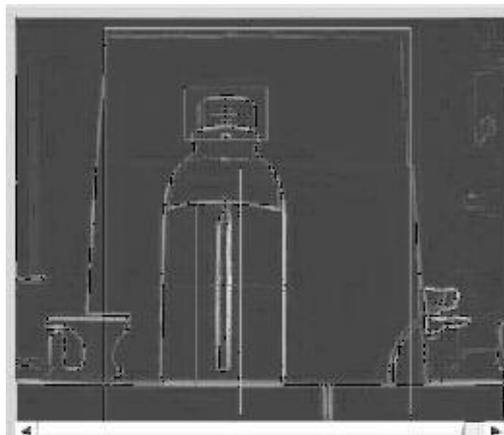
Mediante la opción de “Overlay” se grafica los niveles dentro de la imagen capturada.

Para este objetivo se realiza la calibración dinámica de la imagen, con las mismas coordenadas de la búsqueda de la plantilla, en el envase se procede a trazar una línea horizontal aproximadamente en la mitad del envase.

Por medio del detector de bordes de LabVIEW, son detectados los bordes del envase, es decir se detecta el ancho del envase, al ser éste una medida constante en el

monitoreo del nivel, se usa como medida de calibración, siendo aproximadamente 2.4 cm. en promedio. De esta manera el programa ya puede calcular cuántos centímetros mide el nivel de llenado de cada envase. Figura 83.

**FIGURA 83. Cálculo de Nivel**



**Fuente: Grupo 8, Plataforma para el desarrollo de proyectos con dsPICs y Visión Robótica con LabVIEW, Laboratorio de Microcontroladores**

Con la medida del nivel del líquido de cada envase se procede a verificar si está o no dentro del nivel permitido, configurado al inicio del monitoreo, si no cumple con esta condición serán eliminados de la banda mediante el Brazo “empuja frascos”.

## 5.5. ACTUALIZACIONES Y MEJORAS PARA BRAZO ROBÓTICO

Como todo equipo ya sea este industrial o no, necesita de mejoras y actualizaciones, en esta sección nombraremos algunos de los cambios que en el futuro se le podrán realizar al brazo robótico y al brazo empuja frascos.

- Cambiar la pinza por algún otro accesorio.
- Cambiar la estructura física convirtiéndolos en brazos más robustos, con mayor alcance, y mejor agarre.
- Cambiar servomotores para acompañar la robustez física del brazo.
- Añadir sensores, que ayuden a la autonomía de los brazos
- Añadir módulo RF a la tarjeta controladora de los brazos robóticos, para manejo inalámbrico.
- Incorporar el uso de la memoria EEPROM para almacenar direcciones y comandos fijos en el movimiento del brazo robótico.
- Rediseñar la tarjeta, ya que a mayor robustez del equipo y mayor torque de motores, consumirá más corriente, por ende las pistas de la tarjeta necesitarán ser más gruesas.

Algunos ejemplos de accesorios por los cuales se puede cambiar la pinza del brazo robótico son:

- Rodillo para colocar etiqueta
- Mano, que puede sujetar otro tipo de objetos
- Punta soldadora
- Cámara, para adquisición de imágenes.
- Punta de Soplete, para pintura spray
- Sierra, para corte con discos metálicos.
- Martillo.
- Soplete de corte, para aplicar el sistema de corte llamado Oxicorte.
- Remachadora
- Pulidora

Si nos referimos al cambio de servomotores, para robustecer los sistemas, incrementado el torque de cada uno y por ende el peso que pueden levantar, podríamos decir que los siguientes motores HI-TEC son algunos de los que pueden reemplazar a los servomotores utilizados ya que son compatibles, solo faltaría que el usuario seleccione alguno que cumpla con sus características y exigencias.

HS-311, HS-322, HS-325, HS-422, HS-425BB, HS-475, HS-525BB, HS-525MG, HS-545BB, HS-625MG, HS-635HB, HS-645MG, HS-925MG, HS-945MG, HS-5475HB, HS,5625, HS-5925MG, HS-5945MG, HS-5955TG, HSR-5995TG, HSC-5996TG, HSC-5997TG, HSC-5998TG, HS-6635HB, HS-6965HB, HS-6975HB, HS-6985HB

De la misma manera el servomotor Futaba S3003 utilizado para el empuja frascos puede ser reemplazado por alguno de los siguientes.

S-148, S-3001, S-3003, S-3004, S-3005, S-3010, S-9001, S-9202, S-9206, S-9402, S-9405

## **5.6. ANALISIS DE COSTOS**

Los costos del proyecto del Brazo Robótico y del Brazo Empuja Frascos Controlados por Labview y Dspics se detallan a continuación:

|  |          |
|--|----------|
| Brazo Robótico RA-01                             | \$478.88 |
| Diseño de la tarjeta controladora (dsPIC30F4012) |          |

para el Brazo Robótico y para el Brazo

Empuja Frascos \$ 27.44

Construcción de Brazo Empuja Frascos \$ 41.03

---

TOTAL \$547.35

Se debe de considerar además de los gastos monetarios, el tiempo invertido en la construcción y diseño, los cuales se detallan a continuación:

Ensamblaje del Brazo Robótico RA-01 10 horas

Ensamble de Brazo Empuja Frascos 5 horas

Diseño de la tarjeta 40 horas

Ensamble de tarjeta dsPIC30F4012 5 horas

Programación de la Tarjeta dsPIC30F4012 80 horas

Diseño instrumentos virtuales e interfaz

en LabVIEW 160 horas

Pruebas Realizadas 20 horas

---

TOTAL 320 HORAS

Considerando que el número de horas invertidas en este proyecto fueron 280 horas, y las horas laborables en un mes son 160, podemos concluir que la elaboración de este proyecto equivale a 2 meses de trabajo continuo.

Nos demoramos en el desarrollo de los instrumentos virtuales ya aparte de la programación nos demoramos también en el diseño gráfico de cada uno de ellos y se realizaron pruebas para que sean adaptados a los programas de visión en LabVIEW desarrollados por los compañeros del tópico.

La adquisición del Brazo Robótico nos ahorró tiempo valioso de mano de obra. Información detallada acerca de los costos de los materiales la encontrará en el ANEXO G.

# CAPITULO 6

## 6. PRUEBAS REALIZADAS

En este capítulo como su nombre lo indica se referirá a las pruebas realizadas a los brazos robóticos durante el proceso de ensamblaje.

### 6.1 PRUEBAS REALIZADAS

Las pruebas que se realizaron al brazo robótico fueron principalmente de dos tipos, las cuales se mencionarán a continuación.

- Pruebas de ubicación
- Pruebas de peso
- Pruebas de frecuencia PWM

Las pruebas de ubicación fueron también realizadas al brazo empuja frascos.

#### **6.1.1. Pruebas de ubicación**

Estas se refieren a las pruebas que se hicieron al brazo para conocer que comando se refería a que ángulo y por ende a que posición en el espacio. Ya que el brazo recibirá una serie de comandos que consisten en la nueva dirección que tendrá el brazo, entonces debemos conocer la relación entre el número decimal a ingresar en el programa de LabVIEW y el ángulo al cual se moverá el brazo robótico o el brazo empuja frascos.

La prueba consiste en probar varios números decimales en la ventana de comandos de un servomotor y observar mediante un osciloscopio como cambia el ancho de pulso y a cuantos microsegundos equivale, y de esta manera determinar una fórmula que se aproxime a lineal para el cálculo de ángulos.

### **6.1.2. Pruebas de peso**

Se denomina así a las pruebas que se realizaron para observar cual era el peso óptimo que el brazo robótico puede levantar, sin forzar los servomotores y sin que el objeto resbale y caiga.

La prueba consiste en ejecutar una misma rutina de movimiento con 4 diferentes pesos, desde un peso pluma hasta uno relativamente pesado en comparación con el peso total del cuerpo del brazo robótico.

### **6.1.3. Pruebas de frecuencia PWM**

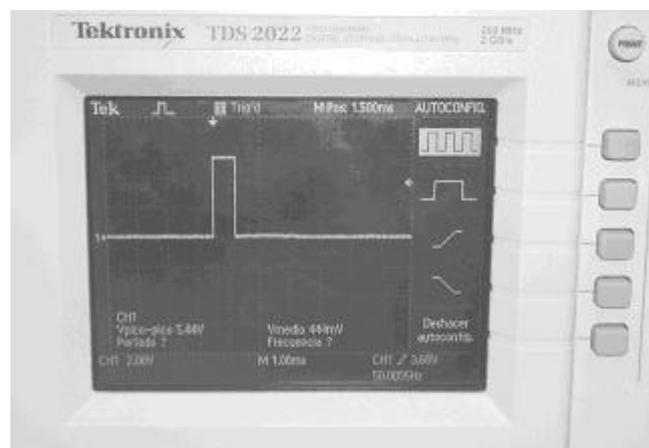
Las pruebas denominadas frecuencia PWM consisten en cambiar la frecuencia en la señal PWM que recibirán los servos alrededor de los 50Hz.

Esta prueba se la realizó con tres frecuencias diferentes y se observó los cambios que ocurrían.

## 6.2 DATOS OBTENIDOS

Los datos que se obtuvieron en la primera prueba denominada prueba de ubicación para el brazo robótico fueron los siguientes:

**FIGURA 84. Prueba a cero Grados**

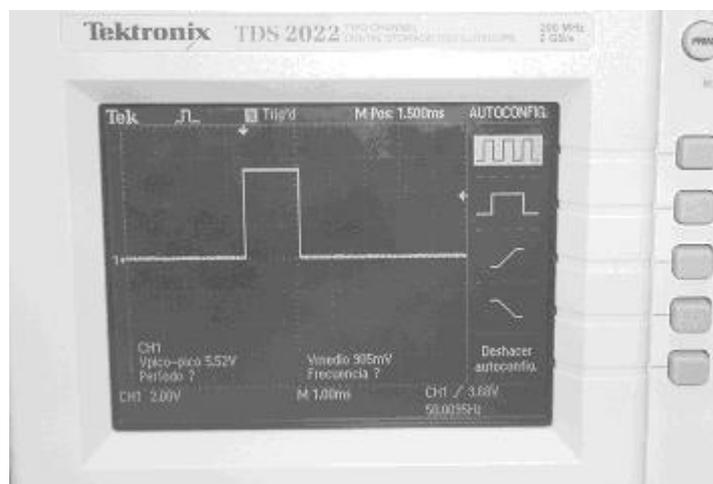


**Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores**

La figura 84 corresponde a 1500-600us-0grados, siendo el primer número el comando decimal que se ingresa en la ventana de LabVIEW, el segundo número el valor en microsegundos del ancho de pulso que pertenece a ese comando, y el tercero pertenece al ángulo en el que se posicionará el servomotor.

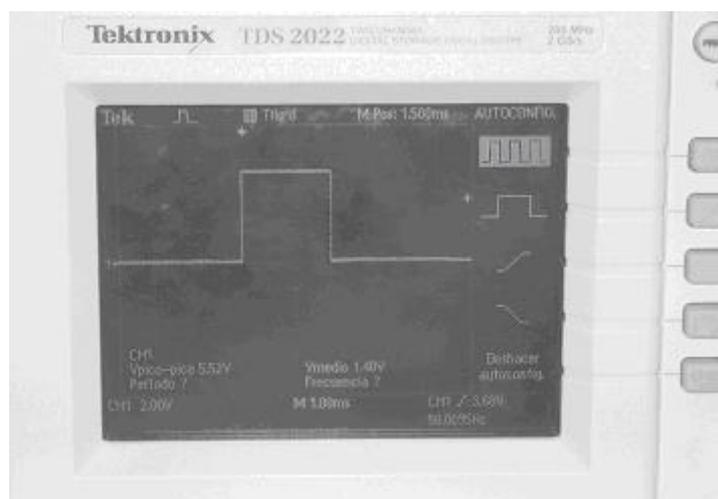
La misma prueba se realizó para otras posiciones representadas en las figuras 85 y 86, llegando a formar la tabla III.

FIGURA 85. Prueba a 90 Grados



Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores

FIGURA 86. Prueba a 180 Grados



Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores

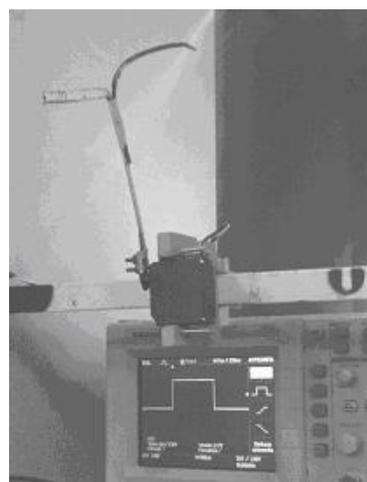
**Tabla III. Resultados de la Prueba de Posición en los servomotores del Brazo Robótico.**

| <b>Comando en decimal</b> | <b>Ancho de Pulso (us)</b> | <b>Posición servomotor</b> | <b>Figura</b> |
|---------------------------|----------------------------|----------------------------|---------------|
| 1500                      | 600                        | 0                          | 84            |
| 3750                      | 1500                       | 90                         | 85            |
| 6000                      | 2400                       | 180                        | 86            |

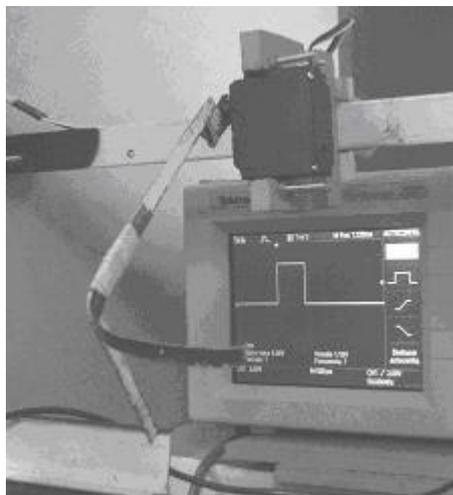
Esta prueba se la realizó a los 3 tipos de servomotores para el brazo robótico, notándose que no existió variación entre las posiciones a cero grados y a 180 grados.

Los datos que se obtuvieron de la prueba de posición para el brazo empuja frascos fueron:

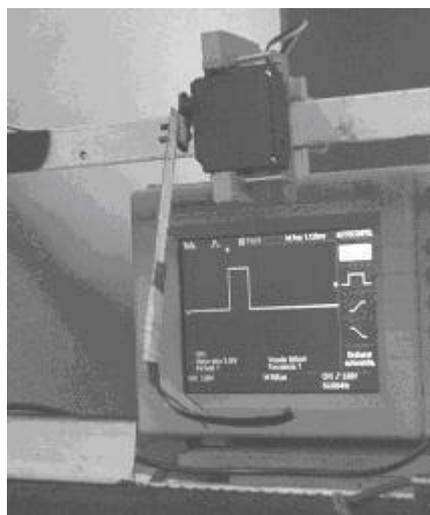
**FIGURA 87. Brazo Posición Inicial**



**Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores**

**FIGURA 88. Brazo Posición intermedio**

Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores

**FIGURA 89. Brazo Posición Final**

Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores

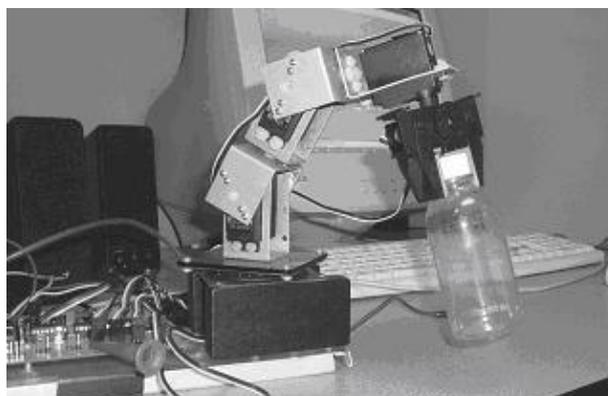
De las figuras 87, 88, y 89 se obtuvo una tabla de resultados, con el número decimal que se debe ingresar para obtener estas posiciones, que son inicial, detener frasco y empujar frasco respectivamente.

**Tabla IV. Resultados de Pruebas de Posición Brazo Empuja Frascos.**

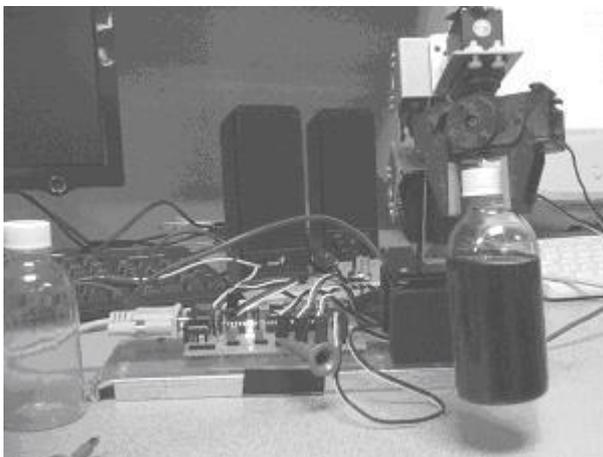
| <b>Posición</b> | <b>Comando decimal</b> | <b>Ancho de pulso</b> | <b>Figura</b> |
|-----------------|------------------------|-----------------------|---------------|
| INICIAL         | 5000                   | 2000                  | 87            |
| CENTRAL         | 2400                   | 950                   | 88            |
| FINAL           | 1700                   | 675                   | 89            |

Los resultados de la prueba de pesos se pueden observar en las figuras 90, 91, 92, 93

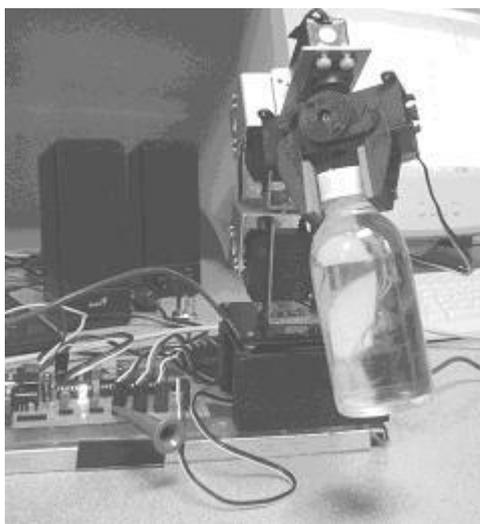
**FIGURA 90. 1(oz.)**



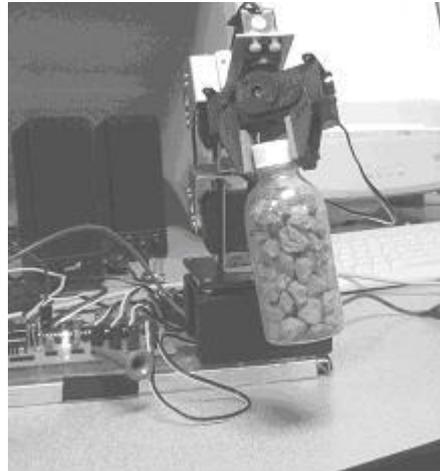
**Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores**

**FIGURA 91. 3 (Oz.)**

**Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores**

**FIGURA 92. 4 (Oz.)**

**Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores**

**FIGURA 93. 7 (Oz.)**

**Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores**

Observando estos resultados se puede presentar la tabla V.

**Tabla V. Resultados de la Prueba de Peso Brazo Robótico**

| <b>Peso</b> | <b>Aceptable o<br/>No Aceptable</b> | <b>Figura</b> |
|-------------|-------------------------------------|---------------|
| 1 Oz.       | √                                   | 90            |
| 3 Oz.       | √                                   | 91            |
| 4 Oz.       | √                                   | 92            |
| 7 Oz.       | X                                   | 93            |

Los resultados de las pruebas de frecuencia PWM se los sintetizó en la tabla VI.

**Tabla VI. Resultados de la prueba de Frecuencia PWM**

| <b>frecuencia</b> | <b>Periodo</b> | <b>Posición</b> |
|-------------------|----------------|-----------------|
| 10Hz              | 100ms          | Invariable      |
| 50Hz              | 20ms           | Invariable      |
| 100Hz             | 10ms           | Invariable      |

### 6.3 ANÁLISIS DE RESULTADOS

Analizando las tres pruebas realizadas en el brazo robótico y el brazo empuja frascos se puede deducir lo siguiente:

Se pudo comprobar que la posición neutral del servo motor es de 1500us o 1.5ms que se especifican en la hoja de datos de los servomotores, esta posición neutral será el valor de 90 grados para nuestra aplicación.

Según la hoja de datos un valor de 400us desde la posición neutral nos proporcionará aproximadamente 40 grados para los servos HS-322HD y un valor de 45 grados para los HS-422 así como para el HS-645MG, al realizar las pruebas se observó que no había una gran diferencia entre la posición de un servo y el otro, por lo cual

se decidió trabajar los mismos valores que se encontró en las pruebas para todos los servos.

Entonces para encontrar las posiciones de 0 y 180 grados se varía desde la posición neutral 900us hacia atrás y 900us hacia delante respectivamente desde la posición central de 90 grados.

Se realizaron pruebas fuera de estos rangos y se obtuvo que se puede variar más de 180 grados pero no es conveniente debido a que se está forzando el servomotor, se deduce esto ya que el servo realiza ruidos que denotan esfuerzo.

Estos son valores en microsegundos, pero en la interfaz de usuario utilizados datos en decimal, es por esto que se diseñó un VI que no ayude a la conversión de los datos en microsegundo a decimal.

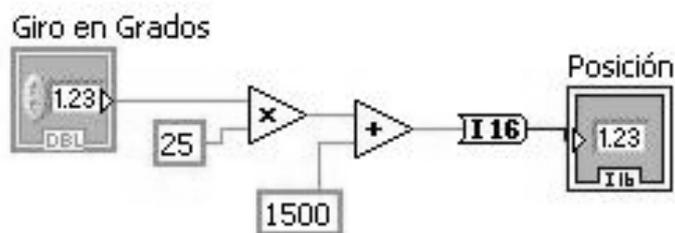
Basándonos en la tabla de resultados de la prueba de posición de los servomotores concluimos que tenemos un rango de 4500 pasos para recorrer 180 grados y se lo puede demostrar con más detalle en el Anexo H. Entonces tenemos la siguiente fórmula:

$$\frac{4500 \text{ pasos}}{180 \text{ grados}} \equiv 25 \text{ pasos cada grado}$$

$$\frac{1}{25} \equiv 0,04 \text{ grados cada paso}$$

Según este resultado pudimos obtener el VI de conversión, que tendrá la siguiente estructura.

**FIGURA 94. Estructura de VI de Conversión de grados a Posición en decimal**



Cabe destacar que el control de giro en grados se incrementará en 0,04.

Esto se da en caso del brazo robótico, en el caso del brazo empuja frascos el rango utilizado es desde 1700 a 5000 que son nuestras posiciones máxima y mínima que no necesariamente son 0 y 180 grados pero son las posiciones óptimas para el uso que se le da. Y

la llamada posición intermedia es de 2400, en la cual el brazo cumple la función de detener la botella a tratar en una posición determinada, o sea justo en frente de la cámara.

La segunda prueba que se hizo fue el análisis de diferentes pesos de frascos, se observó que para frascos con pesos mayores a 4oz el brazo tuvo dificultad al tratar de levantarlo a la posición indicada.

La tercera prueba realizada fue la variación de frecuencia de operación PWM, se denotó que no hubo variación en las posiciones de los servos, esto se debe a que el movimiento del servo no depende de la frecuencia del PWM sino del ciclo de trabajo o ancho de pulso de la señal cuadrada.

Se observó que al querer sacar al servomotor de su posición, se hizo más difícil la tarea usando un periodo de 10ms en la señal PWM, que usando un 20ms que corresponde a 50Hz, y al usar una frecuencia 10Hz o 100ms fue más sencillo sacarlo de su posición.

Cabe recalcar que en los tres casos al quitar la fuerza externa aplicada en el cambio forzado de posición, el servomotor retoma la posición seleccionada.

Esto se debe a que a mayor frecuencia el servomotor recibe mayor potencia por ende ejerce mayor torque para sostener su posición, es por esto que al tratar de sacarlo de su posición muestra dificultad.

# **CONCLUSIONES Y RECOMENDACIONES**

Al desarrollar la tesis podemos concluir que es muy útil la construcción de un brazo robótico a pequeña escala así como del brazo empuja frascos dentro de la plataforma, para realizar diferentes proyectos con fines didácticos en el Laboratorio de Microcontroladores.

## **Conclusiones**

El brazo robótico y el brazo empuja frascos cumplen con las expectativas para los procesos de visión con LabVIEW y otras tareas de simulación dentro de la plataforma. La programación de la tarjeta controladora (Tarjeta dsPIC30F4012 diseñada por nuestro grupo de tesis) puede ser sometida a cambios para realizar nuevas actividades o tareas de carácter educativo tanto dentro como fuera de la plataforma.

En el proceso de desarrollo del brazo robótico se debió cambiar la tarjeta original del brazo adquirido por una tarjeta diseñada con dsPIC, debido a que la tarjeta propia del brazo robótico ofrecía un giro de 180 pasos de movimiento mientras que la nueva tarjeta dsPIC30F4012 nos da la posibilidad de 4500 pasos alrededor de los 180 grados, obteniendo un movimiento de 0,04 grados por paso.

El peso óptimo que puede levantar el brazo robótico es hasta 3oz. más 1oz. de margen de error. Este rango es sin afectar el movimiento y consumiendo una cantidad moderada de corriente que no provoque daños en la tarjeta controladora.

El brazo robótico puede trabajar a 65536 velocidades, considerando el parámetro 0 como la velocidad mas rápida y 65535 como la más lenta. Este parámetro de velocidad puede variar según las aplicaciones que el usuario esté realizando, ya que puede necesitar velocidad o precisión. La velocidad en los diferentes procesos en los que actúa el brazo robótico en la plataforma es más que suficiente para la aplicación actual en el Laboratorio de Microcontroladores.

## Recomendaciones

Basados en las pruebas que se hicieron, las recomendaciones que realizamos son las siguientes:

- No sobrecargar al brazo robótico con objetos muy pesados, debido a que se sobrecalientan los servos motores.
- No colocar objetos grandes que el brazo empuja frascos tenga problemas para empujar o detener.
- No someterlo a un ambiente húmedo, debido a los materiales que se usaron y a la circuitería presente en el proyecto.
- No se recomienda un ambiente muy salino.
- Se debe aislar el proceso de ambientes con polvo, ya que se pueden infiltrar en las juntas de los servos, afectando así su movilidad.

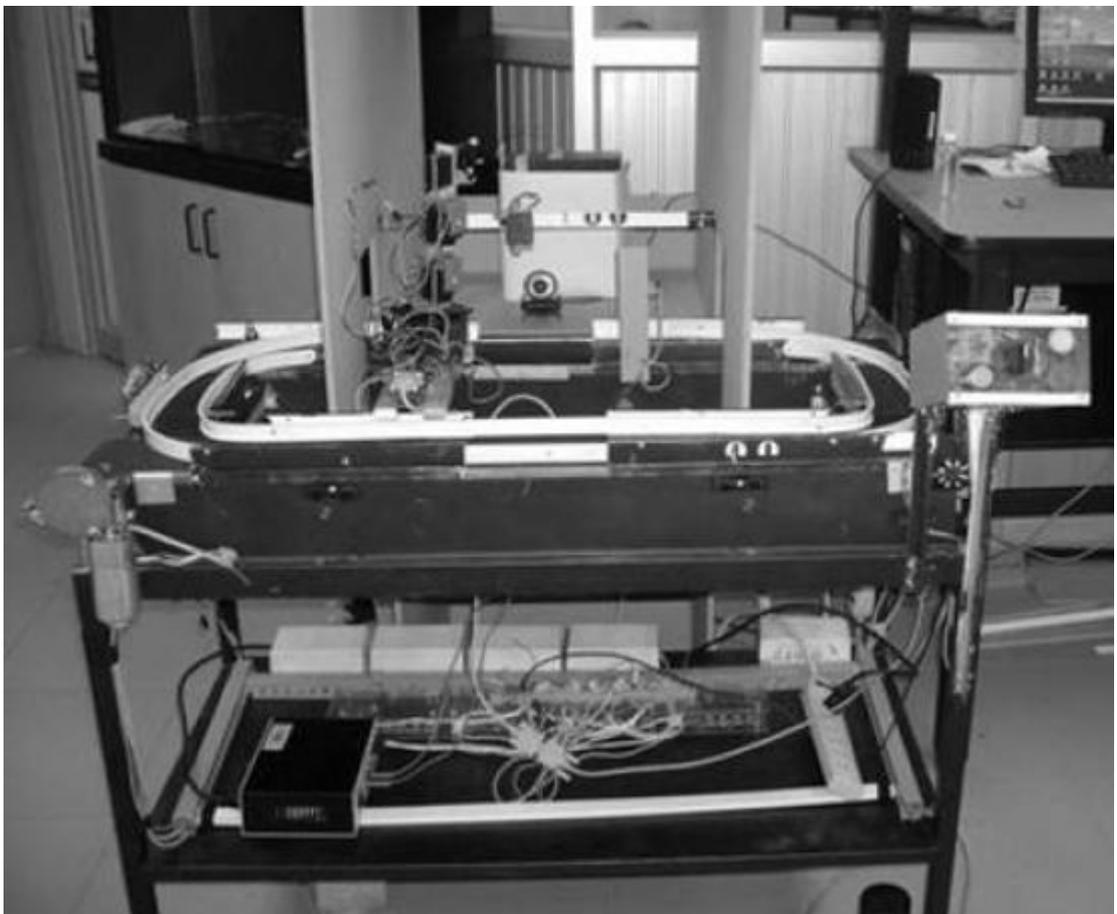
Los brazos objetos de esta tesis son recomendables para procesos en los cuales no se requiera extrema rapidez, ya que el diseño no lo permitirá, y si se lo quiere forzar a hacerlo, lo que se obtendrá es un comportamiento brusco y movimientos poco precisos.

Para futuros proyectos se sugiere utilizar los dsPICs de propósitos generales, por su capacidad y amplio campo de uso, así como el Mikrobasic para la programación de los mismos, ya que su sintaxis es sencilla para el desarrollo y mantenimiento del programa de control.

Se propone también el uso de LabVIEW para diseñar la interfaz de usuario-máquina, ya que se lo realiza de una manera muy sencilla y el usuario no requiere profundos conocimientos de programación.

## ANEXO A

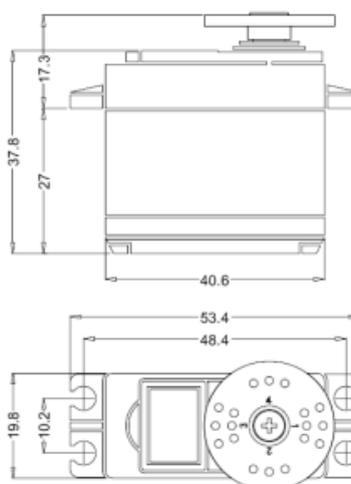
### FOTO DE LA PLATAFORMA DE DESARROLLO



Fuente: Gabriela Sánchez, Laboratorio de Microcontroladores

# ANEXO B-1

## HS-645MG High Torque Metal Gear Servo



**Operating Speed:** 0.24 / 0.20 sec.  
**Output Torque:** 107 / 133 oz. 7.7 / 9.6 kg.  
**Size:** 1.6"x 0.8"x 1.5" 41 x 20 x 38mm  
**Weight:** 1.94 oz. 55.2 g.

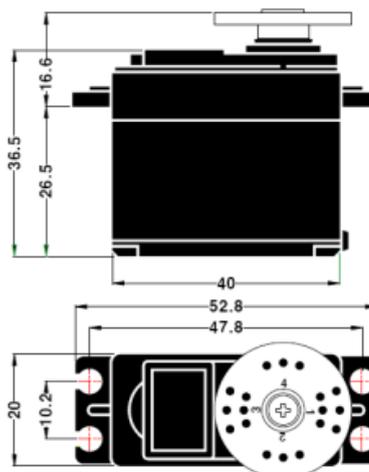
### DETAILED SPECIFICATIONS

Control System: +Pulse Width Control 1500usec Neutral  
 Required Pulse: 3-5 Volt Peak to Peak Square Wave  
 Operating Voltage: 4.8-6.0 Volts  
 Operating Temperature Range: -20 to +60 Degree C  
 Operating Speed (4.8V): 0.24sec/60 degrees at no load  
 Operating Speed (6.0V): 0.20sec/60 degrees at no load  
 Stall Torque (4.8V): 106.93 oz/in. (7.7kg.cm)  
 Stall Torque (6.0V): 133.31 oz/in. (9.6kg.cm)  
 Operating Angle: 45 Deg. one side pulse traveling 400usec  
 360 Modifiable: Yes  
 Direction: Clockwise/Pulse Traveling 1500 to 1900usec  
 Current Drain (4.8V): 8.8mA/idle and 350mA no load operating  
 Current Drain (6.0V): 9.1mA/idle and 450mA no load operating  
 Dead Band Width: 8usec  
 Motor Type: 3 Pole Ferrite  
 Bearing Type: Dual Ball Bearing  
 Gear Type: 3 Metal Gears and 1 Resin Metal Gear  
 Connector Wire Length: 11.81" (300mm)

Dimensions: 1.59" x 0.77" x 1.48" (40.6 x 19.8 x 37.8mm)  
 Weight: 1.94oz. (55.2g)

## ANEXO B-2

### HS-322HD Standard Heavy Duty Servo



**Operating Speed:** 0.15 sec/60° at 6.0V  
**Output Torque:** 3.7 kg/cm (51 oz/in) at 6.0V  
**Size:** 41 x 20 x 37 mm (1.6 x 0.8 x 1.4 in)  
**Weight:** 43 g (1.5 oz)

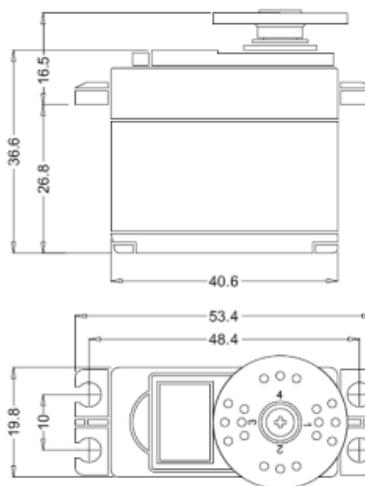
### DETAILED SPECIFICATIONS

Control System: +Pulse Width Control 1500usec Neutral  
 Required Pulse: 3-5 Volt Peak to Peak Square Wave  
 Operating Voltage: 4.8-6.0 Volts  
 Operating Temperature Range: -20 to +60 Degree C  
 Operating Speed (4.8V): 0.19sec/60 degrees at no load  
 Operating Speed (6.0V): 0.15sec/60 degrees at no load  
 Stall Torque (4.8V): 42 oz/in (3.0 kg/cm)  
 Stall Torque (6.0V): 51 oz/in (3.7 kg/cm)  
 Current Drain (4.8V): 7.4mA/idle and 160mA no load operating  
 Current Drain (6.0V): 7.7mA/idle and 180mA no load operating  
 Dead Band Width: 5usec  
 Operating Angle: 40 Deg. one side pulse traveling 400usec  
 Direction: Clockwise/Pulse Traveling 1500 to 1900usec  
 Motor Type: Cored Metal Brush  
 Potentiometer Drive: 4 Slider/Direct Drive  
 Bearing Type: Top/Resin Bushing  
 Gear Type: Heavy Duty Resin  
 360 Modifiable: Yes

Connector Wire Length: 11.81" (300mm)  
 Dimensions: 1.57" x 0.79"x 1.44" (40 x 20 x 36.5mm)  
 Weight: 1.52oz (43g)

## ANEXO B-3

### HS-422 Deluxe Servo



**Operating Speed:** 4.8/6.0v: 0.21 / 0.16 sec.  
**Output Torque:** 4.8/6.0v: 46 / 57 oz 3.3 / 4.1 kg.  
**Size:** 1.6"x0.8"x 1.4" 41 x 20 x 37mm  
**Weight:** 1.6 oz. 45.5 g.

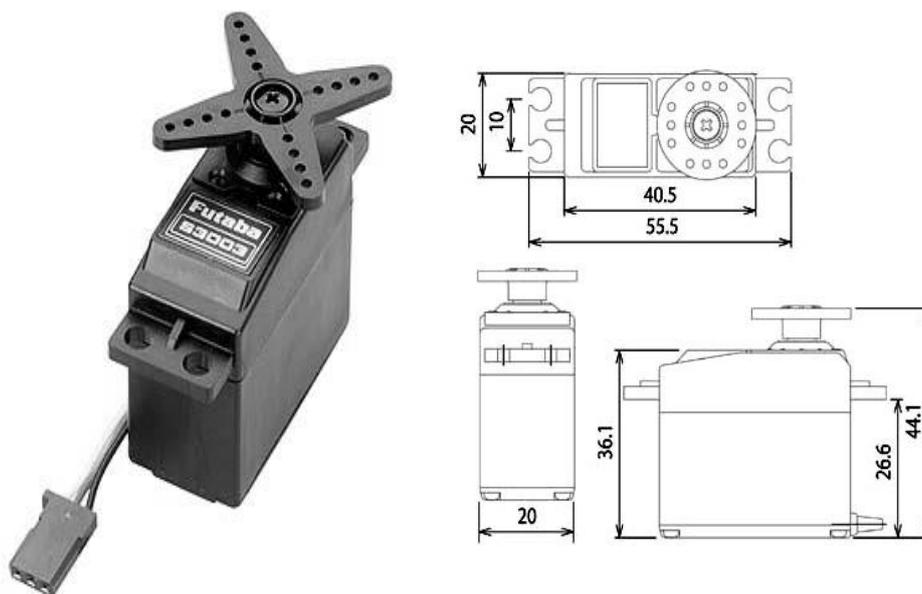
#### DETAILED SPECIFICATIONS

Control System: +Pulse Width Control 1500usec Neutral  
 Required Pulse: 3-5 Volt Peak to Peak Square Wave  
 Operating Voltage: 4.8-6.0 Volts  
 Operating Temperature Range: -20 to +60 Degree C  
 Operating Speed (4.8V): 0.21sec/60 degrees at no load  
 Operating Speed (6.0V): 0.16sec/60 degrees at no load  
 Stall Torque (4.8V): 45.82 oz/in. (3.3kg.cm)  
 Stall Torque (6.0V): 56.93 oz/in. (4.1kg.cm)  
 Operating Angle: 45 Deg. one side pulse traveling 400usec  
 360 Modifiable: Yes  
 Direction: Clockwise/Pulse Traveling 1500 to 1900usec  
 Current Drain (4.8V): 8mA/idle and 150mA no load operating  
 Current Drain (6.0V): 8.8mA/idle and 180mA no load operating  
 Dead Band Width: 8usec  
 Motor Type: 3 Pole Ferrite  
 Potentiometer Drive: Indirect Drive  
 Bearing Type: Dual Oilite Bushing  
 Gear Type: Nylon

Connector Wire Length: 11.81" (300mm)  
 Dimensions: 1.59" x 0.77"x 1.44" (40.6 x 19.8 x 36.6mm)  
 Weight: 1.6oz (45.5g)

## ANEXO B-4

### S3003 Servo Standard



#### DETAILED SPECIFICATIONS

**Control System:** +Pulse Width Control 1520usec Neutral  
**Required Pulse:** 3-5 Volt Peak to Peak Square Wave  
**Operating Voltage:** 4.8-6.0 Volts  
**Operating Temperature Range:** -20 to +60 Degree C  
**Operating Speed (4.8V):** 0.23sec/60 degrees at no load  
**Operating Speed (6.0V):** 0.19sec/60 degrees at no load  
**Stall Torque (4.8V):** 44 oz/in. (3.2kg.cm)  
**Stall Torque (6.0V):** 56.8 oz/in. (4.1kg.cm)  
**Operating Angle:** 45 Deg. one side pulse traveling 400usec  
**360 Modifiable:** Yes  
**Direction:** Counter Clockwise/Pulse Traveling 1520-1900usec  
**Current Drain (4.8V):** 7.2mA/idle  
**Current Drain (6.0V):** 8mA/idle  
**Motor Type:** 3 Pole Ferrite  
**Potentiometer Drive:** Indirect Drive  
**Bearing Type:** Plastic Bearing  
**Gear Type:** All Nylon Gears  
**Connector Wire Length:** 12"

**Dimensions:** 1.6" x 0.8" x 1.4" (41 x 20 x 36mm)  
**Weight:** 1.3oz. (37.2g)

## **ANEXO C**

### **DIAGRAMA PCB DE TARJETA dsPIC30F4012**

# ANEXO D

## FAMILIA DE dsPICs PARA MOTORES

| DISPOSITIVO  | PATITAS                  | MEMORIA PROGRAMA FLASH |                                 | SRAM   | EEPROM | ENTRADA MODO CAPTUR A | SALIDA MODO COMPADOR |            |
|--------------|--------------------------|------------------------|---------------------------------|--------|--------|-----------------------|----------------------|------------|
|              |                          | BYTES                  | INSTRUCCIONES                   |        |        |                       |                      |            |
| dsPIC30F2010 | 28                       | 12kB                   | 4k                              | 512 B  | 1024 B | 4                     | 2                    |            |
| dsPIC30F3010 | 28                       | 24kB                   | 8k                              | 1024 B | 1024 B | 4                     | 2                    |            |
| dsPIC30F4012 | 28                       | 48kB                   | 16k                             | 2048 B | 1024 B | 4                     | 2                    |            |
| dsPIC30F3011 | 40/44                    | 24kB                   | 8k                              | 1024 B | 1024 B | 4                     | 4                    |            |
| dsPIC30F4011 | 40/44                    | 48kB                   | 16k                             | 2048 B | 1024 B | 4                     | 4                    |            |
| dsPIC30F5015 | 64                       | 66kB                   | 22k                             | 2048 B | 1024 B | 4                     | 4                    |            |
| dsPIC30F6010 | 80                       | 144kB                  | 48k                             | 8192 B | 1024 B | 8                     | 8                    |            |
| DISPOSITIVO  | CONTROL DE MOTORES CODEC | A/D 12-BIT 500Ksps     | CODIFICACION DE CUADRATURA UART | UART   | SPI    | I2C                   | CAN                  | LINEAS E/S |
| dsPIC30F2010 | 6 canales                | 6 canales              | Sí                              | 1      | 1      | 1                     |                      | 20         |
| dsPIC30F3010 | 6 canales                | 6 canales              | Sí                              | 1      | 1      | 1                     |                      | 20         |
| dsPIC30F4012 | 6 canales                | 6 canales              | Sí                              | 1      | 1      | 1                     | 1                    | 20         |
| dsPIC30F3011 | 6 canales                | 9 canales              | Sí                              | 2      | 1      | 1                     |                      | 30         |
| dsPIC30F4011 | 6 canales                | 9 canales              | Sí                              | 2      | 1      | 1                     | 1                    | 30         |
| dsPIC30F5015 | 6 canales                | 16 canales             | Sí                              | 1      | 2      | 1                     | 1                    | 52         |
| dsPIC30F6010 | 8 canales                | 16 canales             | Sí                              | 2      | 2      | 1                     | 2                    | 68         |

Fuente: Datasheet dsPIC30F

## ANEXO E

### CÓDIGO FUENTE DEL FIRMWARE DEL dsPIC30F4012 EN BASIC

```

program DAQ_BrazoRobótico_EmpujaFrascos

dim i as word '# del servo.
dim index,rxDato1,rxDato2, rxDato2High, rxDato2Low as word 'Datos que se
obtienen de la comunicación serial.
dim Centro, Inicio, Fin as word 'Posicion del empuja Frascos.
dim MotorPosicion as word[6] 'Posicion Actual.
dim MotorNuevaPosicion as word[6] 'Posición Final.
dim ControlVelocidad, VelocidadSeleccionada as word 'Variables para el
'control de velocidad.

'Procedimiento para la inicialización de Variables
sub procedure Inicio_Reset
  'Se inicializa la posición de los servos cambiando el Ciclo de
  'trabajo
  'de la señal PWM.

  Pwm_Mc_Set_Duty(3700, 1) 'Servo 1.
  Pwm_Mc_Start()
  Pwm_Mc_Set_Duty(3700, 2) 'Servo 2.
  Pwm_Mc_Start()
  Pwm_Mc_Set_Duty(3700, 3) 'Servo 3.
  Pwm_Mc_Start()

  Pwm_Set_Duty(3700,1) 'Servo 4.
  Pwm_Start(1)

  Pwm_Set_Duty(3700,2) 'Servo 5.
  Pwm_Start(2)

  'PR4 se lo utiliza para la frecuencia de 50Hz de la onda de pulsos.
  'PR5 se lo utiliza para variar el ciclo de trabajo de la onda.
  PR4=50000
  Fin=1700
  Centro=2400
  Inicio=5000

  PR5=Inicio
  'Inicialización de la posición y posición futura.
  for i = 1 to 5
    MotorPosicion[i] = 3700
    MotorNuevaPosicion[i] = 3700
  next i
  'Inicialización de la velocidad seleccionada.
  ControlVelocidad = 0

```

```

    VelocidadSeleccionada = 512

end sub

'Timer5Int y Timer4Int: Procedimientos para la generación del PWM por
'software aplicado al Servo del empuja frascos
sub procedure Timer5Int org $40 'Interrupción por TMR5
    PORTE.0=0          ' Mandar a nivel bajo la Señal
    T5CON = $0010      ' Desactivar el temporizador TMR5
    IFS1 = IFS1 and $FFBF ' Limpiar la bandera TMR5IF
end sub

sub procedure Timer4Int org $3E 'Interrupción por TMR4
    PORTE.0=1          ' Mandar a nivel alto la señal
    T5CON = $8010      ' Habilitar el Temporizador TMR5
    IFS1 = IFS1 and $FFDF ' Limpiar la bandera TMR4IF
end sub

main:

    'Setear el puerto E como salida
    TRISE=%10000000
    PORTE=0

    'configurar la señal PWM a 50Hz de los PWM para los tres primeros
    'servos
    'duty_50=
    Pwm_Mc_Init(800,1,%01110000,0)
    PTCON.3=1

    'configurar la señal PWM a 50Hz del PWM para los dos servos
    'restantes
    'a la vez que se inicializa la posición de los servos.
    Pwm_Init(0, 2, 8, 3) 'prescalador de 8
    PR3=49999

    Pwm_Init(0, 1, 8, 2) 'prescalador de 8
    PR2=49999

    'Inicialización de la velocidad del puerto serial
    Uart1_Init(9600)

    Inicio_Reset()

    'Inicialización de los temporizadores TMR4 y TMR5. Los
    'temporizadores se los usa para PWM del empuja frascos.

    IFS1 = IFS1 and $FF9F ' Limpiar banderas TMR4IF y TMR5IF
    IEC1 = IEC1 or $0060 ' Habilitar interrupción por TMR4 y TMR5
    T4CON = $8010 ' Timer4 ON, prescaler 1:8
    while TRUE
        if Uart1_Data_Ready = 1 then 'Verifica dato listo en buffer
            rxDato1 = Uart1_Read_Char() 'Si hay dato, se procede a leer
            while Uart1_Data_Ready = 0
                wend
            rxDato2High = Uart1_Read_Char()
            while Uart1_Data_Ready = 0
                wend
            rxDato2Low = Uart1_Read_Char()

```

```

select case rxDato1
  'Ingreso de la nueva posición de cada servo
  case "1","2","3","4","5"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    rxDato2 = rxDato2High or rxDato2Low
    index=rxDato1-0x0030 'cambia a notacion Decimal
    MotorNuevaPosicion[index] = rxDato2 'nueva posición
  case "B"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    rxDato2 = rxDato2High or rxDato2Low

    select case rxDato2
      case 0
        PR5=Inicio 'Posición Inicial del empuja frascos
      case 1
        PR5=Centro 'Posición Central del empuja frascos
      case 2
        PR5=Fin 'Posición Final del empuja frascos
    end select
  'Ingreso de la velocidad de los servos del brazo robótico
  case "v"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    VelocidadSeleccionada = rxDato2High or rxDato2Low
  'Ingreso de la Posición Inicial del empuja frascos
  case "I"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    Inicio = rxDato2High or rxDato2Low
  'Ingreso de la Posición Final del empuja frascos
  case "F"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    Fin = rxDato2High or rxDato2Low
  'Ingreso de la Posición Central del empuja frascos
  case "C"
    rxDato2Low = rxDato2Low and $00FF
    rxDato2High = rxDato2High << 8
    Centro = rxDato2High or rxDato2Low
  'Identificación del dispositivo
  case "N"
    Uart1_Write_Text("200")
  'Identificación de comando para reseteo del sistema
  case "*"
    Inicio_Reset()
end select

end if

'El control de velocidad produce un retardo en el movimiento
'hasta llegar a la nueva posición deseada
ControlVelocidad = ControlVelocidad + 1

if ControlVelocidad >= VelocidadSeleccionada then
  for i = 1 to 5

```

```

if MotorPosicion[i] <> MotorNuevaPosicion[i] then
  'Verificamos si la nueva posición es menor a la actual
  if MotorPosicion[i] > MotorNuevaPosicion[i] then
    'Si es así, se procede a disminuir la posición
    MotorPosicion[i] = MotorPosicion[i] - 1
  end if
  'Verificamos si nueva posición es mayor a la actual
  if MotorPosicion[i] < MotorNuevaPosicion[i] then

    'Si es asi se procede a aumentar la posición
    MotorPosicion[i] = MotorPosicion[i] + 1

  end if

  if MotorPosicion[i] = MotorNuevaPosicion[i] then
    Uart1_Write_Char($AA)
    Uart1_Write_Char(i)
  end if

  'Se procede a la actualización del PWM
  select case i
    case 1,2,3
      Pwm_Mc_Set_Duty(MotorPosicion[i],i)
    case 4
      Pwm_Set_Duty(MotorPosicion[i],1)
    case 5
      Pwm_Set_Duty(MotorPosicion[i],2)
  end select

  end if
next i
ControlVelocidad = 0 'Se setea el control del retardo
end if

wend
end.

```

## ANEXO F-1

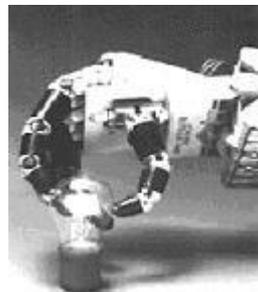
|               | <p align="center"><b>Clasificación de los robots según la</b></p> <p align="center"><b>AFRI Asociación Francesa de Robótica Industrial.</b></p> <p align="center"><b>IFR Federación Internacional de Robótica.</b></p> |
|---------------|--|
| <b>Tipo A</b> | Manipulador con control manual o telemando.  |
| <b>Tipo B</b> | Manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, eléctrico o hidráulico.  |
| <b>Tipo C</b> | Robot programable con trayectoria continua o punto a punto.<br>Carece de conocimiento sobre su entorno.  |
| <b>Tipo D</b> | Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.   |

**Fuente: Universidad de Atacama, Escuela de Ingeniería, Departamento de Industria y Negocios, "Robótica", abril 2005, pp. 14**

## ANEXO F-2

### EJEMPLOS DE CLASIFICACION DE LOS BRAZOS ROBÓTICOS

**Brazo Robótico**



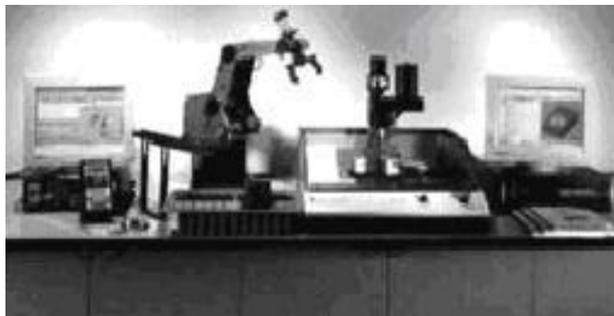
Fuente: [http://cfievalladolid2.net/tecno/cyr\\_01/  
robotica/industrial.htm#control](http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm#control)

**Manipulador**



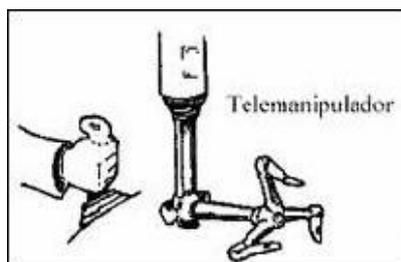
Fuente: [http://cfievalladolid2.net/tecno/cyr\\_01/robotica/  
industrial.htm#control](http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm#control)

### Robot de Repetición



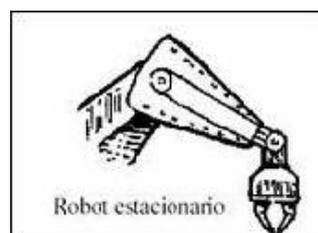
Fuente: [http://cfievalladolid2.net/tecnoc/cyr\\_01/robotica/industrial.htm#control](http://cfievalladolid2.net/tecnoc/cyr_01/robotica/industrial.htm#control)

### Tele manipulador



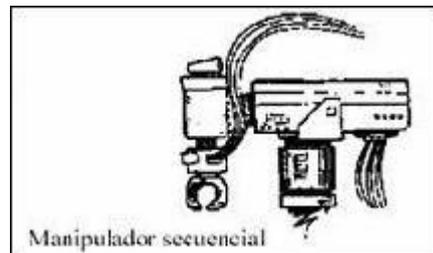
Fuente: Universidad de Atacama, Escuela de Ingeniería,  
Departamento de Industria y Negocios, "Robótica", abril 2005,  
pp. 19

### Robot Estacionario



Universidad de Atacama, Escuela de Ingeniería, Departamento de Industria y Negocios, “Robótica”, abril 2005, pp. 19

#### Robot Secuencial



Fuente: Universidad de Atacama, Escuela de Ingeniería, Departamento de Industria y Negocios, “Robótica”, abril 2005, pp. 19

#### Robot Sensorizado



Fuente: Universidad de Atacama, Escuela de Ingeniería, Departamento de Industria y Negocios, “Robótica”, abril 2005, pp. 19

## ANEXO F-3

### EJEMPLOS DE TIPOS DE ROBOTS

#### *ROBOT TERRESTRE*

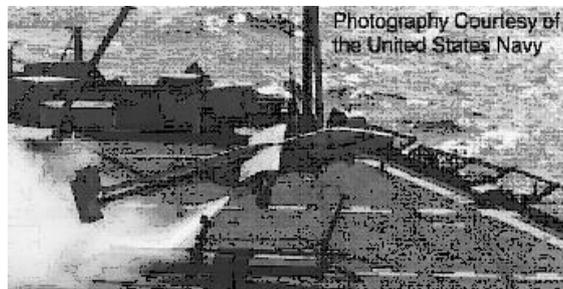
Robot Segway



Fuente: <http://www.roboticspot.com/spot/artic.shtml?todo=&block=6&newspage=robots>

#### *ROBOT AEREO*

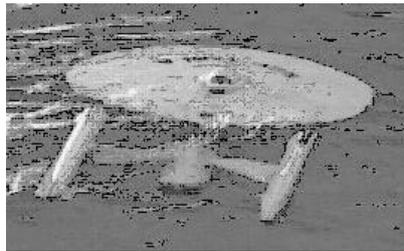
FIGURA 95. Robot Volador Prueba



Fuente: [http://www.pagines.fib.upc.es/~rob/protegit/treballs/Q2\\_03-04/aereos/inicio.html](http://www.pagines.fib.upc.es/~rob/protegit/treballs/Q2_03-04/aereos/inicio.html)

## ***ROBOT ACUATICO***

**Snake II**

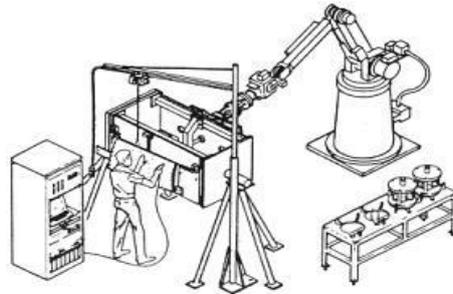


Fuente: <http://www.robotster.org/entry/top-5-underwater-robots/>

## ANEXO F-4

### EJEMPLOS DE APLICACIONES DE BRAZOS ROBÓTICOS

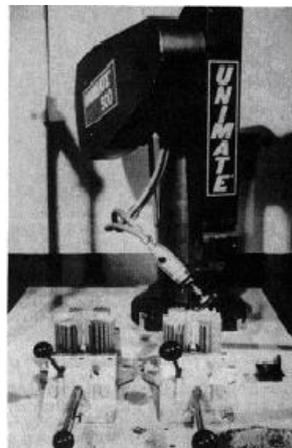
Soldadura por Arco



Fuente:

<http://www.monografias.com/trabajos6/larobo/larobo.shtml>

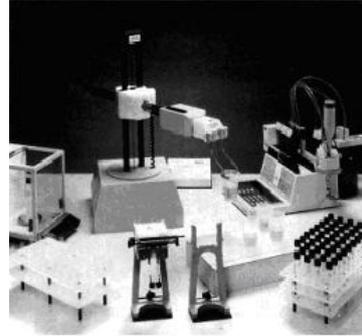
Taladro Unimate



Fuente:

<http://www.monografias.com/trabajos6/larobo/larobo.shtml>

#### Robot Movilizador de Muestras



Fuente:

<http://www.monografias.com/trabajos6/larobo/larobo.shtml>

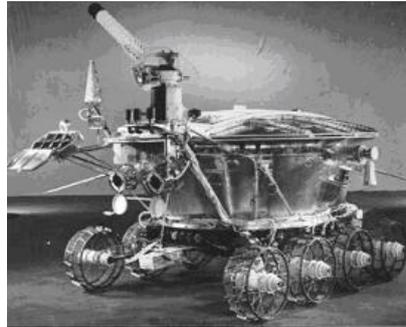
#### Esquiladora de Ovejas



Fuente:

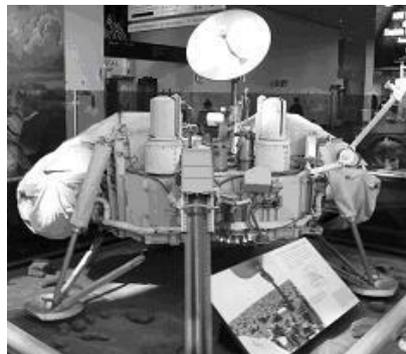
<http://www.monografias.com/trabajos6/larobo/larobo.shtml>

#### Lunokhod I



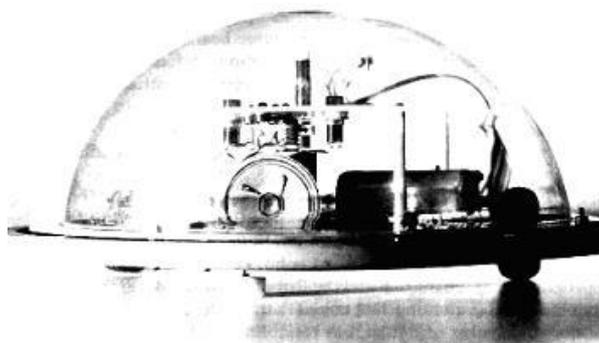
Fuente: [http://en.wikipedia.org/wiki/Lunokhod\\_1](http://en.wikipedia.org/wiki/Lunokhod_1)

### Viking I



Fuente: [http://en.wikipedia.org/wiki/Viking\\_1](http://en.wikipedia.org/wiki/Viking_1)

### Robot Tortuga



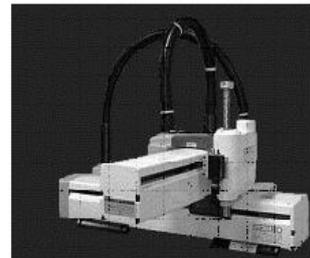
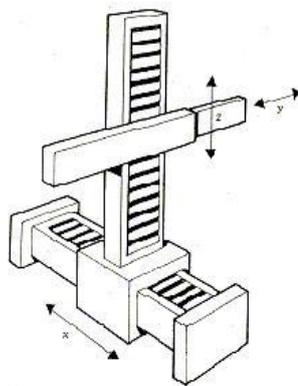
Fuente: <http://www.monografias.com/trabajos6/larobo/larobo.shtml>

ml

## ANEXO F-5

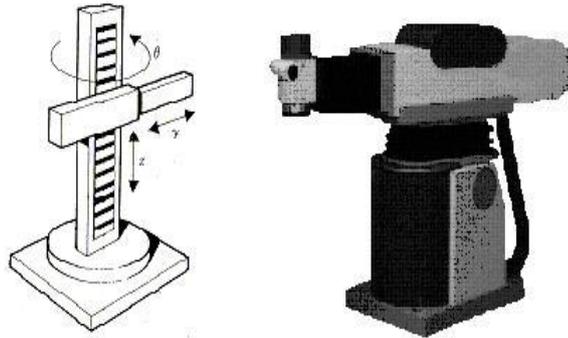
### EJEMPLOS DE CONFIGURACIONES MORFOLÓGICAS DE BRAZOS ROBÓTICOS

Robot Configuración Rectilínea



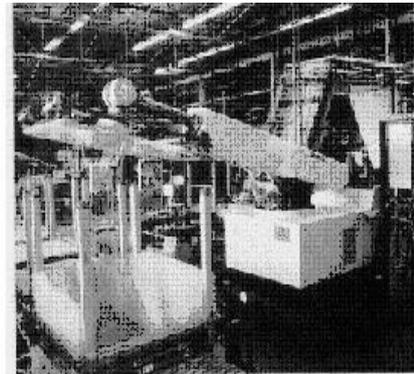
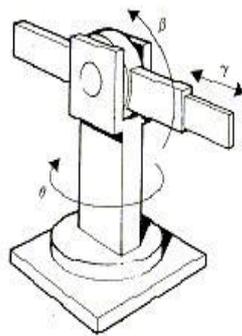
Fuente:[http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm)

Robot Configuración Cilíndrica



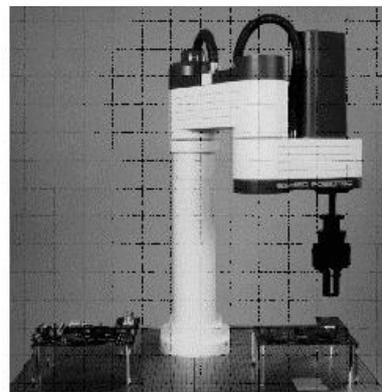
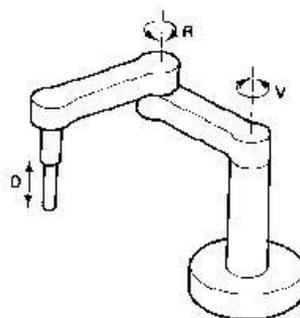
Fuente:[http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm)

### Robot Configuración Esférica



Fuente:[http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm)

### Configuración de Brazo Articulado (SCARA)



Fuente:[http://cfievalladolid2.net/tecno/cyr\\_01/robotica/sistema/morfologia.htm](http://cfievalladolid2.net/tecno/cyr_01/robotica/sistema/morfologia.htm)

## ANEXO G

| DISEÑO DE LA TARJETA CONTROLADORA PARA BRAZOS ROBÓTICOS |                       |                 |              |
|---|-----------------------|-----------------|--------------|
| Cantidad  | Descripción           | Precio Unitario | Precio Total |
| 1   | diseño de tarjeta PCB | 6,00            | 6,00         |
| 1   | dsPIC30F4012          | 9,85            | 9,85         |
| 1   | MAX232                | 3,00            | 3,00         |
| 1   | Cristal 10Mhz         | 1,50            | 1,50         |
| 1   | Led                   | 0,08            | 0,08         |
| 2   | Capacitores 22pF      | 0,30            | 0,60         |

|       |                                |      |       |
|-------|--------------------------------|------|-------|
| 5     | Capacitores Electrolíticos 1uF | 0,35 | 1,75  |
| 1     | Capacitor Electrolítico 10uF   | 0,40 | 0,40  |
| 1     | Capacitor Fijo 0.1uF           | 0,30 | 0,30  |
| 1     | Resistencia 1Kohm 1/4w         | 0,03 | 0,03  |
| 1     | Resistencia 4.7Kohm 1/4w       | 0,03 | 0,03  |
| 1     | Jack                           | 0,50 | 0,50  |
| 1     | Conector serial hembra         | 1,00 | 1,00  |
| 6     | Borneras tornillo              | 0,30 | 1,80  |
|       | Espadines 40 pines             | 0,40 | 0,40  |
| 2     | Puentes                        | 0,05 | 0,10  |
| 1     | Botonera                       | 0,10 | 0,10  |
| TOTAL |                                |      | 27,44 |

| <b>CONSTRUCCION DE BRAZO EMPUJA FRASCOS</b> |  |                        |                     |
|---|--|------------------------|---------------------|
| <b>Cantidad</b>                             | <b>Descripción</b>                             | <b>Precio Unitario</b> | <b>Precio Total</b> |
| 1   | Servomotor Futaba S3003                        | 24,50                  | 24,50               |
| 4   | Pernos cabeza de tornillo 4*1 1/2 con tuerca   | 0,05                   | 0,20                |
|   | MDF  | 1,00                   | 1,00                |
| 1   | Pieza plástica en forma de U                   |                        | 0,00                |
| 2   | Pernos cabeza de tornillo 4*2 con doble tuerca | 0,10                   | 0,20                |
| 1   | Tornillo Tripa de pato 6x1/2                   | 0,03                   | 0,03                |
| 2   | Pernos cabeza de tornillo 4*3/4 con tuerca     | 0,05                   | 0,10                |

|       |                             |       |       |
|-------|-----------------------------|-------|-------|
| 25 cm | Lámina de aluminio 1/2x1/16 | 5,00  | 5,00  |
| 1/2 m | perfil de aluminio          | 10,00 | 10,00 |
| TOTAL |                             |       | 41,03 |

## **ANEXO H**

**RESOLUCIÓN PWM Y NÚMERO DE PASOS DENTRO DE 180 GRADOS EN LOS SERVOMOTORES.**

$$\text{bits de resolución} = n = \frac{\log\left(\frac{2 \times T_{pwm}}{T_{cy}'}\right)}{\log(2)}, \text{ para los 3 primeros servomotores (32)}$$

$T_{cy}' = 16T_{cy}$ , donde  $T_{cy}$  es el tiempo de ciclo de instrucción que es 50ns

$$T_{pwm} = 20ms$$

$$T_{cy} = 50ns$$

$$T_{cy}' = 0.8\mu s$$

$$n = 15.069604$$

$$\text{resolución} = 2^n = 49999.999 \approx 50000$$

Esto nos dice que mi ciclo de trabajo de la onda PWM varia con 50000 pasos.

Esto quiere decir que a cada paso dura  $20ms/50000 \text{ pasos} = 0.4\mu s/\text{paso}$

Como el rango de operación del servomotor es de 900us a 2400us lo que

nos da un recorrido de 1800us para ir de 0 grados a 180 grados. Entonces:

$$\frac{1800\mu s}{0.4\mu s/\text{paso}} = 4500 \text{ pasos}$$

si dividimos los 180 grados por 4500. Entonces:

$$\frac{180 \text{ grados}}{4500 \text{ pasos}} = 0.04 \text{ grados} / \text{pasos}$$

Para los dos servomotores restantes que están conectados a la salida OUTPUT COMPARE, se analiza de forma diferente ya que en la figura 14-10 del Family Reference se muestra que el duty cycle o ciclo de trabajo está gobernado por  $OCxRS$  donde  $x$  es el canal que se va a utilizar que puede ser el 1 ó el 2, y varía desde  $0x0000h$  hasta  $PRy + 1$  que está establecido por el temporizador  $y$  (33). Entonces como  $PRy = 49999$ ,  $OCxRS$  puede variar a 50000 pasos y de esa forma obtenemos los 0.04 grados por paso, siguiendo el análisis anterior.

# BIBLIOGRAFIA

- (1) [http://cfievalladolid2.net/tecno/cyr\\_01/robotica/industrial.htm#control](http://cfievalladolid2.net/tecno/cyr_01/robotica/industrial.htm#control)  
Robots industriales.- Definición.- Clasificación.- Manipuladores, Robots de repetición y aprendizaje, Robots con control por computador, Robots inteligentes, Micro-robots.
- (2) Universidad de Atacama, Escuela de Ingeniería, Departamento de Industria y Negocios, Artículo publicado en revista, "Robótica", abril 2005, pp. 14.
- (3) Universidad de Atacama, Escuela de Ingeniería, Departamento de Industria y Negocios, Artículo publicado en revista, "Robótica", abril 2005, pp. 19.
- (4) <http://www.roboticspot.com/spot/artic.shtml?todo=&block=6&newspage=robots>; Tipos de robots.- Rover, Segway, Humanoide, Bípedo.
- (5) [http://www-pagines.fib.upc.es/~rob/protegit/treballs/Q2\\_03-04/aereos/inicio.html](http://www-pagines.fib.upc.es/~rob/protegit/treballs/Q2_03-04/aereos/inicio.html); Información de Robots Aéreos.
- (6) <http://www.robotster.org/entry/top-5-underwater-robots/>  
Lista de Robots Acuáticos
- (7) <http://www.monografias.com/trabajos6/larobo/larobo.shtml>  
La Robótica.- Introducción.- Breve historia de la robótica.- Automatización y robótica.- Clasificación de los robots.- Aplicaciones.-

Industria.- Aplicación de transferencia de material.- Carga y descarga de maquinas.- Operaciones de procesamiento.- Otras Operaciones de proceso.- Laboratorios.- Manipuladores cinemáticos.- Agricultura.- Espacio.- Vehículos.- Submarinos.- Educación.- El mercado de la robótica y las perspectivas futuras.

- (8) [http://en.wikipedia.org/wiki/Lunokhod\\_1](http://en.wikipedia.org/wiki/Lunokhod_1)  
Información acerca del vehículo de exploración lunar de la serie soviética Lunokhod 1.
- (9) [http://en.wikipedia.org/wiki/Viking\\_1](http://en.wikipedia.org/wiki/Viking_1)  
Información de vehículo de exploración en Marte enviado por la NASA programa Viking.
- (10) <http://www.interempresas.net/Plastico/FeriaVirtual/ResenyaProducto.asp?R=23748>  
Guía de compras de equipos industriales.- Reseña Robot Husky.
- (11) <http://www.interempresas.net/Robotica/FeriaVirtual/ResenyaProducto.asp?R=22246>  
Guía de compras de equipos industriales.- reseña Robots para soldadura por puntos: M-900iA/260L.
- (12) <http://www.fedimetal.com/company/kubiec.php>  
Página oficial de Kubiec y Fedimetal.- Equipos de Corte y Soldadura.

- (13) Kafrissen Edgard, Stephans Mark, "Industrial Robots and Robotics", Reston Publishing, Inc., a Prentice Hall Company, Reston, Virginia 22090, pp, 303-336.
- (14) Shoham Moshe, "A Textbook of Robotics 1", Basic Concepts, Eshed Robotec, Ltd., 1982, pp, 197-202.
- (15) [www.cfievalladolid2.net/tecnocyr\\_01/robotica/sistema/morfologia.htm](http://www.cfievalladolid2.net/tecnocyr_01/robotica/sistema/morfologia.htm)  
Estructura de un Robot Industrial.- Componentes.- Características Principales.- Configuraciones Morfológicas
- (16) Shoham Moshe, "A Textbook of Robotics 1", Basic Concepts, Eshed Robotec, Ltd., 1982, pp, 95-102.
- (17) Kafrissen Edgard, Stephans Mark, "Industrial Robots and Robotics", Reston Publishing, Inc., a Prentice Hall Company, Reston, Virginia 22090, pp, 155-159.
- (18) <http://www.interempresas.net/Robotica/FeriaVirtual/ResenyaProducto.asp?R=22225>  
Guía de compras de equipos industriales.- Reseña Robot para Manipulación de Construcción Modular M-6lb.
- (19) [http://www.chi.itesm.mx/profesional/laboratorios/equipo\\_cim.html](http://www.chi.itesm.mx/profesional/laboratorios/equipo_cim.html)  
Página de robótica industrial.- Ejemplo de Robots Industriales.- Sistemas donde regularmente se usan los mismos.
- (20) [www.imagesco.com](http://www.imagesco.com)  
RA-01 Robotic Arm & Controller, Manual y guía de usuario

- (21) <http://ww1.microchip.com/downloads/en/DeviceDoc/51553E.pdf>  
PICkit. 2 Programmer/Debugger, manual y guía de usuario
- (22) <http://www.ni.com/whatis/esa>  
Información General sobre LabVIEW
- (23) Barcia Orlando, Calle Roger, “Diseño de Tarjetas para el Desarrollo de Aplicaciones con DsPICs”, Proyecto de Tópico de Graduación, 2008
- (24) Hoja de datos de la Familia DsPIC30F4011/4012, [www.microchip.com](http://www.microchip.com), pp., 1-2.
- (25) Hoja de datos de la Familia DsPIC30F4011/4012, [www.microchip.com](http://www.microchip.com), pp., 73.
- (26) DsPIC30F Family Reference Manual.pdf, [www.microchip.com](http://www.microchip.com), Cap.14, pp., 19.
- (27) DsPIC30F Family Reference Manual.pdf, [www.microchip.com](http://www.microchip.com), Cap.5, pp., 5
- (28) DsPIC30F Family Reference Manual.pdf, [www.microchip.com](http://www.microchip.com), Cap.6, pp., 20, 26.
- (29) DsPIC30F Family Reference Manual.pdf, [www.microchip.com](http://www.microchip.com), Cap.12, pp., 4-5.
- (30) Hoja de datos Familia MAXIM, Multichannel & Drivers/Receivers, MAX220–MAX243, pp, 17.
- (31) <http://www.walmaring.com.ar/>  
Página oficial de Walmar SRL.- Equipos de Corte y Soldadura

- (32) DsPIC30F Family Reference Manual.pdf, [www.microchip.com](http://www.microchip.com), Cap.15, pp., 20-22.
- (33) DsPIC30F Family Reference Manual.pdf, [www.microchip.com](http://www.microchip.com), Cap.14, pp., 18, 20.